

## UNIT: II: NETWORK MODELS

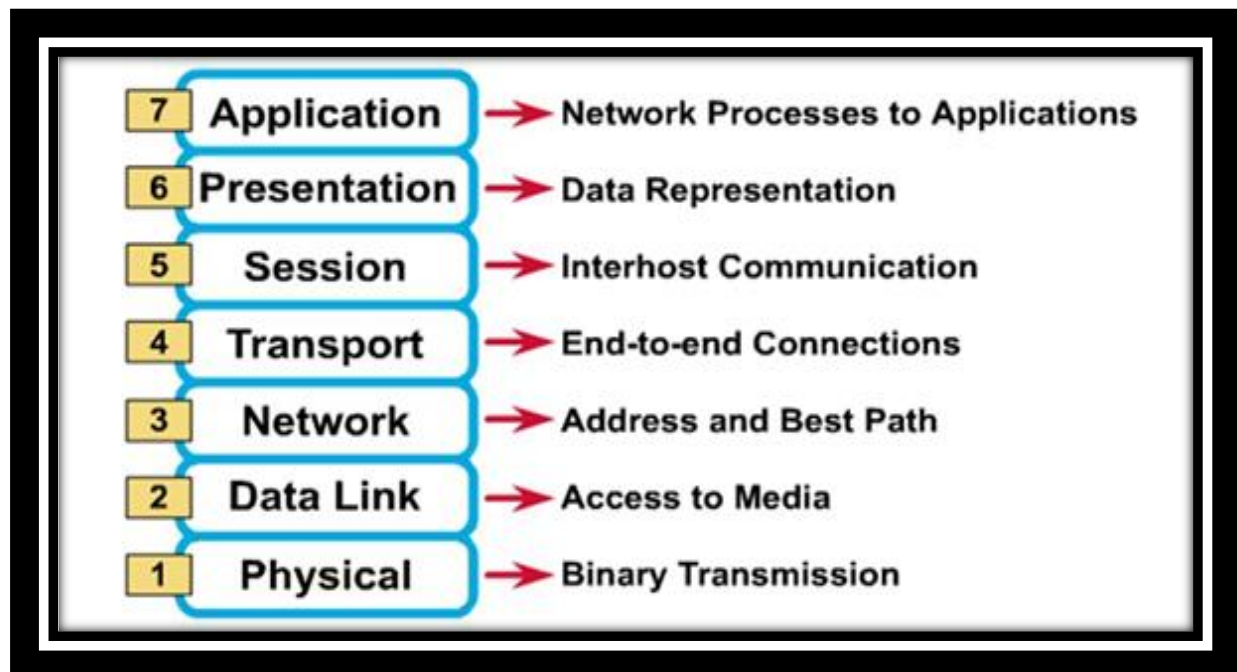
### OSI Model

This model is based on a proposal developed by the International Standards Organization (ISO) as a first step toward international standardization of the protocols used in the various layers (Day and Zimmermann, 1983). It was revised in 1995 (Day, 1995). The model is called the ISO OSI (Open Systems Interconnection) Reference Model because it deals with connecting open systems—that is, systems that are open for communication with other systems. We will just call it the OSI model for short.

