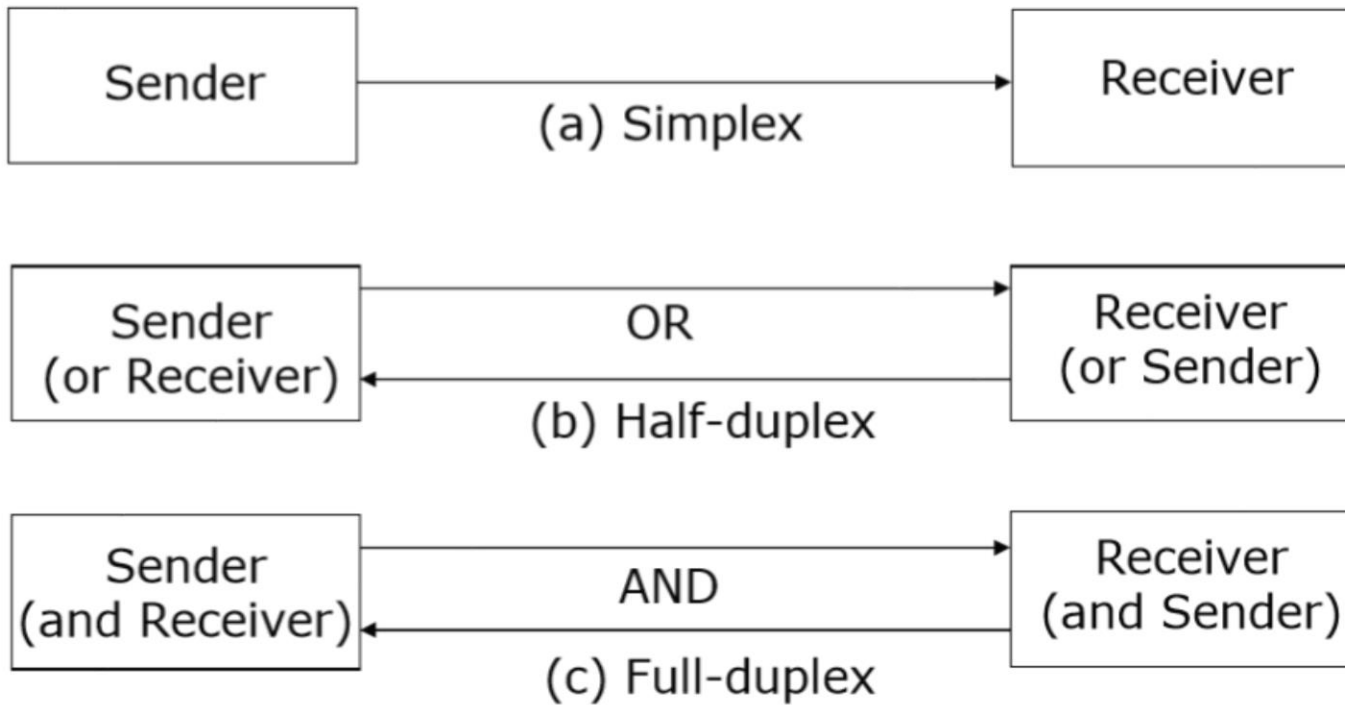




UNIT 1 : INTRODUCTION TO NETWORK

DATA TRANSMISSION MODES



NETWORK SOFTWARE - PROTOCOL HIERARCHY

- 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.
- A protocol is an agreement between the communicating parties on how communication is to proceed.

