University of Southern California

Viterbi School of Engineering

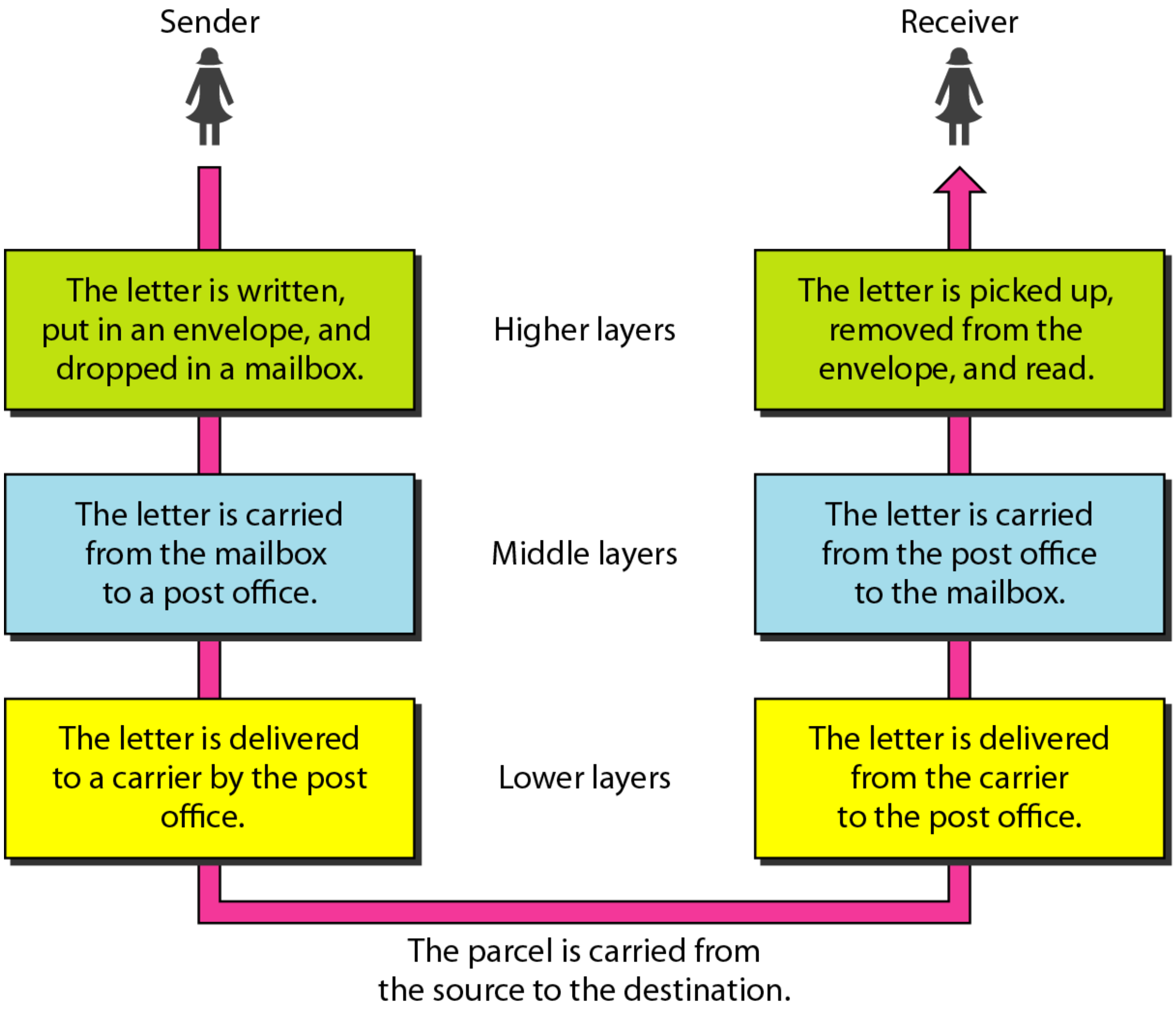
# EE450
# Computer Networks

## Network Protocols and Layering

Shahin Nazarian

Spring 2013

# An Example a Human Protocol: Tasks Involved in Sending a Letter



Sender

Receiver

The letter is written, put in an envelope, and dropped in a mailbox.

Higher layers

The letter is picked up, removed from the envelope, and read.

The letter is carried from the mailbox to a post office.

Middle layers

The letter is carried from the post office to the mailbox.

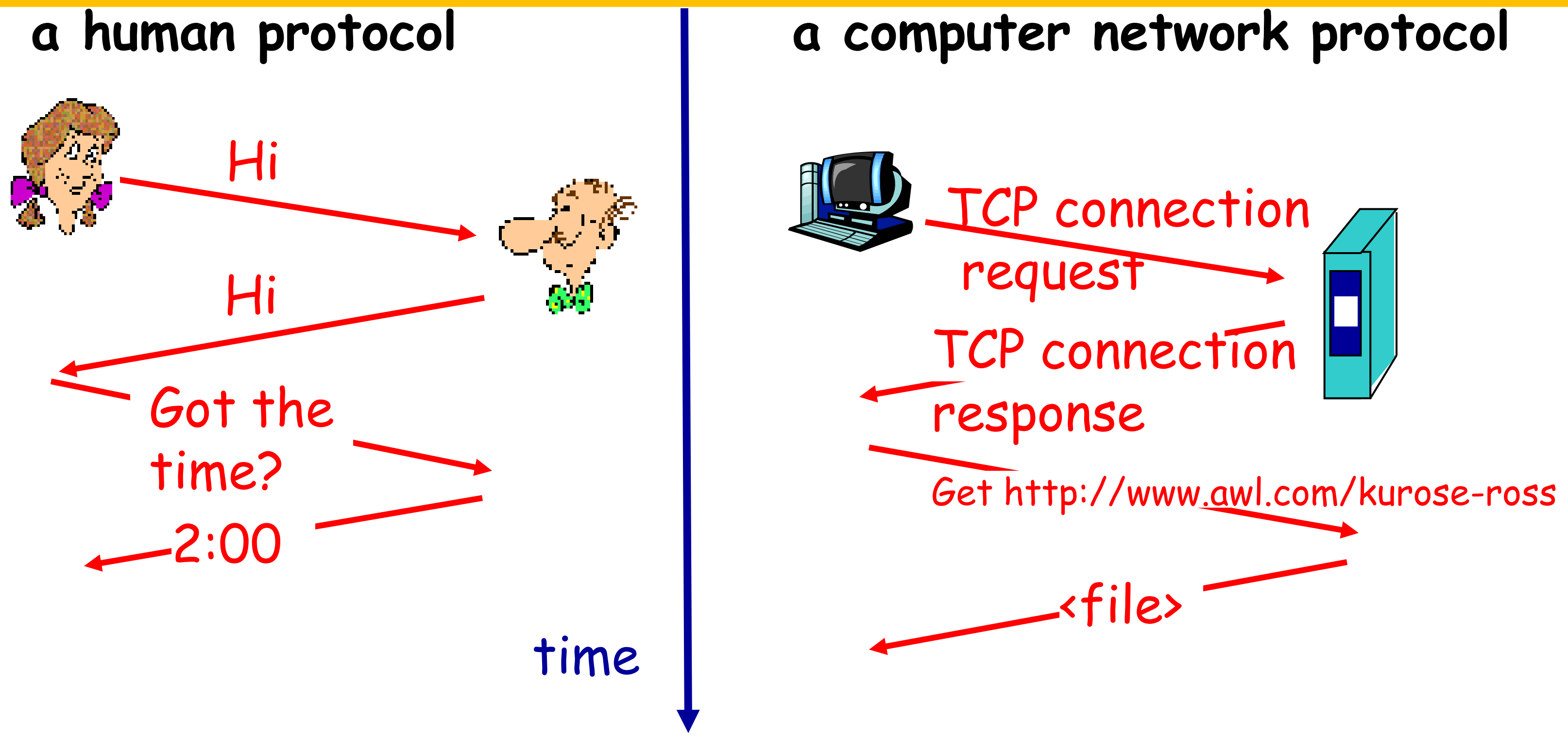
The letter is delivered to a carrier by the post office.