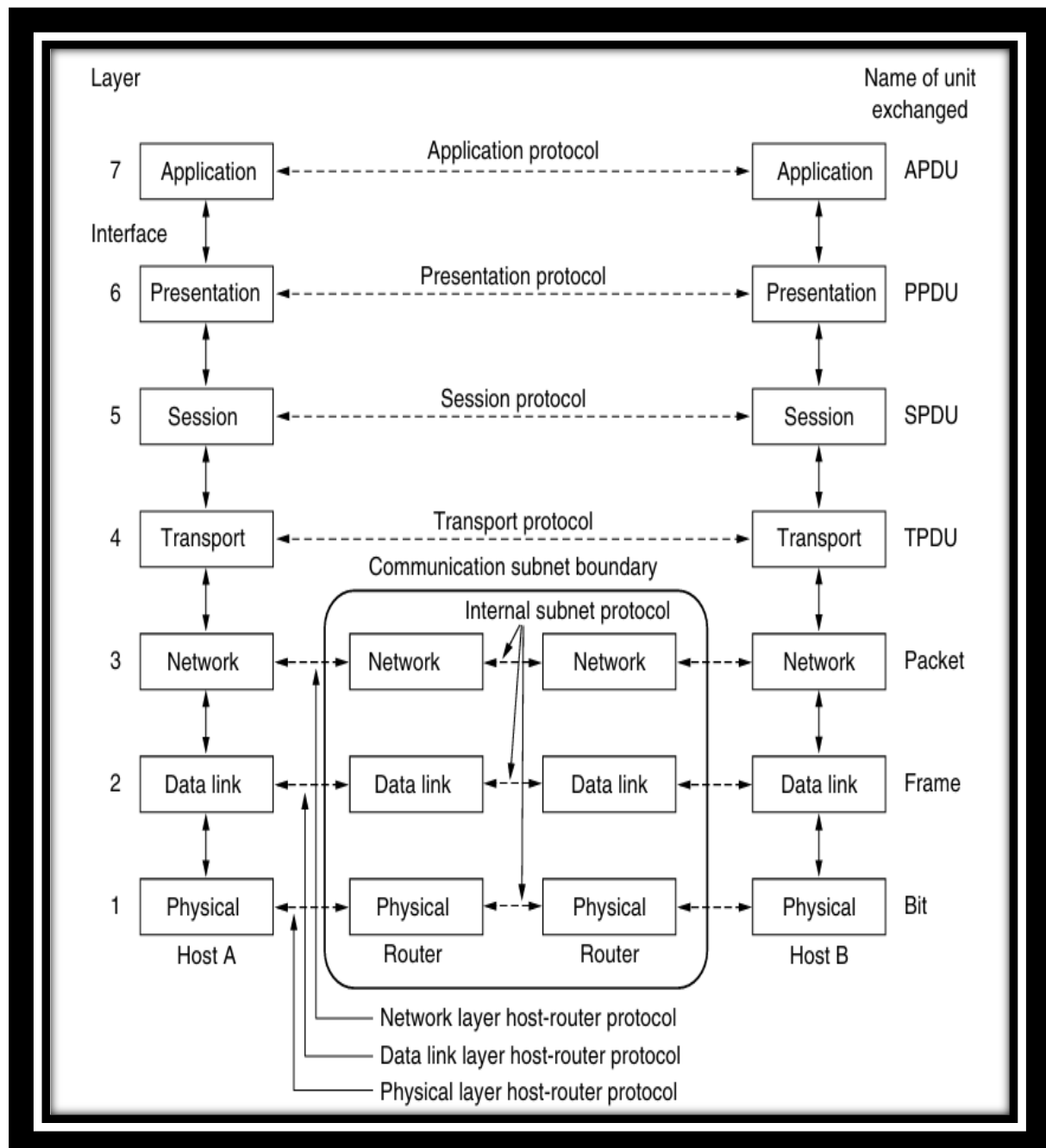
The OSI model has seven layers.

The principles that were applied to arrive at the seven layers can be briefly summarized as follows:

1. A layer should be created where a different abstraction is needed.
2. Each layer should perform a well-defined function.
3. The function of each layer should be chosen with an eye toward defining internationally standardized protocols.
4. The layer boundaries should be chosen to minimize the information flow across the interfaces.
5. The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity and small enough that the architecture does not become unwieldy.



## The OSI reference model.



### *The Physical Layer*

The physical layer is concerned with transmitting raw bits over a communication channel. The design issues have to do with making sure that when one side sends a 1 bit it is received by the other side as a 1 bit, not as a 0 bit. Typical questions here are what electrical signals should be used to represent a 1 and a 0, how many nanoseconds a bit lasts, whether transmission may proceed simultaneously in both directions, how the initial connection is established, how it is torn down when both sides are finished, how many pins the network connector has, and what each pin is used for. These design issues largely deal with mechanical, electrical, and timing interfaces, as well as the physical transmission medium, which lies below the physical layer.

### ***The Data Link Layer***

The main task of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors. It does so by masking the real errors so the network layer does not see them. It accomplishes this task by having the sender break up the input data into data frames (typically a few hundred or a few thousand bytes) and transmit the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by sending back an acknowledgement frame. Another issue that arises in the data link layer (and most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism may be needed to let the transmitter know when the receiver can accept more data. Broadcast networks have an additional issue in the data link layer: how to control access to the shared channel. A special sublayer of the data link layer, the medium access control sublayer, deals with this problem.

### ***The Network Layer***

The network layer controls the operation of the subnet. A key design issue is determining how packets are routed from source to destination. Routes can be based on static tables that are “wired into” the network and rarely changed, or more often they can be updated automatically to avoid failed components. They can also be determined at the start of each conversation, for example, a terminal session, such as a login to a remote machine. Finally, they can be highly dynamic, being determined anew for each packet to reflect the current network load. If too many packets are present in the subnet at the same time, they will get in one another’s way, forming bottlenecks. Handling congestion is also a responsibility of the network layer, in conjunction with higher layers that adapt the load they place on the network. More generally, the quality of service provided (delay, transit time, jitter, etc.) is also a network layer issue. When a packet has to travel from one network to another to get to its destination, many problems can arise. The addressing used by the second network may be different from that used by the first one. The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected. In broadcast networks, the routing problem is simple, so the network layer is often thin or even non-existent.

### ***The Transport Layer***

The basic function of the transport layer is to accept data from above it, split it up into smaller units, if need be, pass these to the network layer, and ensure that the pieces all arrive correctly at the other end. Furthermore, all this must be done efficiently and in a way that isolates the upper layers from the inevitable changes in the hardware technology over the course of time. The transport layer also determines what type of service to provide to the session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an error-free point-to-point channel that delivers messages or bytes in the order in which they were sent. However, other possible kinds of transport service exist, such as the transporting of isolated messages with no guarantee about the order of delivery, and the broadcasting of messages to multiple destinations. The type of service is determined when the connection is established. (As an aside, an error-free channel is completely impossible to achieve; what people really mean by this term is that the error rate is low enough to ignore in practice.) The transport layer is a true end-to-end layer; it carries data all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages. In the lower layers, each protocol is between a machine and its immediate neighbours, and not between the ultimate source and destination machines, which may be separated by many routers. The difference between layers 1 through 3, which are chained, and layers 4 through 7, which are end-to-end.