A five-layer network is in **Fig.**

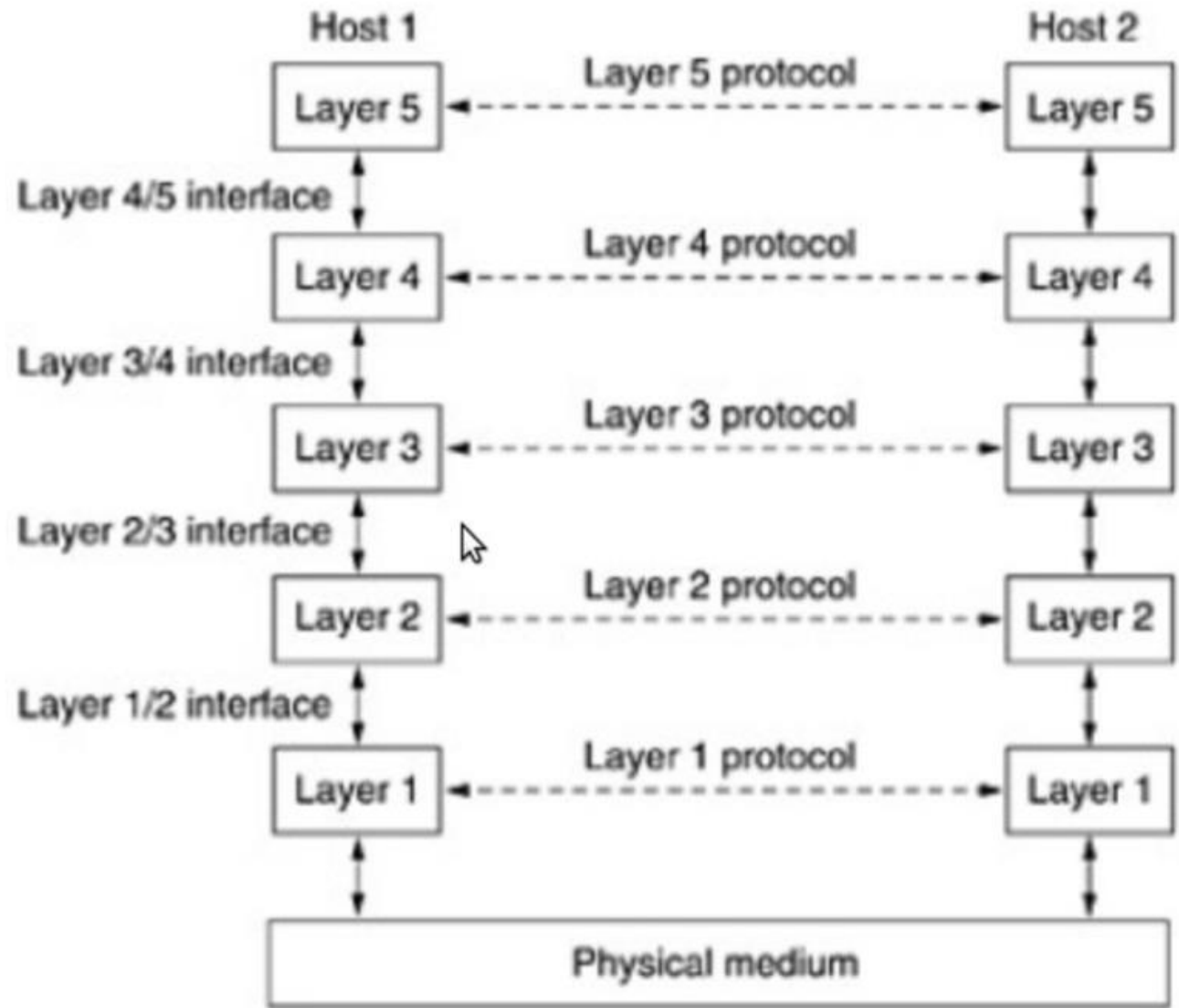
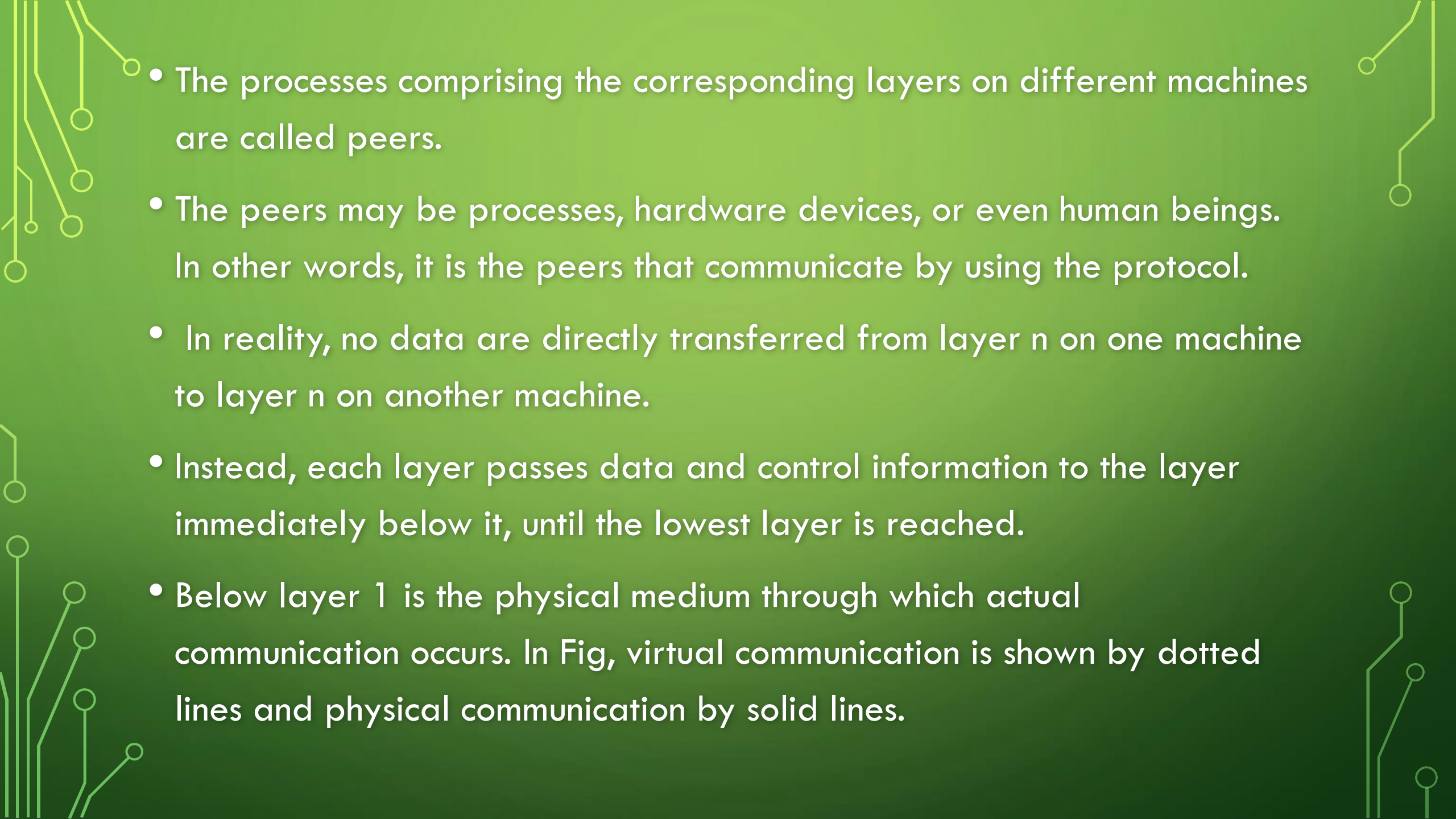


Figure: Layers, protocols, and interfaces.