Lower layers

The letter is delivered from the carrier to the post office.

The parcel is carried from the source to the destination.

# What is a Protocol?



- A protocol can be defined as the rules, conventions, standards governing the syntax, semantics, and synchronization of communication. Another words, protocols are sets of rules, (or a sequence of events) that control or enable (preferably reliable and) recognizable transfer of information among communication end points

# Protocol

- Note that protocol represents an agreement among communication devices. Without a protocol devices may be connected but not communicating
- Protocols are devised such that the system including all its hardware and software products (which in general are from different vendors) will be able to operate as long as they adhere to the standards in the protocol
- This guarantees marketability for vendor products
- One main shortcoming of a protocol is that it takes a long time to be devised and standardized, therefore by the time the protocol is ready to be commercialized, the technology may be outdated

# ISO and OSI

- The International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on international standards
  - ISO was established in 1947

- The Open Systems Interconnection (OSI) model is an ISO standard that covers all aspects of network communications is OSI was first introduced in the late 1970s

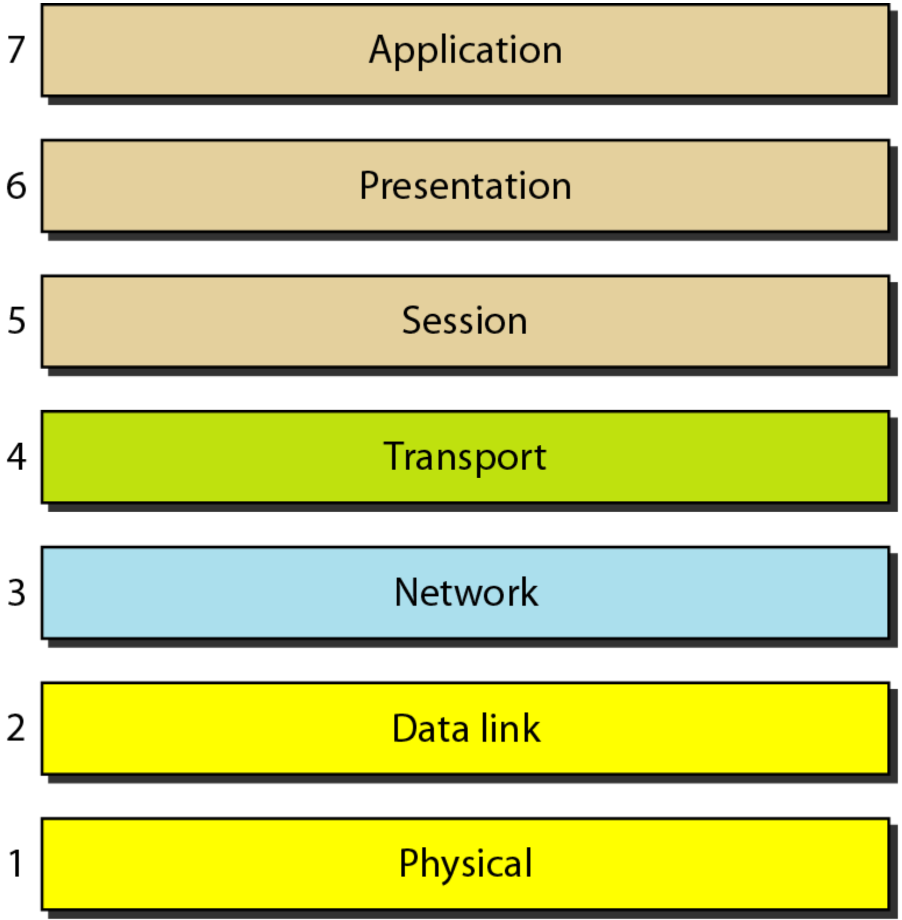
**ISO is the organization, OSI is the model**

# Network Expectations and Goals

- From the user point of view, i.e., application programmer or end user: network delivery should be reliable, and recognizable and in a reasonable (and predictable) time
- The network designer looks for cost-effective designs to make the design affordable; and makes sure the resources such the CPUs, memories and bandwidth are efficiently and fairly used
- The network provider cares about billing issues, simplicity of use, straightforward fault detection and isolation (i.e., the faults that force down the network.) It also cares about administrative and management issues

# OSI Layers

- OSI is the most popular packet-based structure of layers, or protocol stack
- The functionality of communications in OSI is divided into 7 layers which are counted from bottom to top



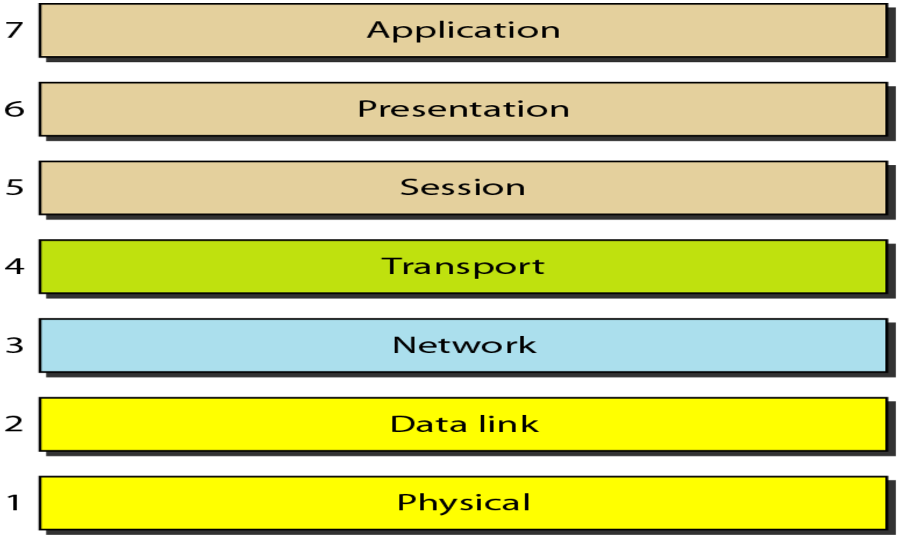
# Why Layering?

- Networking (communications) is overall a very complicated task and involves very complex steps such as transmission, error detection and recovery, routing, synchronization, etc.
- One way to handle networking was to write a big piece of code (program) that would perform all the tasks, without organizing it into subroutines, sub-functions. This would be very hard to enhance in case needed and/or debug in case of error in the program

**Networking Tasks in one layer**

# Why Layering? (Cont.)

- Another choice the network designers had was to divide the networking tasks into subtasks and write programs each of which responsible only for one of those subtasks. Each of those subtasks or the respective program is referred to as a layer
- Each layer is only responsible for the set of functions related to its layer or subtasks. It will perform those functions and pass the results to the layer above it, until it reaches to the top layer and then to the end user



# Why Layering? (Cont.)

- One main advantage of layering is that each layer (i.e., its code or program) is independent of other layers, meaning if you would like to modify or enhance some parts of a certain layer, you only need to pull out that layer, make the changes and put it back without worrying about other layers

- Question: How is this level of layer independence possible?

