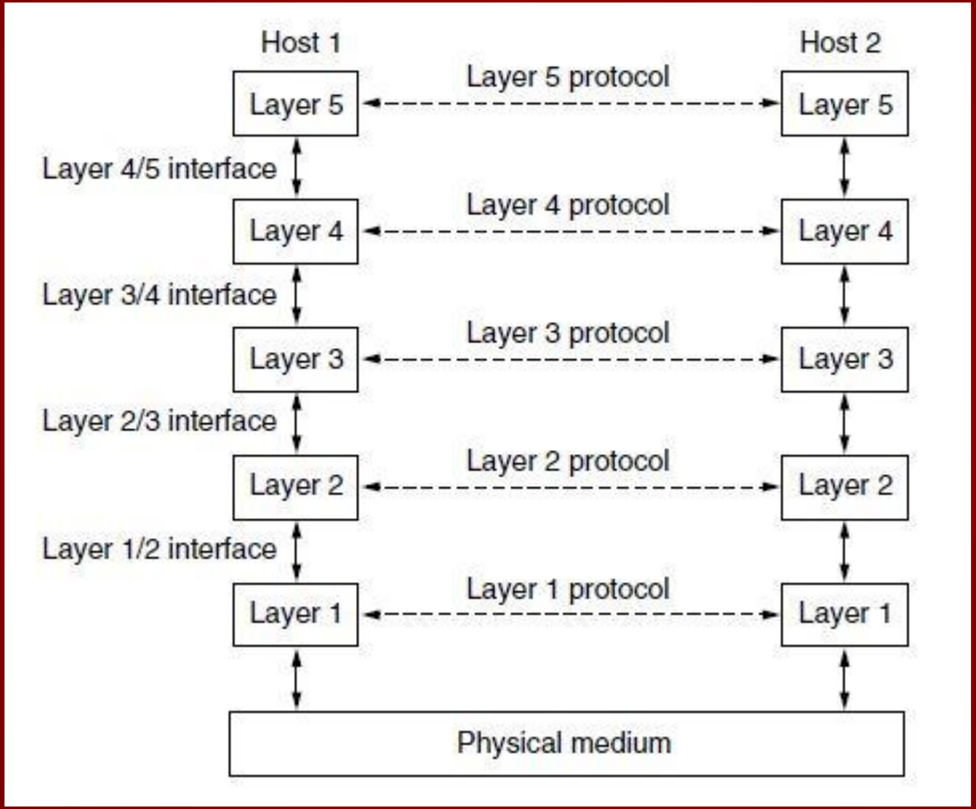
NETWORK SOFTWARE & NETWORK MODEL

By
Abdul Ghofir, M.Kom

Protocol Hierarchies

- To reduce their design complexity, most networks are organized as a stack of **layers** or **levels**, each one built upon the one below it.
- 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.
- In a sense, each layer is a kind of virtual machine, offering certain services to the layer above it.
- Basically, a **protocol** is an agreement between the communicating parties on how communication is to proceed.

Protocol Hierarchies

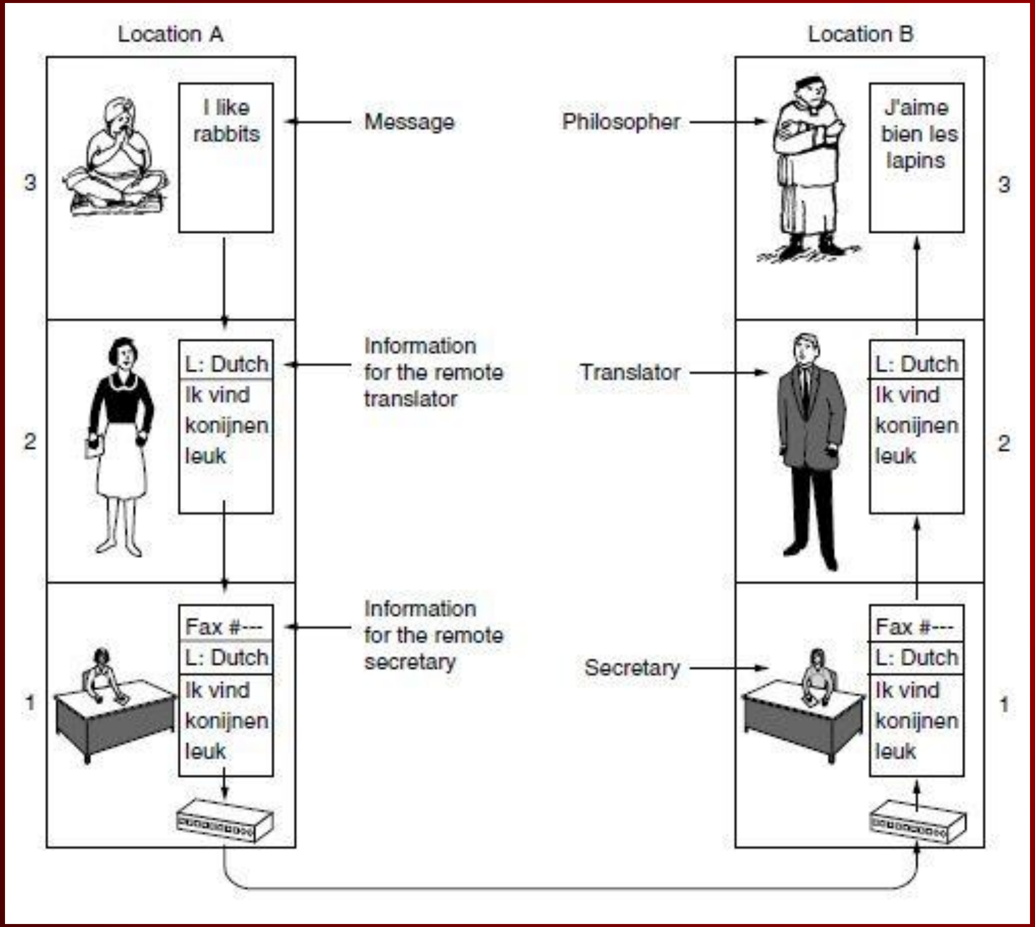


Protocol, layer, and interface

Protocol Hierarchies

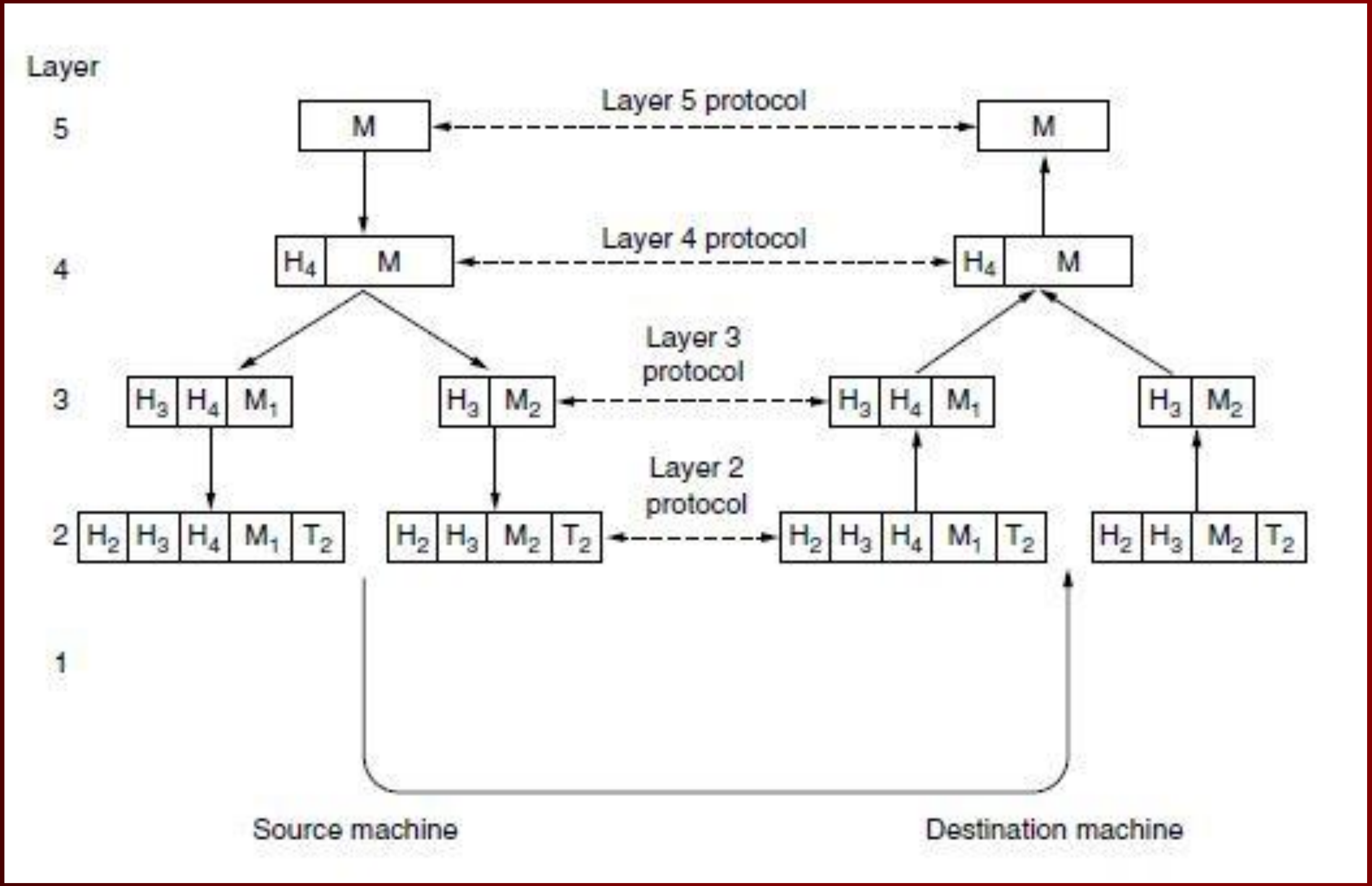
- The entities comprising the corresponding layers on different machines are called **peers**.
- The peers may be software processes, hardware devices, or even human beings.
- Between each pair of adjacent layers is an **interface**

Protocol Hierarchies



The philosopher-translator-secretary architecture.