### ***The Session Layer***

The session layer allows users on different machines to establish sessions between them. Sessions offer various services, including dialog control (keeping track of whose turn it is to transmit), token management (preventing two parties from attempting the same critical operation simultaneously), and synchronization (checkpointing long transmissions to allow them to pick up from where they left off in the event of a crash and subsequent recovery).

### ***The Presentation Layer***

Unlike the lower layers, which are mostly concerned with moving bits around, the presentation layer is concerned with the syntax and semantics of the information transmitted. In order to make it possible for computers with different internal data representations to communicate, the data structures to be exchanged can be defined in an abstract way, along with a standard encoding to be used “on the wire.” The presentation layer manages these abstract data structures and allows higher-level data structures (e.g., banking records) to be defined and exchanged.

### ***The Application Layer***

The application layer contains a variety of protocols that are commonly needed by users. One widely used application protocol is HTTP (Hyper Text Transfer Protocol), which is the basis for the World Wide Web. When a browser wants a Web page, it sends the name of the page it wants to the server hosting the page using HTTP. The server then sends the page back. Other application protocols are used for file transfer, electronic mail, and network news.

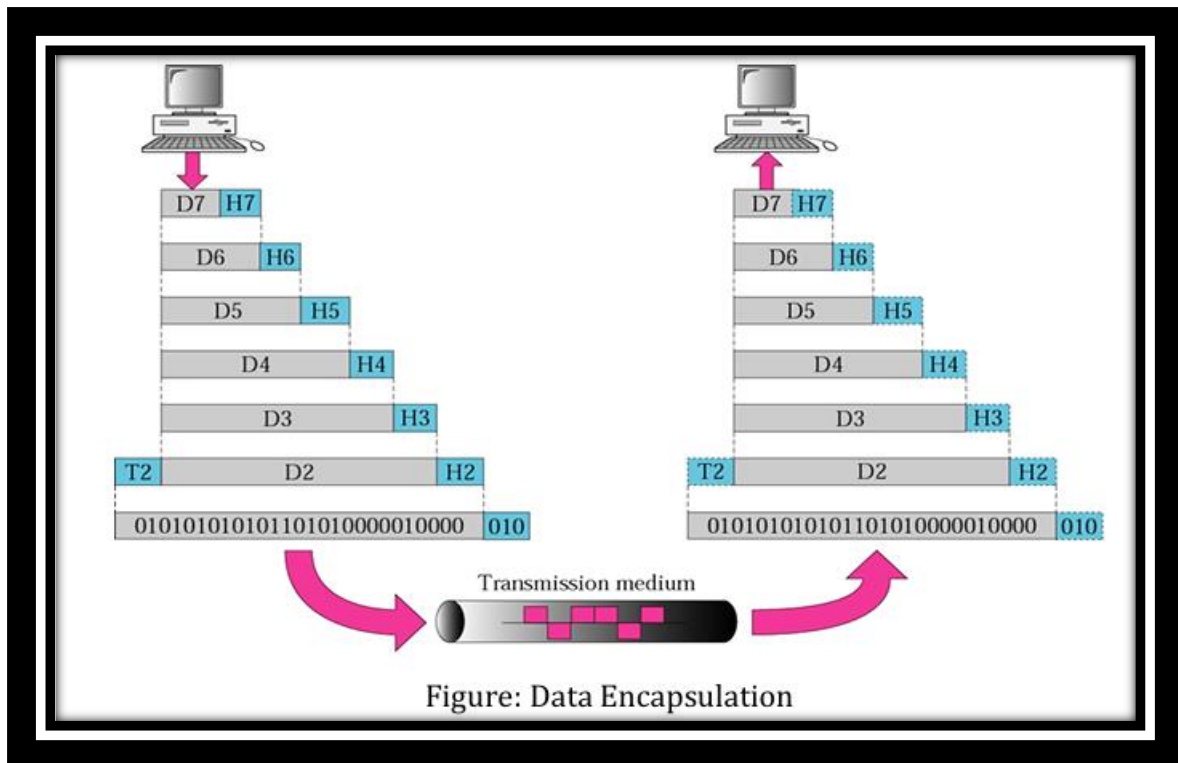
## **PEER-TO-PEER PROCESSES**

The processes on each machine that communicate at a given layer are called peer-to-peer processes.

