OSI Model

Introduction: - The Open System Interconnection (OSI) reference model describes how information from a software application in one computer moves through a network medium to a software application in another computer. The OSI reference model is a conceptual model composed of seven layers, each specifying particular network functions. The model was developed by the International Organization for Standardization (ISO) in 1984, and it is now considered the primary architectural model for intercomputer communications. The OSI model divides the tasks involved with moving information between networked computers into seven smaller, more manageable task groups. A task or group of tasks is then assigned to each of the seven OSI layers. Each layer is reasonably self-contained so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers. The following list details the seven layers of the Open System Interconnection (OSI) reference model:

- Layer 7—Application
- Layer 6—Presentation
- Layer 5—Session
- Layer 4—Transport
- Layer 3—Network
- Layer 2—Data link
- Layer 1—Physical

Application Layer: - The application layer interfaces directly to and performs common application services for the application processes; it also issues requests to the presentation layer. Note carefully that this layer provides services to user-defined application processes, and not to the end user. For example, it defines a file transfer protocol, but the end user must go through an application process to invoke file transfer. The OSI model does not include human interfaces. The common application services sub layer provides functional elements including the Remote Operations Service

Element (comparable to Internet Remote Procedure Call), Association Control, and Transaction Processing (according to the ACID requirements).

Above the common application service sub layer are functions meaningful to user application programs, such as messaging (X.400), directory (X.500), file transfer (FTAM), virtual terminal (VTAM), and batch job manipulation (JTAM). These contrast with user applications that use the services of the application layer, but are not part of the application layer itself.

1. File Transfer applications using FTAM (OSI protocol) or FTP (TCP/IP Protocol)
2. Mail Transfer clients using X.400 (OSI protocol) or SMTP/POP3/IMAP (TCP/IP protocols)
3. Web browsers using HTTP (TCP/IP protocol); no true OSI protocol for web applications

Presentation Layer: - The Presentation layer establishes a context between application layer entities, in which the higher-layer entities can use different syntax and semantics, as long as the Presentation Service understands both and the mapping between them. The presentation service data units are then encapsulated into Session Protocol Data Units, and moved down the stack.

The original presentation structure used the Basic Encoding Rules of Abstract Syntax Notation One (ASN.1), with capabilities such as converting an EBCDIC-coded text file to an ASCII-coded file, or serializing objects and other data structures into and out of XML. ASN.1 has a set of cryptographic encoding rules that allows end-to-end encryption between application entities.

Session Layer:- The Session layer controls the dialogues/connections (sessions) between computers. It establishes, manages and terminates the connections between the local and remote application. It provides for full-duplex, half-duplex, or simplex operation, and establishes check pointing, adjournment, termination, and restart procedures. The OSI model made this layer responsible for "graceful close" of sessions, which is a property of TCP, and also for session check pointing and recovery, which is not usually used in the Internet protocols suite. Session layers are commonly used in application environments that make use of remote procedure calls (RPCs).

iSCSI, which implements the Small Computer Systems Interface (SCSI) encapsulated into TCP/IP packets, is a session layer protocol increasingly used in Storage Area Networks and internally between processors and high-performance storage devices. iSCSI uses TCP for guaranteed delivery, and carries SCSI command descriptor blocks (CDB) as payload to create a virtual SCSI bus between iSCSI initiators and iSCSI targets.

Transport Layer: - The Transport layer provides transparent transfer of data between end users, providing reliable data transfer services to the upper layers. The transport layer controls the reliability of a given link through flow control, segmentation/desegmentation, and error control. Some protocols are state and connection oriented. This means that the transport layer can keep track of the segments and retransmit those that fail.

Although it was not developed under the OSI Reference Model and does not strictly conform to the OSI definition of the Transport Service, the best known example of a layer 4 protocol is the Transmission Control Protocol (TCP). The transport layer is the layer that converts messages into TCP segments or User Datagram Protocol (UDP), Stream Control Transmission Protocol (SCTP), etc. packets.

Of the actual OSI protocols, not merely protocols developed under the model, there are five classes of transport protocols, ranging from class 0 (which is also known as TP0 and provides the least error recovery) to class 4 (which is also known as TP4 and is designed for less reliable networks, similar to the Internet). Class 4 is closest to TCP, although TCP contains functions, such as the graceful close, which OSI assigns to the Session Layer.

