OSI REFERENCE MODEL- INTRODUCTION

Network Software:

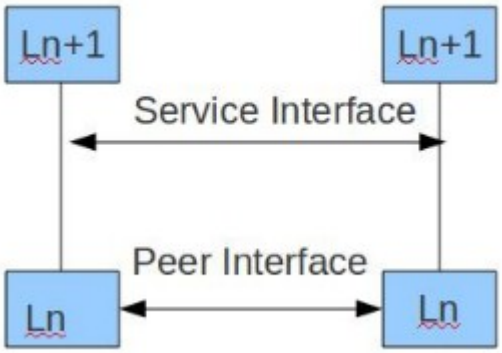
Network Software is a set of primitives that define the protocol between two machines. The network software resolves an ambiguity among different types of network making it possible for all the machines in the network to connect and communicate with one another and share information.

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.

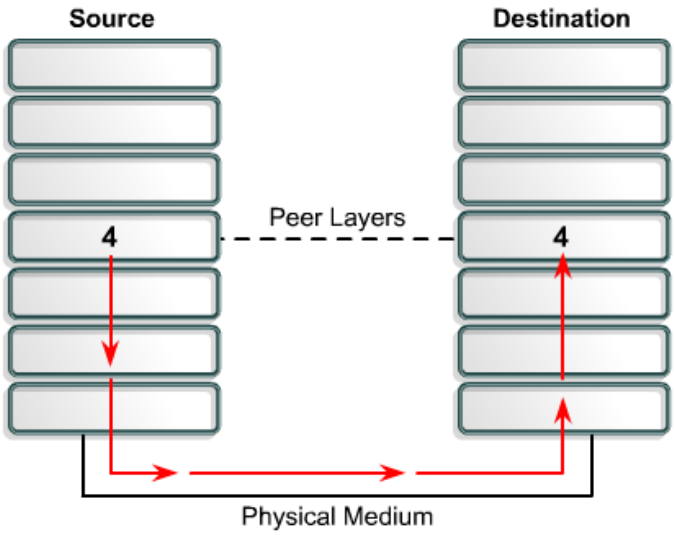
Layer n on one machine carries on a conversation with layer n on another machine. The rules and conventions used in this conversation are collectively known as the layer n protocol. Basically, a protocol is an agreement between the communicating parties on how communication is to proceed. As an analogy, 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:

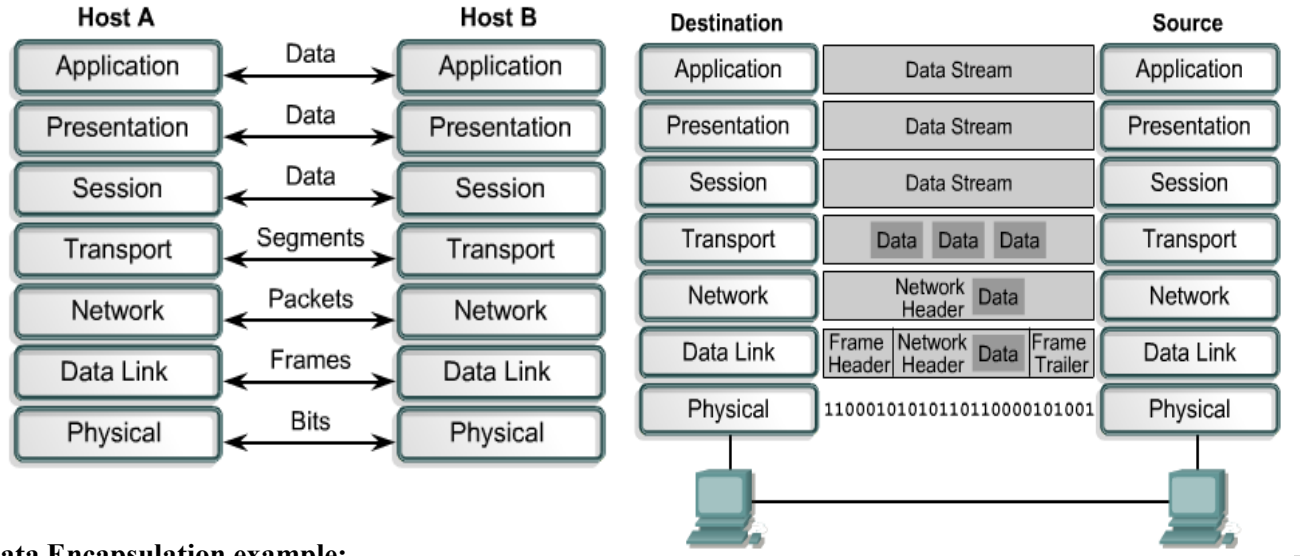
In order for data to travel from the source to the destination, each layer of the OSI model at the source must communicate with its peer layer at the destination. This form of communication is referred to as peer-to-peer. During this process, the protocols of each layer exchange information, called protocol 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.

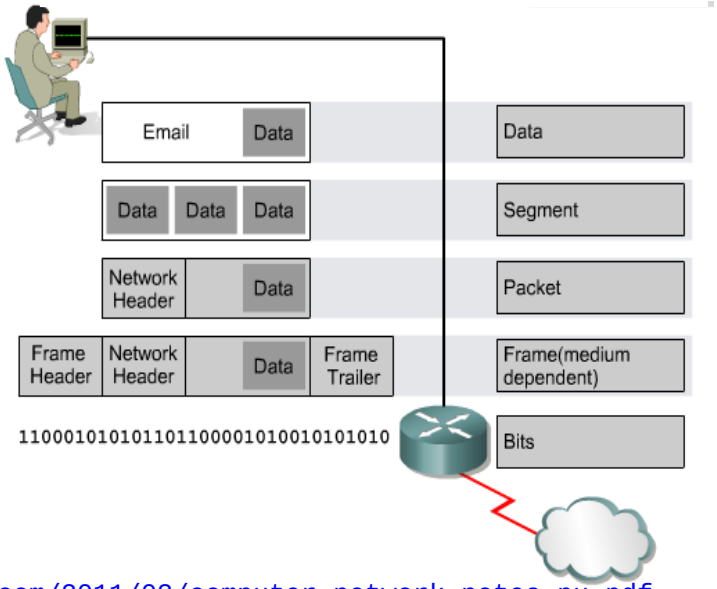
Encapsulation wraps data with the necessary protocol information before network transit. Therefore, as the data packet moves down through the layers of the OSI model, it receives headers, trailers, and other information.



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.



Source : http://dayaramb.files.wordpress.com/2011/03/computer-network-notes-pu.pdf