Module 1

Introduction

Lesson 2

Layered Network Architecture

Specific Functional Objectives

On Completion of this lesson, the students will be able to:

- State the requirement for layered approach
- Explain the basic concept of layering in the network model
- Define entities protocols in networking context
- Describe ISO's OSI Reference Model
- Explain information flow in OSI references Model.
- Explain functions of the seven layers of OSI Model

1.2.1 Basic concept of layering

Network architectures define the standards and techniques for designing and building communication systems for computers and other devices. In the past, vendors developed their own architectures and required that other vendors conform to this architecture if they wanted to develop compatible hardware and software. There are proprietary network architectures such as IBM's SNA (Systems Network Architecture) and there are open architectures like the OSI (Open Systems Interconnection) model defined by the International Organization for Standardization. The previous strategy, where the computer network is designed with the hardware as the main concern and software is afterthought, no longer works. Network software is now highly *structured.*

To reduce the design complexity, most of the networks are organized as a series of **layers** or **levels**, each one build upon one below it. The basic idea of a layered architecture is *to divide the design into small pieces*. Each layer adds to the services provided by the lower layers in such a manner that the highest layer is provided a full set of services to manage communications and run the applications. The benefits of the layered models are modularity and clear interfaces, i.e. open architecture and comparability between the different providers' components.

A basic principle is to ensure independence of layers by defining services provided by each layer to the next higher layer without defining how the services are to be performed. This permits changes in a layer without affecting other layers. Prior to the use of layered protocol architectures, simple changes such as adding one terminal type to the list of those supported by an architecture often required changes to essentially all communications software at a site. The number of layers, functions and contents of each layer differ from network to network. However in all networks, the purpose of each layer is to offer certain services to higher layers, shielding those layers from the details of how the services are actually implemented.

The basic elements of a layered model are services, protocols and interfaces. A *service* is a set of actions that a layer offers to another (higher) layer. *Protocol* is a set of rules that a layer uses to exchange information with a peer entity. These rules concern both the contents and the order of the messages used. Between the layers service interfaces are defined. The messages from one layer to another are sent through those interfaces.

In an n-layer architecture, layer n on one machine carries on conversation with the layer n on other 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. Violating the protocol will make communication more difficult, if not impossible. A five-layer architecture is shown in Fig. 1.2.1**,** the entities comprising the corresponding layers on different machines are called *peers*. In other words, it is the peers that communicate using protocols. In reality, no data is transferred from layer n on one machine to layer n of another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer-1 is the physical layer through which actual communication occurs. The peer process abstraction is crucial to all network design. Using it, the un-manageable tasks of designing the complete network can be broken into several smaller, manageable, design problems, namely design of individual layers.