Protocol Hierarchies



Example information flow supporting virtual communication in layer 5.

Design Issues for the Layers

1. Reliability
   - It is the design issue of making a network that operates correctly even though it is made up of a collection of components that are themselves unreliable.
   - One mechanism for finding errors in received information uses “codes” for **error detection**.
     - ✓ Information that is incorrectly received can then be retransmitted until it is received correctly.
     - ✓ More powerful codes allow for **error correction**, where the correct message is recovered from the possibly incorrect bits that were originally received.
     - ✓ Both of these mechanisms work by adding redundant information.

Design Issues for the Layers

- Another reliability issue is finding a working path through a network.
- Often there are multiple paths between a source and destination, and in a large network, there may be some links or routers that are broken. For instance:
    - ✓ Suppose that the network is down in Germany.
    - ✓ Packets sent from London to Rome via Germany will not get through, but we could instead send packets from London to Rome via Paris.
    - ✓ The network should automatically make this decision.
    - ✓ This topic is called **routing**.

Design Issues for the Layers

2. Evolution of the network.
   - Over time, networks grow larger and new designs emerge that need to be connected to the existing network.
   - Since there are many computers on the network, every layer needs a mechanism for identifying the senders and receivers that are involved in a particular message.
   - This mechanism is called **addressing** or **naming**, in the low and high layers, respectively.

Design Issues for the Layers

3. Allocation
   - Networks provide a service to hosts from their underlying resources, such as the capacity of transmission lines.
   - To do this well, they need mechanisms that divide their resources so that one host does not interfere with another too much.
   - This design is called **statistical multiplexing**, meaning sharing based on the statistics of demand.
   - An allocation problem that occurs at every level is how to keep a fast sender from swamping a slow receiver with data.
   - Feedback from the receiver to the sender is often used. This subject is called **flow control**.

Design Issues for the Layers

4. **Security Networks**
   - The last major design issue is to secure the network by defending it against different kinds of threats.
   - One of the threats is eavesdropping on communications.
     - ✓ Mechanisms that provide **confidentiality** defend against this threat, and they are used in multiple layers.
     - ✓ Mechanisms for **authentication** prevent someone from impersonating someone else.
     - ✓ Other mechanisms for **integrity** prevent surreptitious changes to messages.
     - ✓ All of these designs are based on cryptography.

Connection-Oriented Versus Connectionless Service

1. **Connection-oriented**
   - 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 modeled after the telephone system.
     - ✓ To talk to someone, you pick up the phone, dial the number, talk, and then hang up.
     - ✓ Similarly, to use a connection-oriented network service, the service user first establishes a connection, uses the connection, and then releases the connection.
   - Example of connection oriented is TCP (Transmission Control Protocol) protocol.

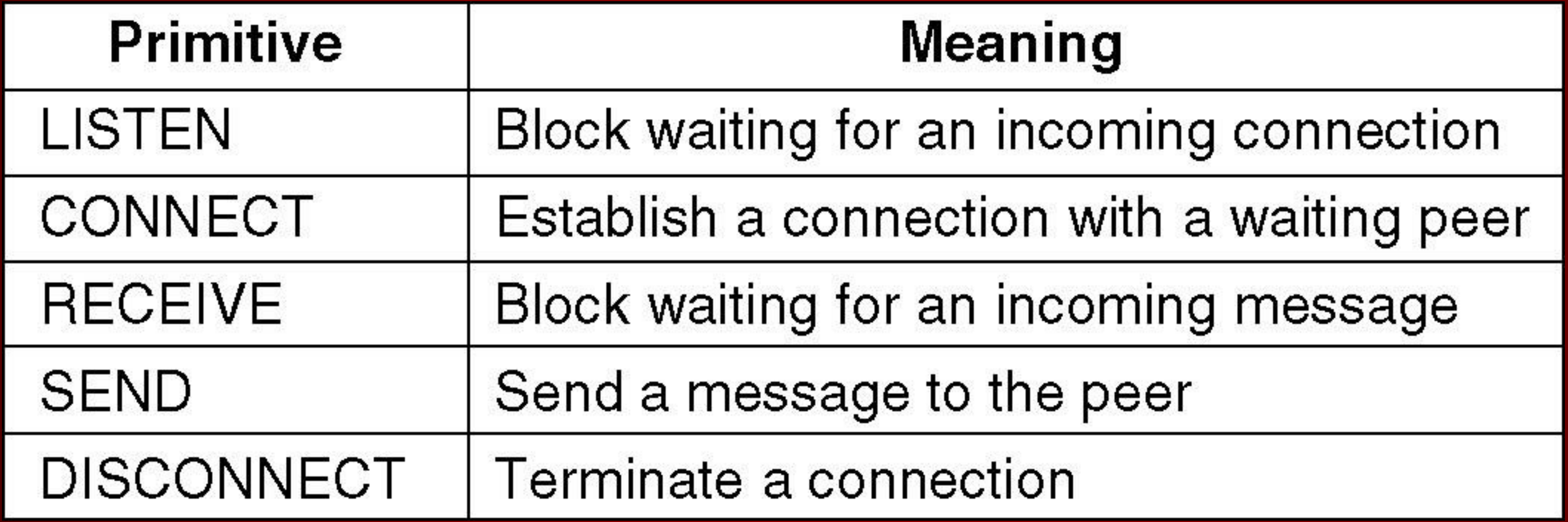
Connection-Oriented Versus Connectionless Service

2. **connectionless** service
   - 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.

Service Primitives

- A service is formally specified by a set of **primitives** (operations) available to user processes to access the service.
- The primitives for connection-oriented service are different from those of connectionless service.

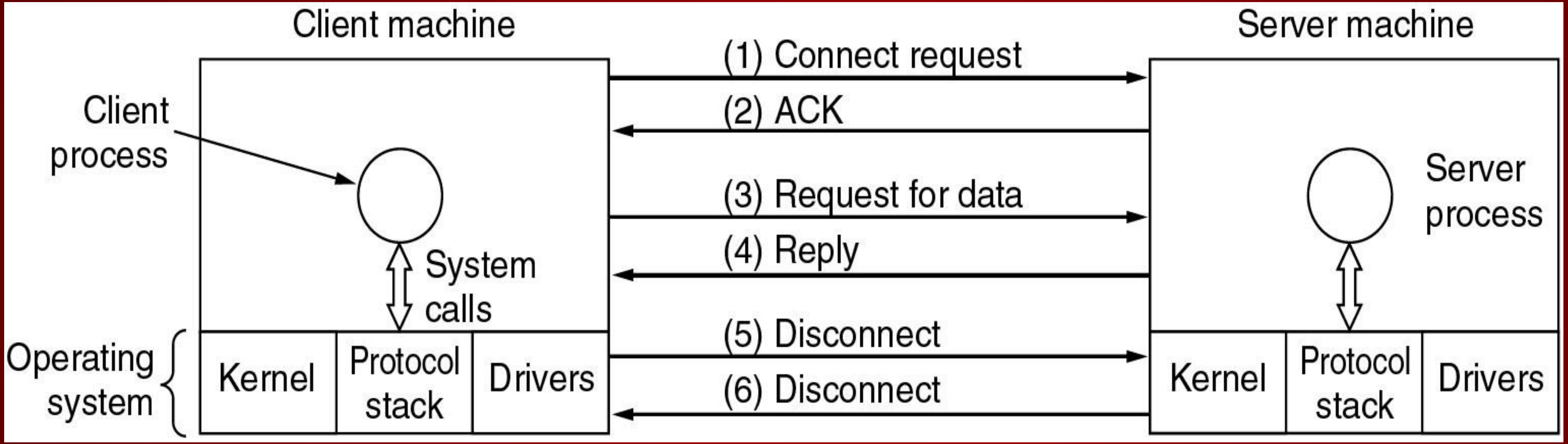
Service Primitives



Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Six service primitives that provide a simple connection-oriented service.

