

1.3 Network Software

Network software in nowadays are highly structured, in the following sections; we examine the software structuring technique in some details.

1.3.1 Protocol hierarchies

In computing, a protocol is a set of rules which is used by computers to communicate with each other across a network. A protocol is a convention or standard that controls or enables the connection, communication, and data transfer between computing endpoints. In its simplest form, a protocol can be defined as the rules governing the syntax, semantics, and synchronization of communication. Protocols may be implemented by hardware, software, or a combination of the two.

To reduce network software design complex, most networks are organized as a series of "layers" or "levels", each one built upon the one below it. The number of layers, the name of each layer, and the function of each layer differ from network to network.

However, in all networks, the purpose of each layer is to offer a certain service to the higher layers, shielding those layers from the details of how the offered services are actually implemented.

Layer *n* in one machine carries on a conversation with layer *n* on another machine, the rules and conventions used in this conversation are collectively known as **layer *n* "protocol"**.

Protocol is an agreement between the communication parties on how communication is to proceed.

A five layer network is illustrated in fig (1-21)

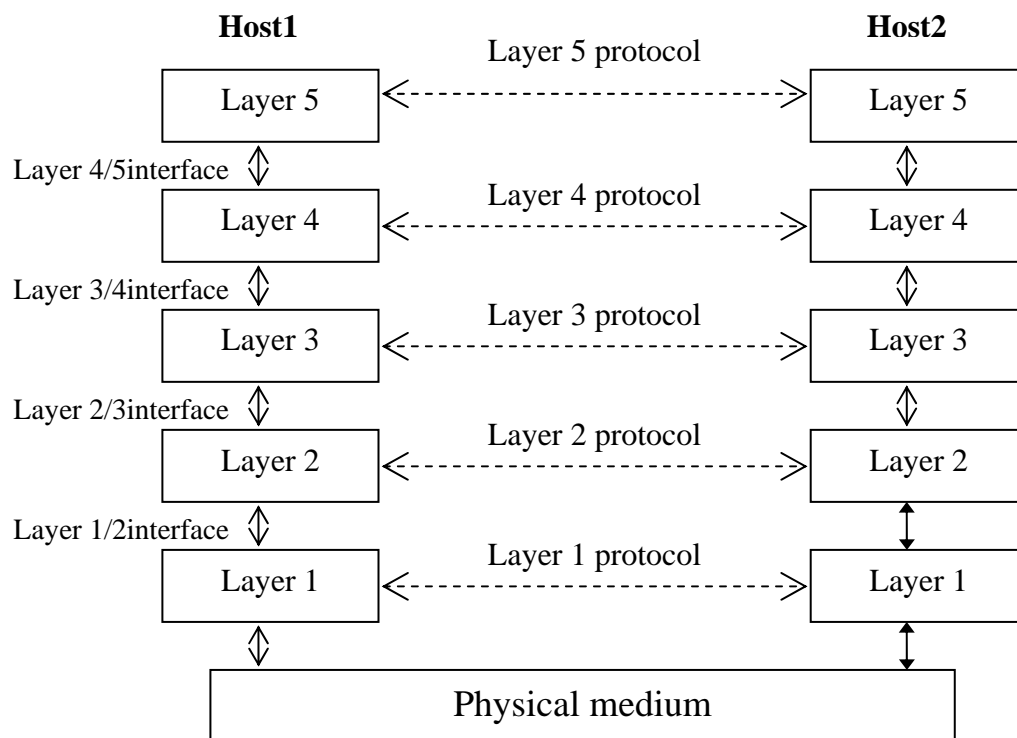


Fig (1-21) layers protocols and interfaces

The entities comprising the corresponding layers on different machines are called "**peers**". On other word it is the peers that communicate using the protocol.

In reality, no data are transferred directly from layer n on one machine to layer n on another machine. Instead; each layer passes data and control information to the layer below it, until the lowest layer is reached.

Below layer 1 is the "physical medium" through which the actual communication occurs.

Note: In fig (1-21) virtual communication is shown by dotted lines (refers to layer protocol) and physical communication by solid line (refer to layer interface).

Between each pair of adjacent layers there is an "interface", the interface defines which operations and services that the lower layer offers to upper layer through it.

A list of protocols used by certain system one protocol per layer is called a "**protocol stack**". Note that the lower layers of protocol hierarchy are frequently implemented by hardware and firmware.

Now, consider how to provide communication to the top layer of five-layer network in fig (1-22).