# Why Layering? (Cont.)

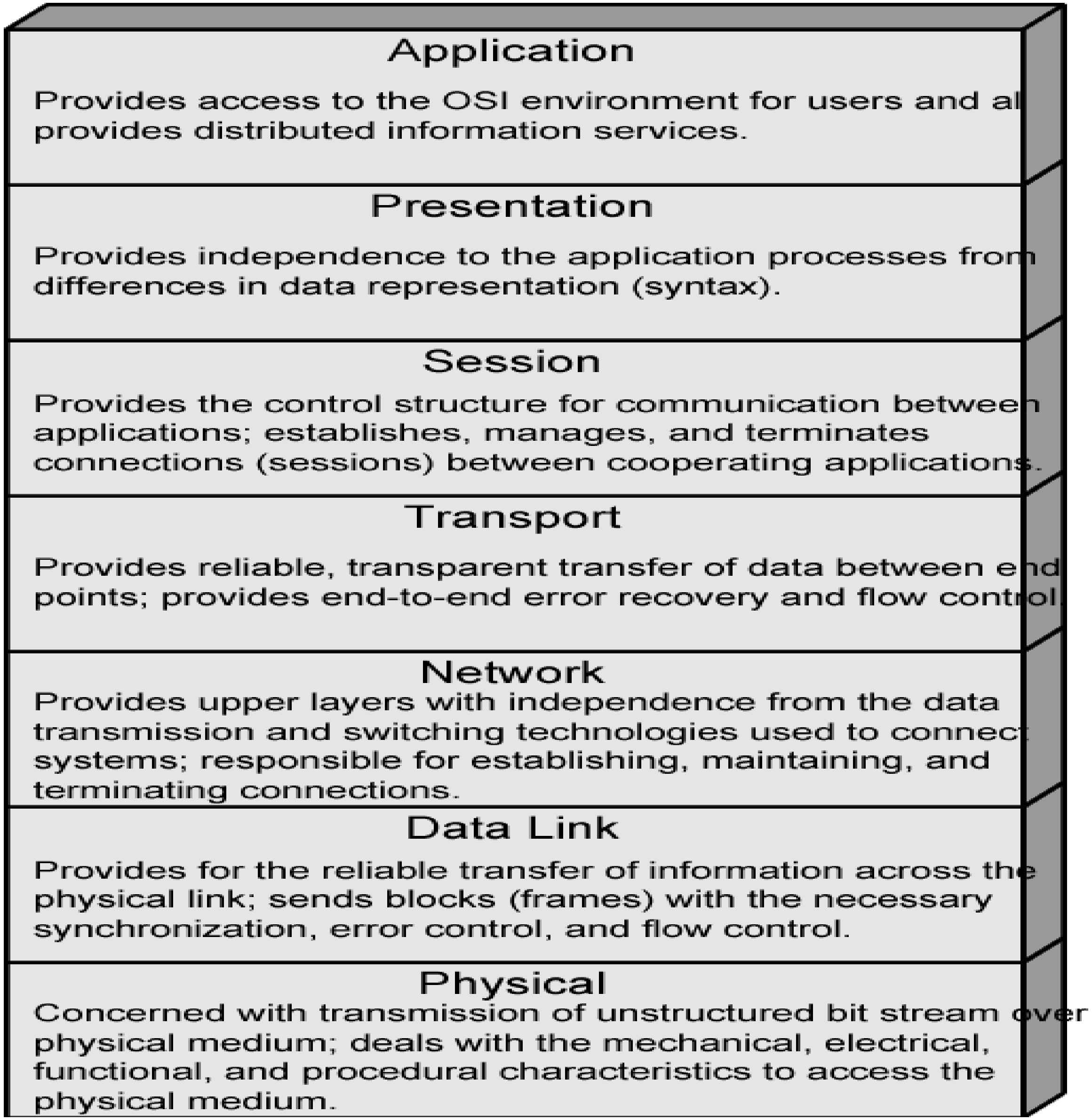
- Note that layers perform their functions independently and each layer provides the services to the layer above it, e.g., layer 3 gives the services to layer 4. Layer 4 is not concerned how layer 3 performed the functions, all matters to layer 4 is the end results which layer 3 passes to layer 4

- This is similar to the school grading system. What matters is the letter grade, and in the future it won't matter any longer how the student received it!

# Why 7 Layers?!

- Why not 70 or 2 or 1 layer for OSI?
- If it had 70 layers, each layer would've been responsible for a smaller set of functionalities, meaning writing the code for those layers would've been more straightforward, however with number of headers 10 times as big, which means it had a large overhead
- If it had one layer or two layers, the overhead would have been less, but the layers more complicated
- Having 7 layers was the designers' and researchers' decision, which happened after doing a lot of research, experiments and simulations