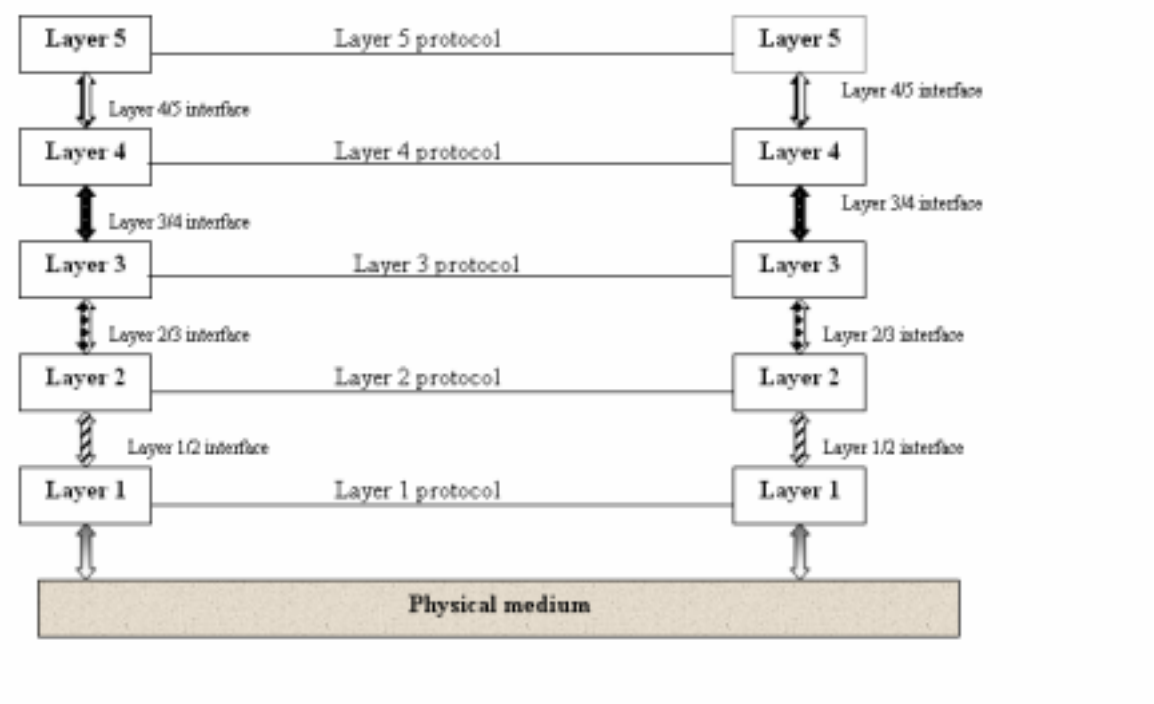

Figure 1.2.1 *Basic five layer architecture*

Between each pair of adjacent layers there is an **interface**. The *interface* defines which primitives operations and services the lower layer offers to the upper layer adjacent to it. When network designer decides how many layers to include in the network and what each layer should do, one of the main considerations is defining clean interfaces between adjacent layers. Doing so, in turns requires that each layer should perform well-defined functions. In addition to minimize the amount of information passed between layers, clean-cut interface also makes it simpler to replace the implementation of one layer with a completely different implementation, because all what is required of new implementation is that it offers same set of services to its upstairs neighbor as the old implementation

(that is what a layer provides and how to use that service from it is more important than knowing how exactly it implements it).

A set of layers and protocols is known as **network architecture**. The specification of architecture must contain enough information to allow an implementation to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol. Neither the details of implementation nor the specification of interface is a part of network architecture because these are hidden away inside machines and not visible from outside. It is not even necessary that the interface on all machines in a network be same, provided that each machine can correctly use all protocols. A list of protocols used by a certain system, one protocol per layer, is called **protocol stack**.

Summary: Why Layered architecture?

1. To make the design process easy by breaking unmanageable tasks into several smaller and manageable tasks (by divide-and-conquer approach).

2. Modularity and clear interfaces, so as to provide comparability between the different providers' components.

3. Ensure independence of layers, so that implementation of each layer can be changed or modified without affecting other layers.

4. Each layer can be analyzed and tested independently of all other layers.

1.2.2 Open System Interconnection Reference Model

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 inter-computer 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 OSI Reference Model includes seven layers:

7. *Application Layer*: Provides Applications with access to network services.

6. *Presentation Layer*: Determines the format used to exchange data among networked computers.

5. *Session Layer*: Allows two applications to establish, use and disconnect a connection between them called a session. Provides for name recognition and additional functions like security, which are needed to allow applications to communicate over the network.

4. *Transport Layer*: Ensures that data is delivered error free, in sequence and with no loss, duplications or corruption. This layer also repackages data by assembling long messages into lots of smaller messages for sending, and repackaging the smaller messages into the original larger message at the receiving end.

3. *Network Layer*: This is responsible for addressing messages and data so they are sent to the correct destination, and for translating logical addresses and names (like a machine name FLAME) into physical addresses. This layer is also responsible for finding a path through the network to the destination computer.

2. *Data-Link Layer*: This layer takes the data frames or messages from the Network Layer and provides for their actual transmission. At the receiving computer, this layer receives the incoming data and sends it to the network layer for handling. The Data-Link Layer also provides error-free delivery of data between the two computers by using the physical layer. It does this by packaging the data from the Network Layer into a frame, which includes error detection information. At the receiving computer, the Data-Link Layer reads the incoming frame, and generates its own error detection information based on the received frames data. After receiving the entire frame, it then compares its error detection value with that of the incoming frames, and if they match, the frame has been received correctly.

1. *Physical Layer*: Controls the transmission of the actual data onto the network cable. It defines the electrical signals, line states and encoding of the data and the connector types used. An example is 10BaseT.

1.2.2.1Characteristics of the OSI Layers

The seven layers of the OSI reference model can be divided into two categories: upper layers and lower layers as shown in Fig. 1.2.2.

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 .

