ENCAPSULATION & DECAPSULATION, ADDRESSING, MULTIPLEXING & DEMULTIPLEXING AND OSI MODEL

ER. NITESH KUMAR JANGID

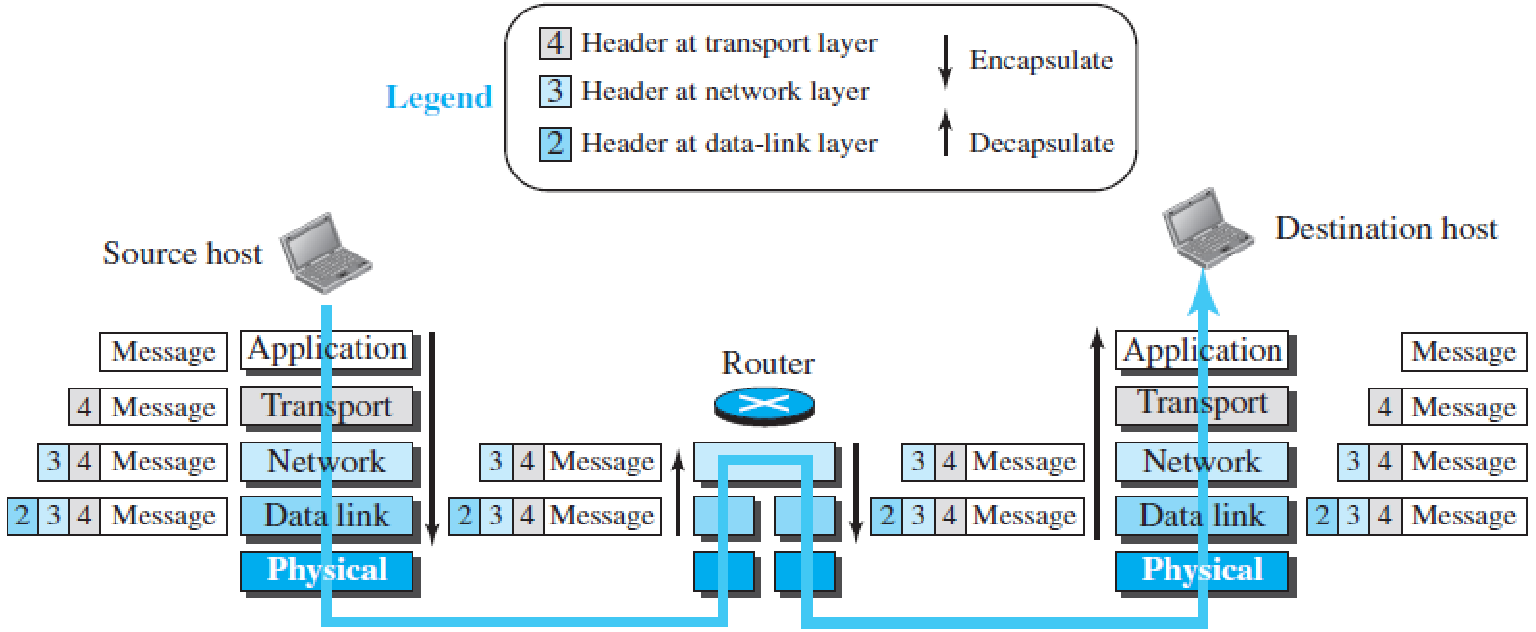
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE

CENTRAL UNIVERSITY OF RAJASTHAN

ENCAPSULATION AND DECAPSULATION

Figure 2.8 *Encapsulation/Decapsulation*



Nitesh Kumar Jangid, Assistant Professor, CS, CURAJ

ENCAPSULATION AT THE SOURCE HOST

- At the source, we have only encapsulation.
- At the application layer, the data to be exchanged is referred to as a message. A message normally does not contain any header or trailer.
- The transport layer takes the message as the payload. It adds the transport layer header to the payload, which contains the identifiers of the source and destination application programs that want to communicate plus some more information that is needed for the end-to-end delivery of the message, such as information needed for flow, error control, or congestion control. The result is the transport-layer packet, which is called the segment (in TCP) and the user datagram (in UDP).

ENCAPSULATION AT THE SOURCE HOST

- The network layer takes the transport-layer packet as data or payload and adds its own header to the payload.The header contains the addresses of the source and destination hosts and some more information used for error checking of the header, fragmentation information, and so on.The result is the network-layer packet, called a datagram.
- The data-link layer takes the network-layer packet as data or payload and adds its own header, which contains the link-layer addresses of the host or the next hop (the router). The result is the link-layer packet, which is called a frame.The frame is passed to the physical layer for transmission.