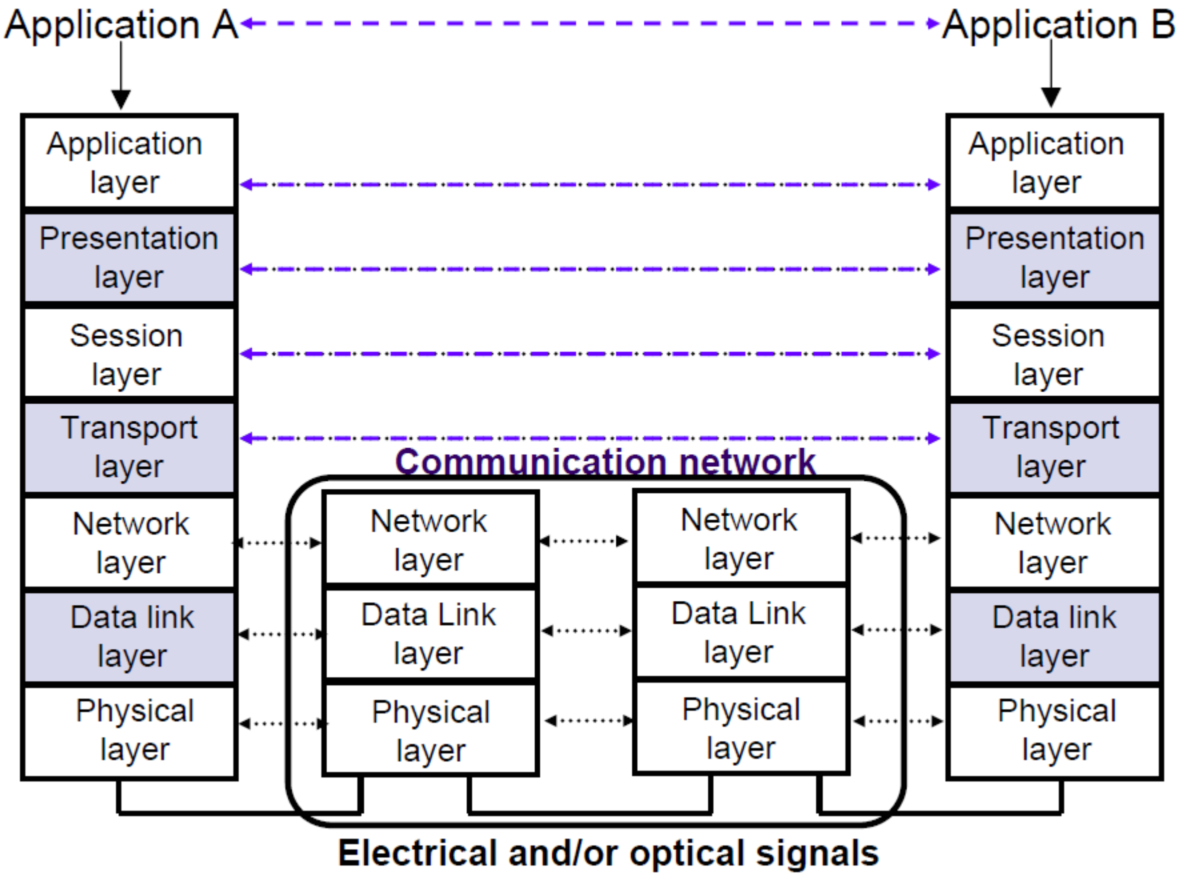
# OSI Layer Function Summary



| Layer # | | |
|---|---|---|
| 7 | Application | Provides access to the OSI environment for users and al provides distributed information services. |
| 6 | Presentation | Provides independence to the application processes from differences in data representation (syntax). |
| 5 | Session | Provides the control structure for communication between applications; establishes, manages, and terminates connections (sessions) between cooperating applications. |
| 4 | Transport | Provides reliable, transparent transfer of data between end points; provides end-to-end error recovery and flow control. |
| 3 | Network | Provides upper layers with independence from the data transmission and switching technologies used to connect systems; responsible for establishing, maintaining, and terminating connections. |
| 2 | Data Link | Provides for the reliable transfer of information across the physical link; sends blocks (frames) with the necessary synchronization, error control, and flow control. |
| 1 | Physical | Concerned with transmission of unstructured bit stream over physical medium; deals with the mechanical, electrical, functional, and procedural characteristics to access the physical medium. |