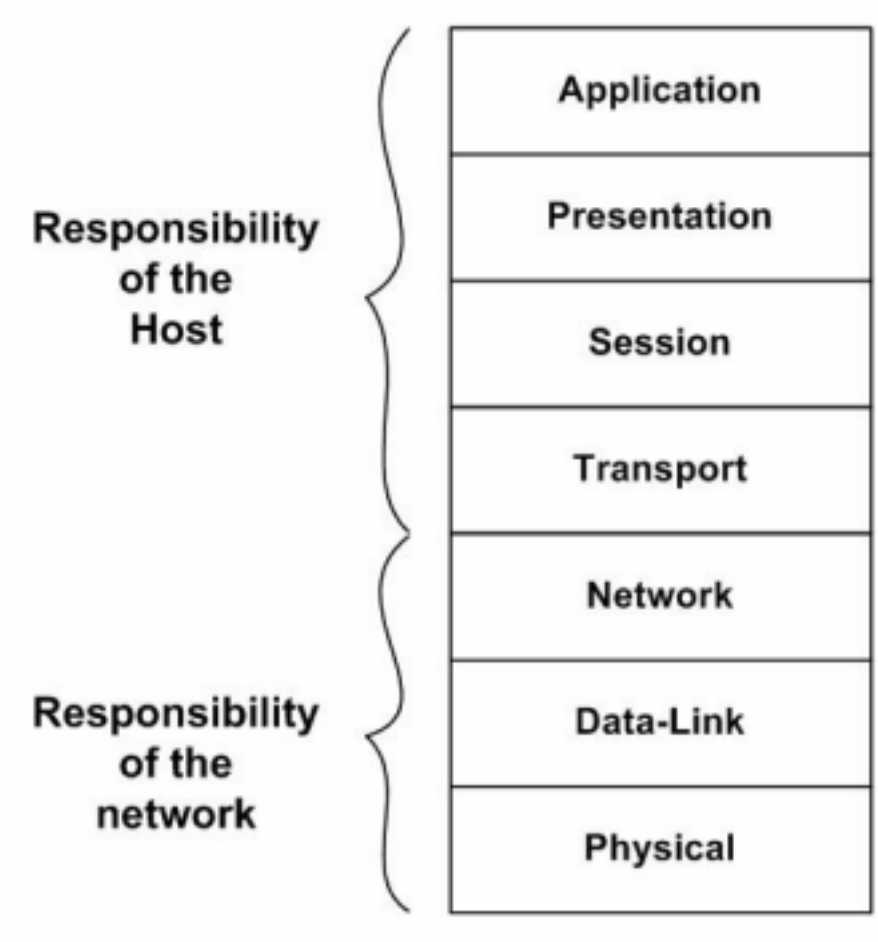

Figure 1.2.2 *Two sets of layers make up the OSI layers*

1.2.2.2 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 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.

1.2.2.3 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.

1.2.2.4 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. Figure1.2.3 illustrates this example.

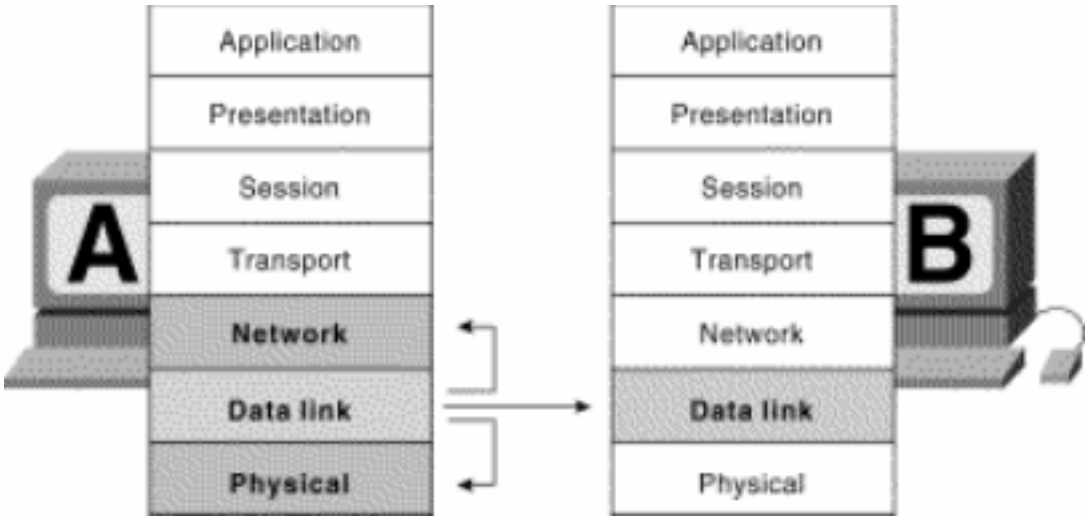

Figure 1.2.3 *OSI Model Layers Communicate with Other Layers*

1.2.3 Services and service access points

One OSI layer communicates with another layer to make use of the services provided by the second layer. The services provided by adjacent layers help a given OSI layer communicate with its peer layer in other computer systems. Three basic elements are involved in layer services: the service user, the service provider, and the service access point (SAP).

In this context, the service user is the OSI layer that requests services from an adjacent OSI layer. The service provider is the OSI layer that provides services to service users. OSI layers can provide services to multiple service users. The SAP is a conceptual location at which one OSI layer can request the services of another OSI layer.