DECAPSULATION AND ENCAPSULATION AT THE ROUTER

- After the set of bits are delivered to the data-link layer, this layer decapsulates the datagram from the frame and passes it to the network layer.
- The network layer only inspects the source and destination addresses in the datagram header and consults its forwarding table to find the next hop to which the datagram is to be delivered. The contents of the datagram should not be changed by the network layer in the router unless there is a need to fragment the datagram if it is too big to be passed through the next link.
- The data-link layer of the next link encapsulates the datagram in a frame and passes it to the physical layer for transmission.

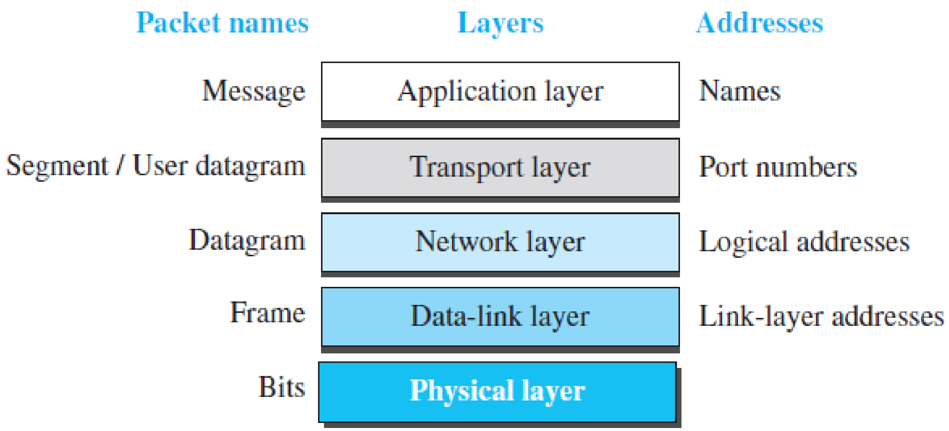
DECAPSULATION AT THE DESTINATION HOST

- At the destination host, each layer only decapsulates the packet received, removes the payload, and delivers the payload to the next-higher layer protocol until the message reaches the application layer. It is necessary to say that decapsulation in the host involves error checking.

ADDRESSING

- Any communication that involves two parties needs two addresses: source address and destination address. Although it looks as if we need five pairs of addresses, one pair per layer, we normally have only four because the physical layer does not need addresses; the unit of data exchange at the physical layer is a bit, which definitely cannot have an address.

Figure 2.9 *Addressing in the TCP/IP protocol suite*

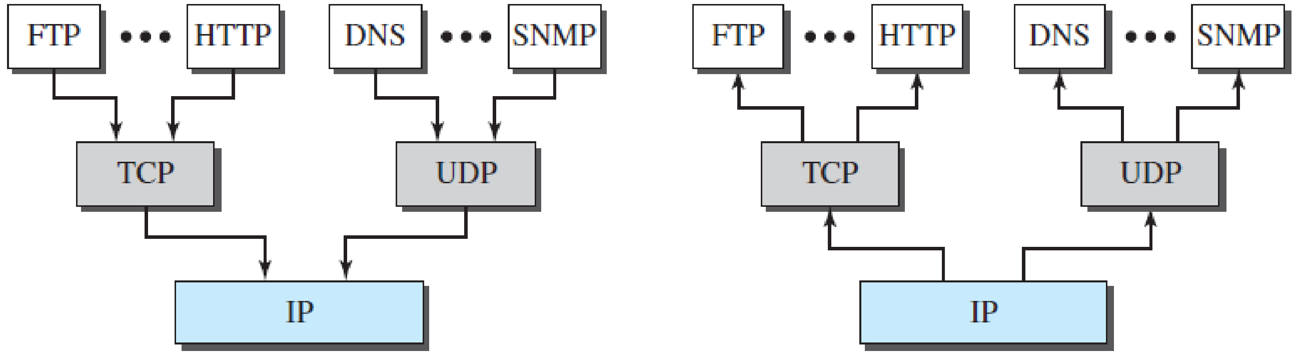


Packet names	Layers	Addresses
Message	Application layer	Names
Segment / User datagram	Transport layer	Port numbers
Datagram	Network layer	Logical addresses
Frame	Data-link layer	Link-layer addresses
Bits	Physical layer	

MULTIPLEXING AND DEMULTIPLEXING

- Since the TCP/IP protocol suite uses several protocols at some layers, we can say that we have multiplexing at the source and demultiplexing at the destination. Multiplexing in this case means that a protocol at a layer can encapsulate a packet from several next-higher layer protocols (one at a time); demultiplexing means that a protocol can decapsulate and deliver a packet to several next-higher layer protocols (one at a time).