- 
- The processes comprising the corresponding layers on different machines are called peers.
 - The peers may be processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol.
 - In reality, no data are directly transferred from layer n on one machine to layer n on 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 medium through which actual communication occurs. In Fig, virtual communication is shown by dotted lines and physical communication by solid lines.

INTERFACE

1. Adjacent-layer interaction on the same computer - On a single computer, one layer provides a service to a higher layer. The software or hardware that implements the higher layer requests that the next lower layer perform the needed function.

2. Same-layer interaction on different computers-The two computers use a protocol to communicate with the same layer on another computer. The protocol defined by each layer uses a header that is transmitted between the computers, to communicate what each computer wants to do.

- A set of layers and protocols is called a Network Architecture. The specification of an architecture must contain enough information to allow an implementer to write the program or build the hardware for each layer so that it will correctly obey the appropriate protocol.
- The details of the implementation nor the specification of the interfaces is part of the architecture because these are hidden away inside the machines and not visible from the outside. It is not even necessary that the interfaces on all machines in a network be the same, provided that each machine can correctly use all the protocols. A list of protocols used by a certain system, one protocol per layer, is called a protocol stack.

How to provide communication to the top layer of the five-layer Network : in **Fig.**

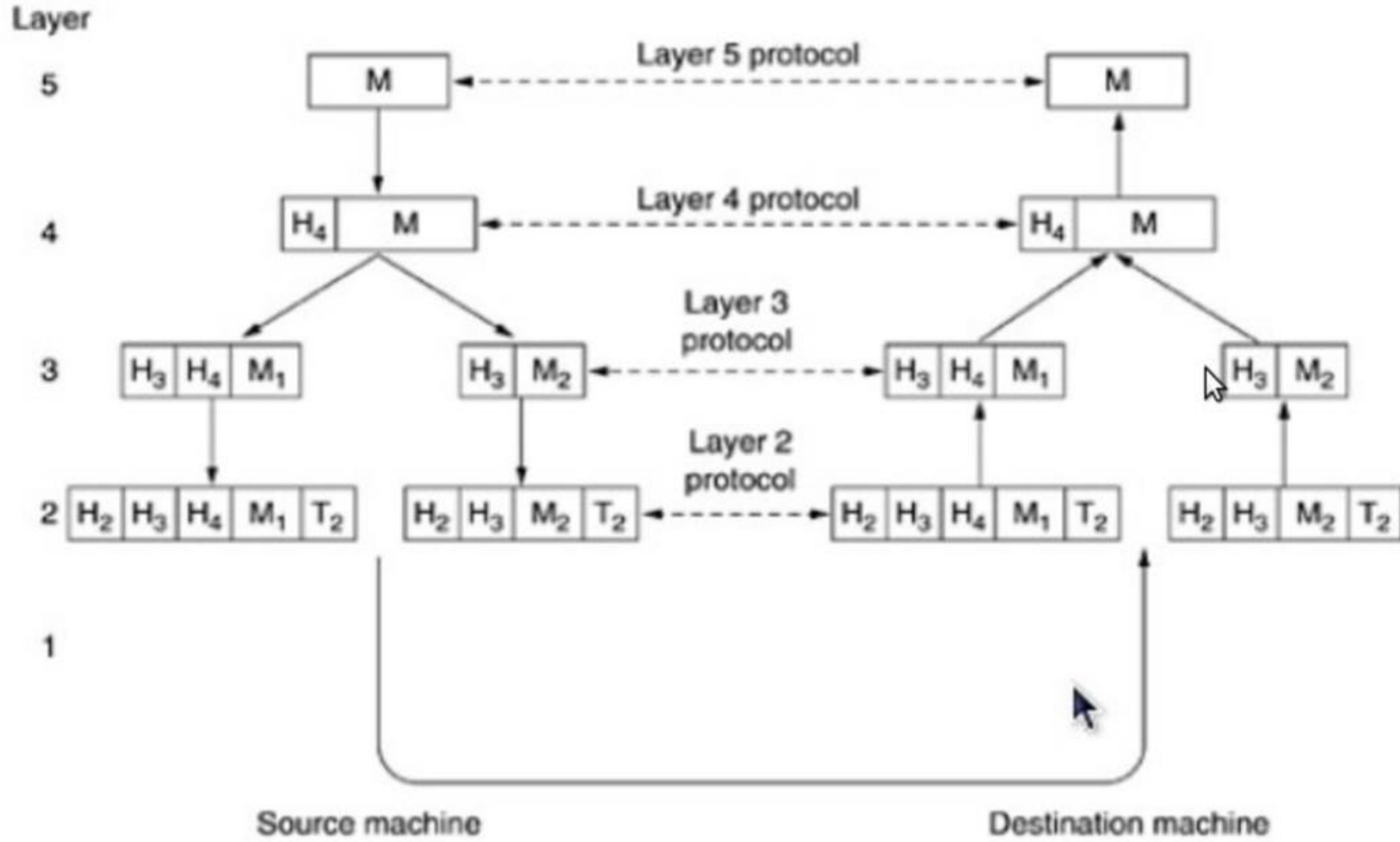


Figure: Information flow supporting virtual communication in layer 5.

- 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 the message to identify the message and passes the result to layer 3. The header includes control information, such as sequence numbers, to allow layer 4 on the destination machine to deliver messages in the right order if the lower layers do not maintain sequence.
- In some layers, headers can also contain sizes, times, and other control fields. layer 3 must break up the incoming messages into smaller units, packets, prepending a layer 3 header to each packet.
- In this example, M is split into two parts, $M1$ and $M2$. Layer 3 decides which of the outgoing lines to use and passes the packets to layer 2.