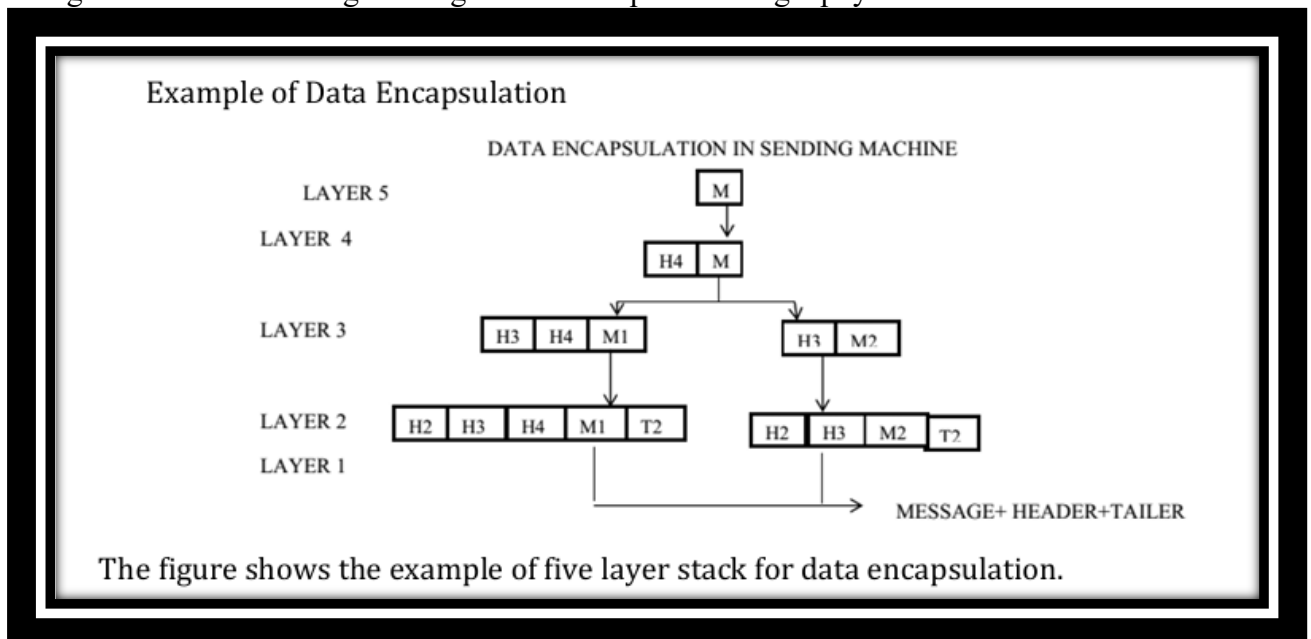
- At the physical layer, communication is direct: Device A sends a stream of bits to device B (through intermediate nodes).
- At higher layers each layer in the sending machines adds its own information to the message it receives from the layer just above it and passes the whole package to the layer just below it.

## **DATA ENCAPSULATION/ENCAPSULATION**

- The outgoing information will travel down through the layers to the lowest layer.
- While moving down on the source machine, it acquires all the control information which is required to reach the destination machine.
- The control information is in the form of Headers and Trailer which surrounds the data received from the layer above.
- This process of adding headers and trailers to the data is called as data encapsulation.
- The information added by each layer is in the form of headers or trailers.
- At layer 1 the entire package is converted to a form that can be transferred to the receiving machine.
- At the receiving machine, the message is unwrapped layer by layer, with each process receiving and removing the data meant for it.
- For example, layer 2 removes the data meant for it, then passes the rest to layer 3.
- Layer 3 then removes the data meant for it and passes the rest to layer 4, and so on.
- The headers and trailers contain control information. The headers and trailers form the envelope which carries the message to the desired destination.



- D7 means the data unit at layer 7, D6 means the data unit at layer 6, and so on.
- The process starts at layer 7 (the application layer), then moves from layer to layer in descending, sequential order.
- At each layer, a header, or possibly a trailer, can be added to the data unit.
- Commonly, the trailer is added only at layer 2.
- When the formatted data unit passes through the physical layer (layer 1), it is changed into an electromagnetic signal and transported along a physical link.



- The fifth layer of sending machine wants to send a message M to the fifth layer of destination machine.
- The message M is produced by layer 5 of machine 1 and given to layer 4 for transmission. Layer 4 adds header H4 in front of the message and pass it to layer 3.

