
Routing
Switching
Tigers
Forum
OSI layer
www.rstforum.net

OSI Layer

Initially host to host communications were proprietary, each vendor controlled its own protocols and hence protocols written by one vendor did not work with others. These proprietary protocols whose standards were not open and not known to other vendors started becoming obsolete. Hence there was a need to create a frame work that will act as standard for vendors developing networking technologies and protocols. OSI reference model was created that became Framework for networking Standards.

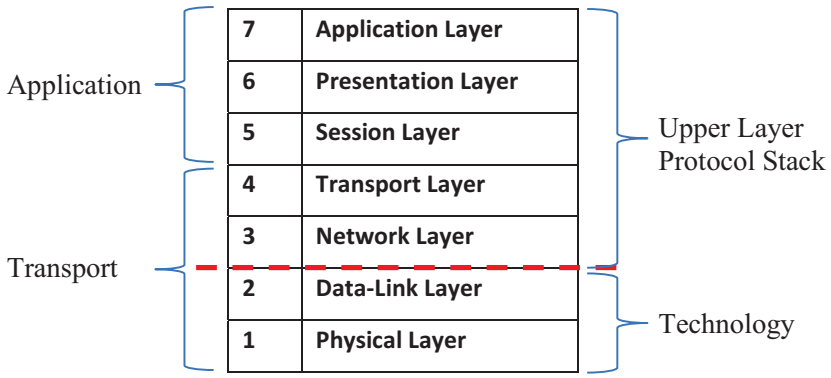
The Open Systems Interconnection (OSI) is a reference model developed by International Organization for Standards (ISO) in 1984. OSI reference model is an architectural model for inter-computer communications. OSI model is a Frame work of networking standards. OSI layer does not dictate on how your technology or protocol should work, but instead is specifies that if you have created any technology or protocol for any aspect of network operations then how and where to open its standards.

With a layered model various vendors can provide solutions for separate layers. Hardware vendors could design hardware and software to support emerging physical-level technologies like Ethernet, Frame-relay etc.

The ISO OSI model is used throughout the network, internet and telecom industries today to describe various networking issues. The OSI model is also of use in a learning or training environment where a novice can use it as a point of reference to learn how various technologies interact, where they reside, what functions they perform and how each protocol communicates with other protocols.

The OSI reference model is a conceptual model composed of seven layers, each specifying a particular network functions.

Figure: The OSI Reference Model Contains Seven Independent Layers



Characteristics of the OSI Layers

The seven layers of the OSI reference model can be divided into two categories: upper layers and lower layers.

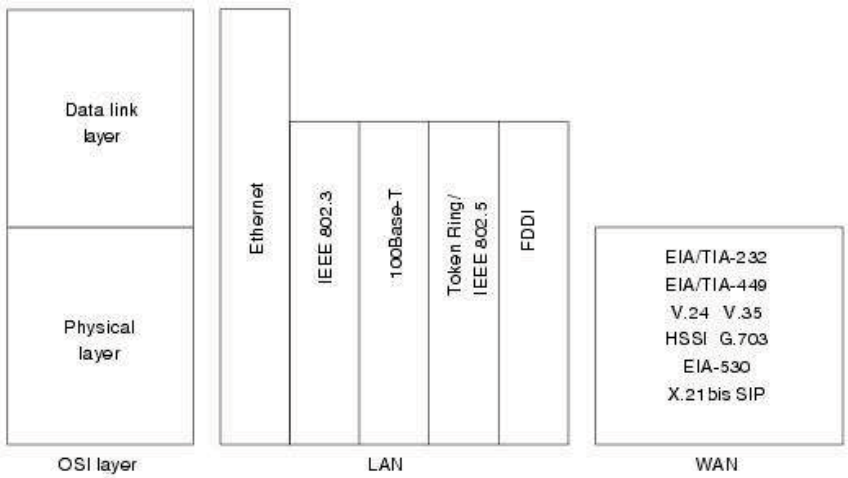
The upper layers of the OSI model deal with application and software. The highest layer, the application layer, is closest to the end user.

The lower layers of the OSI model talks about data transportation. The physical layer and the data link layer belongs to technology and are implemented in hardware and software. The lowest layer, the physical layer, is closest to the physical network medium (the network cabling, for example) and is responsible for actually placing information on the medium.

Physical Layers

As per OSI Layer, if there is anything that is physical or physical in nature used between two communicating devices to form a communication channel then details of that should be opened in physical layer of your technology. Things like cables, wire, connectors, pinouts, voltages, signals, boosting devices, etc. are all physical or physical in nature and specification of these should be specified in physical layer of your technology.

Physical layer specifications define characteristics such as voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, and physical connectors. The physical layer defines the electrical, mechanical, procedural, and functional specifications for activating, maintaining, and deactivating the physical link between communicating network systems.



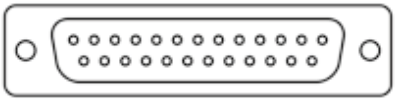
Physical layer implementations

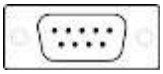
Data Link Layer	**CSMS/CD, 802.3, 802.2, ARPA**	**PPP, HDLC, Frame-relay, X.25, ATM**
Physical Layer	**EIA/TIA Eth Std. CAT 4/5/6, 10B2, 10B5, HUB, RJ45, BNC, AUI**	**EIA/TIA RS232, V.35**
	Ethernet	**Serial**

Ethernets physical layer characterizes the physical layer components like signaling method, physical media type, connectors, boosting devices etc. Ethernets physical layer talks about connectors like RJ45, BNC, AUI, etc. Media like 10B2, 10B5, 100BT, etc. cables specs like CAT4/5/6 etc. mostly defined in EIA/TIA standards.

