Chapter3: OSI Reference Model:

Network Software:

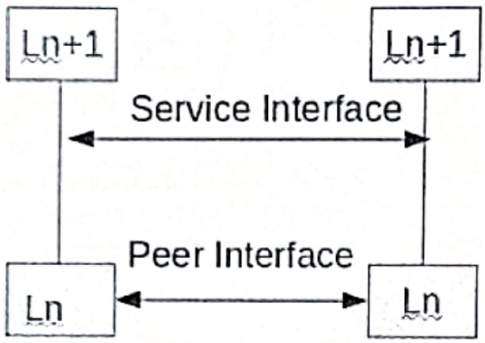
Network Software is a set of primitives that define the protocol between two machines. The network software resolves an ambiguity among different types of network making it possible for all the machines in the network to connect and communicate with one another and share information.

network software is the information, data or programming used to make it possible for computers to communicate or connect to one another.

Network software is used to efficiently share information among computers. It encloses the information to be sent in a "package" that contains a "header" and a "trailer". The header and trailer contain information for the receiving computer, such as the address of that computer and how the information package is coded. Information is transferred between computers as either electrical signals in electric wires, as light signals in fiber-optic cables, or as electromagnetic waves through space.

Protocol Hierarchies

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



This concept is actually a familiar one and used throughout computer science, where it is variously known as information hiding, abstract data types, data encapsulation, and object-oriented programming. The fundamental idea is that a particular piece of software (or hardware) provides a service to its users but keeps the details of its internal state and algorithms hidden from them.

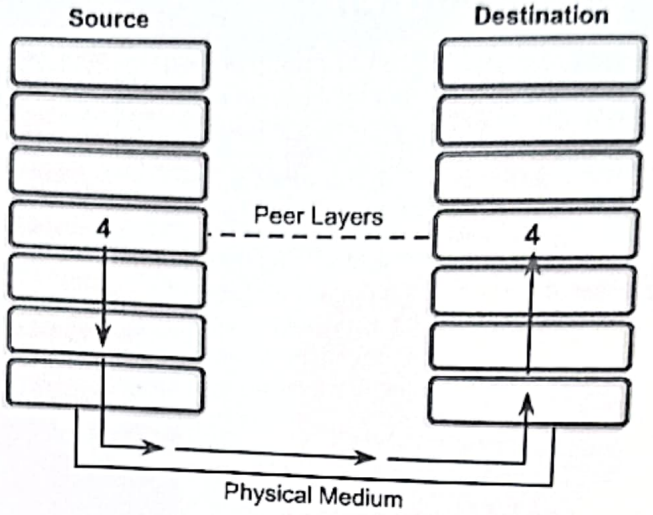
Layer n on one machine carries on a conversation with layer n on another machine. The rules and conventions used in this conversation are collectively known as the layer n protocol. Basically, a protocol is an agreement between the communicating parties on how communication is to proceed. As an analogy, when a woman is introduced to a man, she may choose to stick out her hand. He, in turn, may decide either to shake it or kiss it, depending, for example, on whether she is an American lawyer at a business meeting or a European princess at a formal ball. Violating the protocol will make communication more difficult, if not completely impossible.

Layer Communication:

In order for data packets to travel from a source to a destination on a network, it is important that all the devices on the network speak the same language or protocol. *A protocol is a set of rules that make communication on a network more efficient. For example, while flying an airplane, pilots obey very specific rules for communication with other airplanes and with air traffic control.*

A data communications protocol is a set of rules or an agreement that determines the format and transmission of data.

As shown in fig alongside Layer 4 on the source computer communicates with Layer 4 on the destination computer. The rules and conventions used for this layer are known as Layer 4 protocols. It is important to remember that protocols prepare data in a linear fashion. A protocol in one layer performs a certain set of operations on data as it prepares the data to be sent over the network. The data is then passed to the next layer where another protocol performs a different set of operations.



Once the packet has been sent to the destination, the protocols undo the construction of the packet that was done on the source side. This is done in reverse order. The protocols for each layer on the destination return the information to its original form, so the application can properly read the data.