Service Primitives

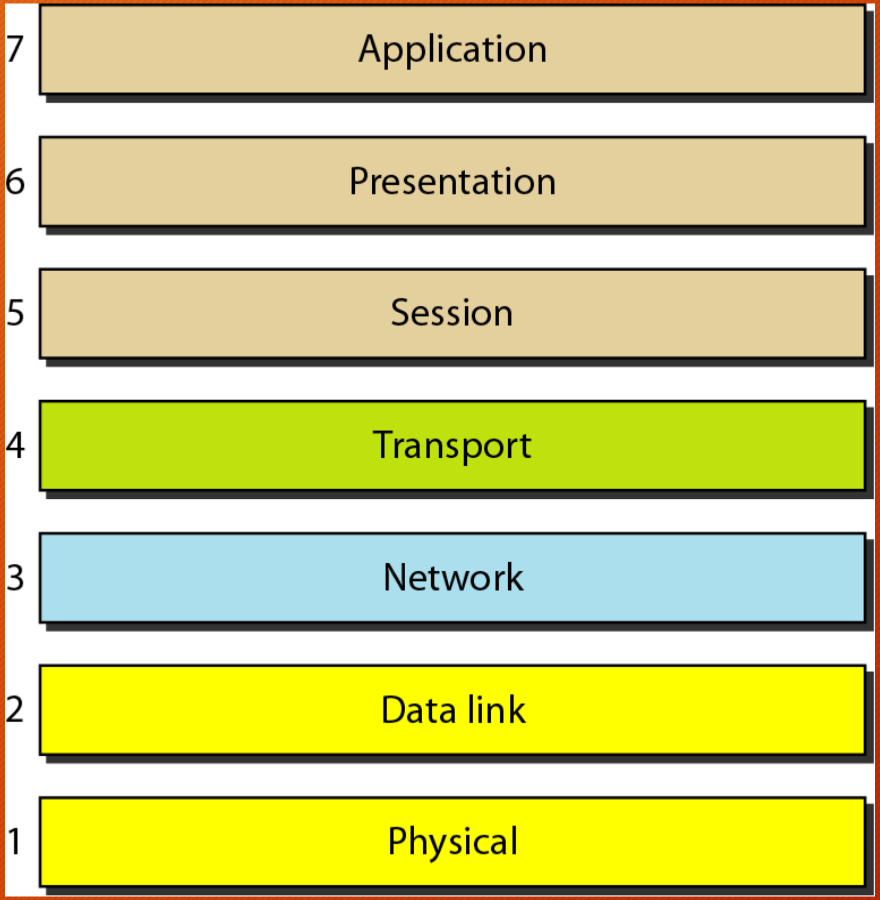


A simple client-server interaction using acknowledged datagrams.

NETWORK MODELS

THE OSI MODEL

- Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards.
- 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

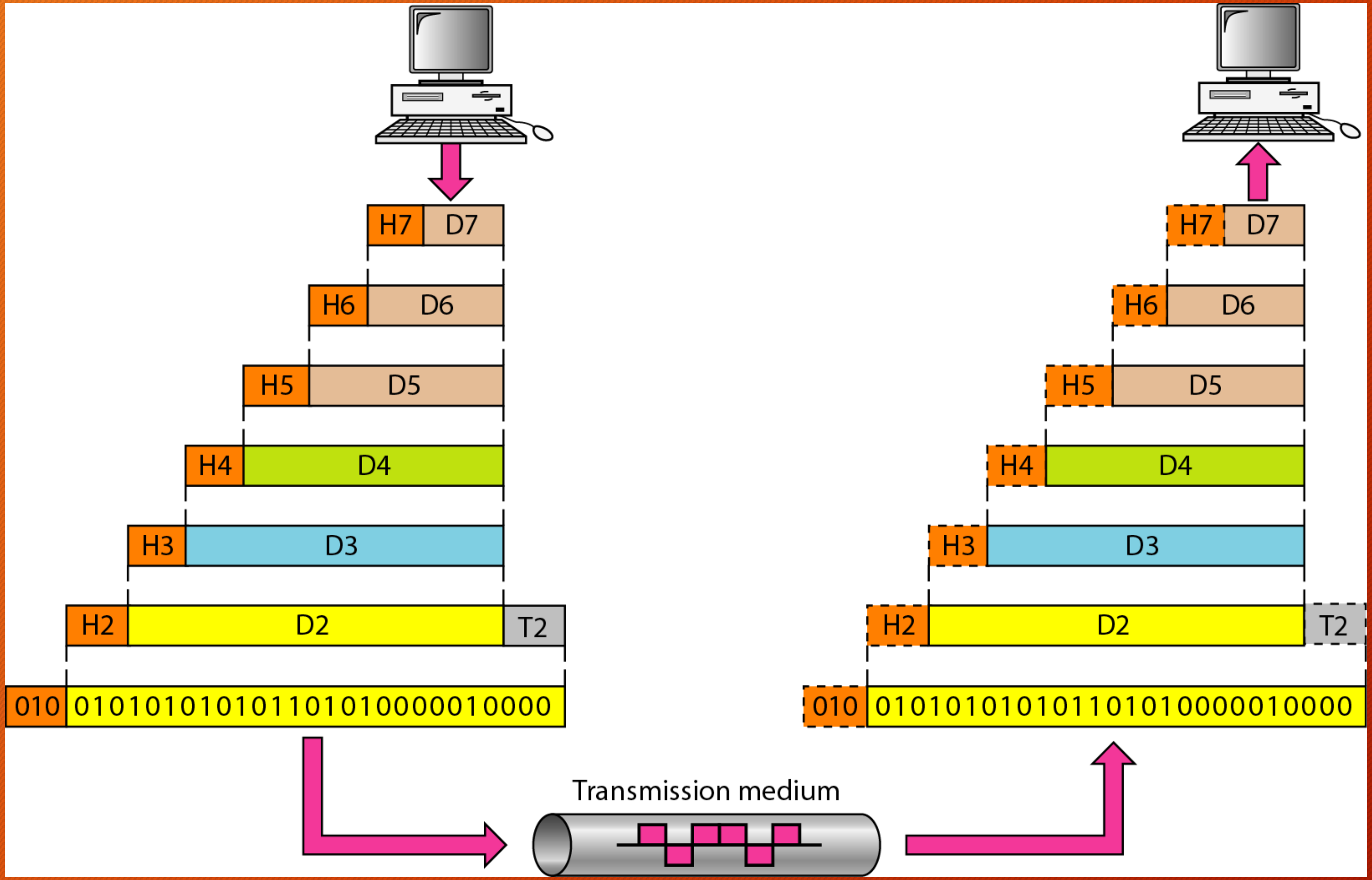


Seven layers of the OSI model

Organization of the Layers

The seven layers can be thought of as belonging to three subgroups:

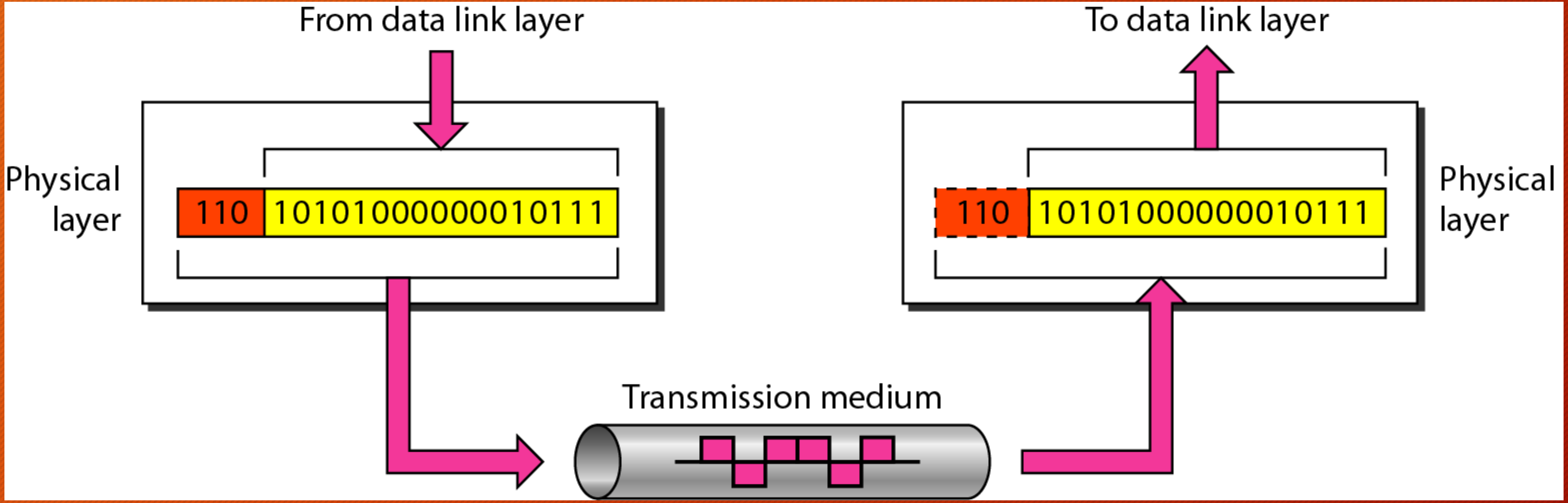
1) Layers l, 2, and 3-physical, data link, and network-are the network support layers.
    - They deal with the physical aspects of moving data from one device to another
    - Such as electrical specifications, physical connections, physical addressing, and transport timing and reliability.
2) Layers 5, 6, and 7-session, presentation, and application-can be thought of as the user support layers.
    - they allow interoperability among unrelated software systems.
3) Layer 4, the transport layer.
    - links the two subgroups and ensures that what the lower layers have transmitted is in a form that the upper layers can use.