Serial technology (WAN) physical layer characterizes the physical layer specification like EIA/TIA-232, 449, RS-232, 449, ITU-T V-Series, I.430, I.431, PDH, SONET/SDH, PON, OTN, DSL, IEEE 802.3, 802.11, 802.15, 802.16, 1394, ITU-T G.hn PHY, USB, Bluetooth, others.

All these specification speaks of physical components used on WAN but differ from one another in the physical specs, end user and vendors can select any spec in their physical layer connection

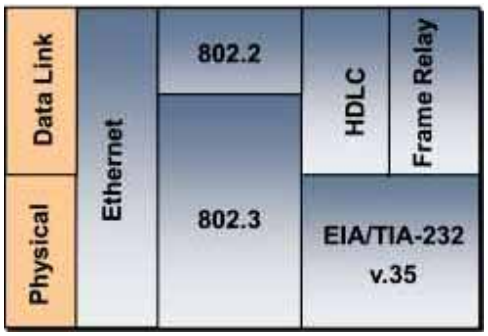




For example RS-232 is a standard for serial communication, and is commonly used in computer serial ports. This standard defines the electrical characteristics and timing of signals, the meaning of signals, and the physical size and pinout of connectors. As per RS232 physical layer specification, we can use either a 9 pin or 25 pin D-type connector. Signals would be transmitted over pin number 2, 3 & 5 on a 9 pin connector or pin number 2, 3 & 7 on a 25 pin connector. Signals can have voltage level of +-12 V and can be carried up to 250 meters over a flat cable.

Similarly the V.35 interface is located on layer 1 of the Open Systems Interconnection (OSI) V.35 has a blocky rectangular 34-pin connector. It achieves better speeds and distance by combining balanced and unbalanced voltage signals on the same interface. Cable distances range up to 1200 m at maximum speeds of 100 kbps. Actual distance depends on equipment and cable quality. V.35 was discontinued and replaced with the V.10 and V.11 recommendations.

Data-Link Layers



Data Link Layer	CSMS/CD, 802.3, 802.2, ARPA	PPP, HDLC, Frame-relay, X.25, ATM
Physical Layer	EIA/TIA Eth Std. CAT 4/5/6, 10B2, 10B5, HUB, RJ45, BNC, AUI	EIA/TIA RS232, V.35
	Ethernet	Serial

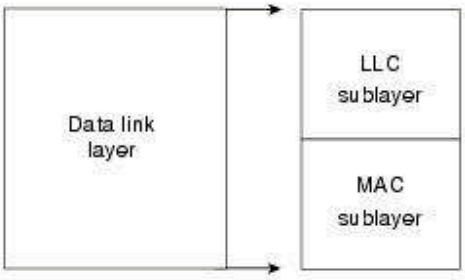
As per OSI Layer, if your technology has any software, tool or protocol, that creates understanding between two communicating devices connected over physical medium before actual communication happens then details of that should be opened in data-link layer of your technology.

Data link layer provides functional and procedural means to transfer data between networked devices, a lot of understanding is required between the two communicating pairs connected over physical medium before actual communication can happen. Understanding on parameters like start bit, stop bit, authentication, error checking, compression, correct transmission errors, activation, maintenance, and deactivation of data link connections, grouping of bits into characters and message frames, character and frame synchronization, etc. Communicating devices should have common understanding on the above parameters for communication to happen.

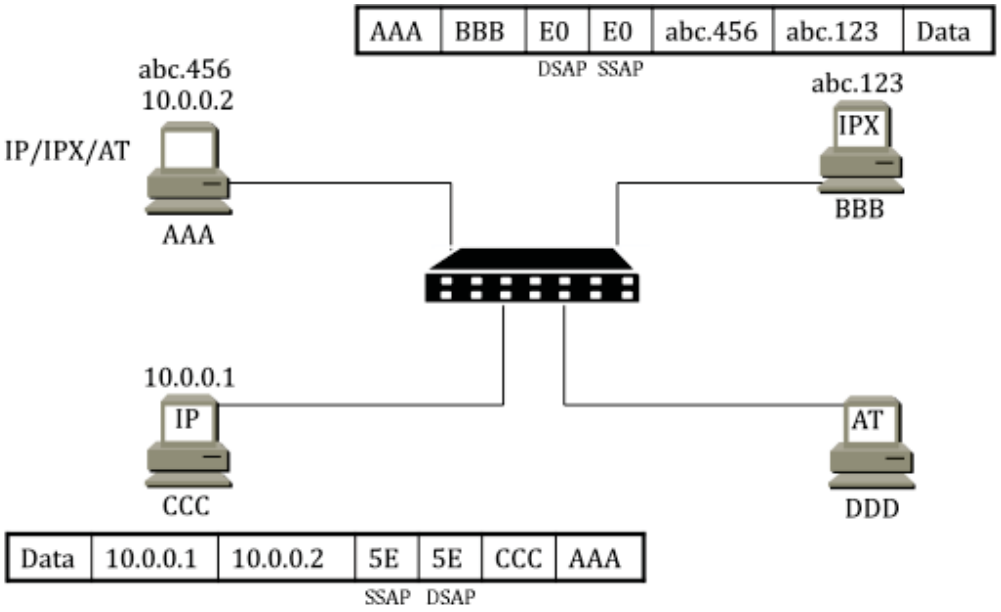
Serial technologies were created by various groups and hence there are multiple protocols available that create common understanding between communicating devices. Protocol like PPP, HDLC, Frame-relay X.25, ATM, etc. (WAN Protocols) were specified at layer 2 (L2) of serial technology. Different data link layer specifications define different network and protocol characteristics. All these protocol do the same job of building understanding between communicating devices but they do it in different way, for example HDLC does not support compression while PPP does, Frame-relay has capabilities of frame shaping but HDLC does not support frame shaping. It is up to users and OEM to decide which protocol they desire to use based on the features that these WAN protocols provides.