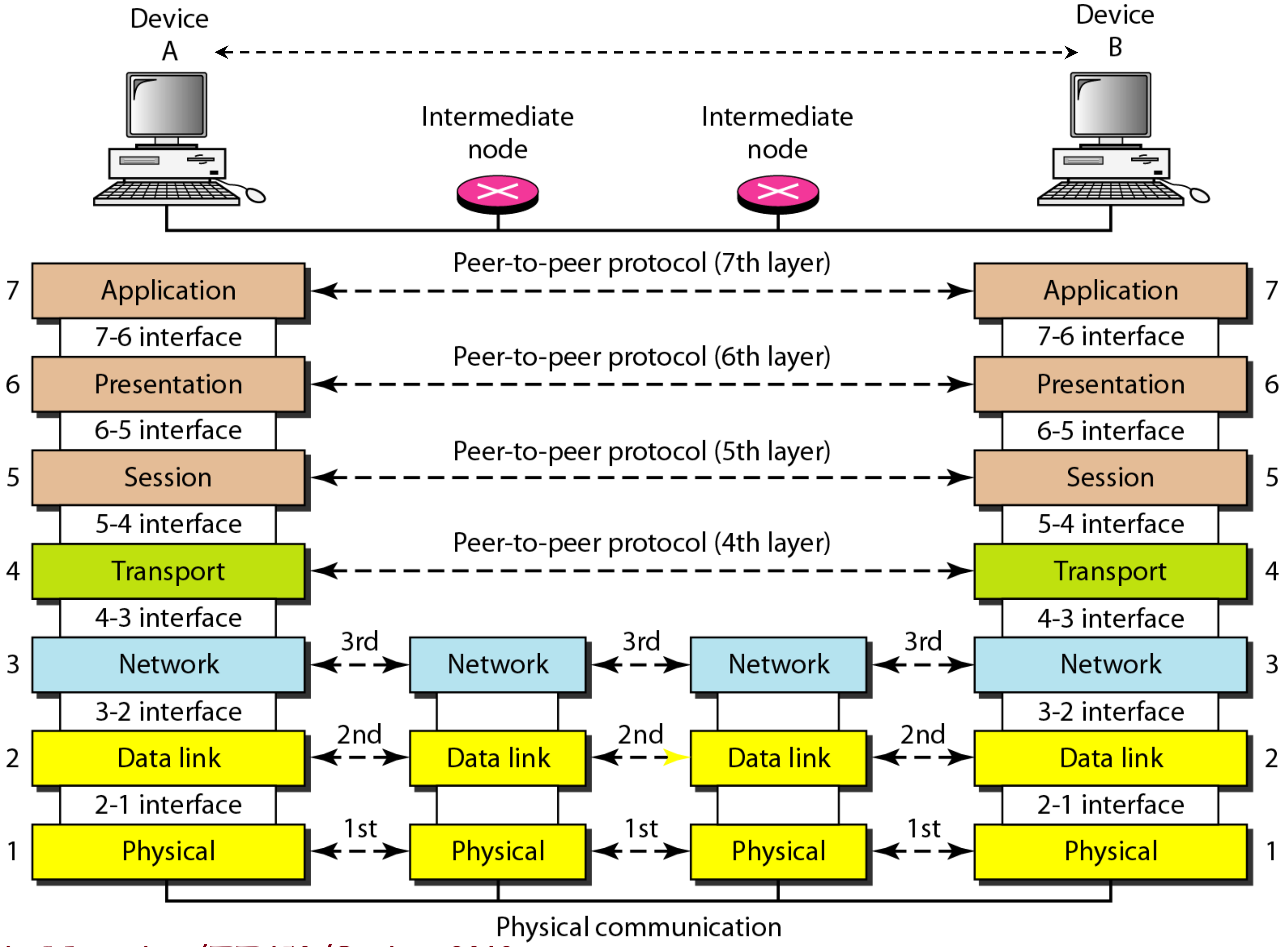
# Interaction between OSI Layers

- OSI is a seven-layer abstract reference model for a network architecture

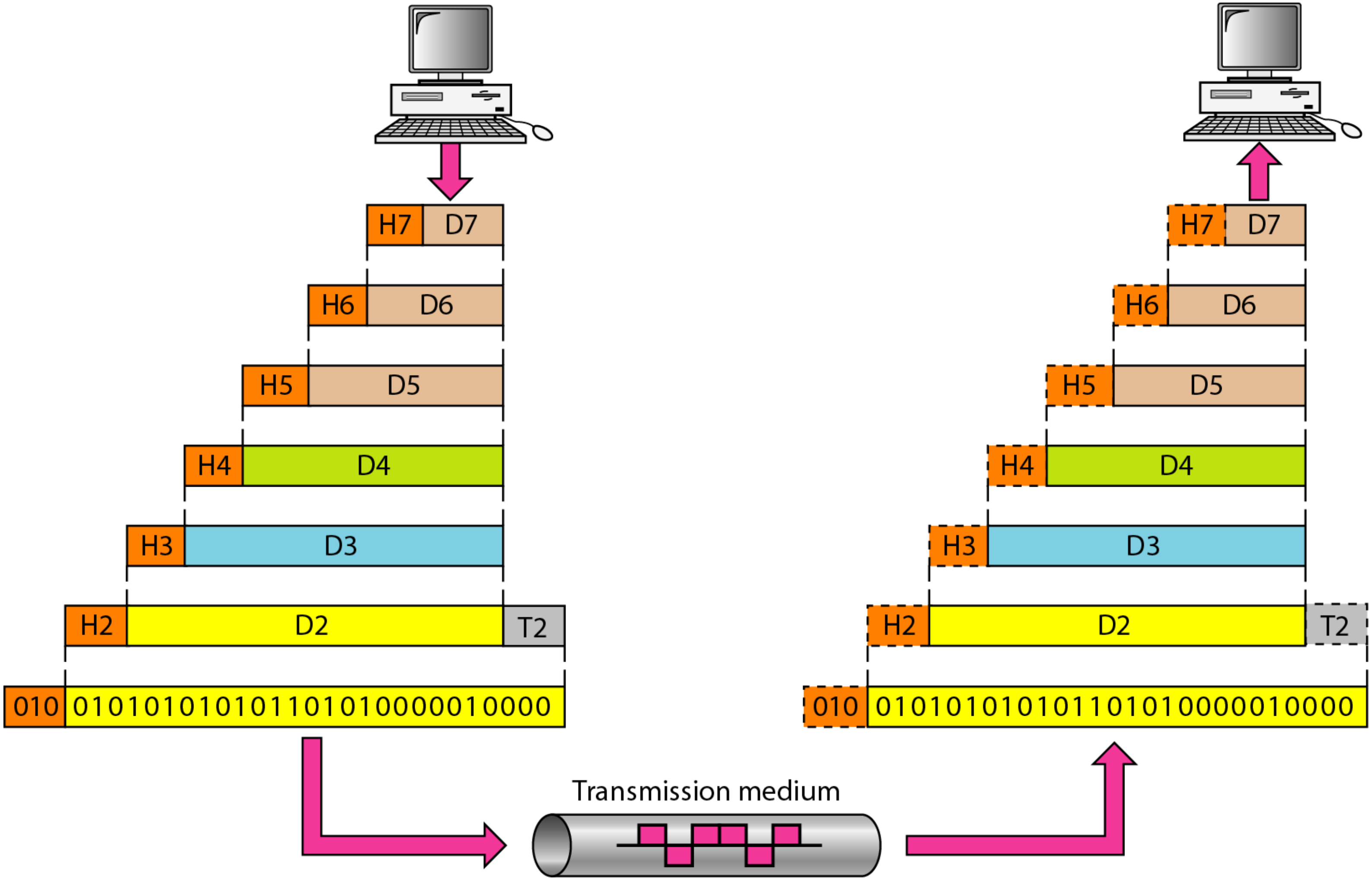


# Interaction between OSI Layers (Cont.)

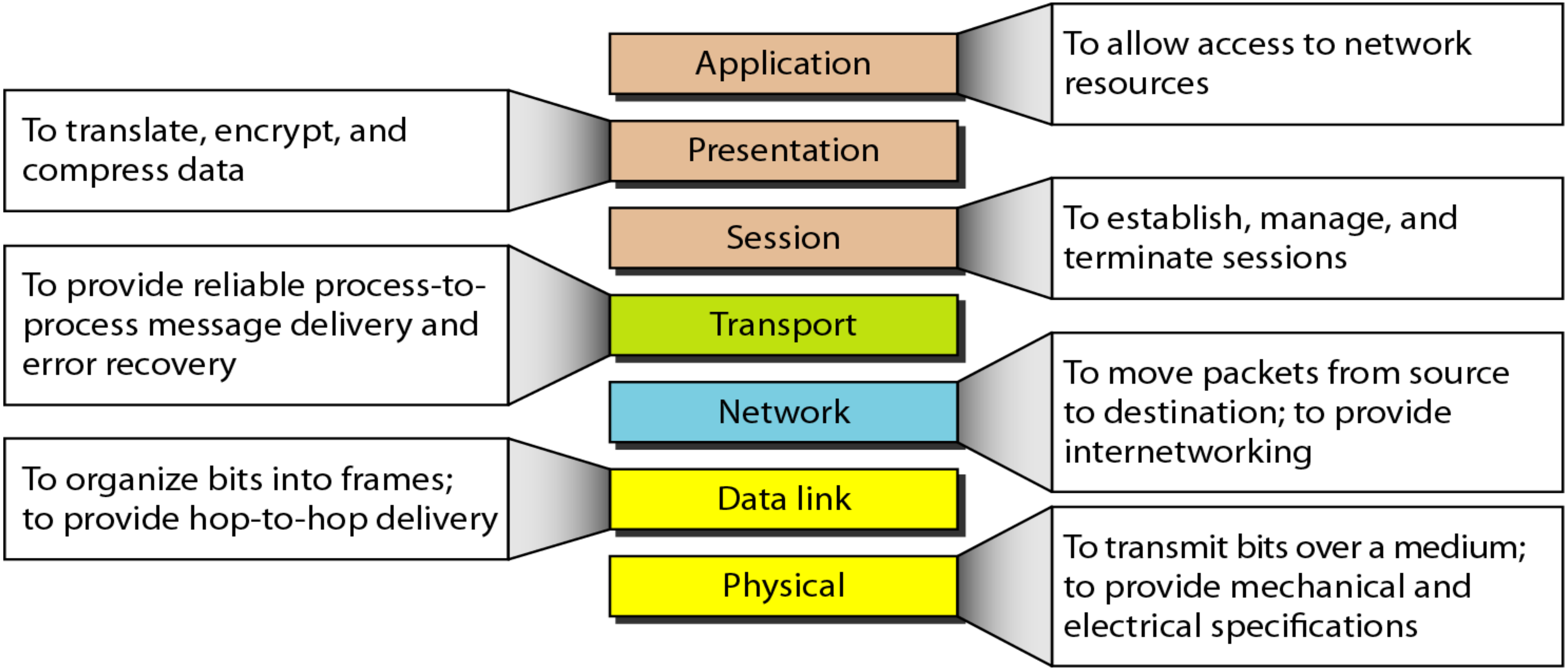
## Interaction between nodes considering intermediate nodes



# Exchange Using OSI Model



# Layer Summary

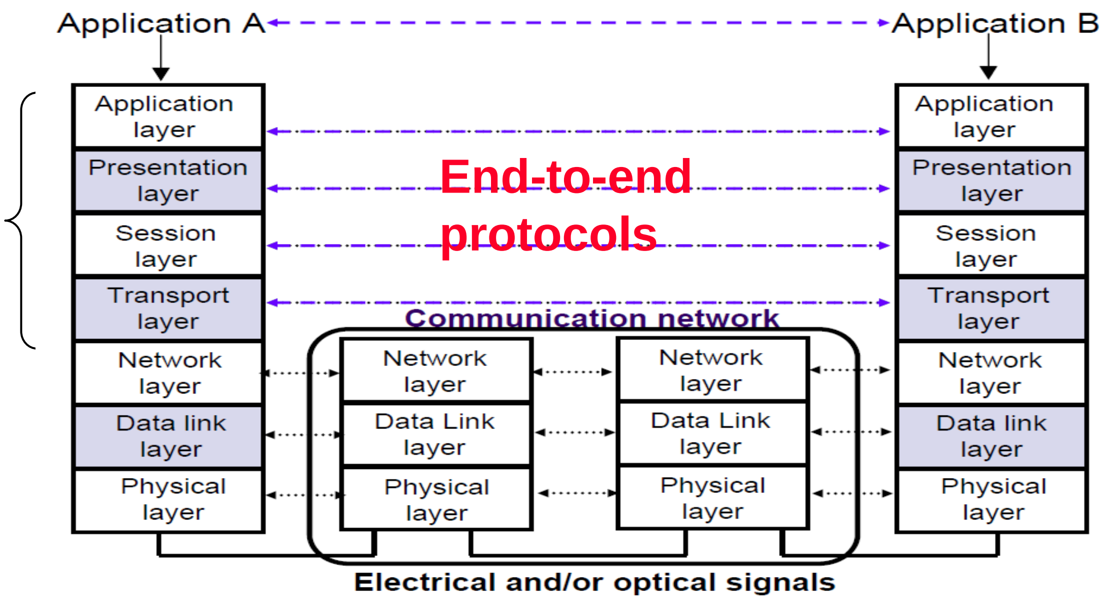

Application
To allow access to network resources
Presentation
To translate, encrypt, and compress data
Session
To establish, manage, and terminate sessions
Transport
To provide reliable process-to-process message delivery and error recovery
Network
To move packets from source to destination; to provide internetworking
Data link
To organize bits into frames; to provide hop-to-hop delivery
Physical
To transmit bits over a medium; to provide mechanical and electrical specifications