Perhaps an easy way to visualize the Transport Layer is to compare it with a Post Office, which deals with the dispatch and classification of mail and parcels sent. Do remember, however, that a post office manages the outer envelope of mail. Higher layers may have the equivalent of double envelopes, such as cryptographic presentation services that can be read by the addressee only. Roughly speaking, tunneling protocols operate at the transport layer, such as carrying non-IP protocols such as IBM's SNA or Novell's IPX over an IP network, or end-to-end encryption with IPsec. While Generic Routing Encapsulation (GRE) might seem to be a network layer protocol, if the encapsulation of the payload takes place only at endpoint, GRE becomes closer to a transport protocol that uses IP headers but contains complete frames or packets to deliver to an endpoint. L2TP carries PPP frames inside transport packets.

Network Layer: - The Network layer provides the functional and procedural means of transferring variable length data sequences from a source to a destination via one or more networks while maintaining the quality of service requested by the Transport layer. The Network layer performs network routing functions, and might also perform fragmentation and reassembly, and report delivery errors. Routers operate at this layer—sending data throughout the extended network and making the Internet possible. This is a logical addressing scheme – values are chosen by the network engineer. The addressing scheme is hierarchical.

The best-known example of a layer 3 protocol is the Internet Protocol (IP). It manages the connectionless transfer of data one hop at a time, from end system to ingress router, to router to router, and from egress router to destination end system. It is not responsible for reliable delivery to a next hop, but only for the detection of errored packets so they may be discarded. When the medium of the next hop cannot accept a packet in its current length, IP is responsible for fragmenting into sufficiently small packets that the medium can accept it.

A number of layer management protocols, a function defined in the Management Annex, ISO 7498/4, belong to the network layer. These include routing protocols, multicast group management, network layer information and error, and network layer address assignment. It is the function of the payload that makes these belong to the network layer, not the protocol that carries them.

Data-Link Layer: - The Data Link layer provides the functional and procedural means to transfer data between network entities and to detect and possibly correct errors that may occur in the Physical layer. Originally, this layer was intended for point-to-point and point-to-multipoint media, characteristic of wide area media in the telephone system. Local area network architecture, which included broadcast-capable multi-access media, was developed independently of the ISO work, in IEEE Project 802. IEEE work assumed sub layering and management functions not required for WAN use. In modern practice, only error detection, not flow control using sliding window, is present in modern data link protocols such as Point-to-Point Protocol (PPP), and, on local area networks, the IEEE 802.2 LLC layer is not used for most protocols on Ethernet, and, on other local area networks, its flow control and acknowledgment mechanisms are rarely used. Sliding window flow control and acknowledgment is used at the transport layers by protocols such as TCP, but is still used in niches where X.25 offers performance advantages.

Both WAN and LAN services arrange bits, from the physical layer, into logical sequences called frames. Not all physical layer bits necessarily go into frames, as some of these bits are purely intended for physical layer functions. For example, every fifth bit of the FDDI bit stream is not used by the data link layer.

Physical Layer: - The Physical layer defines all the electrical and physical specifications for devices. In particular, it defines the relationship between a device and a physical medium. This includes the layout of pins, voltages, cable specifications, Hubs, repeaters, network adapters, Host Bus Adapters (HBAs used in Storage Area Networks) and more.

To understand the function of the physical layer in contrast to the functions of the data link layer, think of the physical layer as concerned primarily with the interaction of a single device with a medium, where the data link layer is concerned more with the interactions of multiple devices (i.e., at least two) with a shared medium. The physical layer will tell one device how to transmit to the medium, and another device how to receive from it (in most cases it does not tell the device how to connect to the medium). Obsolescent physical layer standards such as RS-232 do use physical wires to control access to the medium.

The major functions and services performed by the physical layer are:

- Establishment and termination of a connection to a communications medium.
- Participation in the process whereby the communication resources are effectively shared among multiple users. For example, contention resolution and flow control.
- Modulation, or conversion between the representation of digital data in user equipment and the corresponding signals transmitted over a communications channel. These are signals operating over the physical cabling (such as copper and optical fiber) or over a radio link.

Parallel SCSI buses operate in this layer, although it must be remembered that the logical SCSI protocol is a transport-layer protocol that runs over this bus. Various physical-layer Ethernet standards are also in this layer;

Ethernet incorporates both this layer and the data-link layer. The same applies to other local-area networks, such as Token ring, FDDI, and IEEE 802.11, as well as personal area networks such as Bluetooth and IEEE 802.15.4.