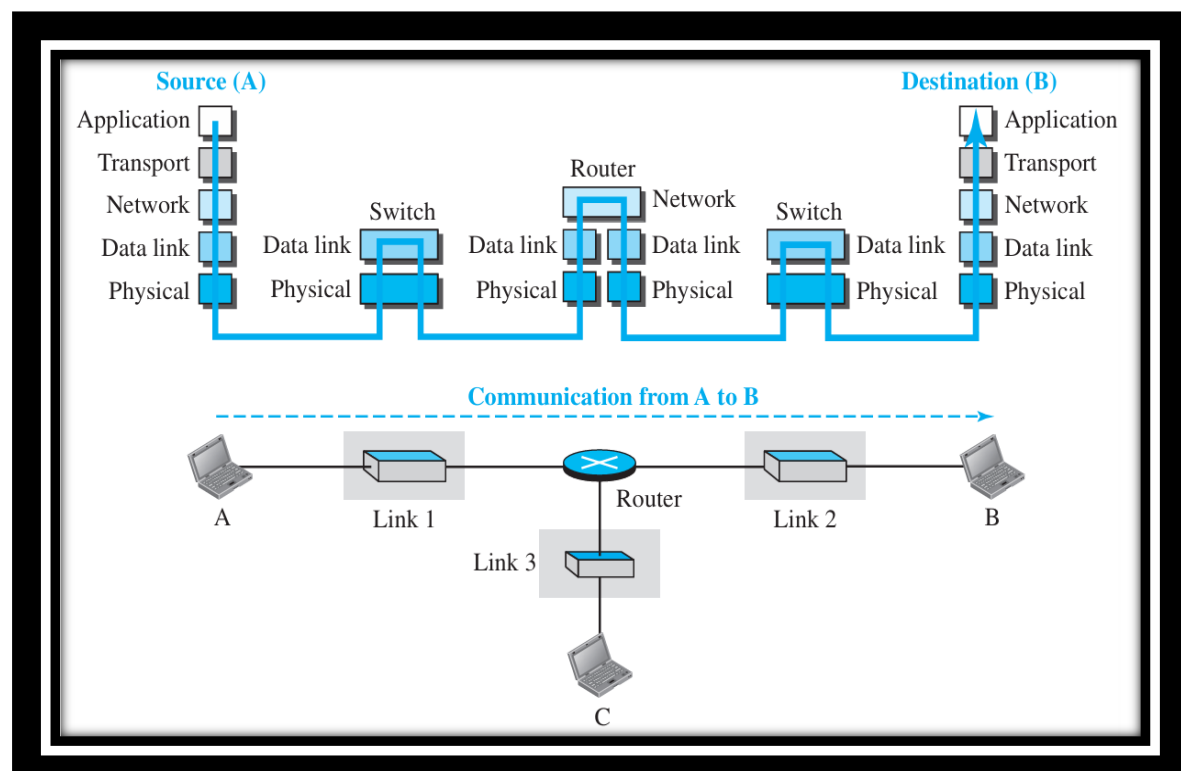
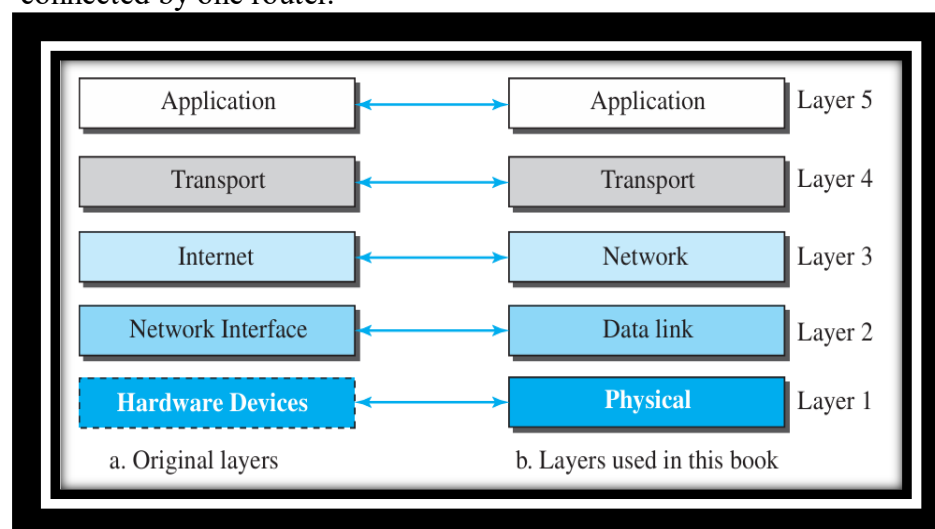
- Layer 3 breaks up the incoming message into small units as M1 and M2 and pass these packets to layer 2.
- Layer 2 adds the header as well as footer to each packet obtained from layer 3 and pass it to layer 1 for physical transmission.

## TCP/IP MODEL

### TCP/IP PROTOCOL SUITE

#### Layered Architecture

To show how the layers in the TCP/IP protocol suite are involved in communication between two hosts, we assume that we want to use the suite in a small internet made up of three LANs (links), each with a link-layer switch. We also assume that the links are connected by one router.