- Layer 2 adds not only a header to each piece, but also a trailer, and gives the resulting unit to layer 1 for physical transmission.
- At the receiving machine the message moves upward, from layer to layer, with headers being stripped off as it progresses. The important thing to understand about Fig. is the relation between the virtual and actual communication and the difference between protocols and interfaces.

DESIGN ISSUES FOR THE LAYERS OF COMPUTER NETWORKS

- A number of design issues exist for the layer to layer approach of computer networks. Some of the main design issues are as follows:

Reliability

- Network channels and components may be unreliable, resulting in loss of bits while data transfer. So, an important design issue is to make sure that the information transferred is not distorted.

Scalability

- Networks are continuously evolving. The sizes are continually increasing leading to congestion. Also, when new technologies are applied to the added components, it may lead to incompatibility issues. Hence, the design should be done so that the networks are scalable and can accommodate such additions and alterations.

Addressing

- At a particular time, innumerable messages are being transferred between large numbers of computers. So, a naming or addressing system should exist so that each layer can identify the sender and receivers of each message.

Error Control

- Unreliable channels introduce a number of errors in the data streams that are communicated. So, the layers need to agree upon common error detection and error correction methods so as to protect data packets while they are transferred.

Flow Control

- If the rate at which data is produced by the sender is higher than the rate at which data is received by the receiver, there are chances of overflowing the receiver. So, a proper flow control mechanism needs to be implemented.

Resource Allocation

- Computer networks provide services in the form of network resources to the end users. The main design issue is to allocate and deallocate resources to processes. The allocation/deallocation should occur so that minimal interference among the hosts occurs and there is optimal usage of the resources.

Statistical Multiplexing

- It is not feasible to allocate a dedicated path for each message while it is being transferred from the source to the destination. So, the data channel needs to be multiplexed, so as to allocate a fraction of the bandwidth or time to each host.

Routing

- There may be multiple paths from the source to the destination. Routing involves choosing an optimal path among all possible paths, in terms of cost and time. There are several routing algorithms that are used in network systems.

Security

- A major factor of data communication is to defend it against threats like eavesdropping and surreptitious alteration of messages. So, there should be adequate mechanisms to prevent unauthorized access to data through authentication and cryptography.

CONNECTION ORIENTED AND CONNECTIONLESS SERVICES

- These are the two services given by the layers to layers above them. These services are:

Connection Oriented Service

Connectionless Services

○ Connection Oriented Services

- There is a sequence of operation to be followed by the users of connection oriented service. These are:
 - ❖ Connection is established.
 - ❖ Information is sent.
 - ❖ Connection is released.
- In connection oriented service we have to establish a connection before starting the communication. When connection is established, we send the message or the information and then we release the connection.
- Connection oriented service is more reliable than connectionless service. We can send the message in connection oriented service if there is an error at the receivers end. Example of connection oriented is TCP (Transmission Control Protocol) protocol.

CONNECTION LESS SERVICES

- It is similar to the postal services, as it carries the full address where the message (letter) is to be carried. Each message is routed independently from source to destination. The order of message sent can be different from the order received.
- In connectionless the data is transferred in one direction from source to destination without checking that destination is still there or not or if it prepared to accept the message. Authentication is not needed in this. Example of Connectionless service is UDP (User Datagram Protocol) protocol.

DIFFERENCE: CONNECTION ORIENTED AND CONNECTIONLESS SERVICE

- In connection oriented service authentication is needed, while connectionless service does not need any authentication.
- Connection oriented protocol makes a connection and checks whether message is received or not and sends again if an error occurs, while connectionless service protocol does not guarantees a message delivery.
- Connection oriented service is more reliable than connectionless service.
- Connection oriented service interface is stream based and connectionless is message based.

WHAT ARE SERVICE PRIMITIVES?

- A service is formally specified by a set of primitives (operations) available to a user process to access the service. These primitives tell the service to perform some action or report on an action taken by a peer entity. If the protocol stack is located in the operating system, as it often is, the primitives are normally system calls. These calls cause a trap to kernel mode, which then turns control of the machine over to the operating system to send the necessary packets. The set of primitives available depends on the nature of the service being provided. The primitives for connection-oriented service are different from those of connection-less service. There are five types of service primitives :

- **LISTEN** : When a server is ready to accept an incoming connection it executes the LISTEN primitive. It blocks waiting for an incoming connection.
- **CONNECT** : It connects the server by establishing a connection. Response is awaited.
- **RECIEVE**: Then the RECIEVE call blocks the server.
- **SEND** : Then the client executes SEND primitive to transmit its request followed by the execution of RECIEVE to get the reply. Send the message.
- **DISCONNECT** : This primitive is used for terminating the connection. After this primitive one can't send any message. When the client sends DISCONNECT packet then the server also sends the DISCONNECT packet to acknowledge the client. When the server package is received by client then the process is terminated.

CONNECTION ORIENTED SERVICE PRIMITIVES

There are 5 types of primitives for Connection Oriented Service :

LISTEN	Block waiting for an incoming connection
CONNECTION	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Sending a message to the peer
DISCONNECT	Terminate a connection