Any layer 2 protocols of serial technology can work on any serial technology L1 specification but Ethernet technology has only one protocol which can works on any Ethernet mediums.

Ethernet Data-link Layer: IEEE has subdivided the data link layer into two sublayers: Logical Link Control (LLC) and Media Access Control (MAC).



Initially Ethernet technology was created to support only single upper layer protocol and did not support multiple upper layer protocols, but as multiple protocols got created Data Link layer was broken into two parts, viz. LLC (Logical Link Control) and MAC (Media Access Control).



The Logical Link Control (LLC) sublayer of the data link layer is responsible for up-linking with upper-layer protocols (IP/IPX/AT) and enable higher-layer protocols to share a single physical link. To do this 802.3 introduced two additional field in the 802.3 frame i.e. SSAP, DSAP these fields were taken from 802.2 frame to identify multiple higher layer protocols.

The Media Access Control (MAC) sublayer of the data link layer manages protocol access to the physical network medium. The IEEE MAC specification defines function like: append/remove MAC address, Error Checking and discarding of corrupt/collided frames, Frame delimiting and recognition, Protection against errors, generally by means of generating and checking frame check, sequences, Control of access to the physical transmission medium, etc.

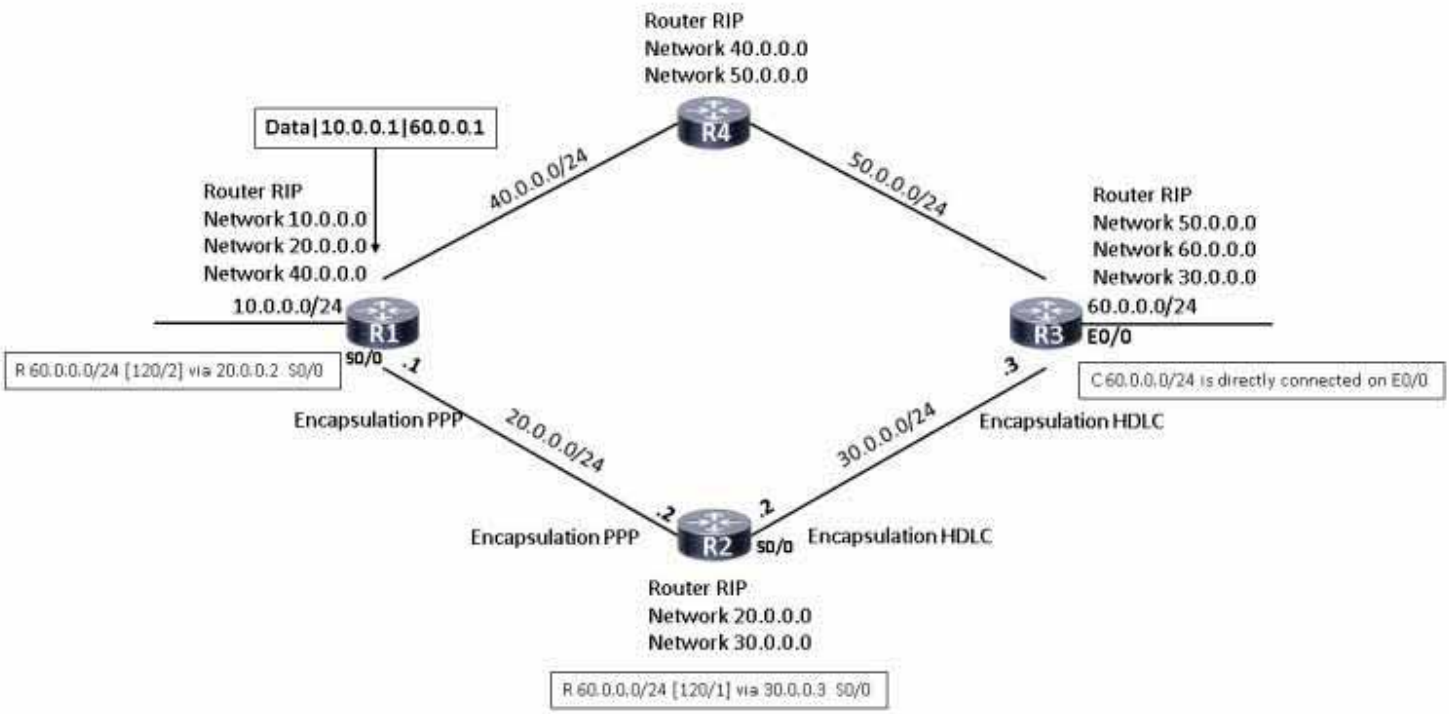
Network Layers

As per OSI layer, if your upper layer protocol stack has any software or protocol that helps to learn about all networks, all paths to reach all networks and select the best path to reach all networks then details of that should be specified in the layer 3 of your upper layer protocol stack.