An exchange using the OSI model

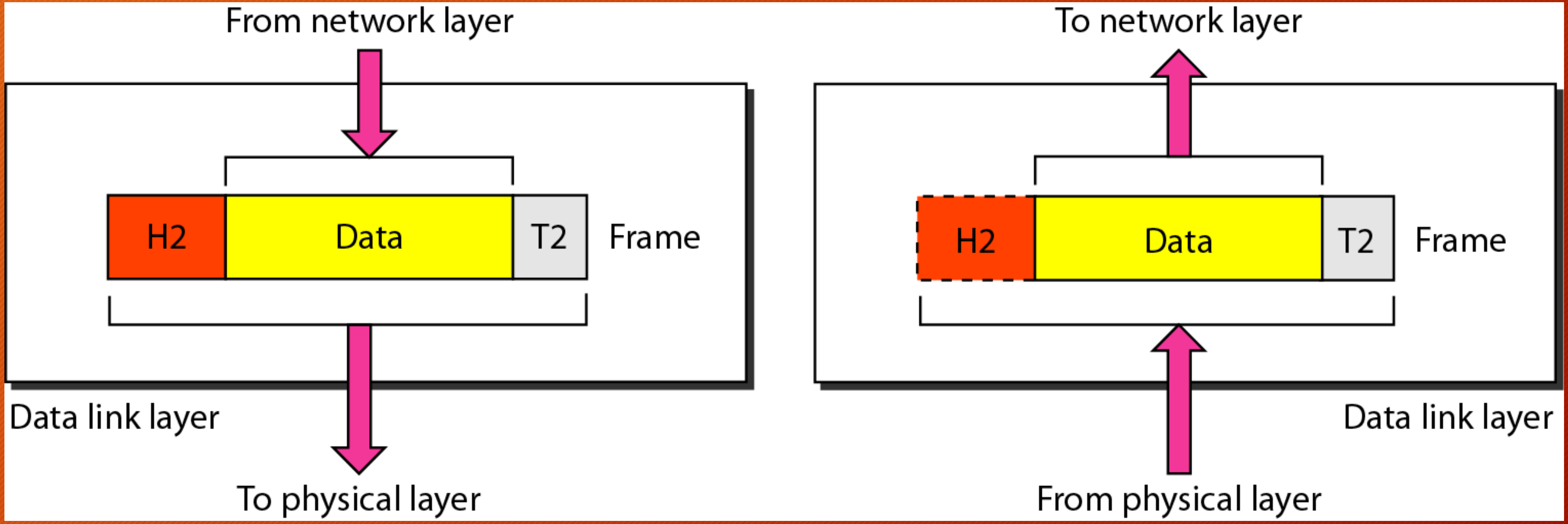
LAYERS IN THE OSI MODEL

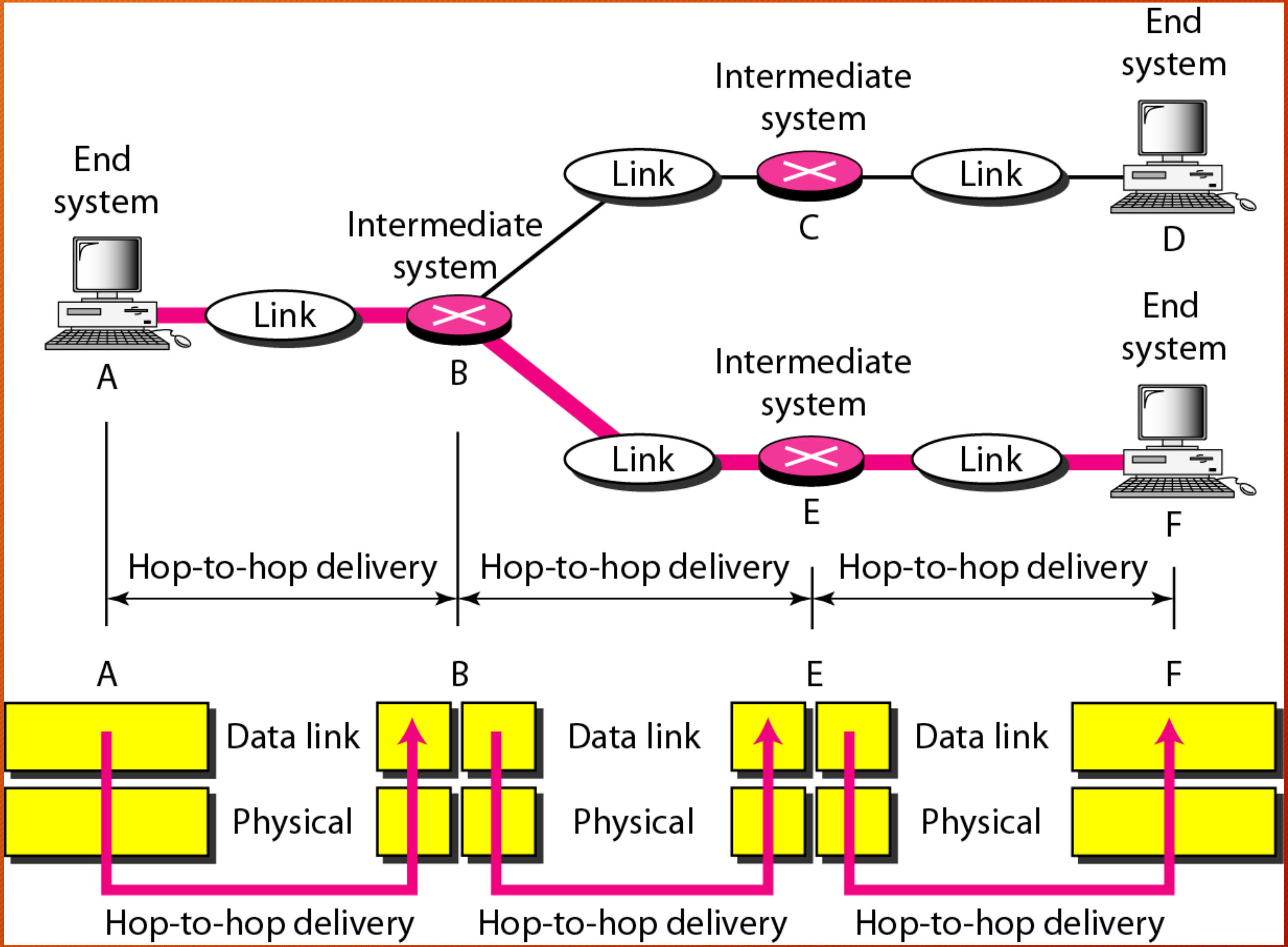
1. Physical Layer
   - The physical layer is responsible for movements of individual bits from one hop (node) to the next.