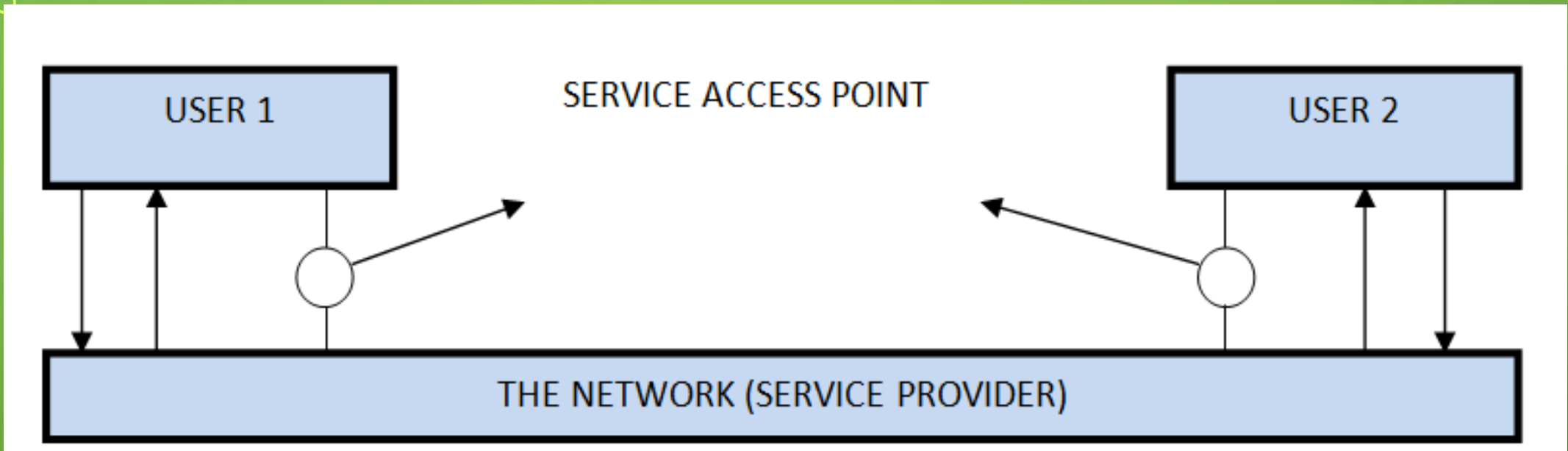
CONNECTIONLESS SERVICE PRIMITIVES

THERE ARE 4 TYPES OF PRIMITIVES FOR CONNECTIONLESS ORIENTED SERVICE:

UNIDATA	This primitive sends a packet of data
FACILITY, REPORT	Primitive for enquiring about the performance of the network, like delivery statistics.

RELATIONSHIP OF SERVICES TO PROTOCOL

- In this section we will learn about how services and protocols are related and why they are so important for each other.
-
- What are Services?
- These are the operations that a layer can provide to the layer above it in the OSI Reference model. It defines the operation and states a layer is ready to perform but it does not specify anything about the implementation of these operations.



PROTOCOLS

These are set of rules that govern the format and meaning of frames, messages or packets that are exchanged between the server and client.

NETWORK MODELS

- The most important reference models are:
- OSI reference model.
- TCP/IP reference model.

INTRODUCTION TO ISO-OSI MODEL

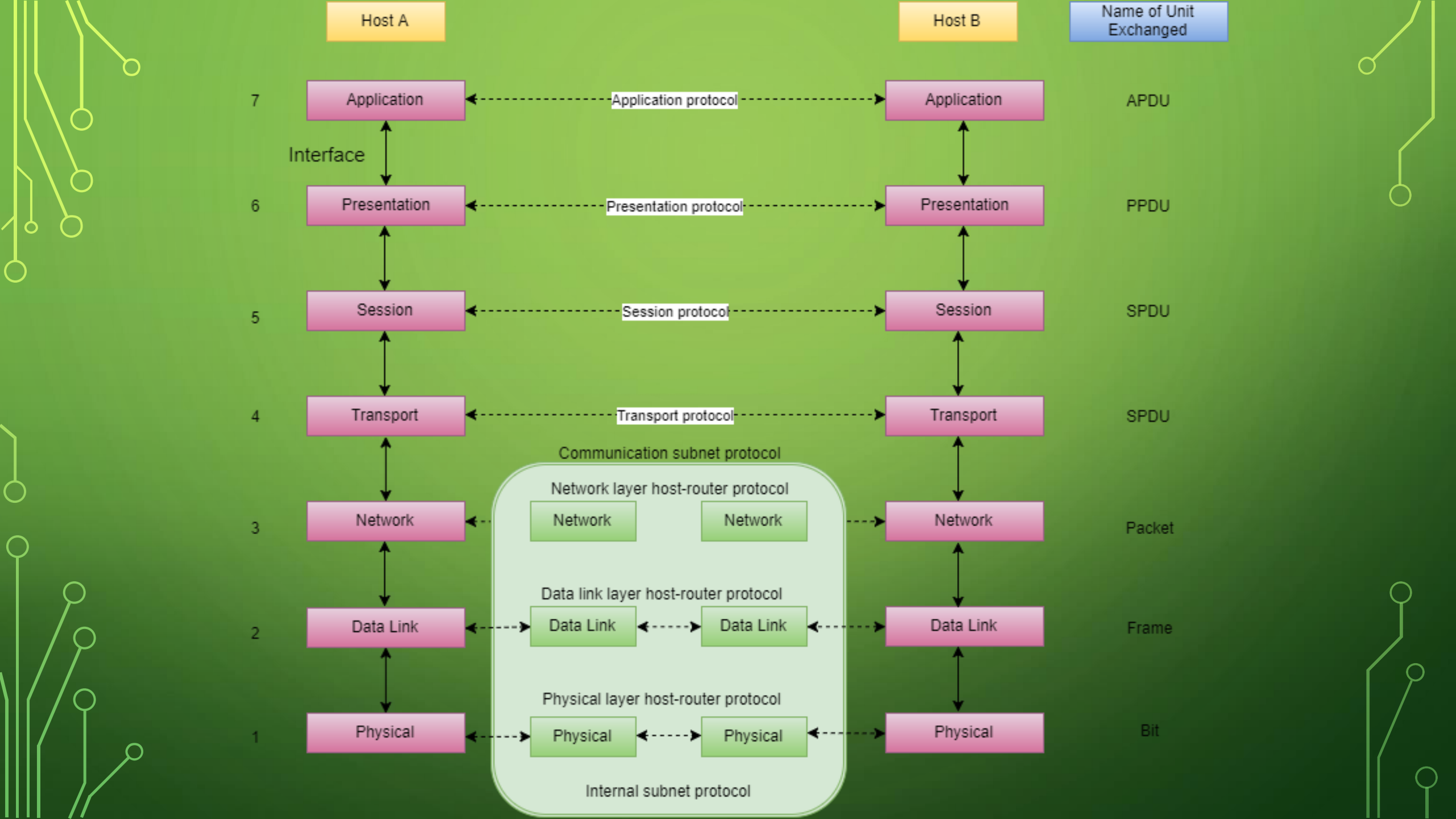
International Organization for Standardization (IOS) In 1984, the ISO released a revision of this model and called it the Open Systems Interconnection (OSI) reference model. The 1984 revision has become an international standard and serves as a guide for networking. The concept of a seven layer model was provided by the work of **Charles Bachman**.