OSI Model

An architectural model for open networking systems that was developed by the International Organization for Standardization (ISO) in Europe in 1974. The Open Systems Interconnection (OSI) reference model was intended as a basis for developing universally accepted networking protocols, but this initiative essentially failed for the following reasons.

- The standards process was relatively closed compared with the open standards process used by the Internet Engineering Task Force (IETF) to develop the TCP/IP protocol suite.
- The model was overly complex. Some functions (such as connectionless communication) were neglected, while others (such as error correction and flow control) were repeated at several layers.
- The growth of the Internet and TCP/IP—a simpler, real-world protocol model—pushed the OSI reference model out.

The OSI reference model is best seen as an idealized model of the logical connections that must occur in order for network communication to take place. Most protocol suites used in the real world, such as TCP/IP, DECnet, and Systems Network Architecture (SNA), map somewhat loosely to the OSI reference model. The OSI model is a good starting point for understanding how various protocols within a protocol suite function and interact.

Benefits of OSI Model:

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

Peer-to-Peer Communication:

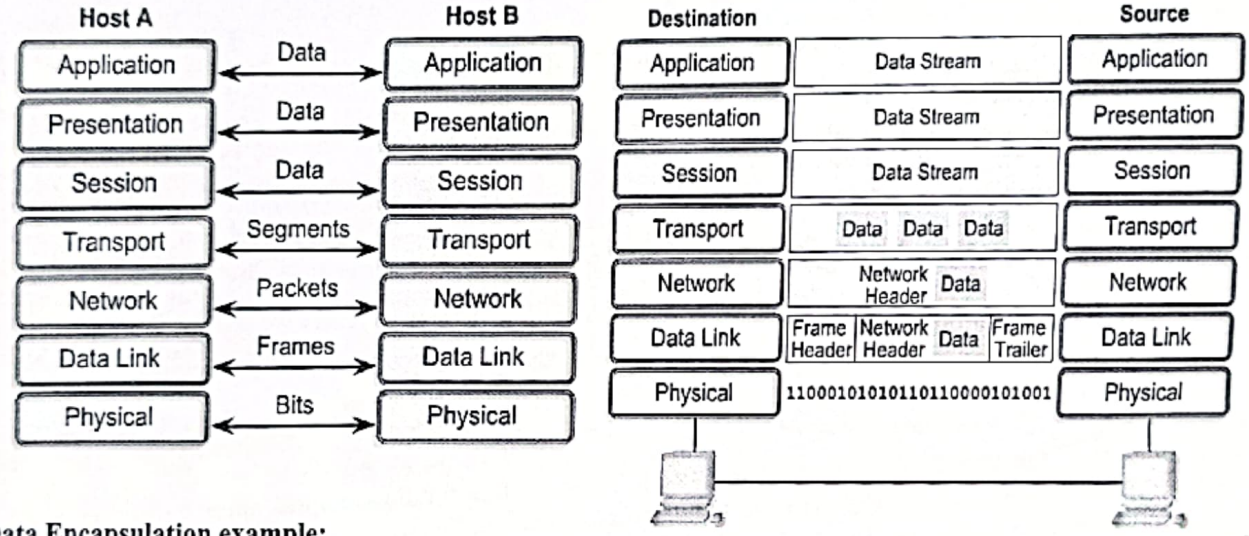
In order for data to travel from the source to the destination, each layer of the OSI model at the source must communicate with its peer layer at the destination. This form of communication is referred to as peer-to-peer. During this process, the protocols of each layer exchange information, called protocol data units (PDUs). Each layer of communication on the source computer communicates with a layer-specific PDU, and with its peer layer on the destination computer as illustrated in Figure

Data packets on a network originate at a source and then travel to a destination. Each layer depends on the service function of the OSI layer below it. To provide this service, the lower layer uses encapsulation to put the PDU from the upper layer into its data field. Then it adds whatever headers and trailers the layer needs to perform its function. Next, as the data moves down through the layers of the OSI model, additional headers and trailers are added.