Figure 2.10 *Multiplexing and demultiplexing*



a. Multiplexing at source

b. Demultiplexing at destination

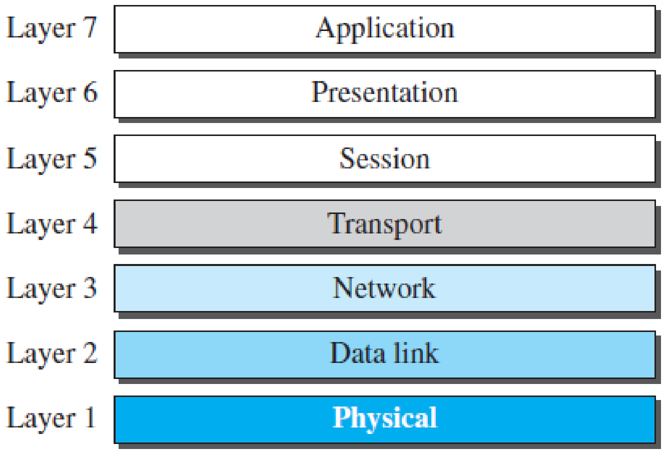
THE OSI MODEL

- Established in 1947, the International Organization for Standardization (ISO) is a multinational body dedicated to worldwide agreement on international standards.
- An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s.
- An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable. The OSI model was intended to be the basis for the creation of the protocols in the OSI stack.

THE OSI MODEL

- The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.

Figure 2.11 *The OSI model*



Layer 7 Application

Layer 6 Presentation

Layer 5 Session

Layer 4 Transport

Layer 3 Network

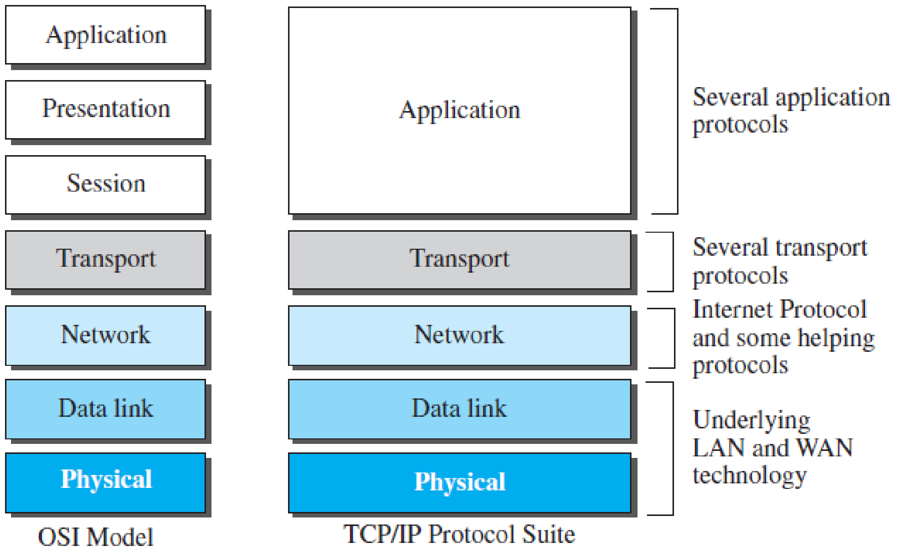
Layer 2 Data link

Layer 1 **Physical**

OSI VERSUS TCP/IP

- When we compare the two models, we find that two layers, session and presentation, are missing from the TCP/IP protocol suite. These two layers were not added to the TCP/IP protocol suite after the publication of the OSI model. The application layer in the suite is usually considered to be the combination of three layers in the OSI model.

Figure 2.12 *TCP/IP and OSI model*



LACK OF OSI MODEL’S SUCCESS

- The OSI model appeared after the TCP/IP protocol suite. Most experts were at first excited and thought that the TCP/IP protocol would be fully replaced by the OSI model. This did not happen for several reasons, but we describe only three, which are agreed upon by all experts in the field.
  - First, OSI was completed when TCP/IP was fully in place and a lot of time and money had been spent on the suite; changing it would cost a lot.
  - Second, some layers in the OSI model were never fully defined. For example, although the services provided by the presentation and the session layers were listed in the document, actual protocols for these two layers were not fully defined, nor were they fully described, and the corresponding software was not fully developed.
  - Third, when OSI was implemented by an organization in a different application, it did not show a high enough level of performance to entice the Internet authority to switch from the TCP/IP protocol suite to the OSI model.