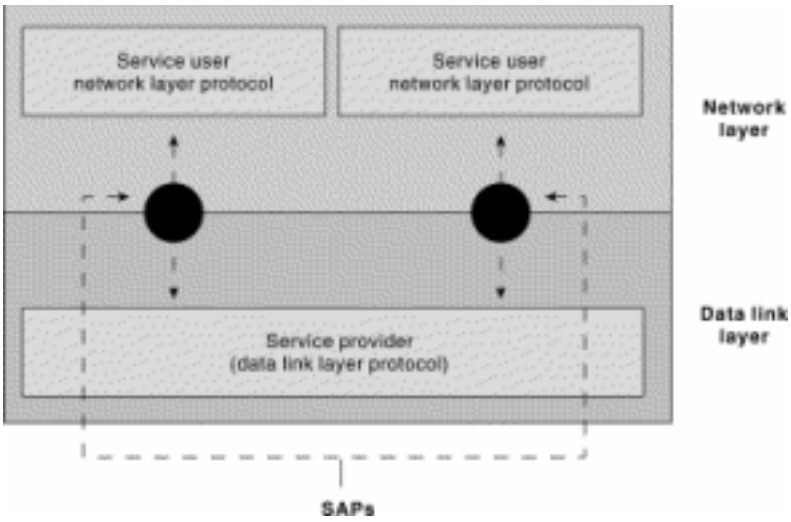

Figure 1.2.4 *Service Users, Providers, and SAPs interact at the Network and Data Link Layers*

1.2.3.1 OSI Model Layers and Information Exchange

The seven OSI layers use various forms of control information to communicate with their peer layers in other computer systems. This control information consists of specific requests and instructions that are exchanged between peer OSI layers.

Control information typically takes one of two forms: headers and trailers. Headers are prepended to data that has been passed down from upper layers. Trailers are appended to data that has been passed down from upper layers. An OSI layer is not required to attach a header or a trailer to data from upper layers.

Headers, trailers, and data are relative concepts, depending on the layer that analyzes the information unit. At the network layer, for example, an information unit consists of a

Layer 3 header and data. At the data link layer, however, all the information passed down by the network layer (the Layer 3 header and the data) is treated as data.

In other words, the data portion of an information unit at a given OSI layer potentially can contain headers, trailers, and data from all the higher layers. This is known as encapsulation. Figure 1-6 shows how the header and data from one layer are encapsulated into the header of the next lowest layer.