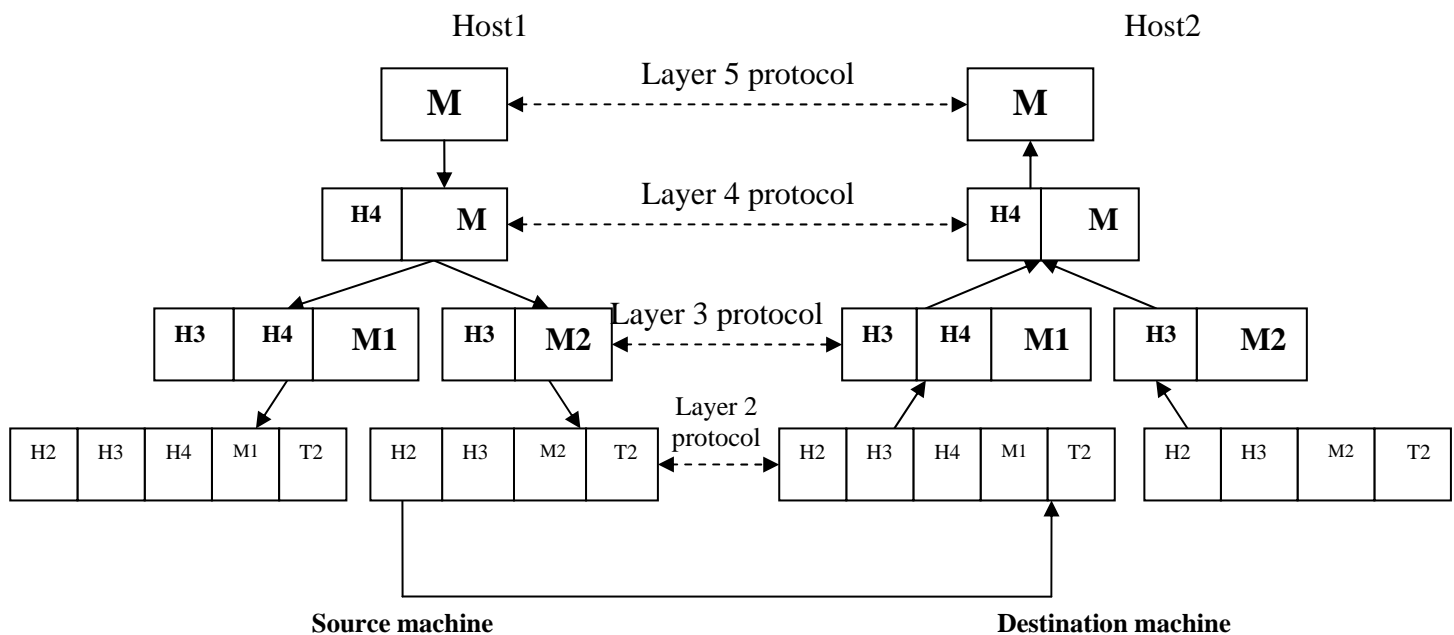


Fig (1-22) information flow supporting in 5-layer network

A message **M** is produced by an application process running in layer 5 and given to layer 4 for transmission.

Layer 4 puts a "**header**" in front of message **M** to identify the message and passes the result to layer 3. (the header contains control information such as sequence numbers, size, times and other control fields).

In many networks, there is no limit to the size of message in layer 4-protocol, but there is nearly always a limit imposed by layer 3 protocol. Hence, layer 3 must break up the incoming message into small units (packets) prepending layer 3 header to each packet. In this example the message **M** is split into two parts, **M1**, and **M2**. Layer 3 decides which of out

going lines to use and pass the packet to layer 2.

Layer 2 adds a header and trailer to each packet and gives the result to layer 1 for physical transmission. At receiving machine, the message move up word from layer to layer which header being stripped off as it progress till the message m arrives to application layer 5.

1.3.2 Protocol Data Unit and Encapsulation

As application data is passed down the protocol stack on its way to be transmitted across the network media, various protocols add information to it at each level. This is commonly known as the encapsulation process.

The form that a piece of data takes at any layer is called a Protocol Data Unit (PDU). During encapsulation, each succeeding layer encapsulates the PDU that it receives from the layer above in accordance with the protocol being used. At each stage of the process, a PDU has a different name to reflect its new appearance. Although there is no universal naming convention for PDUs, in this course, the PDUs are named according to the protocols of the TCP/IP suite.

- Data - The general term for the PDU used at the Application layer
- Segment - Transport Layer PDU
- Packet - Internetwork Layer PDU
- Frame - Network Access Layer PDU
- Bits - A PDU used when physically transmitting data over the medium

1.3.3 Design Issues for the layers

Some of design issues that occur in computer networking are present in several layers. Some of the more impotent ones are listed below:

1. Every layer needs mechanism for identifying senders and

receivers. Since a network normally has many computers, a means is needed for a process on one machine to specify with whom it want to talk. Example, network layer uses IP address for identifying sender and receiver.