LAYERS IN THE OSI MODEL

2. Data Link Layer
   - The data link layer is responsible for moving frames from one hop (node) to the next.

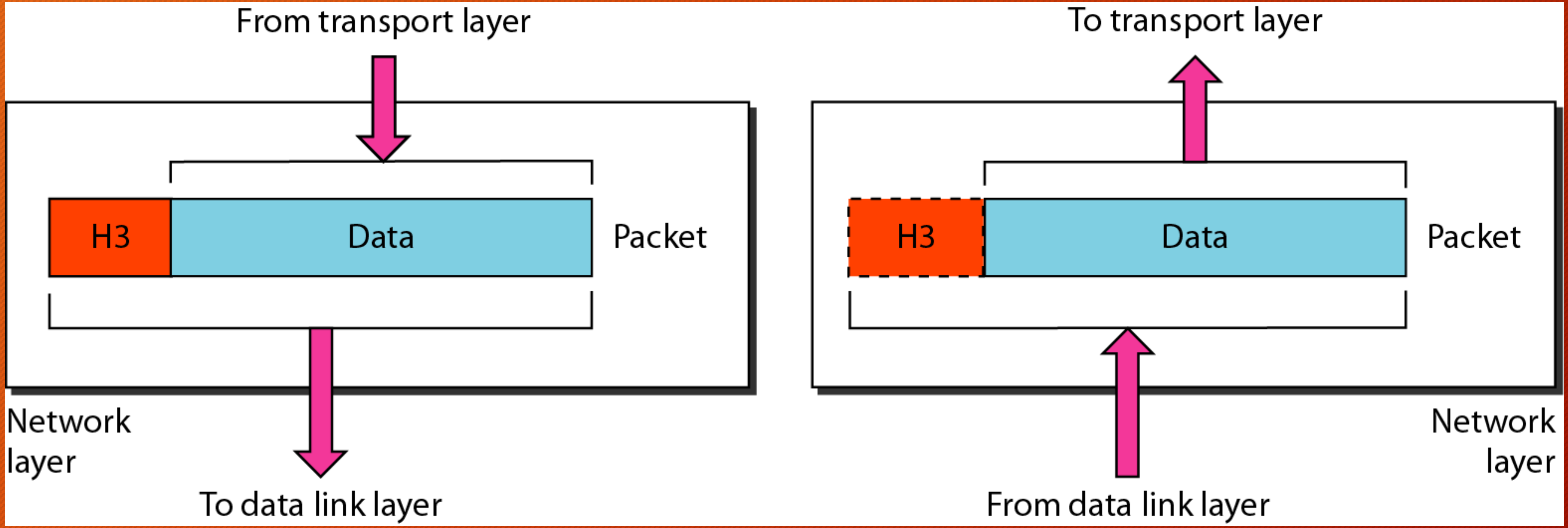


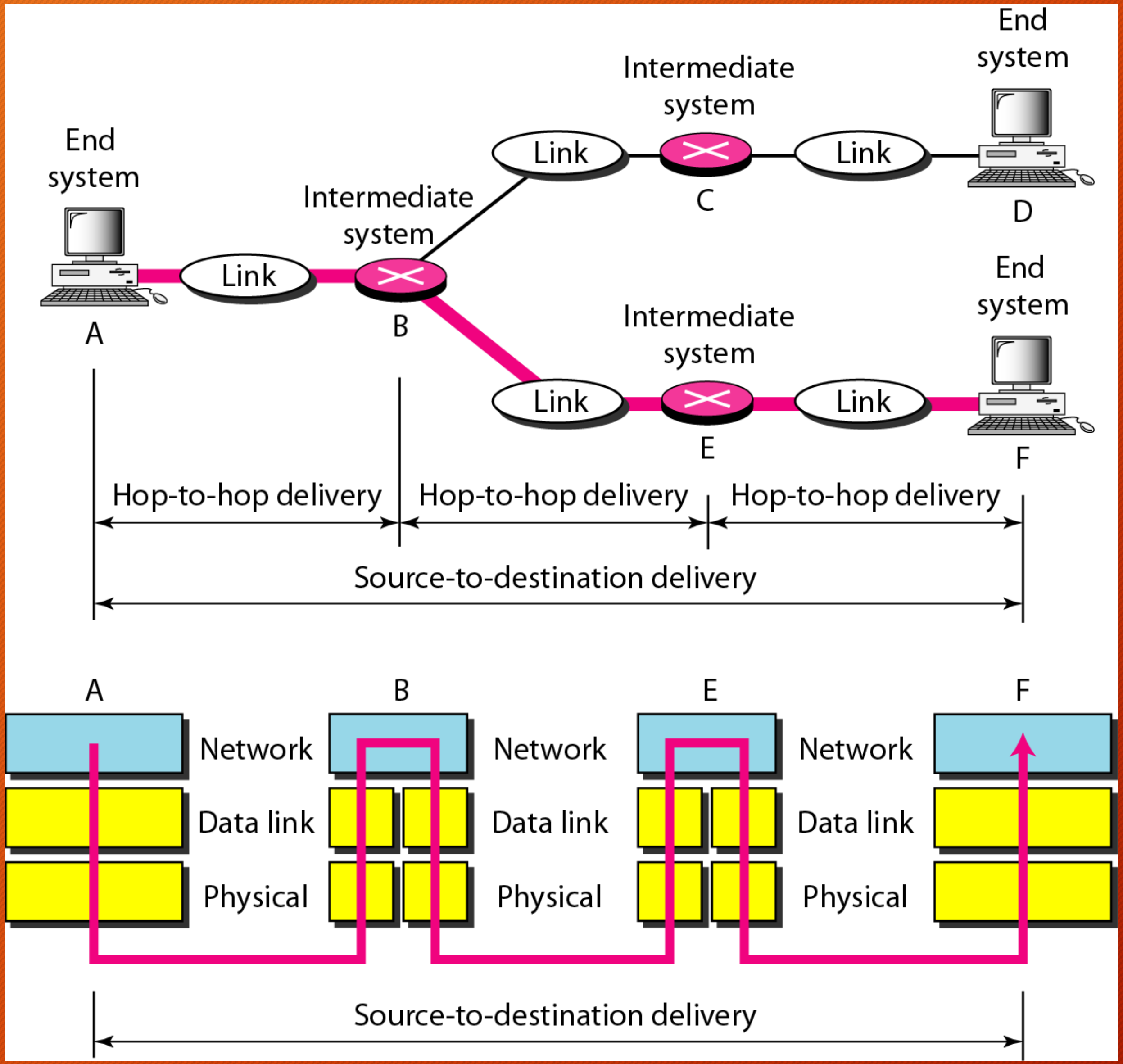


Hop-to-hop delivery

LAYERS IN THE OSI MODEL

3. Network Layer
   - The network layer is responsible for the delivery of individual packets from the source host to the destination host.