OSI MODEL, LAYERS & PROTOCOLS

7 Applications

Web Browser, Email, Print Serivces, SIP, SSH and SCP, NFS, RTSP, Feed, XMPP, Whois, SMB; DNS; FTP; TFTP; BOOTP; SNMP;RLOGIN; SMTP; MIME; NFS; FINGER; TELNET; NCP; APPC; AFP; SMB

6 Presentations

XDR, ASN.1, SMB, AFP, NCP, MIDI, HTML, GIF, TIFF, JPEG, ASCII, EBCDIC

5 Sessions

TLS, SSH, X.225, RPC, NetBIOS, ASP, Winsock, BSD

4 Transports

TCP, UDP, RTP, SCTP, SPX, ATP
Gateway, Advanced Cable Tester, Brouter

3 Networks

IP, ICMP, IGMP, BGP, OSPF, RIP, IGRP, EIGRP, ARP, RARP, X.25, NETBEUI
Brouter, Router, Frame Relay Device, ATM Switch, Advanced Cable Tester, DDP

2 Data Link

Ethernet, Token ring, StarLAN, HDLC, Frame relay, ISDN, ATM, 802.11 WiFi, FDDI, PPP, Bridge, Switch, ISDN Router, Intelligent Hub, NIC, Advanced Cable Tester, ARCNET, LocalTalk, FDDI, ATM. NIC Drivers: Open Datalink Interface (ODI), Network Independent Interface Specification (NDIS)

1 Physical

NIC, Twisted Pair, Coax, Fiber Optic, Wireless Media, Repeater, Multiplexer, Hubs, (Passive/Active), TDR, Oscilloscope, Amplifier, Carrier pigeon

TCP LAYERS

4 Applications (OSI - Layers5 through 7)

HTTP, FTP, DNS
(Routing protocols like BGP and RIP, which for a variety of reasons run over TCP and UDP respectively, may also be considered part of the Internetwork layer)

3 Transports (OSI - Layers4 and 5)

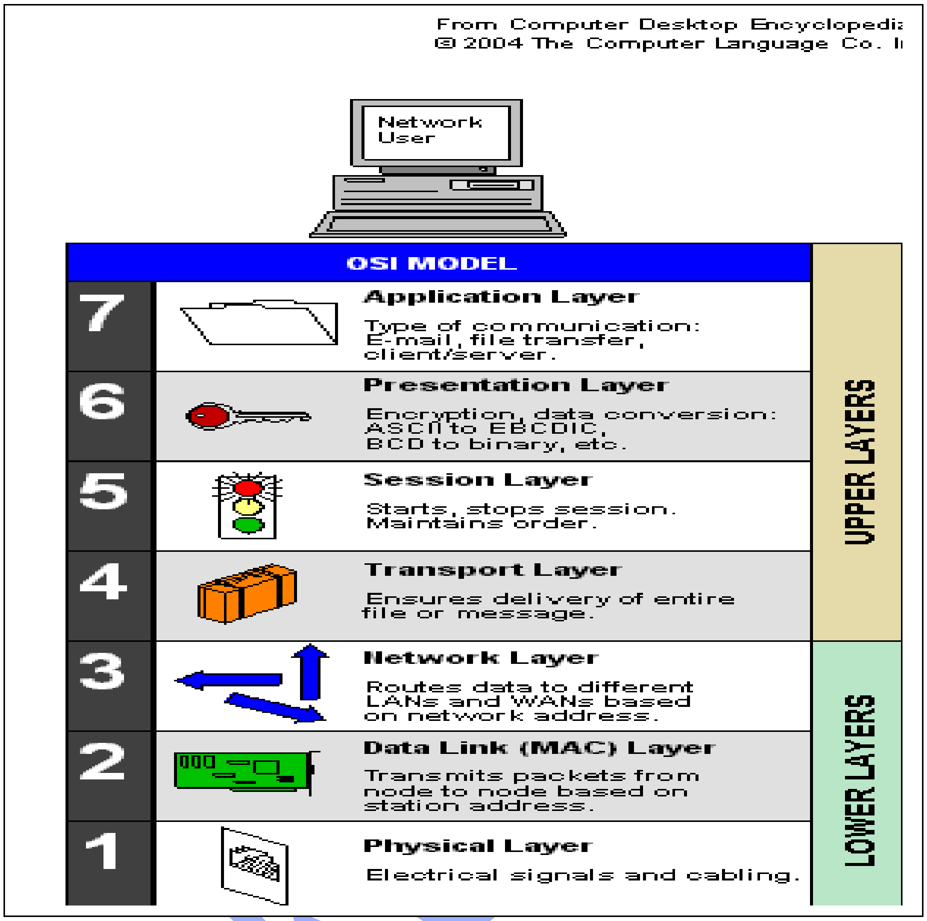
TCP, UDP, RTP, SCTP
(Routing protocols like OSPF, which run over IP, may also be considered part of the Internetwork layer)