2. The rules for data transfer must be specified. In some system data only traveled in one direction (simplex communication), in others they can travel in either directions but not simultaneously (half-duplex communication), or they can travel in both directions at the same time (full-duplex communication).
3. Error control is an important issue because physical communication circuits are not perfect .Thus, ends of the connection must agree with a certain error-detecting or error correcting code.
4. Not all communication channels preserve the order messages sent to them. To deal with possible loss of sequencing, the protocol must make explicit provision for receiver to allow the pieces of data to put back together properly.
5. An issue that occurs at every level is how to keep a fast sender form swamping a slow receiver with data. One possible solution is by limiting the sender to an agreed upon transmission rate.
6. The inability of all processes to accept arbitrarily long messages. This property leads to mechanisms for disassembling, transmitting and then reassembling messages. This is called fragmentation.
7. When there are multiple paths between source and destination, a route must be chosen. Some times this decision must be split over two or more layers.

1.3.4 Interfaces and services

The function of each layer is to provide service to the layer above it. The active element in each layer is often called "**entities**" which could be software entity or hardware entity. Entities in the same layer on different machines are called **peer entities**.

The entity on layer n implement a service used by layer $n+1$. In this case layer n called "service provider" and layer $n+1$ is called "service user". Layer n may use the service of layer $n-1$ in order to provide its services.

Services are available at SAPs (**Service Access Points**). The layer n SAPs are placed where layer $n+1$ can access the service offered. Each SAP has an address that uniquely identifies it.

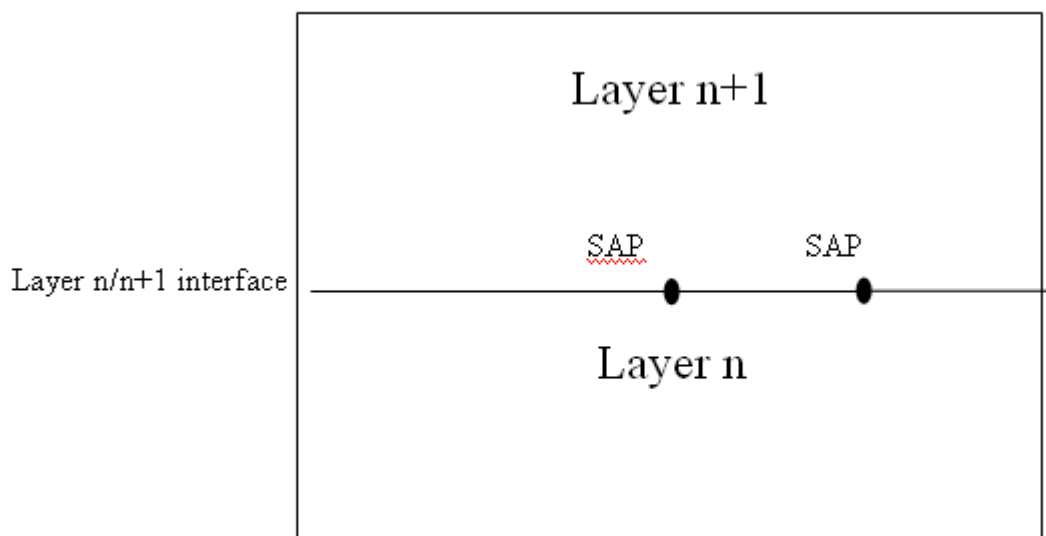


Fig (1-23) Relation between layers at an interface.

Layers can offers two different types of services to the layers a above them: connection-oriented and connection less.

* **Connection oriented service:** is implemented by service user first establish a connection, use the connection and then release the connection. Thus sender pushes objects (bits) in at one end, and the

receiver takes them out in the same order at other end (for example to a connection- oriented networks is the telephone system).

* **Connection less service:** in which each message carries the full destination address and each one is routed in the system independent of all the others. Hence, there is a possibility that the first one sent can be delayed so that the second one arrives first. This is impossible with a connection oriented service.

In connection less there is no need to establish connection first as the provider and destination always on line.

(For example to a connection less networks is ISDN network "Integrated Service Digital Network").

1.3.5 The Relationship of services to protocols

Service and protocol are distinct concepts.

A "service" is a set of primitive operations that the layer provides to the layer above it.

A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

A "protocol" is a set of rules governing the format and the meaning of the frames or messages that are exchanged by the peer entities within the layer. Entities use their protocols in order to implement their service definitions.