IP protocol stack has multiple such protocols which does the above function. Protocols like RIP, IGRP, EIGRP, OSPF, BGP, etc. were specified in Layer 3 of IP upper layer protocol stack.

These protocols do the similar functions but in different ways and they differ from one another in the way they work.

For example, as per OSPF best path is path with highest bandwidth and for RIP best path is path with least hops. So it is up to administrator to decide what he wants in his network, whether he wants best path selection on basis of hops or bandwidth and select protocol accordingly.



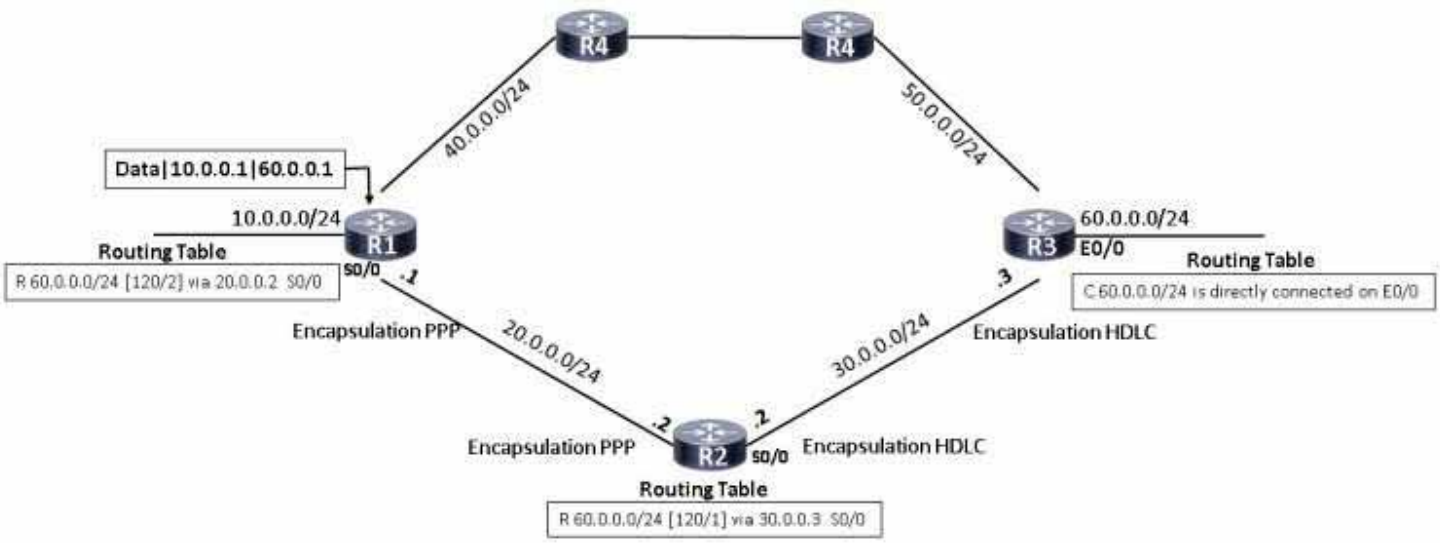
Technology has point to point visibility of network it does not have end to end visibility, end to end visibility of network is job of upper layer protocol.

For example if administrator selects to use PPP on R1 routers serial link then he will have use same PPP protocol on R2 routers S1/0 link. So that if PPP on R1 router send compresses data then the same protocol is available at the other end to decompress it.

Now if he is using PPP protocol on R2 routers S1/0 link does not mean he has to use PPP on R2 routers S0/0 link.so if plans to use HDLC on R2 routers S0/0 link then he will have use same HDLC protocol on R3 routers S1/0 link. Technology has point to point visibility of network it does not have end to end visibility,

End to end visibility of network is job of upper layer protocol. To do this job upper layer protocol has multiple routing protocols like RIP, IGRP, EIGRP, OSPF, ISIS, and BGP. These protocols are responsible to learn about all networks, to learn about all paths to reach all networks and select best path to reach all networks. They do same job but in different ways,

In the above example if we decide to use RIP routing protocol to learn all routes. Then every router will have 6 routes in their routing table. Every route will have 2 paths and only best path will reflect in routers routing table.



In the Diagram above from R1 router the best path to reach 60.0.0.0/24 network is via R2 & R3.

If R1 receives a data packet for 60.0.0.1, it will refer routing table to take forwarding decision. Routing table has path to reach 60.0.0.0/24 network via s0/0 port. So layer 3 will forward this packet to Layer 2 protocol of s0/0 interface. L2 protocol of s0/0 interface on R1 router is PPP. If compression is enabled then the PPP will compress bits and put it on wire.

At the other end on R2 router, PPP protocol will receive this compressed frame. It will decompress this frame and forward it to Layer 3, which in turn will refer routing table to take forwarding decision.

Routing table has path to reach 60.0.0.0/24 network via s0/0 port. So layer 3 will forward this packet to Layer 2 protocol of s0/0 interface. L2 protocol of s0/0 interface on R2 router is HDLC. HDLC will apply PADS and put it on wire.

At the other end on R3 router, HDLC protocol will receive this frame. It will remove PADs and forward it to Layer 3, which in turn will refer routing table and forward this packet to Layer 2 protocol of E0/0 interface.

In this way at every hop it is the Layer 3 protocol that will select the path on which data will be sent and use the technology of that path to deliver it to the other end and at the other end Layer 3 will again collect the data from technology, refer routing table, select forwarding path and use technology of that path to deliver data.