The OSI reference model represents the seven layers of the process by which data is packaged and transmitted from a sending application through the physical wires to the receiving application. Each Specifying particular network functions. OSI defines a large collection of protocols that allow computers to communicate.

OSI Model is a set of protocols that define standardize Data commutation process.

The lower 4 layers (transport, network, data link and physical —Layers 4, 3, 2, and 1) are concerned with the flow of data from end to end through the network.

The upper three layers of the OSI model (application, presentation and session— Layers 7, 6 and 5) are orientated more toward services to the applications.



In the table below, we have specified the **protocols** used and the **data unit** exchanged by each layer of the OSI Model.

Layer	Name of Protocol	Name of Unit exchanged
Application	Application Protocol	APDU - Application Protocol Data Unit
Presentation	Presentation Protocol	PPDU - Presentation Protocol Data Unit
Session	Session Protocol	SPDU - Session Protocol Data Unit
Transport	Transport Protocol	TPDU - Transport Protocol Data Unit
Network	Network layer host-router Protocol	Packet
Data Link	Data link layer host-router Protocol	Frame
Physical	Physical layer host-router Protocol	Bit

APPLICATION LAYER

- User Support Layer 7 provides an interface between the communications software and any applications that need to communicate outside the computer . application layer is the OSI layer that is closest to the user.
- The OSI model describes how information or data makes its way from application programmes (such as spreadsheets) through a network medium (such as wire) to another application programme located on another network.
- Application-layer protocols can be programs in themselves, such as File Transfer Protocol (FTP), or they can be used by other programs, such as Simple Mail Transfer Protocol (SMTP), used by most e-mail programs, to redirect data to the network.
- The user application itself does not reside at the Application layer – the protocol does. The user interacts with the application, which in turn interacts with the application protocol.

Examples of Application layer protocols include:

- FTP, via an FTP client
- HTTP, via a web browser
- POP3 and SMTP, via an email client
- Telnet

The Application layer provides a variety of functions:

- Identifies communication partners
- Determines resource availability
- Synchronizes communication

PRESENTATION LAYER : USER SUPPORT

- This layer's main purpose is to define and negotiate data formats, such as ASCII text, EBCDIC text, binary, BCD, and JPEG. Encryption also is defined by OSI as a presentation layer service. The presentation layer is responsible for converting protocols, translating the data, encrypting the data, changing or converting the character set, and expanding graphics commands. The presentation layer also manages data compression to reduce the number of bits that need to be transmitted.

- The Presentation layer (Layer-6) controls the formatting and syntax of user data for the application layer. This ensures that data from the sending application can be understood by the receiving application.
- Standards have been developed for the formatting of data types, such as text, images, audio, and video. Examples of Presentation layer formats include:
 - • Text - TXT, ASCII, EBCDIC
 - • Images - GIF, JPG, JPEG, MPEG
 - • Audio - MIDI, MP3, WAV
 - • Movies - MPEG, AVI, MOV

PRESENTATION LAYER : USER SUPPORT

- If two devices do not support the same format or syntax, the Presentation layer can provide conversion or translation services to facilitate communication.
- Additionally, the Presentation layer can perform encryption and compression of data, as required. However, these functions can also be performed at lower layers as well. For example, the Network layer can perform encryption, using IPSec