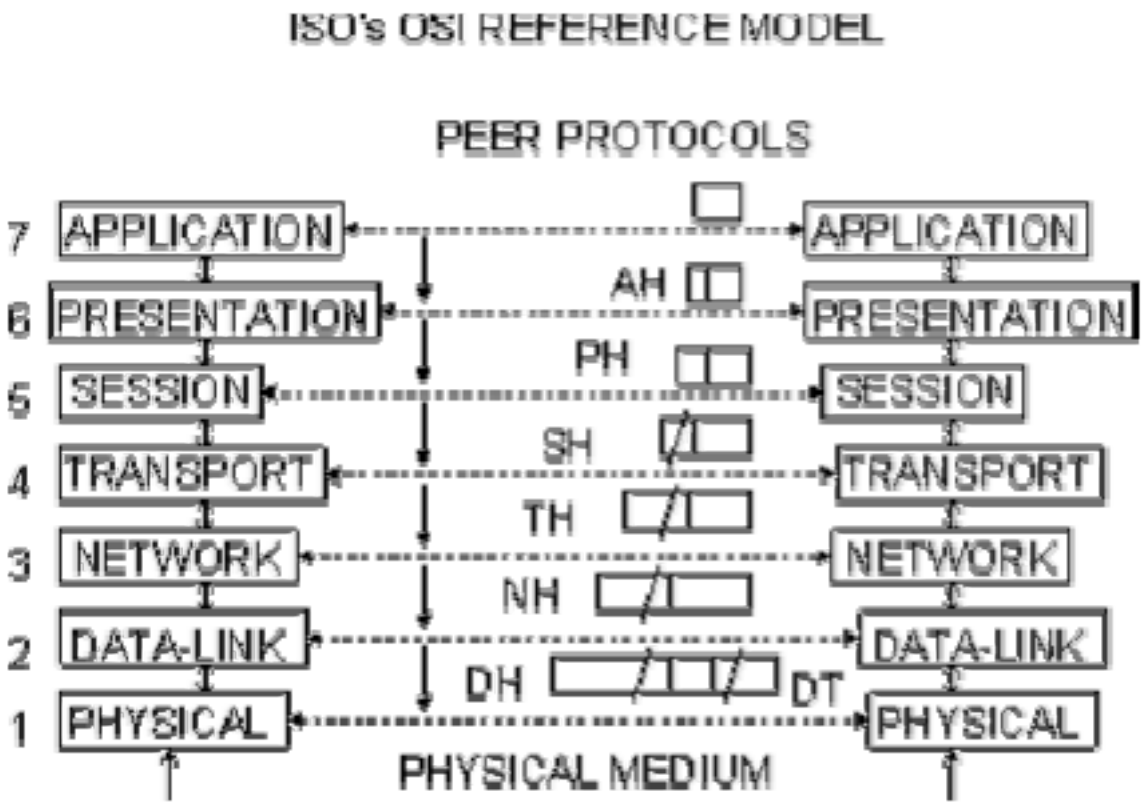

Figure 1.2.6 *Headers and Data can be encapsulated during Information exchange*

1.2.3.2 Information Exchange Process

The information exchange process occurs between peer OSI layers. Each layer in the source system adds control information to data, and each layer in the destination system analyzes and removes the control information from that data.

If system A has data from software application to send to System B, the data is passed to the application layer. The application layer in System A then communicates any control information required by the application layer in System B by pre-pending a header to the data. The resulting information unit (a header and the data) is passed to the presentation layer, which pre-pends its own header containing control information intended for the presentation layer in System B. The information unit grows in size as each layer pre pends its own header (and, in some cases, a trailer) that contains control information to be

used by its peer layer in System B. At the physical layer, the entire information unit is placed onto the network medium.

The physical layer in System B receives the information unit and passes it to the data link layer. The data link layer in System B then reads the control information contained in the

header pre-pended by the data link layer in System A. The header is then removed, and the remainder of the information unit is passed to the network layer. Each layer performs the same actions: The layer reads the header from its peer layer, strips it off, and passes the remaining information unit to the next highest layer. After the application layer performs these actions, the data is passed to the recipient software application in System B, in exactly the form in which it was transmitted by the application in System A.

1.2.4 Functions of the OSI Layers

Functions of different layers of the OSI model are presented in this section.

1.2.4.1 Physical Layer

The physical layer is concerned with transmission of raw bits over a communication channel. It specifies the mechanical, electrical and procedural network interface specifications and the physical transmission of bit streams over a transmission medium connecting two pieces of communication equipment. In simple terns, the physical layer decides the following:

- Number of pins and functions of each pin of the network connector (Mechanical) • Signal Level, Data rate (Electrical)
- Whether simultaneous transmission in both directions
- Establishing and breaking of connection
- Deals with physical transmission

There exist a variety of physical layer protocols such as RS-232C, Rs-449 standards developed by Electronics Industries Association (EIA).

1.2.4.2 Data Link Layer

The goal of the data link layer is to provide reliable, efficient communication between adjacent machines connected by a single communication channel. Specifically:

1. Group the physical layer bit stream into units called frames. Note that frames are nothing more than ``packets'' or ``messages''. By convention, we shall use the term ``frames'' when discussing DLL packets.

2. Sender calculates the checksum and sends checksum together with data. The checksum allows the receiver to determine when a frame has been damaged in transit or received correctly.

3. Receiver recomputes the checksum and compares it with the received value. If they differ, an error has occurred and the frame is discarded.

4. Error control protocol returns a positive or negative acknowledgment to the sender. A positive acknowledgment indicates the frame was received without errors, while a negative acknowledgment indicates the opposite.

5. Flow control prevents a fast sender from overwhelming a slower receiver. For example, a supercomputer can easily generate data faster than a PC can consume it.

6. In general, data link layer provides service to the network layer. The network layer wants to be able to send packets to its neighbors without worrying about the details of getting it there in one piece.

1.2.4.2.1 Design Issues

Below are the some of the important design issues of the data link layer:

a). Reliable Delivery:

Frames are delivered to the receiver reliably and in the same order as generated by the sender. Connection state keeps track of sending order and which frames require retransmission. For example, receiver state includes which frames have been received, which ones have not, etc.

b). Best Effort:

The receiver does not return acknowledgments to the sender, so the sender has no way of knowing if a frame has been successfully delivered.

When would such a service be appropriate?

1. When higher layers can recover from errors with little loss in performance. That is, when errors are so infrequent that there is little to be gained by the data link layer performing the recovery. It is just as easy to have higher layers deal with occasional loss of packet.