- **Example: Suppose A attempts to send an encrypted file to B. The encryption is done in the Presentation layer, because changing the format of the file and adding a key to encrypt it is part of Presentation functionality. The Network layer does not need to know and does not worry whether the file is encrypted or not, or whether it's an email, a photo, etc.!**

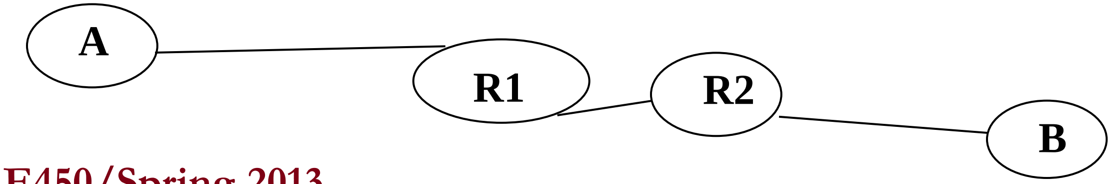
# End-to-End Layers, End-to-End Protocols

- End-to-end layers are not implemented in the intermediate nodes or routers

End-to-end layers

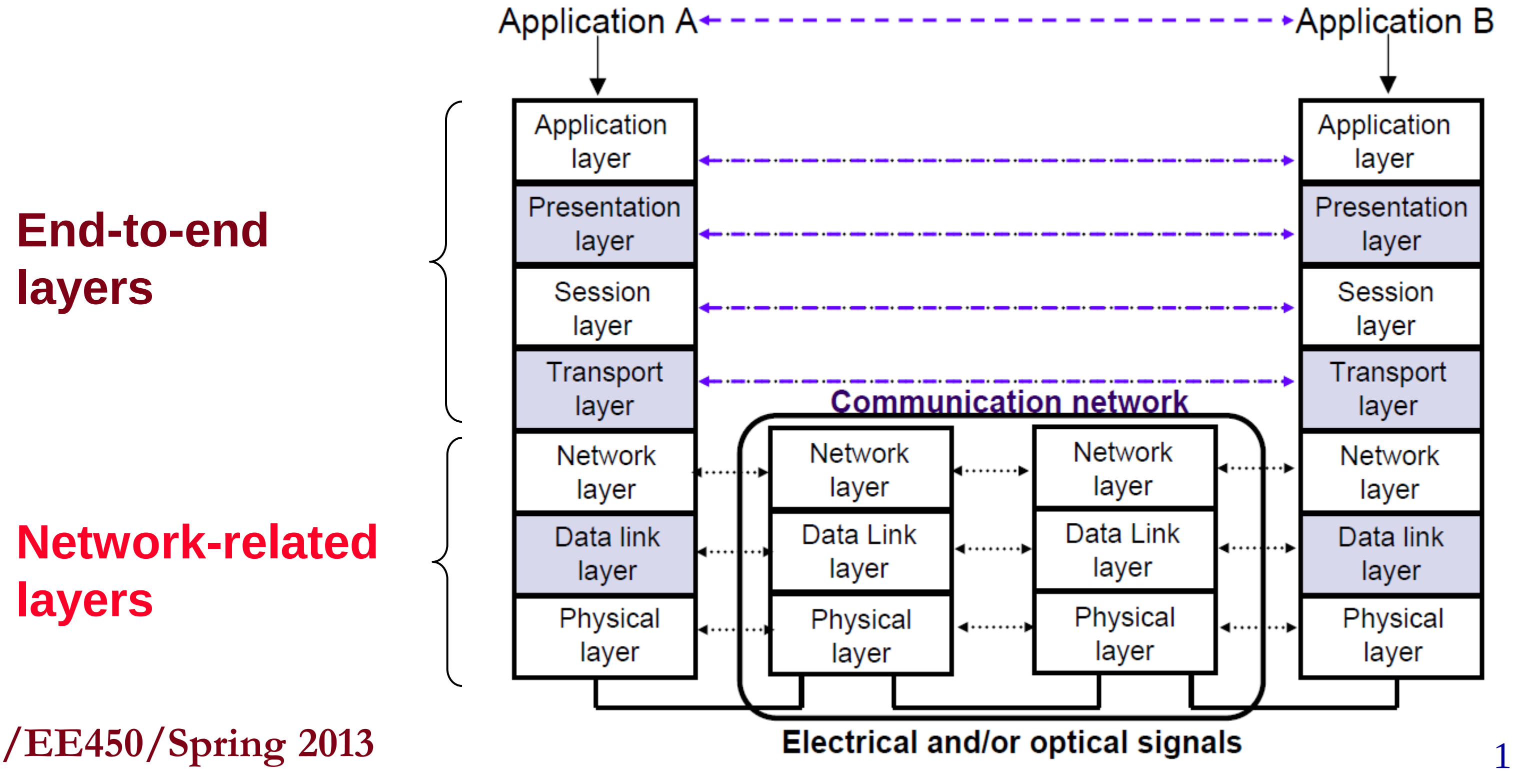


End-to-end protocols



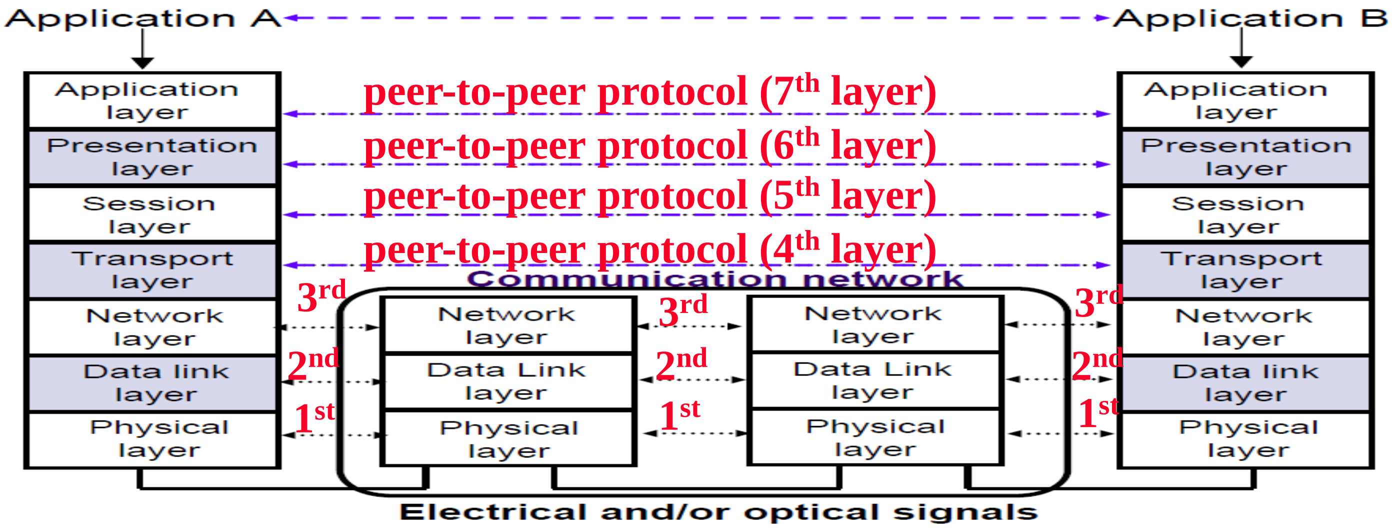
# Network-Related Layers

- The lower 3 layers are Network-related layers. This means they are implemented in every node, in end hosts and the intermediate routers all
- Remember that each layer is using a set of rules (protocols)