2 Internetworks (OSI - Layer 3)

For TCP/IP this is the Internet Protocol (IP)
(Required protocols like ICMP and IGMP run over IP, but may still be considered part of the Internetwork layer; ARP does not run over IP)

1 Link (OSI - Layers 1 and 2)

Ethernet, WI-Fi, MPLS, etc.


From Computer Desktop Encyclopedia
© 2004 The Computer Language Co. I
Network User
OSI MODEL
7
Application Layer
Type of communication: E-mail, file transfer, client/server.
6
Presentation Layer
Encryption, data conversion: ASCII to EBCDIC, BCD to binary, etc.
5
Session Layer
Starts, stops session. Maintains order.
4
Transport Layer
Ensures delivery of entire file or message.
3
Network Layer
Routes data to different LANs and WANs based on network address.
2
Data Link (MAC) Layer
Transmits packets from node to node based on station address.
1
Physical Layer
Electrical signals and cabling.
UPPER LAYERS
LOWER LAYERS

Layer	Function	Protocols	Network Components
Application **User Interface**	• used for applications specifically written to run over the network • allows access to network services that support applications; • directly represents the services that directly support user applications • handles network access, flow control and error recovery • Example apps are file transfer,e-mail, NetBIOS-based applications	DNS; FTP; TFTP; BOOTP; SNMP;RLOGIN; SMTP; MIME; NFS; FINGER; TELNET; NCP; APPC; AFP; SMB	**Gateway**
Presentation	• Translates from application to network format and vice-versa		**Gateway**

Translation	• all different formats from all sources are made into a common uniform format that the rest of the OSI model can understand • responsible for protocol conversion, character conversion,data encryption / decryption, expanding graphics commands, data compression • sets standards for different systems to provide seamless communication from multiple protocol stacks • not always implemented in a network protocol		**Redirector**
Session **"syncs and sessions"**	• establishes, maintains and ends sessions across the network • responsible for name recognition (identification) so only the designated parties can participate in the session • provides synchronization services by planning check points in the data stream => if session fails, only data after the most recent checkpoint need be transmitted • manages who can transmit data at a certain time and for how long • Examples are interactive login and file transfer connections, the session would connect and re-connect if there was an interruption; recognize names in sessions and register names in history	NetBIOS Names Pipes Mail Slots RPC	**Gateway**
Transport **packets; flow control & error-handling**	• additional connection below the session layer • manages the flow control of data between parties across the network • divides streams of data into chunks or packets; the transport layer of the receiving computer reassembles the message from packets • "train" is a good analogy => the data is divided into identical units	TCP, ARP, RARP; SPX NWLink NetBIOS / NetBEUI ATP	**Gateway** **Advanced Cable Tester** **Brouter**

	• provides error-checking to guarantee error-free data delivery, with on losses or duplications • provides acknowledgment of successful transmissions; requests retransmission if some packets don’t arrive error-free • provides flow control and error-handling		
Network **addressing; routing**	• translates logical network address and names to their physical address (e.g. computername ==> MAC address) • responsible for o addressing o determining routes for sending o managing network problems such as packet switching, data congestion and routing • if router can’t send data frame as large as the source computer sends, the network layer compensates by breaking the data into smaller units. At the receiving end, the network layer reassembles the data • think of this layer stamping the addresses on each train car	**IP**; ARP; RARP, ICMP; RIP; OSFP; IGMP; **IPX** NWLink NetBEUI OSI DDP DECnet	**Brouter** **Router** **Frame Relay Device** **ATM Switch** **Advanced Cable Tester**