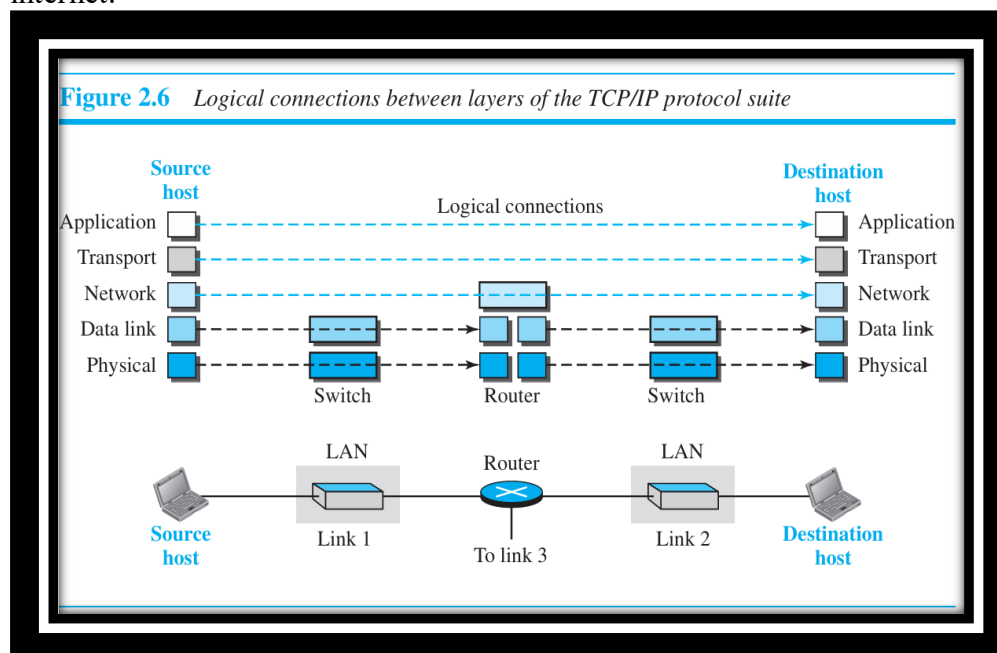
Let us assume that computer A communicates with computer B. As the figure shows, we have five communicating devices in this communication: source host (computer A), the link-layer switch in link 1, the router, the link-layer switch in link 2, and the destination host (computer B). Each device is involved with a set of layers depending on the role of the device in the internet. The two hosts are involved in all five layers; the source host needs to create a message in the application layer and send it down the layers so that it is physically sent to the destination host. The destination host needs to receive the communication at the physical layer and then deliver it through the other layers to the application layer.

The router is involved in only three layers; there is no transport or application layer in a router as long as the router is used only for routing. Although a router is always involved in one network layer, it is involved in  $n$  combinations of link and physical layers in which  $n$  is the number of links the router is connected to. The reason is that each link may use its own data-link or physical protocol. For example, in the above figure, the router is involved in three links, but the message sent from source A to destination B is involved in two links. Each link may be using different link-layer and physical-layer protocols; the router needs to receive a packet from link 1 based on one pair of protocols and deliver it to link 2 based on another pair of protocols.

A link-layer switch in a link, however, is involved only in two layers, data-link and physical. Although each switch in the above figure has two different connections, the connections are in the same link, which uses only one set of protocols. This means that, unlike a router, a link-layer switch is involved only in one data-link and one physical layer.