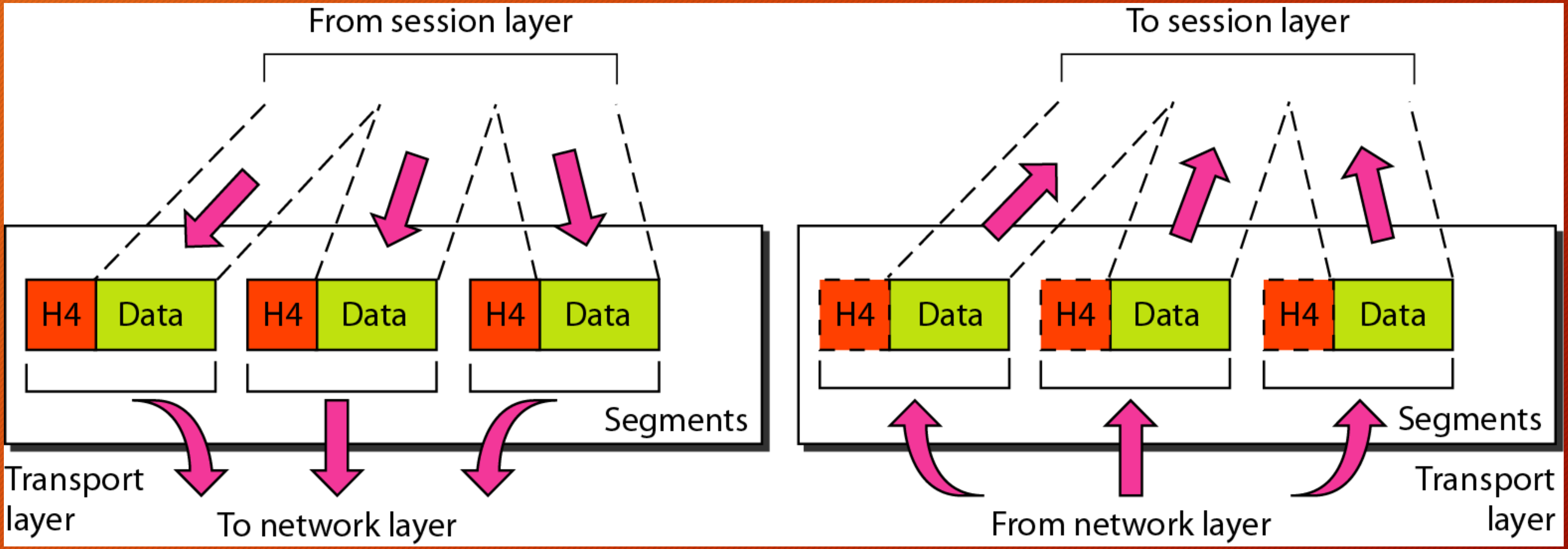


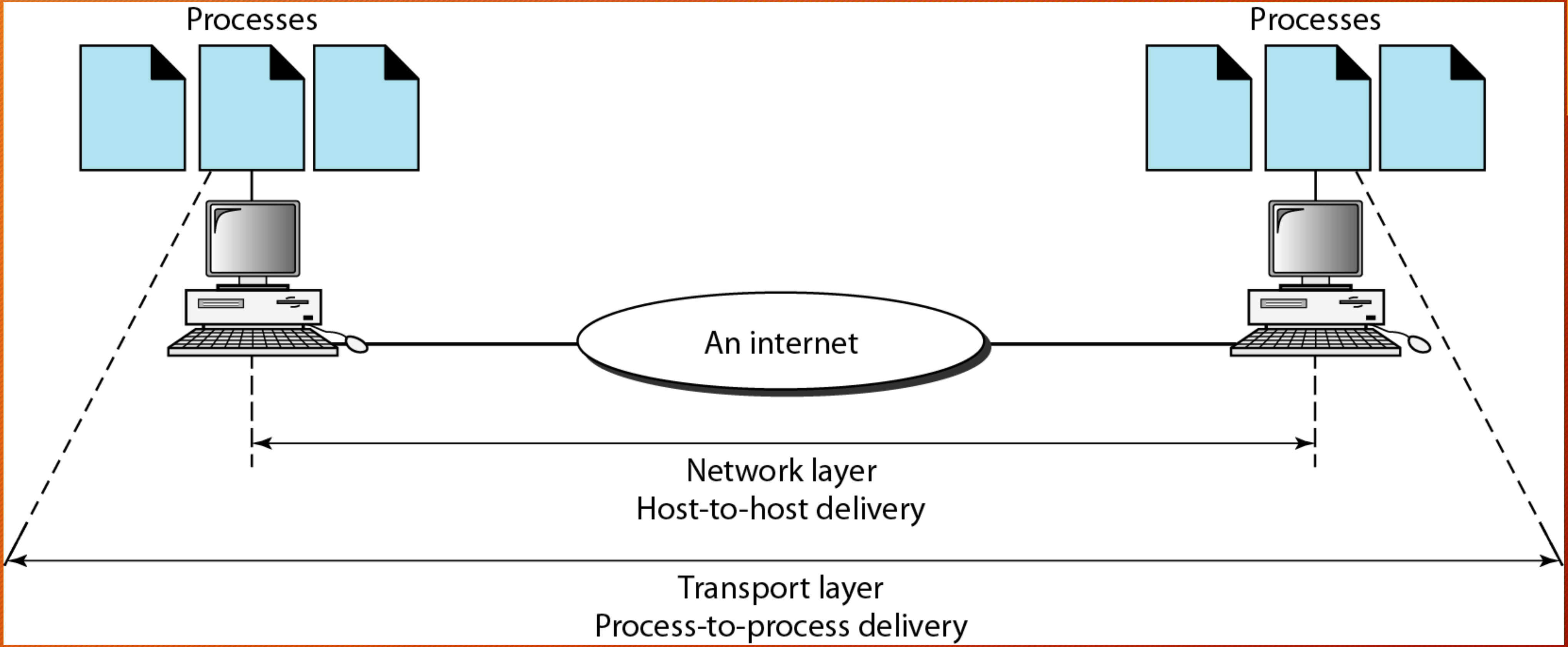


Source-to-destination delivery

LAYERS IN THE OSI MODEL

4. Transport Layer
   - The transport layer is responsible for the delivery of a message from one process to another.

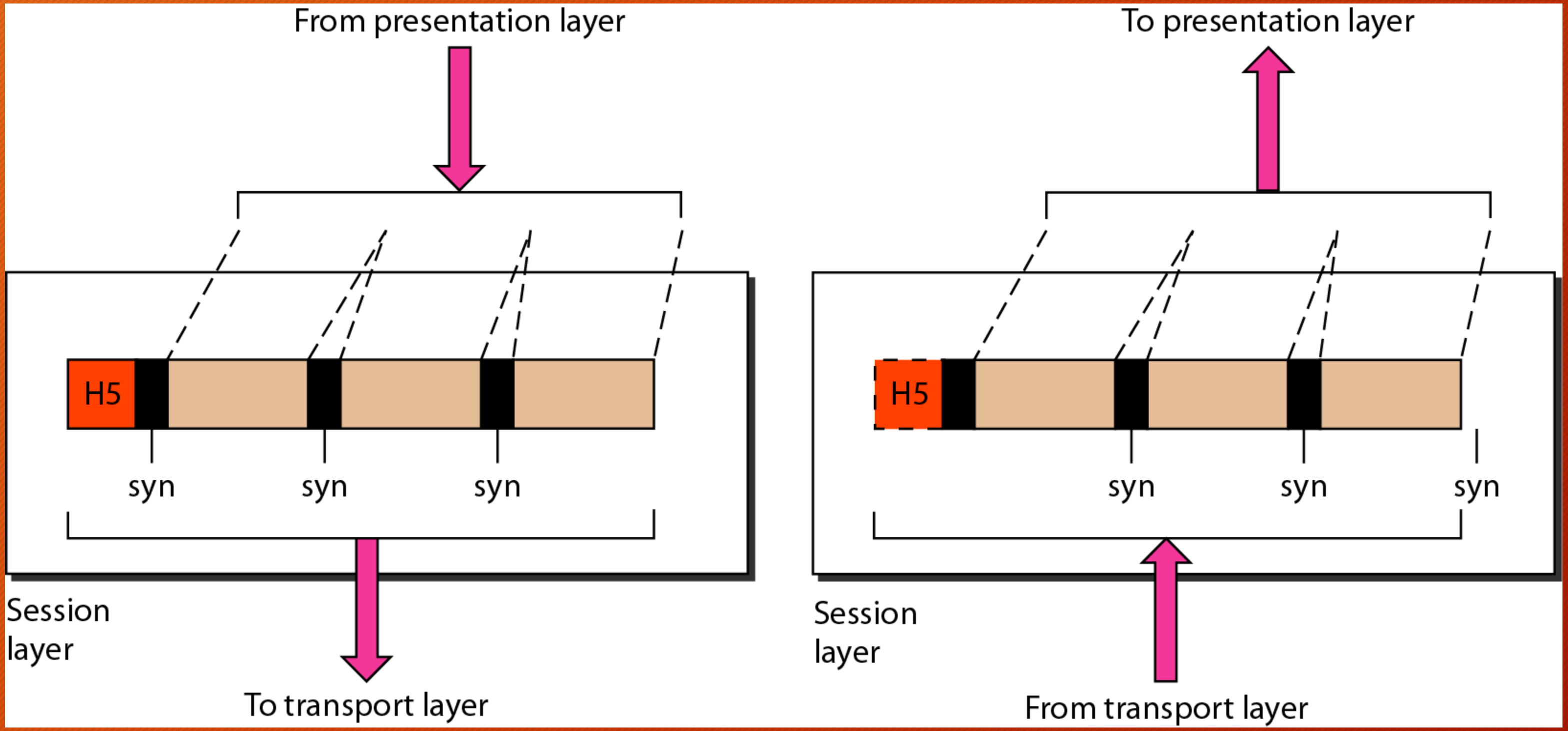




Reliable process-to-process delivery of a message

LAYERS IN THE OSI MODEL

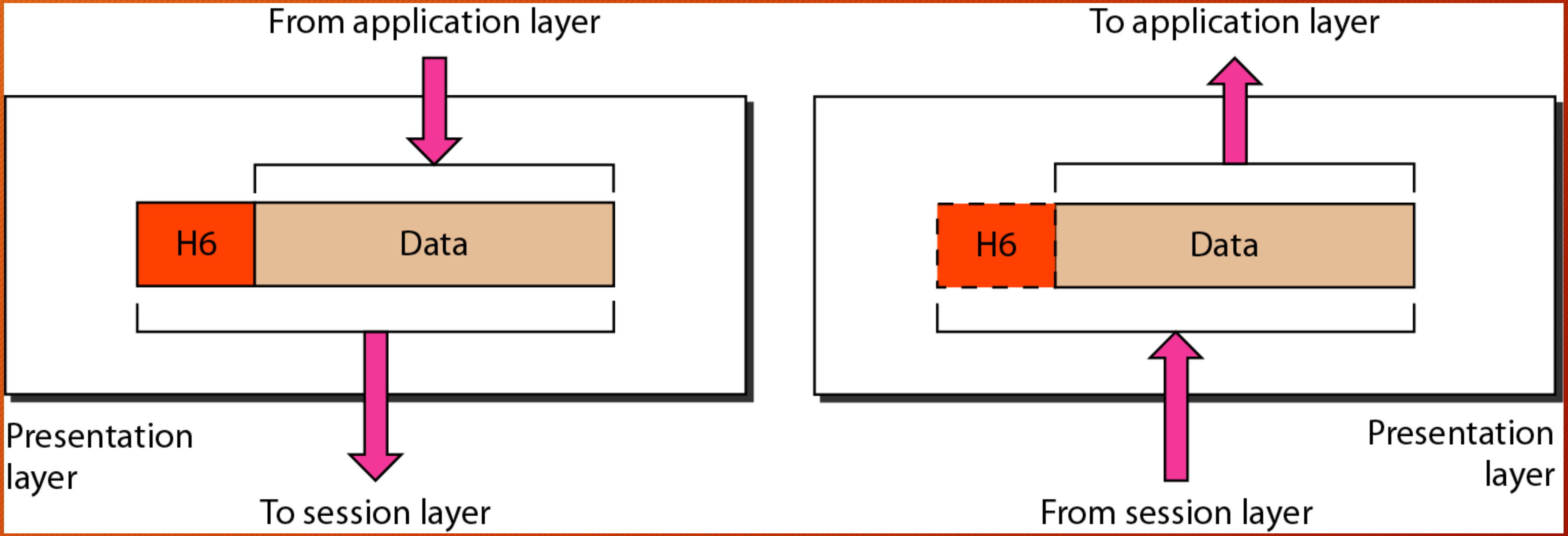
5. Session Layer
   - The session layer is responsible for dialog control and synchronization.