Application Layers

The application layer is closest to the end user and it is all about direct user interaction with the software application.

As per OSI layer, if your upper layer protocol stack has any software, tool or protocol that it provides to its users to use and communicate then details of that should be opened in application layer of your upper layer protocol stack.

IP upper layer protocol stack has multiple such tool, application or protocol that it provides to its users to use and communicate. So details of these protocols were opened in layer 7 of IP upper layer protocol stack. Examples of application layer implementations include Telnet, File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP).

Presentation Layers

The presentation layer provides a variety of coding and conversion functions that are applied to data. As per OSI layer if your upper layer protocol stack has any tool software or protocol that is responsible for converting code before sending and at the other end restoring the codes before sending it to user then details of that should be provided at Layer 6 of your upper layer protocol stack.

Some examples of presentation layer coding and conversion schemes include standard image, sound, and video formats. Conversion schemes like EBCDIC, ASCII, etc. are used for data representations. Some well-known standards for video include QuickTime and Motion Picture Experts Group (MPEG). QuickTime is an Apple Computer specification for video and audio, and MPEG is a standard for video compression and coding. Among the well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Tagged Image File Format (TIFF). GIF is a standard for compressing and coding graphic images. JPEG is another compression and coding standard for graphic images, and TIFF is a standard coding format for graphic images.

Code conversion and bit compression are two different phenomenon, bit compression takes place at layer 2 and code conversion happens at layer 6.

Presentation layer implementations are not typically associated with a particular protocol stack. IP upper layer protocol stack does not support code conversion, and hence there is no presentation layer in IP upper layer protocol stack.

Session Layers

The session layer establishes, manages, and terminates communication sessions. Communication sessions consist of service requests and service responses that occur between applications located in different network devices. These requests and responses are coordinated by protocols implemented at the session layer.

As per OSI Layer if your upper layer protocol stack has any tool software of protocol that creates session before communication then details of that should be opened in session layer of your upper layer protocol stack.

Some examples of session-layer implementations include Zone Information Protocol (ZIP), the AppleTalk protocol that coordinates the name binding process; and Session Control Protocol (SCP), the DECnet Phase IV session layer protocol.

In IP upper layer protocol stack session development and maintenance is done at transport layer and hence there is no session layer in IP Upper layer protocol stack.

Transport Layers

As per OSI Layer if your upper layer protocol stack has any tool, software or protocol that is responsible for end to end, error-free, successful delivery of data then details of that should be opened in transport layer of your upper layer protocol stack.

Transport layer protocol accepts data from the session layer and segments the data to transport across the network. Generally, the transport layer is responsible for making sure that the data is delivered error-free and in the proper sequence. Flow control generally occurs at the transport layer.

IP upper layer protocol talks of TCP (Transmission Control Protocol) at this layer, TCP is connection oriented transport protocol that provides guaranteed delivery of data. TCP is responsible for end-to-end, error-free, successful delivery of data.

IP upper layer protocol stack also talks of UDP (User Datagram Protocol) at this layer. UDP is a connectionless and unreliable transport protocol and used when a reliable delivery is not required. UDP is never used to send important data such as web-pages, database information, etc. Streaming media such as video, audio and others use UDP because it offers speed. The reason UDP is faster than TCP is

because there is no form of flow control. No error checking, error correction, or acknowledgment is done by UDP.

WWW, FTP, SSH, etc. are all TCP based applications, almost all data communication applications are TCP based. TCP is optimized for accurate delivery rather than timely delivery and hence it is not suitable for Real Time applications like Voice over IP. UDP is typically used for applications such as streaming media (audio, video, etc.).

TCP Working:

Transmission Control Protocol accepts data from a data stream, divides it into chunks, mark the segment with sequence number, apply checksum header (CRC) and use the IP layers below to deliver it to the other end. At the other end on receiving this segment TCP will check for errors and packet loss if all is received perfect then receiver will send acknowledgement with the next expected sequence number. The sender will now clear buffer take next sequenced segment or segments apply CRC code and send it to other end at other end on receiving frames it will check for errors and packet loss if all is received perfect then receiver will send acknowledgement with the next expected sequence number. In this way when all segments are received at the other end, TCP on both devices will exchange control message with FIN flag to close connection.

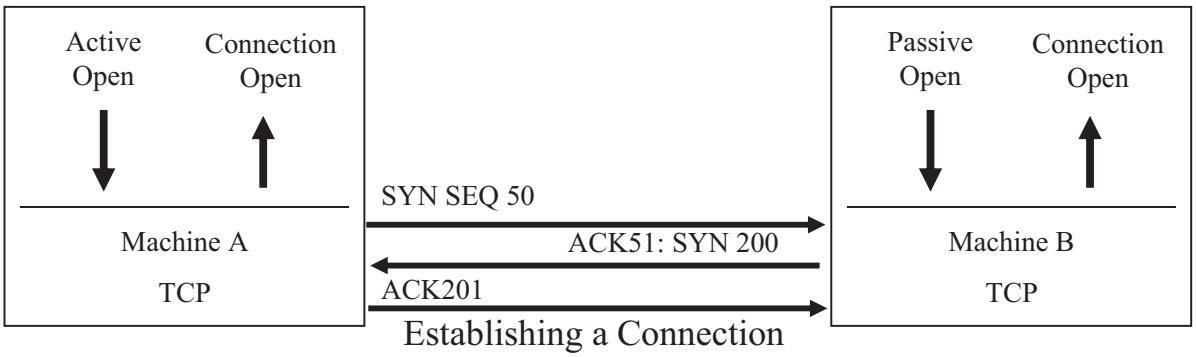
TCP Flow Example:

If user writes following command

FTP 10.0.0.2
Put ash.jpg

This means he wants to send ash.jpg file to 10.0.0.2, computer will fetch ash.jpg file from HDD, and FTP will send request (Active open) to TCP for connection establishment.