SESSION LAYER :USER SUPPORT

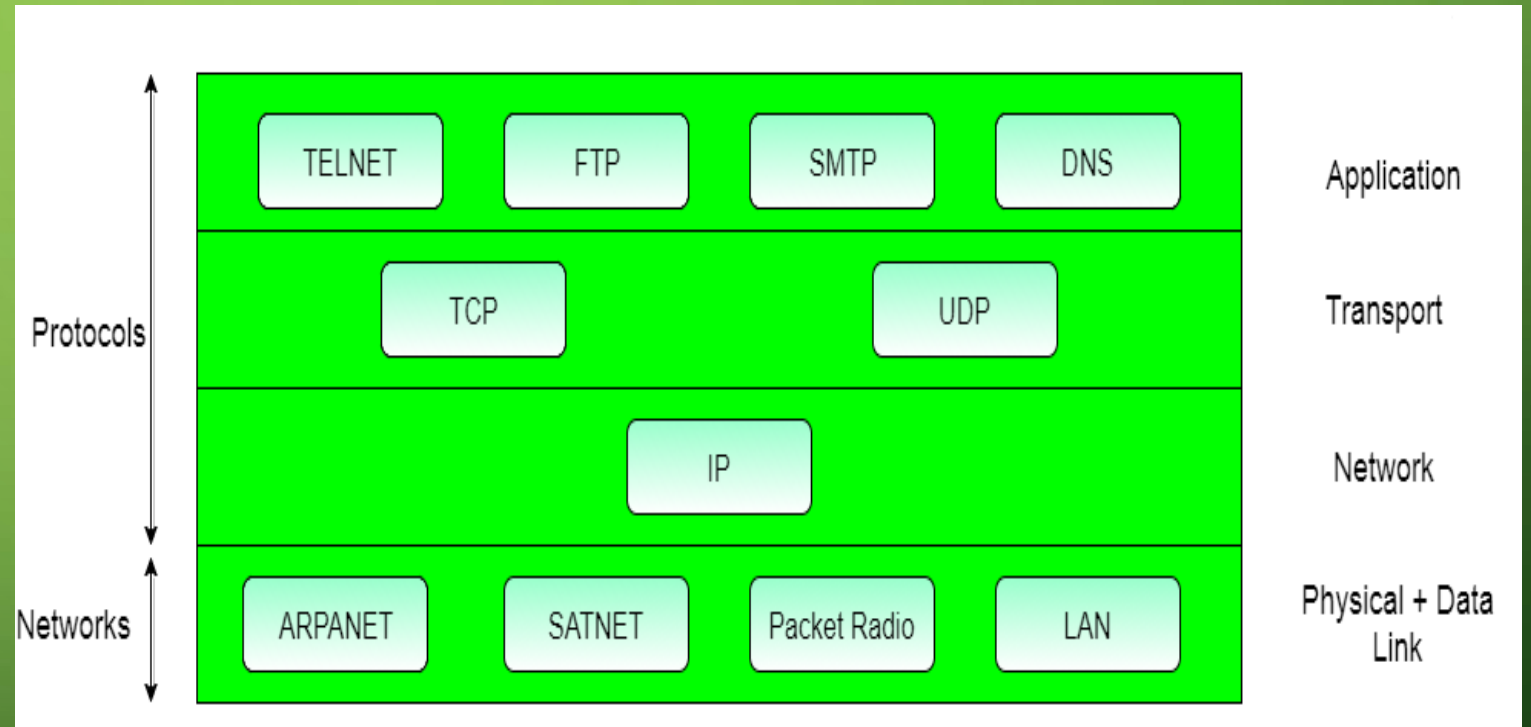
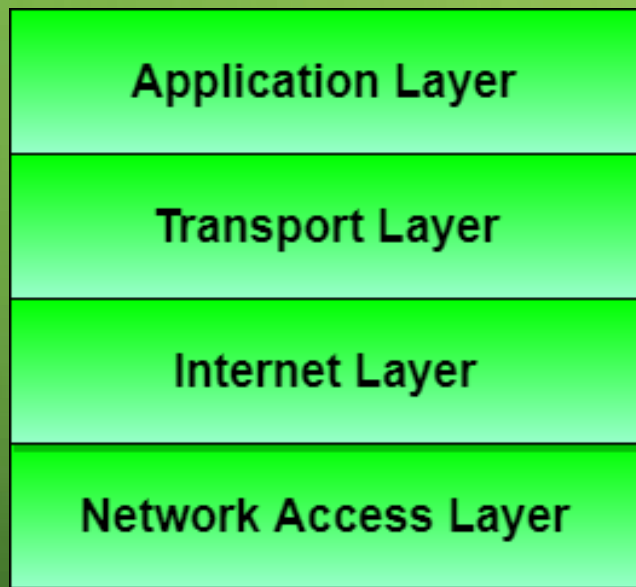
- The session layer defines how to start, control, and end conversations (called sessions). This includes the control and management of multiple bidirectional messages so that the application can be notified if only some of a series of messages are completed. This allows the presentation layer to have a seamless view of an incoming stream of data.
- The session layer is responsible for managing this dialog. It performs name recognition and other functions, such as security, that are needed to allow two applications to communicate over the network.

- Sessions communication falls under one of three categories:
 - Full-Duplex – simultaneous two-way communication
 - Half-Duplex – two-way communication, but not simultaneous
 - Simplex – one-way communication
- Many modern protocol suites, such as TCP/IP, do not implement Session layer protocols. Connection management is often controlled by lower layers, such as the Transport layer.

TCP/IP REFERENCE MODEL

- TCP/IP means Transmission Control Protocol and Internet Protocol. It is the network model used in the current Internet architecture as well. **Protocols** are set of rules which govern every possible communication over a network. These protocols describe the movement of data between the source and destination or the internet. They also offer simple naming and addressing schemes.

- Protocols and networks in the TCP/IP model:

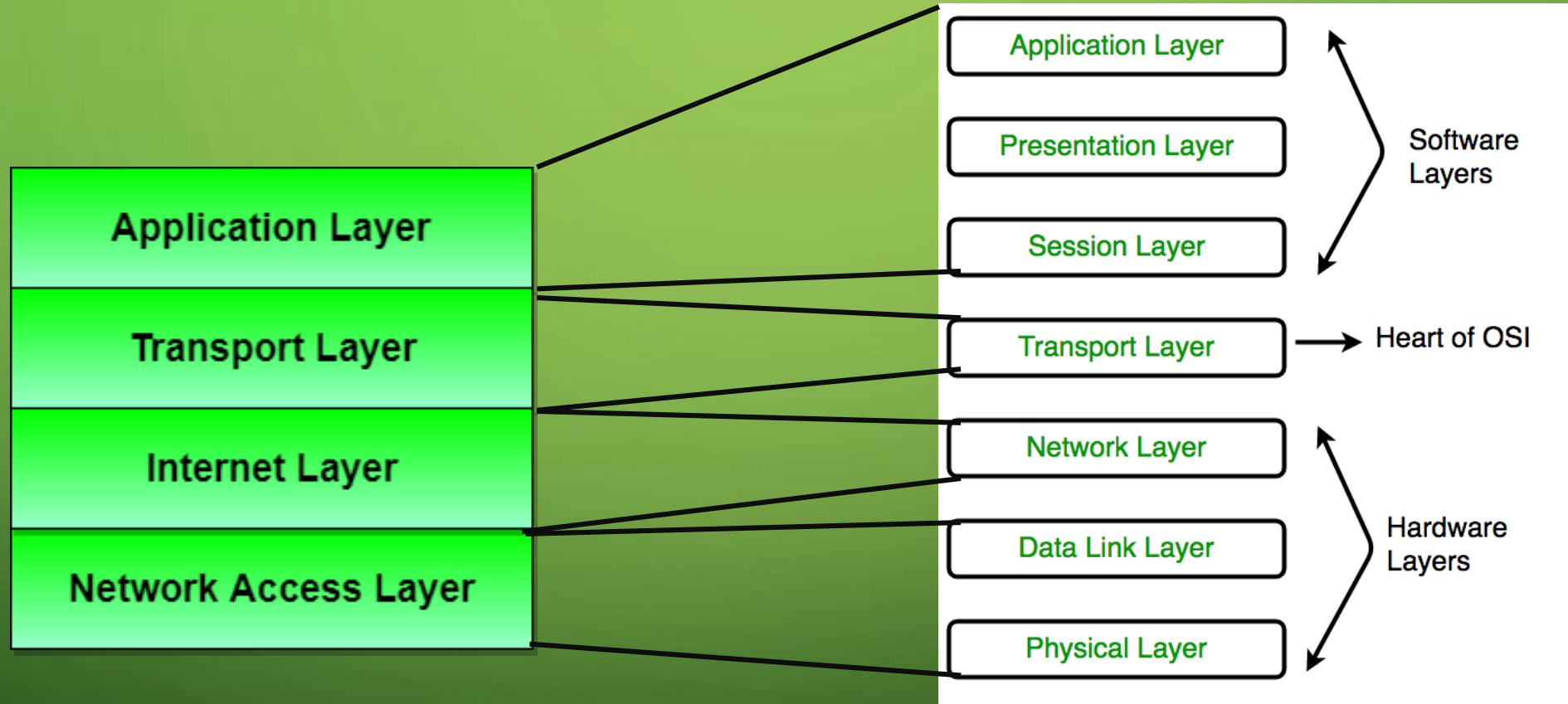


OVERVIEW OF TCP/IP REFERENCE MODEL

- TCP/IP that is Transmission Control Protocol and Internet Protocol was developed by Department of **Defence's Project Research Agency** (ARPA, later DARPA) as a part of a research project of network interconnection to connect remote machines.
- The features that stood out during the research, which led to making the TCP/IP reference model were:
 - **Support for a flexible architecture. Adding more machines to a network was easy.**
 - **The network was robust, and connections remained intact until the source and destination machines were functioning.**
- The overall idea was to allow one application on one computer to talk to (send data packets) another application running on different computer.

DIFFERENT LAYERS OF TCP/IP REFERENCE MODEL

- Below we have discussed the 4 layers that form the TCP/IP reference model:



LAYER 1: HOST-TO-NETWORK LAYER

- Lowest layer of the all.
- Protocol is used to connect to the host, so that the packets can be sent over it.
- Varies from host to host and network to network.

LAYER 2: INTERNET LAYER

- Selection of a packet switching network which is based on a connectionless internetwork layer is called a internet layer.
- It is the layer which holds the whole architecture together.
- It helps the packet to travel independently to the destination.
- Order in which packets are received is different from the way they are sent.
- IP (Internet Protocol) is used in this layer.
- The various functions performed by the Internet Layer are:
 - Delivering IP packets
 - Performing routing
 - Avoiding congestion

LAYER 3: TRANSPORT LAYER

- It decides if data transmission should be on parallel path or single path.
- Functions such as multiplexing, segmenting or splitting on the data is done by transport layer.
- The applications can read and write to the transport layer.
- Transport layer adds header information to the data.
- Transport layer breaks the message (data) into small units so that they are handled more efficiently by the network layer.
- Transport layer also arrange the packets to be sent, in sequence.

LAYER 4: APPLICATION LAYER

- The TCP/IP specifications described a lot of applications that were at the top of the protocol stack. Some of them were TELNET, FTP, SMTP, DNS etc.
- **TELNET** is a two-way communication protocol which allows connecting to a remote machine and run applications on it.
- **FTP**(File Transfer Protocol) is a protocol, that allows File transfer amongst computer users connected over a network. It is reliable, simple and efficient.
- **SMTP**(Simple Mail Transport Protocol) is a protocol, which is used to transport electronic mail between a source and destination, directed via a route.
- **DNS**(Domain Name Server) resolves an IP address into a textual address for Hosts connected over a network.
- It allows peer entities to carry conversation.
- It defines two end-to-end protocols: TCP and UDP
 - **TCP(Transmission Control Protocol)**: It is a reliable connection-oriented protocol which handles byte-stream from source to destination without error and flow control.
 - **UDP(User-Datagram Protocol)**: It is an unreliable connection-less protocol that do not want TCPs, sequencing and flow control. Eg: One-shot request-reply kind of service.

MERITS OF TCP/IP MODEL

- It operated independently.
- It is scalable.
- Client/server architecture.
- Supports a number of routing protocols.
- Can be used to establish a connection between two computers.

DEMERITS OF TCP/IP

- In this, the transport layer does not guarantee delivery of packets.
- The model cannot be used in any other application.
- Replacing protocol is not easy.
- It has not clearly separated its services, interfaces and protocols.