LAYERS IN THE OSI MODEL

6. Presentation Layer

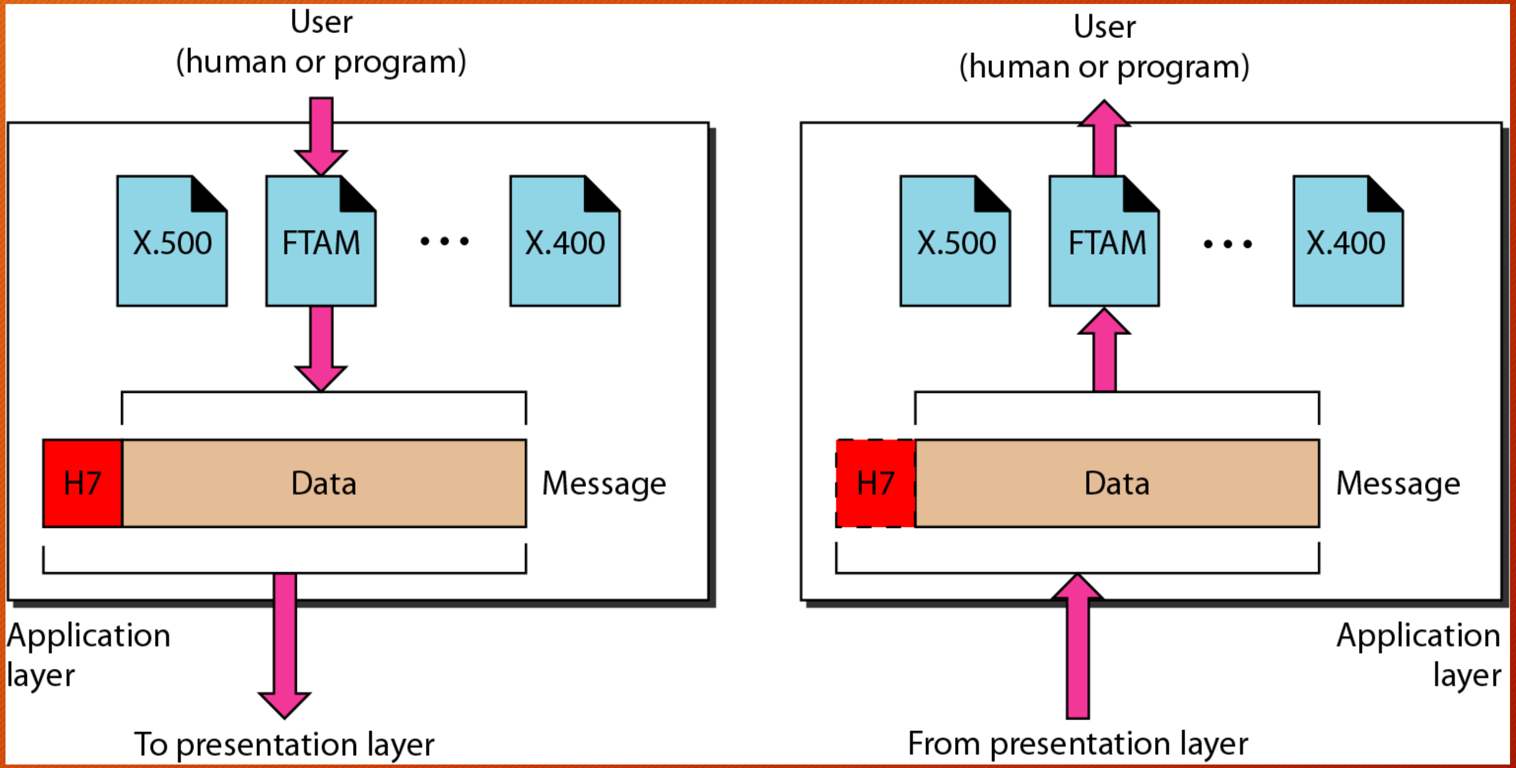
   The presentation layer is responsible for translation, compression, and encryption.



LAYERS IN THE OSI MODEL

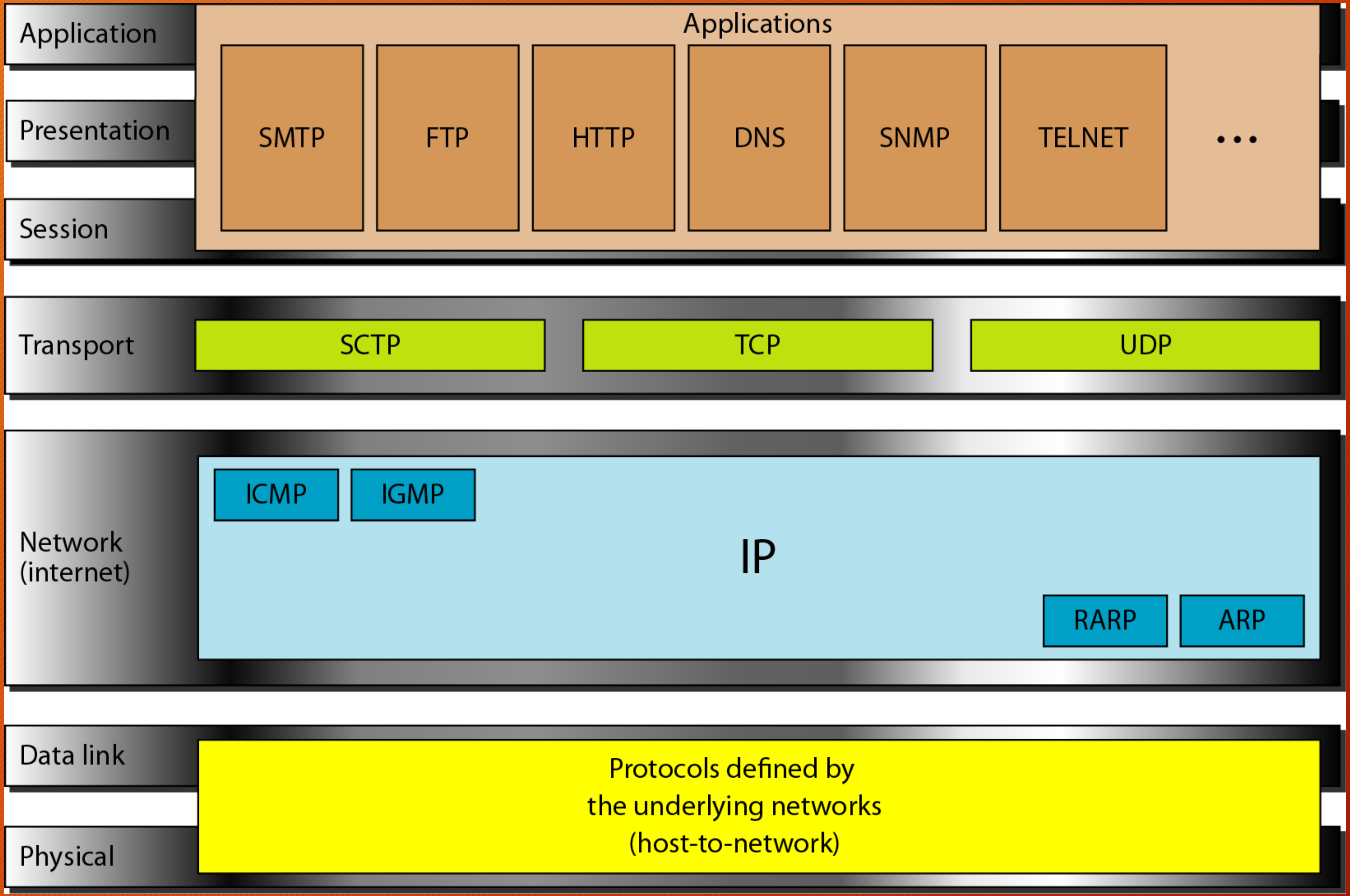
7. Application Layer

   The application layer is responsible for providing services to the user.