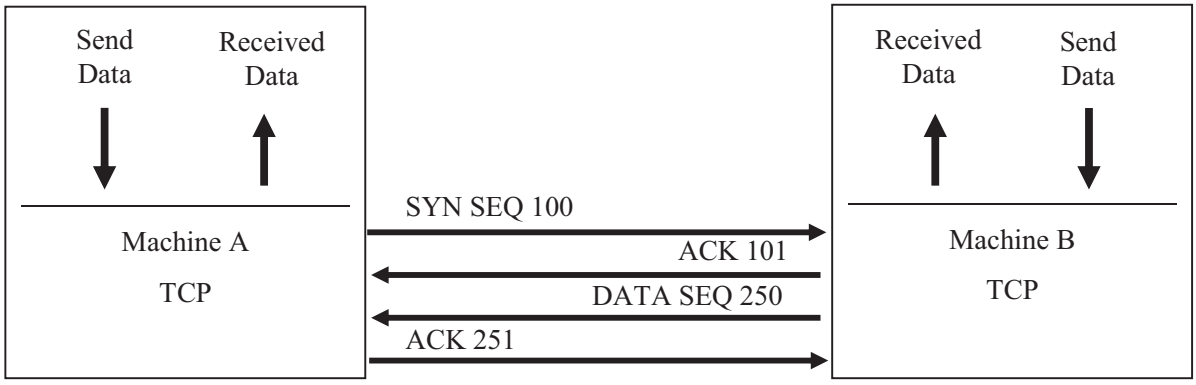
Connection Establishment: As process of connection establishment, devices will exchange SYN and ACK messages and decide on the initial sequence number they wants to use for its first transmission. The diagram below shows that machine A has sent SYN SEQ 50 indicating Initial Send Sequence number is 50. (Any number could have been chosen.)



Establishing a Connection

On receiving this, Application on machine B will send Passive Open, now machine B's TCP will send an ACK back to Machine A with the sequence number of 51. Now because it a 2 way communication, Machine B will also send its own Initial Send Sequence (ISS) number. The diagram shows this message as ACK 51; SYN 200 indicating that the message is an acknowledgment with sequence number 51, and has an ISS of 200. Upon receipt, Machine A sends back its own acknowledgment message with the sequence number set to 201. Then, having opened and acknowledged the connection, Machine A and Machine B both send connection open messages to the requesting applications (FTP in this scenario).

Data Transfer: Now Application will forward information to be delivered. Data is broken in to small blocks, TCP encapsulates it with its headers and sends it to Machine B with an increasing sequence number in this case 100. After Machine B receives the segment, it will check CRC and send acknowledges with next sequence number that it is expecting to receive (and hence indicates that it received everything up to the acknowledged sequence number). Figure below shows the transfer of only one segment of information - one each way.



Data Transfers

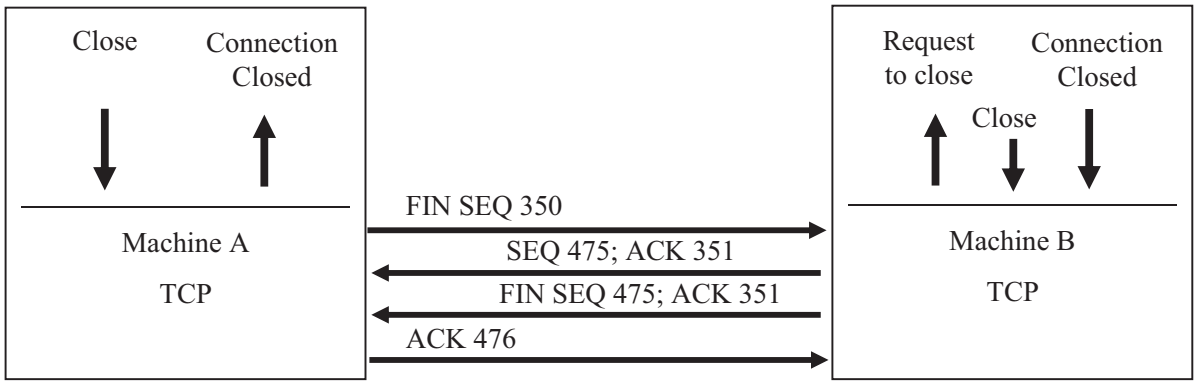
In our example Machine B will send ACK 101 to the Sender, which in turn will release the buffer and take next block apply header and send Data segment 101. At the other end Machine B will check CRC code of received segment, if OK it will send ACK 102. Now say for some reason ACK 102 is not received by sender within stipulated time then sender will collect the data block from buffer apply header and resend Data segment 101. At the other end on receiving the previously received segment it will over write the segment and send ACK 102 for next chunk.

On receiving ACK102, sender will send next Data segment 102. At the other end Machine B will check CRC code of received segment, if OK it will send ACK 103. Now sender will send Data segment 103 but for some reason the At the lower levels of the protocol stack, due to network congestion, traffic load balancing, or other unpredictable network behavior, IP packets may be lost, duplicated, or delivered out of order. TCP will detect these problems, requests retransmission (ARQ) of lost data, rearranges out-of-order data, and even helps minimize network congestion to reduce the occurrence of the other problems. If the data still remains undelivered, its source is notified of this failure. Once all segments of

the flow is received, the TCP receiver will assemble the segments and pass them to the receiving application.