2. For real-time applications requiring ``better never than late" semantics. Old data may be worse than no data.

c). Acknowledged Delivery

The receiver returns an acknowledgment frame to the sender indicating that a data frame was properly received. This sits somewhere between the other two in that the sender keeps connection state, but may not necessarily retransmit unacknowledged frames. Likewise, the receiver may hand over received packets to higher layer in the order in

which they arrive, regardless of the original sending order. Typically, each frame is assigned a unique sequence number, which the receiver returns in an acknowledgment frame to indicate which frame the ACK refers to. The sender must retransmit unacknowledged (e.g., lost or damaged) frames.

d). Framing

The DLL translates the physical layer's raw bit stream into discrete units (messages) called *frames*. How can the receiver detect frame boundaries? Various techniques are used for this: Length Count, Bit Stuffing, and Character stuffing.

e). Error Control

Error control is concerned with insuring that all frames are eventually delivered (possibly in order) to a destination. To achieve this, three items are required: Acknowledgements, Timers, and Sequence Numbers.

f). Flow Control

Flow control deals with throttling the speed of the sender to match that of the receiver. Usually, this is a dynamic process, as the receiving speed depends on such changing factors as the load, and availability of buffer space.

1.2.4.2.2 Link Management

In some cases, the data link layer service must be ``opened" before use:

- The data link layer uses open operations for allocating buffer space, control blocks, agreeing on the maximum message size, etc.
- Synchronize and initialize send and receive sequence numbers with its peer at the other end of the communications channel.

1.2.4.2.3 Error Detection and Correction

In data communication, error may occur because of various reasons including attenuation, noise. Moreover, error usually occurs as bursts rather than independent, single bit errors. For example, a burst of lightning will affect a set of bits for a short time after the lightning strike. Detecting and correcting errors requires redundancy (i.e., sending additional information along with the data).

There are two types of attacks against errors:

• Error Detecting Codes: Include enough redundancy bits to detect errors and use ACKs and retransmissions to recover from the errors. Example: parity encoding. • Error Correcting Codes: Include enough redundancy to detect and correct errors. Examples: CRC checksum, MD5.

1.2.4.3 Network Layer

The basic purpose of the network layer is to provide an end-to-end communication capability in contrast to machine-to-machine communication provided by the data link layer. This end-to-end is performed using two basic approaches known as connection oriented or connectionless network-layer services.

1.2.4.3.1 Four issues:

1. Interface between the host and the network (the network layer is typically the boundary between the host and subnet)

2. Routing

3. Congestion and deadlock

4. Internetworking (A path may traverse different network technologies (e.g., Ethernet, point-to-point links, etc.)

1.2.4.3.2 Network Layer Interface

There are two basic approaches used for sending packets, which is a group of bits that includes data plus source and destination addresses, from node to node called *virtual circuit* and *datagram* methods. These are also referred to as *connection-oriented* and *connectionless* network-layer services. In virtual circuit approach, a *route*, which consists of logical connection, is first established between two users. During this establishment phase, the two users not only agree to set up a connection between them but also decide upon the quality of service to be associated with the connection. The well-known virtual circuit protocol is the ISO and CCITT *X.25* specification. The datagram is a self contained message unit, which contains sufficient information for routing from the source node to the destination node without dependence on previous message interchanges between them. In contrast to the virtual-circuit method, where a fixed path is explicitly set up before message transmission, sequentially transmitted messages can follow completely different paths. The datagram method is analogous to the postal system and the virtual-circuit method is analogous to the telephone system.

1.2.4.3.3 Overview of Other Network Layer Issues:

The network layer is responsible for routing packets from the source to destination. The *routing algorithm* is the piece of software that decides where a packet goes next (e.g., which output line, or which node on a broadcast channel).

For connectionless networks, the routing decision is made for each datagram. For connection-oriented networks, the decision is made once, at circuit setup time.

1.2.4.3.4 Routing Issues:

The routing algorithm must deal with the following issues:

- Correctness and simplicity: networks are never taken down; individual parts (e.g., links, routers) may fail, but the whole network should not.
- Stability: if a link or router fails, how much time elapses before the remaining routers recognize the topology change? (Some never do.)
- Fairness and optimality: an inherently intractable problem. Definition of optimality usually doesn't consider fairness. Do we want to maximize channel usage? Minimize average delay?

When we look at routing in detail, we'll consider both adaptive--those that take current traffic and topology into consideration--and non-adaptive algorithms.

1.2.4.3.4 Congestion

The network layer also must deal with congestion:

- When more packets enter an area than can be processed, delays increase and performance decreases. If the situation continues, the subnet may have no alternative but to discard packets.
- If the delay increases, the sender may (incorrectly) retransmit, making a bad situation even worse.
- Overall, performance degrades because the network is using (wasting) resources processing packets that eventually get discarded.