Data Encapsulation:

All communications on a network originate at a source, and are sent to a destination. The information sent on a network is referred to as data or data packets. If one computer (host A) wants to send data to another computer (host B), the data must first be packaged through a process called encapsulation.

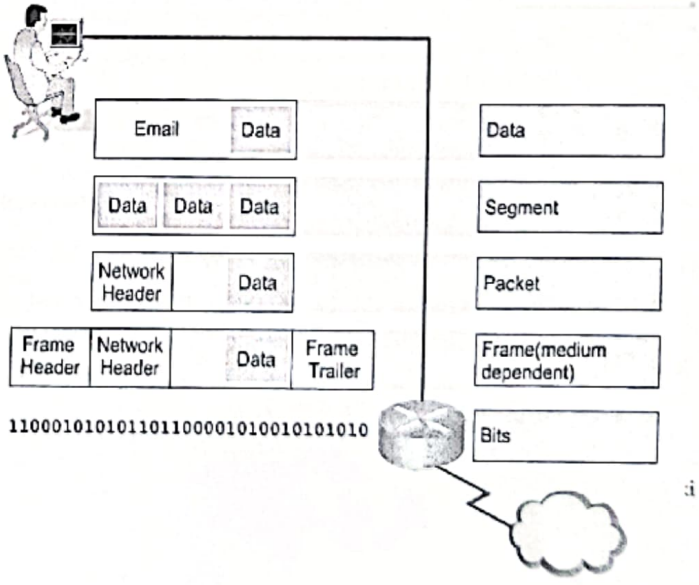
Encapsulation wraps data with the necessary protocol information before network transit. Therefore, as the data packet moves down through the layers of the OSI model, it receives headers, trailers, and other information.



Data Encapsulation example:

Perform the following five conversion steps in order to encapsulate the data.

1. Build the data.
2. Package the data for end-to-end transport.
3. Add the network IP address to the header.
4. Add the data link layer header and trailer.
5. Convert to bits for transmission.



Seven Layers of OSI Reference Model:

1. Physical Layer:

physical layer is the bottom layer of the OSI reference model. The physical layer has four improtant characteristics.

Mechanical. Relates to the physical properties of the interface to a transmission medium. Typically, the specification is of a pluggable connector that joins one or more signal conductors, called circuits.

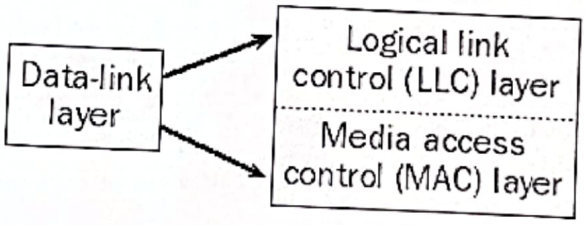
Electrical. Relates to the representation of bits (e.g., in terms of voltage levels) and the data transmission rate of bits. It defines the voltage, current, modulation, bit synchronization, connection activation and deactivation, and various electrical characteristics for the transmission media (such as unshielded or shielded twisted-pair cabling, coaxial cabling, and fiber-optic cabling).

Functional. Specifies the functions performed by individual circuits of the physical interface between a system and the transmission medium.

Procedural. Specifies the sequence of events by which bit streams are exchanged across the physical medium.

2. Data Link Layer:

The physical layer provides only a raw bit-stream service, the data link layer attempts to make the physical link reliable while providing the means to activate, maintain, and deactivate the link .
For LANs, the Project 802 standards of the Institute of Electrical and Electronics Engineers (IEEE) separate the data-link layer into two sublayers:



- The logical link control (LLC) layer, the upper of the two layers, which is responsible for flow control, error correction, and resequencing functions for connection-oriented communication, but which also supports connectionless communication
- The media access control (MAC) layer, the lower of the two layers, which is responsible for providing a method for stations to gain access to the medium

Functions:

Framing. The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

Physical addressing. If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.

Flow control. If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.