Closing Connections: The application will now send a close primitive to TCP, which in-turn will send control message with the FIN flag set on. This is shown in the Figure below. In the figure below, Machine A's TCP sends the request to close the connection to Machine B with the next sequence number. Machine B will then send back an acknowledgment of the request and its next sequence number. Following this, Machine B sends the close message to its own application and waits for the application to acknowledge the closure. This step is not strictly necessary; TCP can close the connection without the application's approval, but a well-behaved system would inform the application of the change in state.

After receiving approval to close the connection from the application (or after the request has timed out), Machine B's TCP sends a control message back to Machine A with the FIN flag set. Finally, Machine A acknowledges the closure and the connection is terminated.



Closing a connection

TCP Window and Window Scaling: One of the functions of TCP is session development, Session development is for guaranteed delivery, deliveries guaranteed is provided by TCP with help of ACK, hence session development is only in TCP based communication. As part of session development, TCP hosts agree on how many segments (data) can be sent before receiver will acknowledge receipt of data. This is referred to as the TCP window size, and is communicated via a 16-bit field in the TCP header.

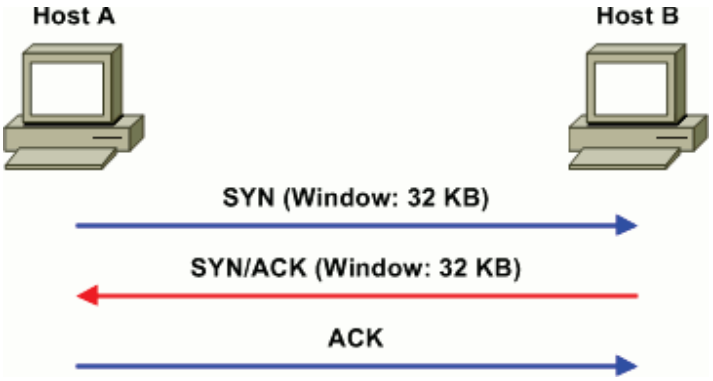
TCP windowing is important as TCP is a connection-oriented protocol and both ends of a connection should keep strict track of all transmitted data, so that any lost or jumbled segments can be retransmitted or reordered as necessary to maintain reliable transport.

There are many type of windowing mechanism like, fixed window, sliding windows, etc.

If sender and receiver have agreed on a fixed window size 1(say 1460 bytes) which means after every one segment that a sender sends, receiver will have to acknowledge receipt. For e.g. If sender send segment 1 then receiver will send ACK 2. Receiver will always send acknowledgement number of the segment that it wants next. Now sender will send segment 2 and on receipt receiver will send ACK3, if sender does not receive the acknowledgement within the expected time then it will retransmit the same segment again. On receipt receiver will send ACK 3 again. In this way with smaller window size benefit is that every segment is immediately getting verified and fixed, but drawback is it makes it slower and more bandwidth is used. Hence best would be to have larger window size.

If sender and receiver agree on larger window size say window size 3 (3 X 1460 bytes) which means after every 3 segment that a sender sends, receiver will acknowledgement the receipt. For e.g. If sender sends segment 1; 2; 3, then receiver will send ACK 4. Now sender will send segment 4; 5; 6, and if segment number 5 got dropped so receiver will only receive segment 4;__; 6. In this scenario receiver will send ACK5 so now sender will send 5; 6; 7. Segment no 6 is received twice computer will overwrite it and now send ACK 8. In this way with larger window size more segments are sent with lesser acknowledgements, so communication is faster and lesser bandwidth is utilized due to lesser ACK. Drawback is if segments are lost, duplicated, or delivered out of order then sender sends more then what is required.

Keeping this is mind sliding window was created. Which means while communicating, the sender and receiver can reduce or increase their window size. For example if A and B form a TCP connection. At the start of the connection, both hosts allocate 32 KB of buffer space for incoming data, so the initial window size for each is 32,768.



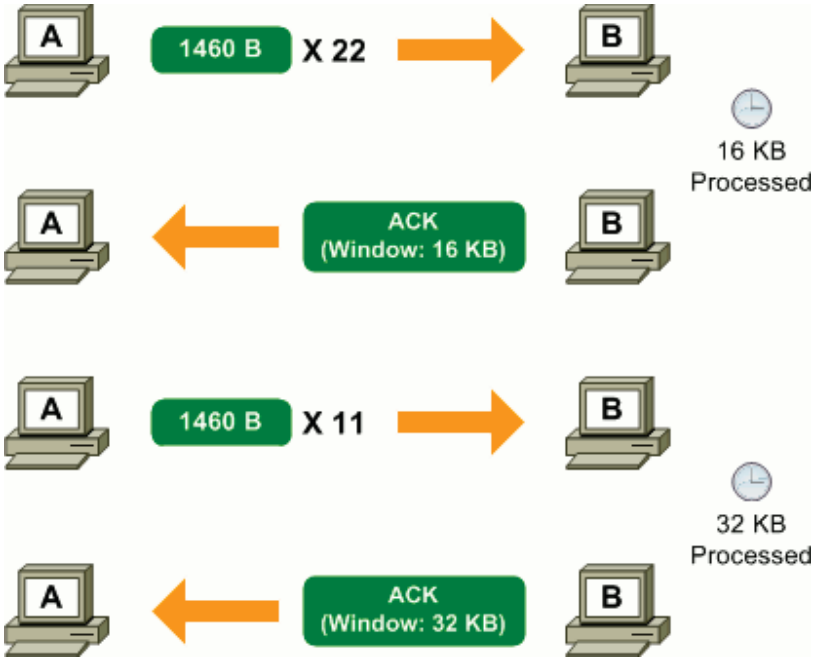
Host A needs to send data to host B. It will automatically understand from host B's advertised window size that it can only transmit up to 32,768 bytes of data (in intervals of the maximum segment size, or MSS) and wait for an acknowledgment.