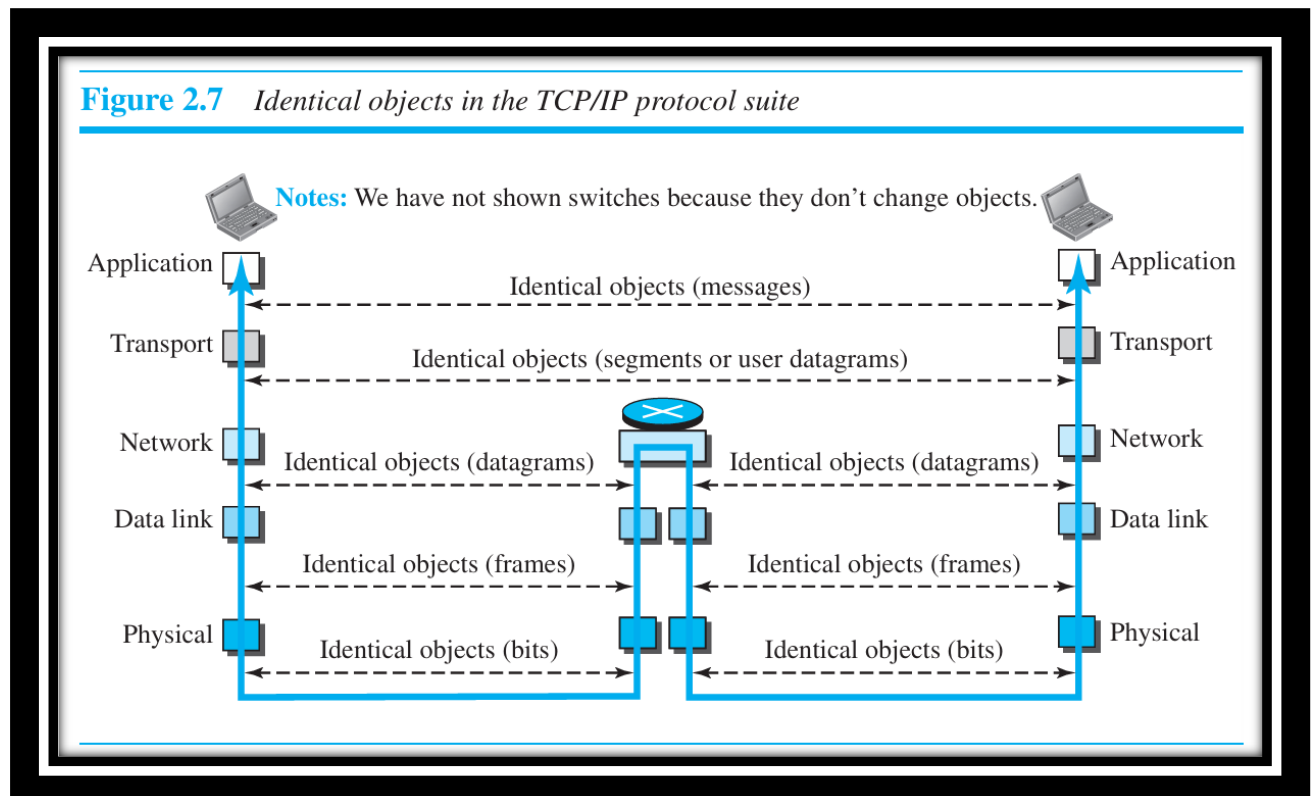
### Layers in the TCP/IP Protocol Suite

After the above introduction, we briefly discuss the functions and duties of layers in the TCP/IP protocol suite. Each layer is discussed in detail in the next five parts of the book. To better understand the duties of each layer, we need to think about the logical connections between layers. Figure 2.6 shows logical connections in our simple internet.



Using logical connections makes it easier for us to think about the duty of each layer. As the figure shows, the duty of the application, transport, and network layers is

end-to-end. However, the duty of the data-link and physical layers is hop-to-hop, in which a hop is a host or router. In other words, the domain of duty of the top three layers is the internet, and the domain of duty of the two lower layers is the link. Another way of thinking of the logical connections is to think about the data unit created from each layer. In the top three layers, the data unit (packets) should not be changed by any router or link-layer switch. In the bottom two layers, the packet created by the host is changed only by the routers, not by the link-layer switches. Figure 2.7 shows the second principle discussed previously for protocol layering. We show the identical objects below each layer related to each device.



Note that, although the logical connection at the network layer is between the two hosts, we can only say that identical objects exist between two hops in this case because a router may fragment the packet at the network layer and send more packets than received. Note that the link between two hops does not change the object.

### **Description of Each Layer**

After understanding the concept of logical communication, we are ready to briefly discuss the duty of each layer.

#### **Physical Layer**

We can say that the physical layer is responsible for carrying individual bits in a frame across the link. Although the physical layer is the lowest level in the TCP/IP protocol suite, the communication between two devices at the physical layer is still a logical communication because there is another, hidden layer, the transmission media, under the physical layer. Two devices are connected by a transmission medium (cable or air). We need to know that the transmission medium does not carry bits; it carries electrical or optical signals. So, the bits received in a frame from the data-link layer are transformed and sent through the transmission media, but we can think that the logical unit between two physical layers in two devices is a bit. There are several protocols that



transform a bit to a signal.

### ***Data-link Layer***

We have seen that an internet is made up of several links (LANs and WANs) connected by routers. There may be several overlapping sets of links that a datagram can travel from the host to the destination. The routers are responsible for choosing the best links. However, when the next link to travel is determined by the router, the data-link layer is responsible for taking the datagram and moving it across the link. The link can be a wired LAN with a link-layer switch, a wireless LAN, a wired WAN, or a wireless WAN. We can also have different protocols used with any link type. In each case, the data-link layer is responsible for moving the packet through the link.

TCP/IP does not define any specific protocol for the data-link layer. It supports all the standard and proprietary protocols. Any protocol that can take the datagram and carry it through the link suffices for the network layer. The data-link layer takes a datagram and encapsulates it in a packet called a frame.

Each link-layer protocol may provide a different service. Some link-layer protocols provide complete error detection and correction; some provide only error correction.

### ***Network Layer***

The network layer is responsible for creating a connection between the source computer and the destination computer. The communication at the network layer is host-to-host. However, since there can be several routers from the source to the destination, the routers in the path are responsible for choosing the best route for each packet. We can say that the network layer is responsible for host-to-host communication and routing the packet through possible routes. Again, we may ask ourselves why we need the network layer. We could have added the routing duty to the transport layer and dropped this layer. One reason, as we said before, is the separation of different tasks between different layers. The second reason is that the routers do not need the application and transport layers. Separating the tasks allow us to use fewer protocols on the routers.

The network layer in the Internet includes the main protocol, Internet Protocol (IP), that defines the format of the packet, called a datagram at the network layer. IP also defines the format and the structure of addresses used in this layer. IP is also responsible for routing a packet from its source to its destination, which is achieved by each router forwarding the datagram to the next router in its path.

IP is a connectionless protocol that provides no flow control, no error control, and no congestion control services. This means that if any of these services is required for an application, the application should rely only on the transport-layer protocol. The network layer also includes unicast (one-to-one) and multicast (one-to-many) routing protocols. A routing protocol does not take part in routing (it is the responsibility of IP), but it creates forwarding tables for routers to help them in the routing process.

The network layer also has some auxiliary protocols that help IP in its delivery and routing tasks. The Internet Control Message Protocol (ICMP) helps IP to report some problems when routing a packet. The Internet Group Management Protocol (IGMP) is another protocol that helps IP in multitasking. The Dynamic Host Configuration Protocol (DHCP) helps IP to get the network-layer address for a host. The Address Resolution Protocol (ARP) is a protocol that helps IP to find the link-layer address of a host or a router when its network-layer address is given.