1.2.4.3.5 Internetworking

Finally, when we consider internetworking -- connecting different network technologies together -- one finds the same problems, only worse:

- Packets may travel through many different networks
- Each network may have a different frame format
- Some networks may be connectionless, other connection oriented

1.2.4.3.6 Routing

Routing is concerned with the question: Which line should router J use when forwarding a packet to router K?

There are two types of algorithms:

- **Adaptive algorithms** use such dynamic information as current topology, load, delay, etc. to select routes.
- In **non-adaptive algorithms**, routes never change once initial routes have been selected. Also called static routing.

Obviously, adaptive algorithms are more interesting, as non-adaptive algorithms don't even make an attempt to handle failed links.

1.2.4.4 Transport Layer

The transport level provides end-to-end communication between processes executing on different machines. Although the services provided by a transport protocol are similar to those provided by a data link layer protocol, there are several important differences between the transport and lower layers:

1. User Oriented. Application programmers interact directly with the transport layer, and from the programmers perspective, the transport layer is the ``network''. Thus, the transport layer should be oriented more towards user services than simply reflect what the underlying layers happen to provide. (Similar to the beautification principle in operating systems.)

2. Negotiation of Quality and Type of Services. The user and transport protocol may need to negotiate as to the quality or type of service to be provided. Examples? A user may want to negotiate such options as: throughput, delay, protection, priority, reliability, etc.

3. Guarantee Service. The transport layer may have to overcome service deficiencies of the lower layers (e.g. providing reliable service over an unreliable network layer).

4. Addressing becomes a significant issue. That is, now the user must deal with it; before it was buried in lower levels.

Two solutions:

- Use well-known addresses that rarely if ever change, allowing programs to ``wire in'' addresses. For what types of service does this work? While this works for services that are well established (e.g., mail, or telnet), it doesn't allow a user to easily experiment with new services.

- Use a name server. Servers register services with the name server, which clients contact to find the transport address of a given service.

In both cases, we need a mechanism for mapping high-level service names into low-level encoding that can be used within packet headers of the network protocols. In its general

form, the problem is quite complex. One simplification is to break the problem into two parts: have transport addresses be a combination of machine address and local process on that machine.

***5. Storage capacity* of the subnet.** Assumptions valid at the data link layer do not necessarily hold at the transport Layer. Specifically, the subnet may buffer messages for a potentially long time, and an ``old'' packet may arrive at a destination at unexpected times.

6. We need a dynamic flow control mechanism. The data link layer solution of reallocating buffers is inappropriate because a machine may have hundreds of connections sharing a single physical link. In addition, appropriate settings for the flow control parameters depend on the communicating end points (e.g., Cray supercomputers vs. PCs), not on the protocol used.

Don't send data unless there is room. Also, the network layer/data link layer solution of simply not acknowledging frames for which the receiver has no space is unacceptable. Why? In the data link case, the line is not being used for anything else; thus retransmissions are inexpensive. At the transport level, end-to-end retransmissions are needed, which wastes resources by sending the same packet over the same links multiple times. If the receiver has no buffer space, the sender should be prevented from sending data.

7. *Deal with congestion control.* In connectionless Internets, transport protocols must exercise congestion control. When the network becomes congested, they must reduce rate at which they insert packets into the subnet, because the subnet has no way to prevent itself from becoming overloaded.

8. *Connection establishment.* Transport level protocols go through three phases: establishing, using, and terminating a connection. For data gram-oriented protocols, opening a connection simply allocates and initializes data structures in the operating system kernel.

Connection oriented protocols often exchanges messages that negotiate options with the remote peer at the time a connection are opened. Establishing a connection may be tricky because of the possibility of old or duplicate packets.

Finally, although not as difficult as establishing a connection, terminating a connection

presents subtleties too. For instance, both ends of the connection must be sure that all the data in their queues have been delivered to the remote application.

1.2.4.5 Session Layer

This layer allows users on different machines to establish session between them. A session allows ordinary data transport but it also provides enhanced services useful in some applications. A session may be used to allow a user to log into a remote time

sharing machine or to transfer a file between two machines. Some of the session related services are:

1. This layer manages *Dialogue Control*. Session can allow traffic to go in both direction at the same time, or in only one direction at one time.

2. *Token management*. For some protocols, it is required that both sides don't attempt same operation at the same time. To manage these activities, the session layer provides tokens that can be exchanged. Only one side that is holding token can perform the critical operation. This concept can be seen as entering into a critical section in operating system using semaphores.

3. *Synchronization*. Consider the problem that might occur when trying to transfer a 4-hour file transfer with a 2-hour mean time between crashes. After each transfer was aborted, the whole transfer has to start again and again would probably fail. To Eliminate this problem, Session layer provides a way to insert checkpoints into data streams, so that after a crash, only the data transferred after the last checkpoint have to be repeated.