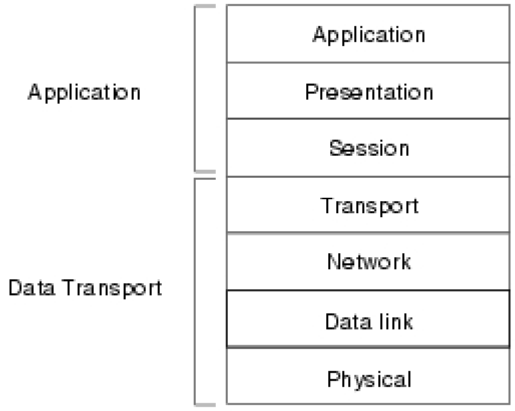
Data Link **data frames to bits**	• turns packets into raw bits 100101 and at the receiving end turns bits into packets. • handles data frames between the Network and Physical layers • the receiving end packages raw data from the Physical layer into data frames for delivery to the Network layer • responsible for error-free transfer of frames to other computer via the Physical Layer • this layer defines the methods used to transmit and receive data on the network. It consists of the wiring, the devices use to connect the NIC to the wiring, the signaling involved to transmit / receive data and the ability to detect signaling errors on the network media	**Logical Link Control** • error correction and flow control • manages link control and defines SAPs 802.1 OSI Model 802.2 Logical Link Control **Media Access Control** • communicates with the adapter card • controls the type of media being used: 802.3 CSMA/CD (Ethernet) 802.4 Token Bus (ARCnet) 802.5 Token Ring 802.12 Demand Priority	**Bridge** **Switch** **ISDN Router** **Intelligent Hub** **NIC** **Advanced Cable Tester**
Physical **hardware; raw bit stream**	• transmits raw bit stream over physical cable • defines cables, cards, and physical aspects • defines NIC attachments to hardware, how cable is attached to NIC • defines techniques to transfer bit stream to cable	IEEE 802 IEEE 802.2 ISO 2110 ISDN	**Repeater** **Multiplexer** **Hubs** • **Passive** • **Active** **TDR** **Oscilloscope** **Amplifier**

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 issues and generally are implemented only in software. The highest layer, the application layer, is closest to the end user. Both users and application layer processes interact with software applications that contain a communications component. The term upper layer is sometimes used to refer to any layer above another layer in the OSI model.

The lower layers of the OSI model handle data transport issues. The physical layer and the data link layer 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. Two Sets of Layers Make Up the OSI Layers



Protocols

The OSI model provides a conceptual framework for communication between computers, but the model itself is not a method of communication. Actual communication is made possible by using communication protocols. In the context of data networking, a protocol is a formal set of rules and conventions that governs how computers exchange information over a network medium. A protocol implements the functions of one or more of the OSI layers.

A wide variety of communication protocols exist. Some of these protocols include LAN protocols, WAN protocols, network protocols, and routing protocols. LAN protocols operate at the physical and data link layers of the OSI model and define communication over the various LAN media. WAN protocols operate at the lowest three layers of the OSI model and define communication over the various wide-area media. Routing protocols are network layer protocols that are responsible for exchanging information between routers so that the routers can select the proper path for network traffic. Finally, network protocols are the various upper-layer protocols that exist in a given protocol suite. Many protocols rely on others for operation. For example, many routing protocols use network protocols to exchange information between routers. This concept of building upon the layers already in existence is the foundation of the OSI model.

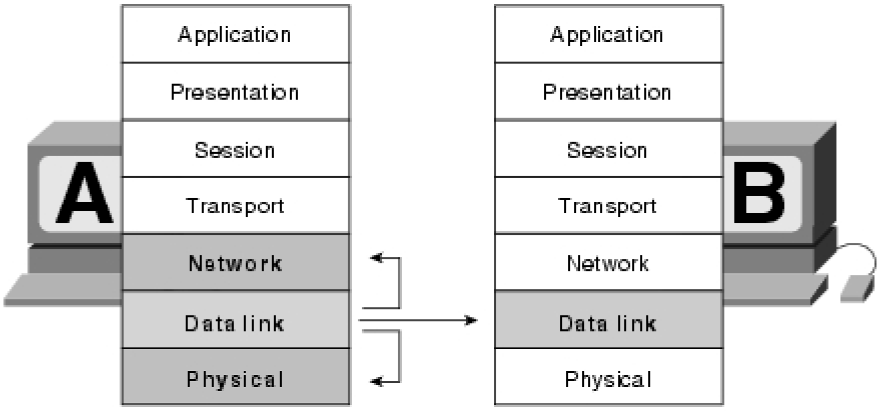
OSI Model and Communication between Systems