TCP/IP PROTOCOL SUITE

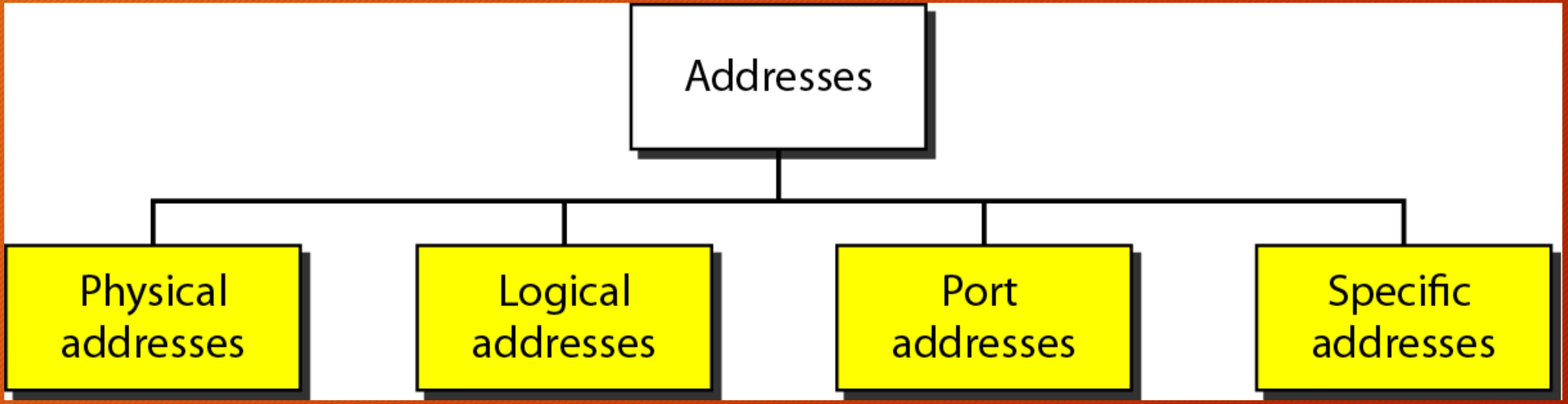
- The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application.
- However, when TCP/IP is compared to OSI, we can say that the host-to-network layer is equivalent to the combination of the physical and data link layers.
- The internet layer is equivalent to the network layer, and the application layer is roughly doing the job of the session, presentation, and application layers with the transport layer in TCP/IP taking care of part of the duties of the session layer.
- So, we assume that the TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application.



TCP/IP and OSI model

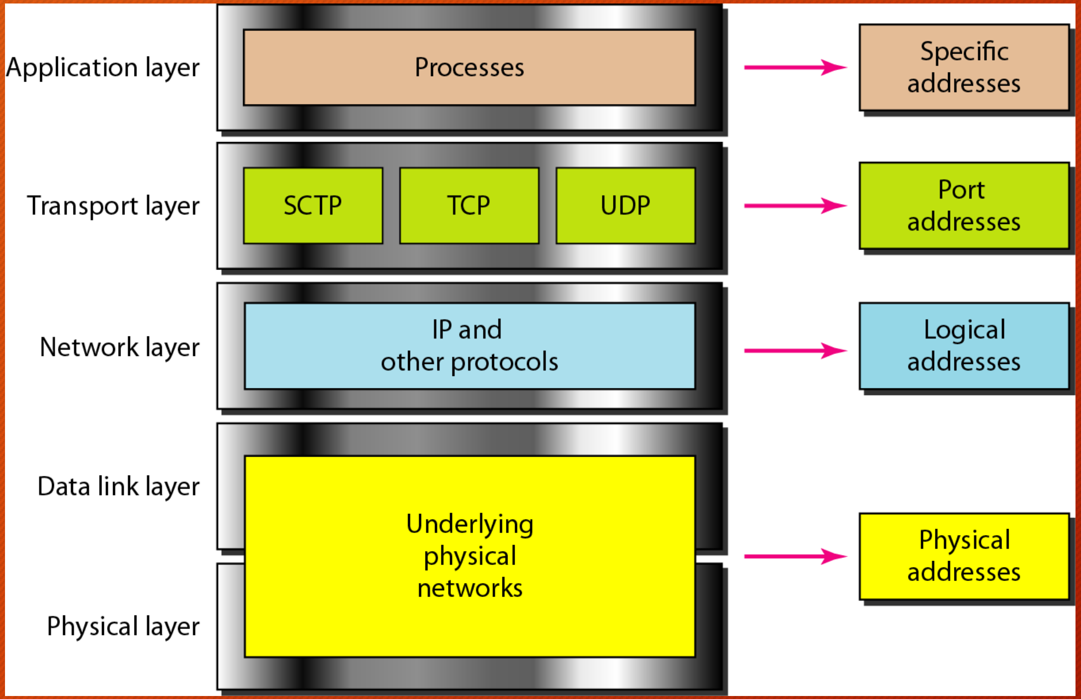
ADDRESSING

- *Four levels of addresses are used in an internet employing the TCP/IP protocols: physical, logical, port, and specific.*



ADDRESSING

- **Each address is related to a specific layer in the TCP/IP architecture, as shown here:**



Relationship of layers and addresses in TCP/IP

ADDRESSING

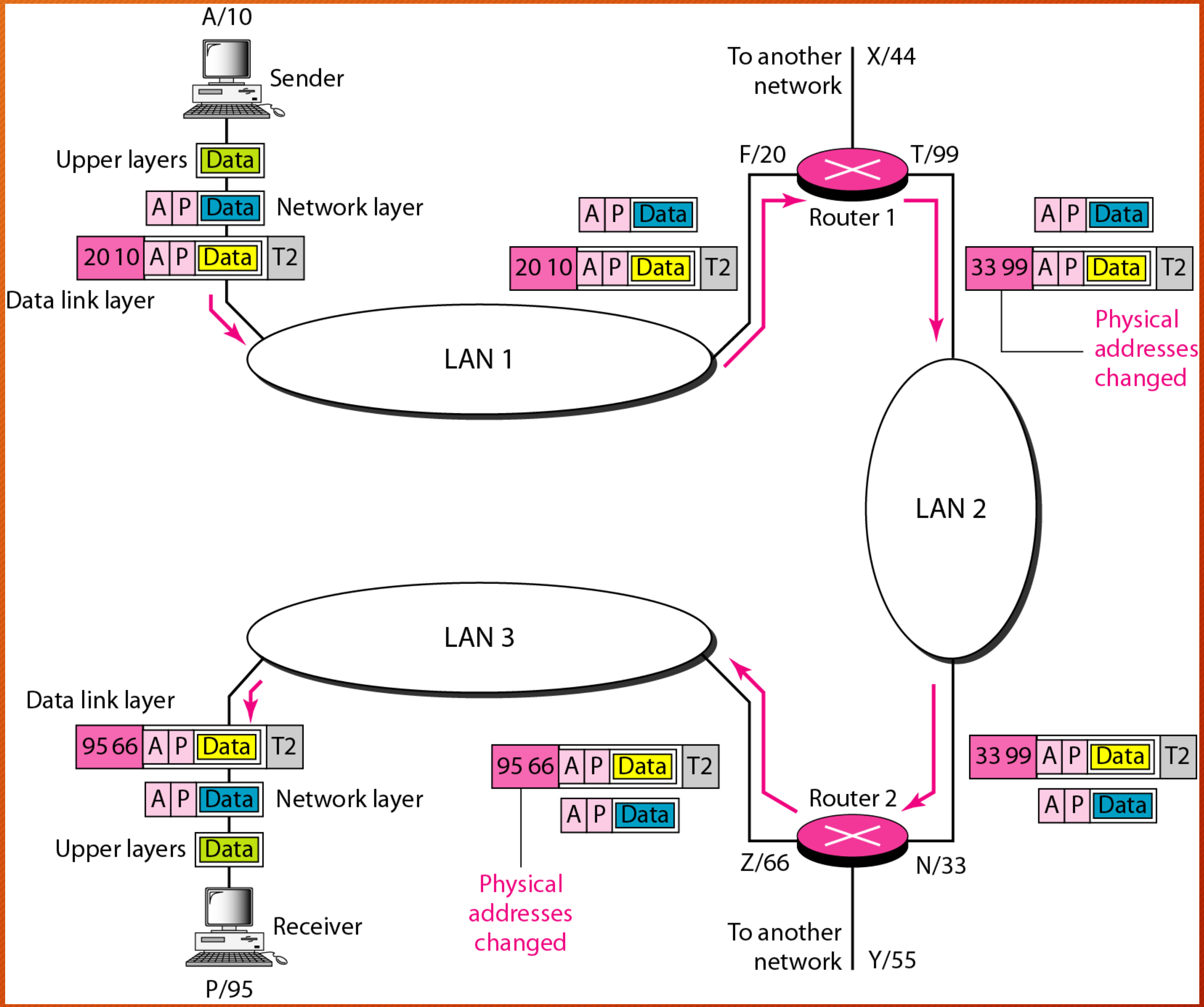
- *Most local-area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:*

07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address.

ADDRESSING

- The next figure *shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection.*



IP addresses