1.2.4.6 Presentation Layer

This layer is concerned with Syntax and Semantics of the information transmitted, unlike other layers, which are interested in moving data reliably from one machine to other. Few of the services that Presentation layer provides are:

1. Encoding data in a standard agreed upon way.

2. It manages the abstract data structures and converts from representation used inside computer to network standard representation and back.

1.2.4.7 Application Layer

The application layer consists of what most users think of as programs. The application does the actual work at hand. Although each application is different, some applications are so useful that they have become standardized. The Internet has defined standards for:

- File transfer (FTP): Connect to a remote machine and send or fetch an arbitrary

file. FTP deals with authentication, listing a directory contents, ASCII or binary files, etc.

- Remote login (telnet): A remote terminal protocol that allows a user at one site to establish a TCP connection to another site, and then pass keystrokes from the local host to the remote host.
- Mail (SMTP): Allow a mail delivery agent on a local machine to connect to a mail delivery agent on a remote machine and deliver mail.
- News (NNTP): Allows communication between a news server and a news client. • Web (HTTP): Base protocol for communication on the World Wide Web.

Review questions

Q-1. Why it is necessary to have layering in a network?

Ans: A computer network is a very complex system. It becomes very difficult to implement as a single entity. The layered approach divides a very complex task into small pieces each of which is independent of others and it allow a structured approach in implementing a network. The basic idea of a layered architecture is *to divide the design into small pieces*. Each layer adds to the services provided by the lower layers in such a manner that the highest layer is provided a full set of services to manage communications and run the applications.

Q-2. What are the key benefits of layered network?

Ans: Main benefits of layered network are given below:
i) Complex systems can be broken down into understandable subsystems. ii) Any facility implemented in one layer can be made visible to all other layers. iii) Services offered at a particular level may share the services of lower level. iv) Each layer may be analyzed and tested independently.
v) Layers can be simplified, extended or deleted at any time.
vi) Increase the interoperability and compatibility of various components build by different vendors.

Q-3. What do you mean by OSI?

Ans: 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 Standardization Organization (ISO) in 1984, and it is now considered the primary architectural model for inter-computer communications.

Q-4. What are the seven layers of ISO's OSI model?

Ans:- The seven layers are:
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

Q-5. Briefly write functionalities of different OSI layers?

Ans: The OSI Reference Model includes seven layers. Basic functionality of each of them is as follows:

7. ***Application Layer:*** Provides Applications with access to network services.

6. ***Presentation Layer:***
Determines the format used to exchange data among networked computers.

5. ***Session Layer:*** Allows two applications to establish, use and disconnect a connection between them called a session. Provides for name recognition and additional functions like security, which are needed to allow applications to communicate over the network.

4. ***Transport Layer:*** Ensures that data is delivered error free, in sequence and with no loss, duplications or corruption. This layer also repackages data by assembling long messages into lots of smaller messages for sending, and repackaging the smaller messages into the original larger message at the receiving end.

3. ***Network Layer:*** This is responsible for addressing messages and data so they are sent to the correct destination, and for translating logical addresses and names (like a machine name FLAME) into physical addresses. This layer is also responsible for finding a path through the network to the destination computer.

2. ***Data-Link Layer:*** This layer takes the data frames or messages from the Network Layer and provides for their actual transmission. At the receiving computer, this layer receives the incoming data and sends it to the network layer for handling. The Data-Link Layer also provides error-free delivery of data between the two computers by using the physical layer. It does this by packaging the data from the Network Layer into a frame, which includes error detection information. At the receiving computer, the Data-Link Layer reads the incoming frame, and generates its own error detection information based on the received frames data. After receiving the entire frame, it then compares its error

detection value with that of the incoming frames, and if they match, the frame has been received correctly.

1. *Physical Layer*: Controls the transmission of the actual data onto the network cable. It defines the electrical signals, line states and encoding of the data and the connector types used. An example is 10BaseT.

Q-6. How two adjacent layers communicate in a layered network? (or What do you mean by Service Access Point?)

Ans: In layered network, each layer has various entities and entities of layer i provide service to the entities of layer i+1. The services can be accessed through service access

point (SAP), which has some address through which the layer i+1 will access the services provided by layer i.

Q-7. What are the key functions of data link layer?

Ans: Data link layer transfers data in a structured and reliable manner so that the service provided by the physical layer is utilized by data link layer. Main function of data link layer is framing and media access control.

Q8. What do you mean by Protocol?

Ans: 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.

Version 2 CSE IIT, Kharagpur