Information being transferred from a software application in one computer system to a software application in another must pass through the OSI layers. For example, if a software application in System A has information to transmit to a software application in System B, the application program in System A will pass its information to the application layer (Layer 7) of System A. The application layer then passes the information to the presentation layer (Layer 6), which relays the data to the session layer (Layer 5), and so on down to the physical layer (Layer 1). At the physical layer, the information is placed on the physical network medium and is sent across the medium to System B. The physical layer of System B removes the information from the physical medium, and then its physical layer passes the information up to the data link layer (Layer 2), which passes it to the network layer (Layer 3), and so on, until it reaches the application layer (Layer 7) of System B. Finally, the application layer of System B passes the information to the recipient application program to complete the communication process.

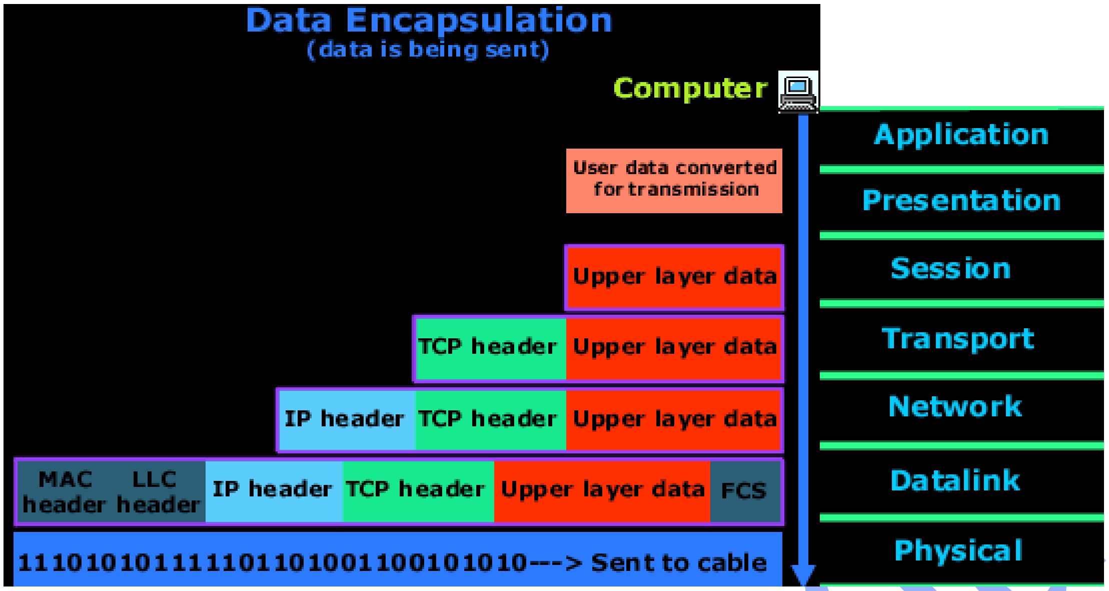
Interaction between OSI Model Layers

A given layer in the OSI model generally communicates with three other OSI layers: the layer directly above it, the layer directly below it, and its peer layer in other networked computer systems. The data link layer in System A, for example, communicates with the network layer of System A, the physical layer of System A, and the data link layer in System B. Figure 1-4 illustrates this example.

OSI Model Layers Communicate with Other Layers



Encapsulation and Decapsulation



The computer in the above picture needs to send some data to another computer. The Application layer is where the user interface exists, here the user interacts with the application he or she is using, then this data is passed to the Presentation layer and then to the Session layer. These three layer add some extra information to the original data that came from the user and then passes it to the Transport layer. Here the data is broken into smaller pieces (one piece at a time transmitted) and the TCP header is a added. At this point, the data at the Transport layer is called a segment.