### ***Transport Layer***

The logical connection at the transport layer is also end-to-end. The transport layer at the source host gets the message from the application layer, encapsulates it in a transport layer packet (called a segment or a user datagram in different protocols) and sends it,

through the logical (imaginary) connection, to the transport layer at the destination host. In other words, the transport layer is responsible for giving services to the application layer: to get a message from an application program running on the source host and deliver it to the corresponding application program on the destination host. We may ask why we need an end-to-end transport layer when we already have an end-to-end application layer. The reason is the separation of tasks and duties, which we discussed earlier.

The transport layer should be independent of the application layer. In addition, we will see that we have more than one protocol in the transport layer, which means that each application program can use the protocol that best matches its requirement.

As we said, there are a few transport-layer protocols in the Internet, each designed for some specific tasks. The main protocol, Transmission Control Protocol (TCP), is a connection-oriented protocol that first establishes a logical connection between transport layers at two hosts before transferring data. It creates a logical pipe between two TCPs for transferring a stream of bytes. TCP provides flow control (matching the sending data rate of the source host with the receiving data rate of the destination host to prevent overwhelming the destination), error control (to guarantee that the segments arrive at the destination without error and resending the corrupted ones), and congestion control to reduce the loss of segments due to congestion in the network. The other common protocol, User Datagram Protocol (UDP), is a connectionless protocol that transmits user datagrams without first creating a logical connection. In UDP, each user datagram is an independent entity without being related to the previous or the next one (the meaning of the term connectionless). UDP is a simple protocol that does not provide flow, error, or congestion control. Its simplicity, which means small overhead, is attractive to an application program that needs to send short messages and cannot afford the retransmission of the packets involved in TCP, when a packet is corrupted or lost. A new protocol, Stream Control Transmission Protocol (SCTP) is designed to respond to new applications that are emerging in the multimedia.

### ***Application Layer***

As Figure 2.6 shows, the logical connection between the two application layers is end-to-end. The two application layers exchange messages between each other as though there was a bridge between the two layers. However, we should know that the communication is done through all the layers.

Communication at the application layer is between two processes (two programs running at this layer). To communicate, a process sends a request to the other process and receives a response. Process-to-process communication is the duty of the application layer. The application layer in the Internet includes many predefined protocols, but a user can also create a pair of processes to be run at the two hosts.

The Hypertext Transfer Protocol (HTTP) is a vehicle for accessing the World Wide Web (WWW). The Simple Mail Transfer Protocol (SMTP) is the main protocol used in electronic mail (e-mail) service. The File Transfer Protocol (FTP) is used for transferring files from one host to another. The Terminal Network (TELNET) and Secure Shell (SSH) are used for accessing a site remotely. The Simple Network Management Protocol (SNMP) is used by an administrator to manage the Internet at global and local levels. The Domain Name System (DNS) is used by other protocols to find the network-layer address of a computer. The Internet Group Management Protocol (IGMP) is used to collect membership in a group.

## ADDRESSING

### ◆ 1. Physical Address

- **Definition:** The address that is **hard-coded into the network interface card (NIC)** by the manufacturer.
- **Also Called:** **MAC Address** (Media Access Control)
- **Format:** 48-bit address written in hexadecimal (e.g., 00:1A:2B:3C:4D:5E)
- **Used At:** **Data Link Layer (Layer 2)** of the OSI model
- **Purpose:** Identifies devices within the **same local network**.

✓ **Example:** A printer and a computer connected to the same LAN use MAC addresses to communicate.

---

### ◆ 2. Logical Address

- **Definition:** An address assigned by software, used for **routing across networks**.
- **Also Called:** **IP Address**
- **Format:**
  - IPv4: 32-bit address (e.g., 192.168.1.1)
  - IPv6: 128-bit address (e.g., 2001:0db8:85a3::8a2e:0370:7334)
- **Used At:** **Network Layer (Layer 3)**
- **Purpose:** Enables **global identification** and communication between devices on different networks.

✓ **Example:** Your computer is assigned IP address 192.168.1.5 by a router in a home network.

---

### ◆ 3. Port Address

- **Definition:** A **number assigned to specific processes or services** on a device.
- **Used At:** **Transport Layer (Layer 4)**
- **Purpose:** Distinguishes between **multiple services** on the same device.
- **Range:** 0 to 65535
  - 0–1023: Well-known ports (e.g., 80 for HTTP, 443 for HTTPS)
  - 1024–49151: Registered
  - 49152–65535: Dynamic/private

✓ **Example:** When you visit <https://example.com>, your computer connects to **IP: 93.184.216.34, Port: 443** (for HTTPS).

---

### ◆ 4. Specific Address

- **Definition:** Refers to **application-level or service-specific addresses**, often used in **high-level protocols or multicast/broadcast scenarios**.
- **Purpose:** Targets a **specific service, user, or group** beyond the transport layer.
- **Examples:**
  - **Email Address** (e.g., user@example.com) in SMTP
  - **URL or URI** (e.g., <https://openai.com/api>) in web access
  - **Multicast Address** (e.g., 224.0.0.1) for group communications

✓ **Use Case:** Sending an email to student@university.edu uses logical addressing (IP), port addressing (SMTP uses port 25), and the **specific address** is the email ID.