By default MTU for Ethernet frame is 1500bytes, which has 20bytes of IP header and 20 bytes of TCP header and remaining is MSS (Maximum Segment size)

MSS = MTU - 20(IP header) - 20(TCP Header) = 1460

Assuming an MSS of 1460 bytes, host A can transmit 22 segments before exhausting host B's receive window.

When acknowledging receipt of the data received, host B can adjust its window size. For example, if the upper-layer application has only processed half of the buffer, host B would have to lower its window size to 16 KB. If the buffer was still entirely full, host B would set its window size to zero, indicating that it cannot yet accept more data.



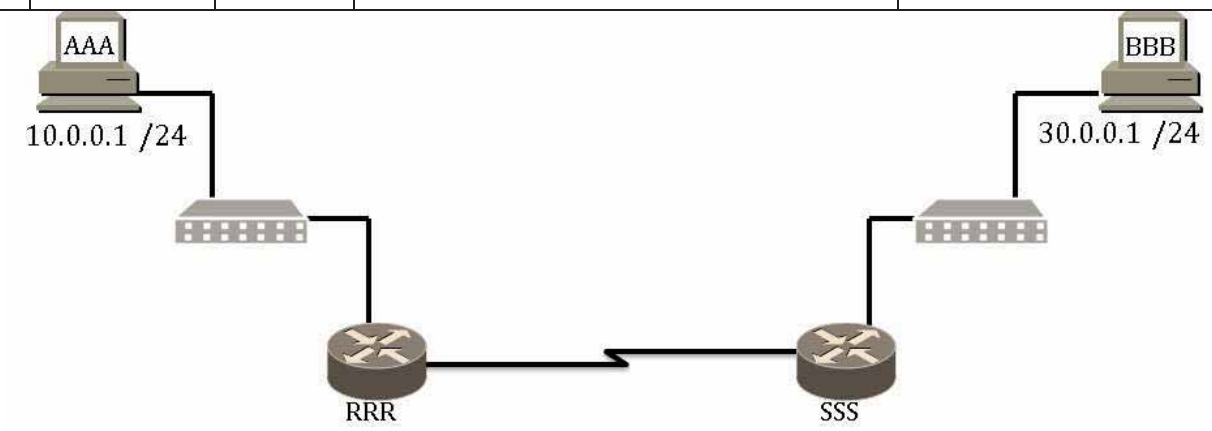
On a LAN with high bandwidth and extremely low delay, because our maximum receive window is only 65,535 bytes, host A must stop transmitting once this number has been reached and wait for an acknowledgment. This delay wastes potential throughput, unnecessarily inflating the time it takes to reliably transfer data across the network. TCP window scaling was created to address this problem. Which is out of scope of this document.

Error checking takes place at layer 4 and Layer 2, but error correction takes place only at Layer 4. If technology detects corruption of frames it will discard it but if Layer 4 detects corruption of segments or misses ACK it will correct it by retransmitting the segments.

We as a user cannot select to communicate in TCP or UDP, these are not user selectable parameters, we cannot select windowing mechanism (Fixed window or sliding window) nor can we select window size. These are defined by application developers and system.

Reference Model and Communication between Systems Layer

Layer	OSI Model	PDU	Functional Responsibility	Examples
7	Application	-----	User Interface	Telnet, HTTP, WWW & FTP
6	Presentation	-----	Define How Data is Presented	ASCII, EBCDIC
5	Session	-----	Keeping Different Application's Data Separate	Operating System
4	Transport	Segment	Defines Reliable or Un-Reliable Delivery, Error Detection & Recovery	TCP / UDP
3	Network	Packet	Provide Logical Addressing which Routers Use for Path Determination	IP, IPX, AT, RIP, IGRP, EIGRP, OSPF, ISIS, BGP
2	Data Link	Frame	Combines Bits into Bytes into Frames, Offer Access to Media using MAC Address and Performs Error Detection not Correction	802.3 / 802.2 HDLC
1	Physical	Bit	Specify Voltage, Wire Speed Pin-Out Cables and Moves Bits Between Devices	TIA/EIA-232 V.35



All layers of the reference model talks of some or the other software or protocol, except for the physical layer which talks of hardware. All IP enabled devices will have all these software installed on them, but depending on the functionality of devices some of the layers will be more elaborated on some devices and on others it may be less elaborated. For example, A PC's job is not to get involved in extensive

routing and hence network layer on a PC is lesser elaborated that means PC may not support RIP, EIGRP, OSPF, BGP routing protocols for routing, but instead it may only use a static route or default route for routing purposes. At the same time as PC is an edge device hence it has highly elaborated application layer for end users.

Similarly a routers main job is to exchange routes hence it will have highly elaborated Network Layer that supports all routing protocols, but less elaborated Application Layer as no one would want to sit on a router and browse internet.

L2 Switch is a technology device and hence will only have feature and functions offered by layer 2 software's.