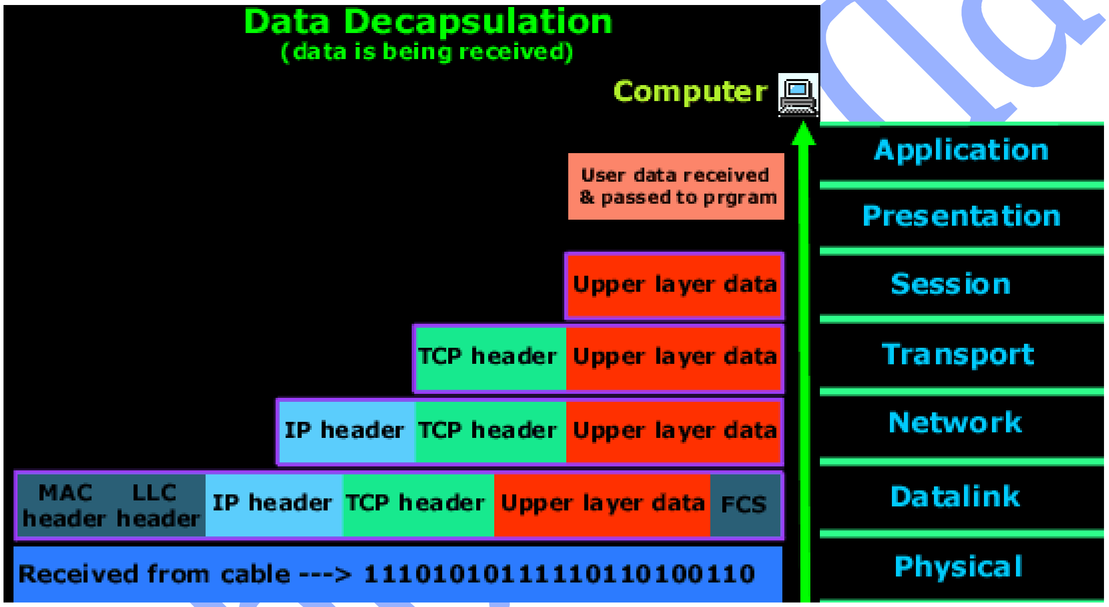
Each segment is sequenced so the data stream can be put back together on the receiving side exactly as transmitted. Each segment is then handed to the Network layer for network addressing (logical addressing) and routing through the internet network. At the Network layer, we call the data (which includes at this point the transport header and the upper layer information) a packet.

The Network layer adds its IP header and then sends it off to the Data link layer. Here we call the data (which includes the Network layer header, Transport layer header and upper layer information) a frame. The Data link layer is responsible for taking packets from the Network layer and placing them on the network medium (cable). The Data link layer encapsulates each packet in a frame which contains the hardware address (MAC) of the source and destination computer (host) and the LLC information which identifies to which protocol in the previous layer (Network layer) the packet should be passed when it arrives to its destination. Also, at the end, you will notice the

FCS field which is the Frame Check Sequence. This is used for error checking and is also added at the end by the Data link layer.

If the destination computer is on a remote network, then the frame is sent to the router or gateway to be routed to the destination. To put this frame on the network, it must be put into a digital signal. Since a frame is really a logical group of 1's and 0's, the Physical layer is responsible for encapsulating these digits into a digital signal which is read by devices on the same local network.

There are also a few 1's and 0's put at the beginning of the frame, only so the receiving end can synchronize with the digital signal it will be receiving.



The receiving computer will firstly synchronize with the digital signal by reading the few extra 1's and 0's as mentioned above. Once the synchronization is complete and it receives the whole frame and passes it to the layer above it which is the Data link layer.

The Data link layer will do a Cyclic Redundancy Check (CRC) on the frame. This is a computation which the computer does and if the result it gets matches the value in the FCS field, then it assumes that the frame has been received without any errors. Once that's out of the way, the Data link layer will strip off any information or header which was put on by the remote system's Data link layer and pass the rest (now we are moving from the Data link layer to the Network layer, so we call the data a packet) to the above layer which is the Network layer.

At the Network layer the IP address is checked and if it matches (with the machine's own IP address) then the Network layer header or IP header if you like, is stripped off from the packet and the rest is passed to the above layer which is the Transport layer. Here the rest of the data is now called a segment.

The segment is processed at the Transport layer, which rebuilds the data stream (at this level on the sender's computer it was actually split into pieces so they can be transferred) and acknowledges to the transmitting computer that it received each piece. It is obvious that since we are sending an ACK back to the sender from this layer that we are using TCP and not UDP. Please refer to the Protocols section for more clarification. After all that, it then happily hands the data stream to the upper-layer application.

You will find that when analyzing the way data travels from one computer to another most people never analyze in detail any layers above the Transport layer. This is because the whole process of getting data from one computer to another involves usually layers 1 to 4 (Physical to Transport) or layer 5 (Session) at the most, depending on the type of data.