Error control. The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

Access control. When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time

Examples of data-link protocols for local area networking include the following:

- IEEE 802.3, which provides the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method for baseband Ethernet networks
- IEEE 802.5, which provides the token-passing access method for baseband token ring implementations

For WANs, data-link layer protocols encapsulate LAN traffic into frames suitable for transmission over WAN links. Common data-link encapsulation methods for WAN transmission include the following:

- Point-to-point technologies such as Point-to-Point Protocol (PPP) and High-level Data Link Control (HDLC) protocol
- Multipoint technologies such as frame relay, Asynchronous Transfer Mode (ATM), Switched Multimegabit Data Services (SMDS), and X.25

3. Network Layer:

Layer 3 of the Open Systems Interconnection (OSI) reference model for networking. The network layer is responsible for functions such as the following:

- Logical addressing and routing of packets over the network
- Establishing and releasing connections and paths between two nodes on a network
- Transferring data, generating and confirming receipts, and resetting connections

The network layer also supplies connectionless and connection-oriented services to the transport layer above it. The network layer functions closely with the physical layer (layer 1) and data-link layer (layer 2) in most real-world network protocol implementations.

On TCP/IP-based networks, IP addresses and network numbers are used at the network layer, and IP routers perform their routing functions at this layer. An example of an OSI model network layer protocol is the X.25 packet-switching network layer protocol, which is built on the X.21 physical layer protocol.

4. Transport Layer:

Layer 4 of the Open Systems Interconnection (OSI) reference model. The transport layer is responsible for providing reliable transport services to the upper-layer protocols. These services include the following:

- Flow control to ensure that the transmitting device does not send more data than the receiving device can handle.

- Packet sequencing for segmentation of data packets and remote reassembly.
- Error handling and acknowledgments to ensure that data is retransmitted when required.
- Multiplexing for combining data from several sources for transmission over one data path.
- Virtual circuits for establishing sessions between communicating stations.

The Transmission Control Protocol (TCP) of the TCP/IP protocol suite resides at the transport layer.

A connection between two devices that acts as though it's a direct connection even though it may physically be circuitous. The term is used most frequently to describe connections between two hosts in a packet-switching network. In this case, the two hosts can communicate as though they have a dedicated connection even though the packets might actually travel very different routes before arriving at their destination. An X.25 connection is an example of a virtual circuit.
Virtual circuits can be either permanent (called PVCs) or temporary (called SVCs).

5. Session Layer:

Layer 5 of the Open Systems Interconnection (OSI) reference model, which enables sessions between computers on a network to be established and terminated. The session layer does not concern itself with issues such as the reliability and efficiency of data transfer between stations because these functions are provided by the first four layers of the OSI reference model.

Functions:

Dialog control: The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half- duplex (one way at a time) or full-duplex (two ways at a time) mode.

Synchronization: The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need
not be resent.

6. Presentation Layer:

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems .

Specific responsibilities of the presentation layer include the following:

Translation. The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.

Encryption. To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.

Compression. Data compression reduces the number of bits contained in the information. Data compression

becomes particularly important in the transmission of multimedia such as text, audio, and video.

7. Application layer:

Layer 7 of the Open Systems Interconnection (OSI) reference model, in which network-aware, user-controlled software is implemented—for example, e-mail, file transfer utilities, and terminal access. The application layer represents the window between the user and the network. Examples of protocols that run at the application layer include File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), telnet, and similar protocols that can be implemented as utilities the user can interface with.

File transfer, access, and management. This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.

Mail services. This application provides the basis for e-mail forwarding and storage.

Directory services. This application provides distributed database sources and access for global information about various objects and services.

Summary:
Physical Layer: How to transmit bits.
Data Link Layer: How to transmits frames
Network: How to route packets to the node.
Transport: How to send packets to the applications.
Session: Manage connections.
Presentation: Encode/Decode messages, security.
Application: Everything else.

Devices and Protocols on each Layer:

Layer	Protocols	Devices
Physical		Hub, Repeater, Cables
Data-link	LLC,MAC,Ethernet	Switch
Network	IP,Routing protocol(RIP, OSPF)	Router
Transport	TCP,UDP	
Session		
Presentation	ASCII,Encryption, Decryption	
Application	DNS,NFS,TELNET,NFS	