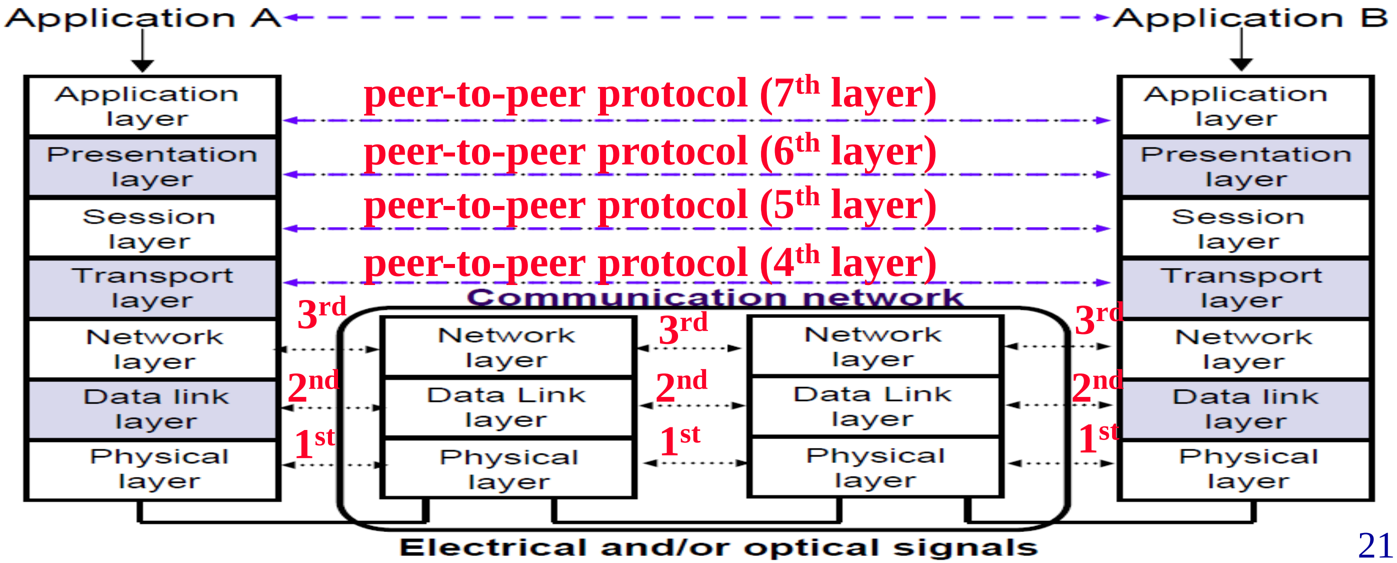
# Peer-to-Peer Protocols

- The dashed lines in the following figure
- Question: Transport layer of A uses a set of rules. Which layer the upper one (Session layer) or the bottom one (Network layer) will interpret those protocols to understand what they are all about? None
  - Answer: B's Transport layer (aka peer of A's Transport layer)
- Each Peer-to-peer protocol is just an agreement on how we are going to interpret that protocol



# Peer-to-Peer Protocols (Cont.)

- Again note that peer layers communicate via protocols and there exists no physical communication between them
- Example: For the Application layer, its peer Application layer understands it, e.g., for HTTP in the application layer of A, there is another HTTP in the application layer of B that understands it. Similarly for SMTP the other Application layer also needs to know SMTP
- Only solid lines are the physical connections between end system A to router 1, router 1 to router 2 and router 2 to the end system B. All the rest of the lines are dashed lines (do not exist physically)



# Peer-to-Peer Protocols (Cont.)

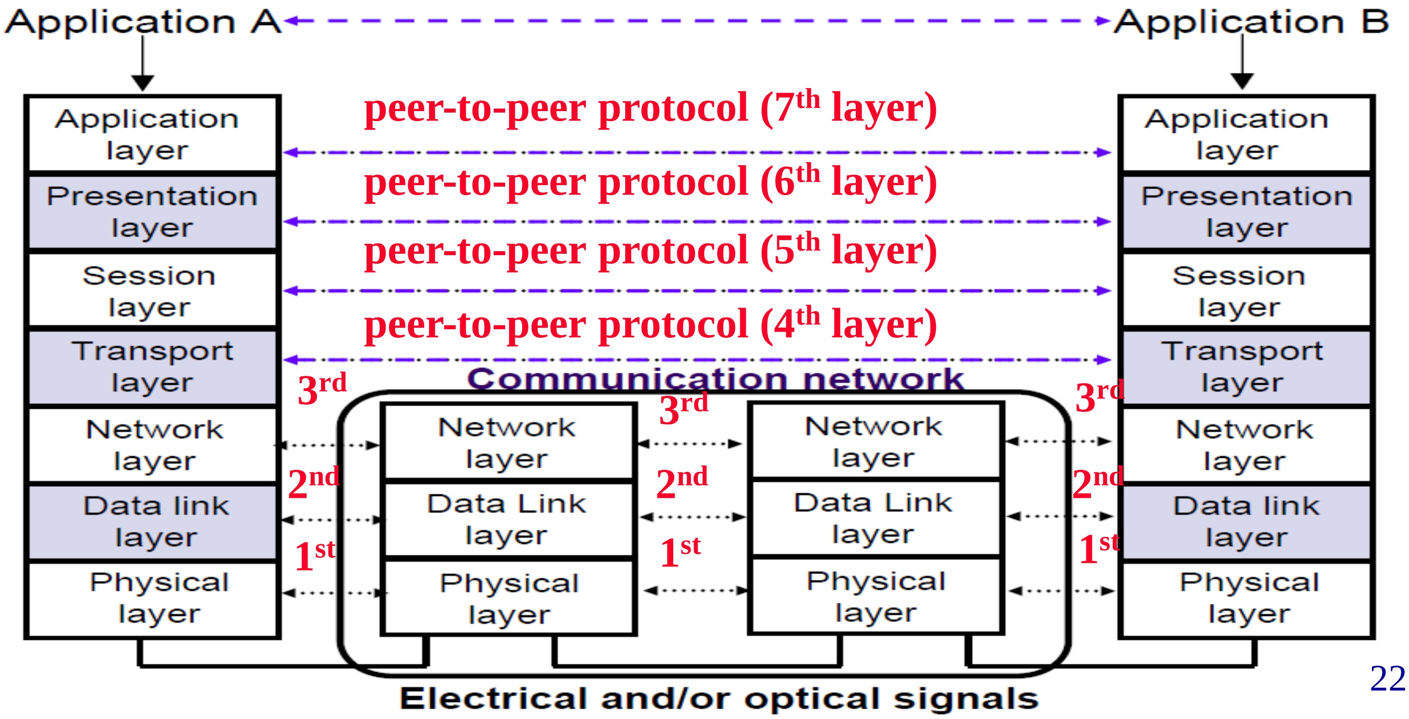
- Question: Which one is the peer of A's Network layer? Answer:

a) A's Data Link layer

b) A's Transport layer

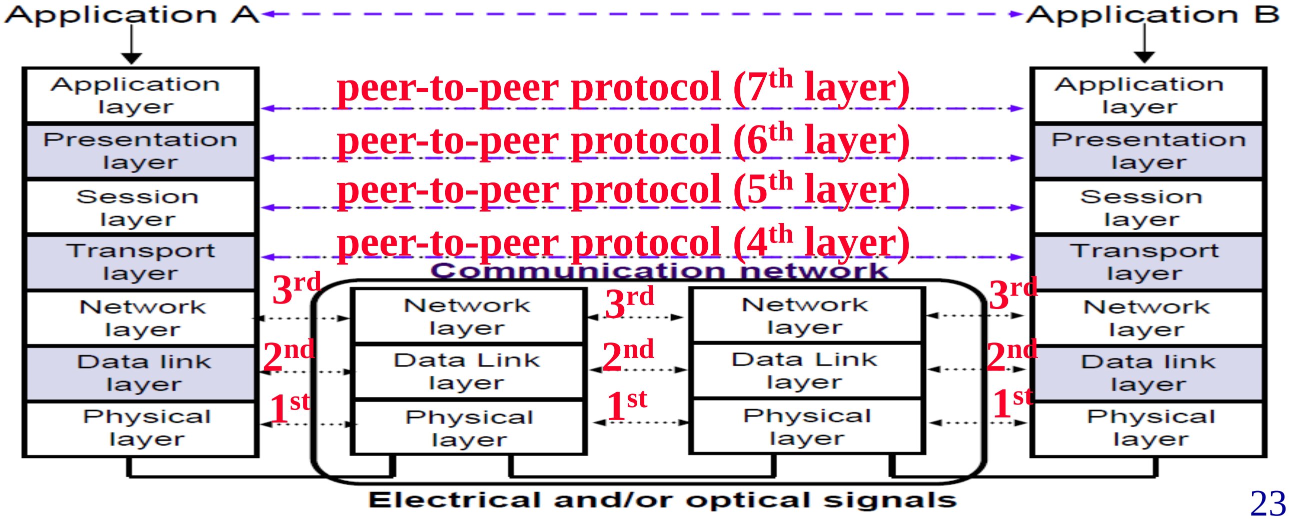
c) Router 1's Network layer

d) B's Network layer



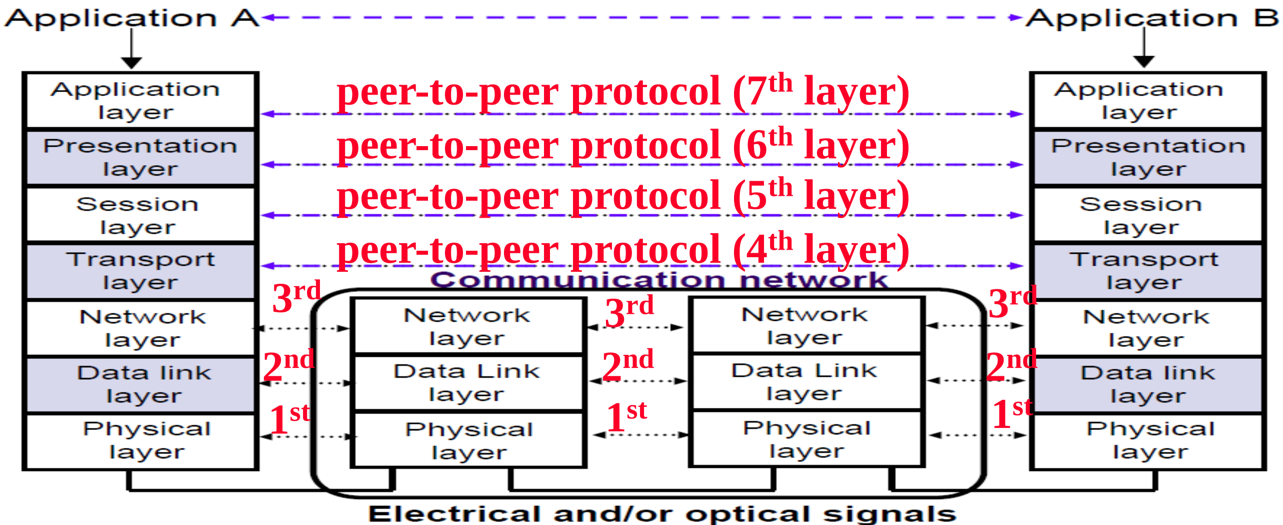
# Peer-to-Peer Protocols (Cont.)

- Only peer protocols need to understand each other, i.e., the protocol used in A's Network layer does not need to be the same as the one used in B's Network layer, but it needs to be the same as Router 1's Network layer (only neighboring boxes are peers
  - We will see that in TCP/IP, the major protocols in the Network layer are IP protocols and all the nodes use the same protocol in their Network layer. Now we know this is not necessary!
- Note that we do not call the network-related protocols end-to-end protocols, because they can be end-to-router, router-to-router or router-to-end protocols. The top four layers are the ones that are end-to-end (or host-to-host)



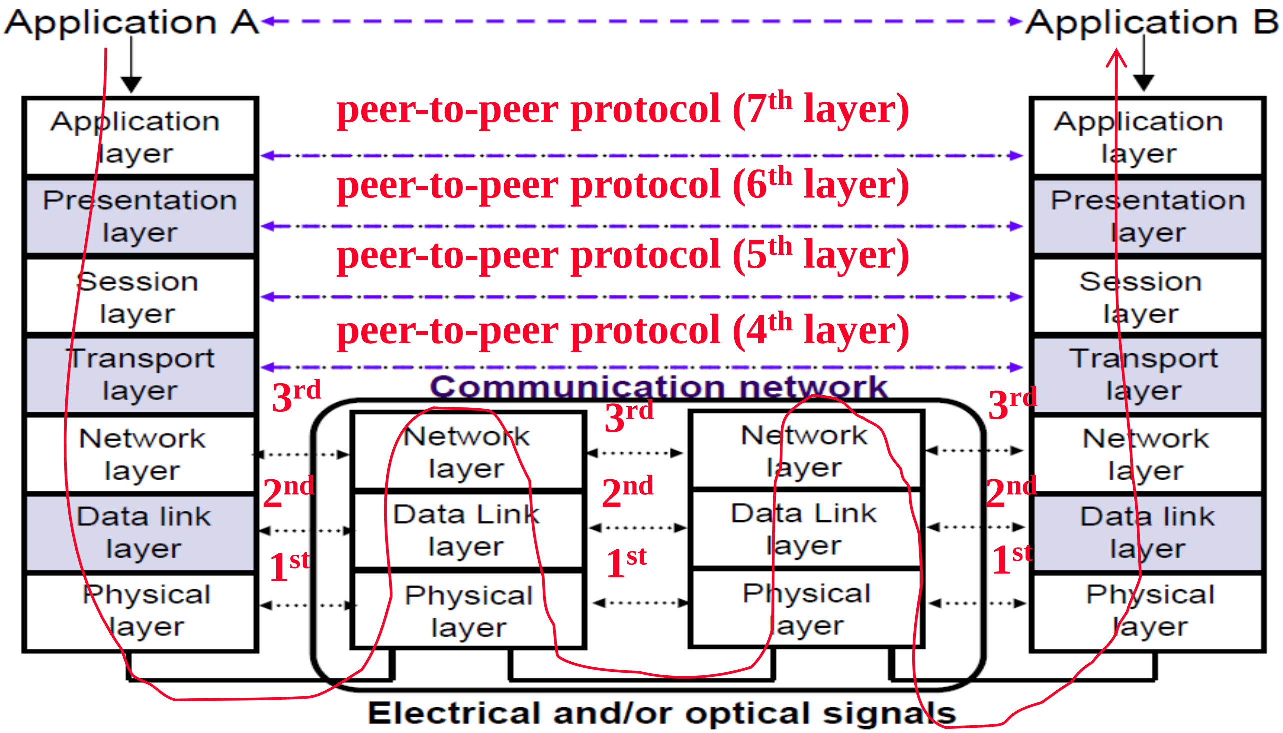
# Adjacent Layers

- Adjacent layers are the neighboring layers in the same node. Adjacent layers communicate with each other via interfaces also called Service Access Interfaces
- Question: what is interface 4-3? It's the interface between Layer 4 (Transport) and Layer 3 (Network) in a certain node
- Note that unlike peer layers, adjacent layers do NOT communicate with each other via protocols, because they do not understand each other, e.g., if Application layer uses SMTP, Presentation layer does not understand what SMTP is, so it needs an interface

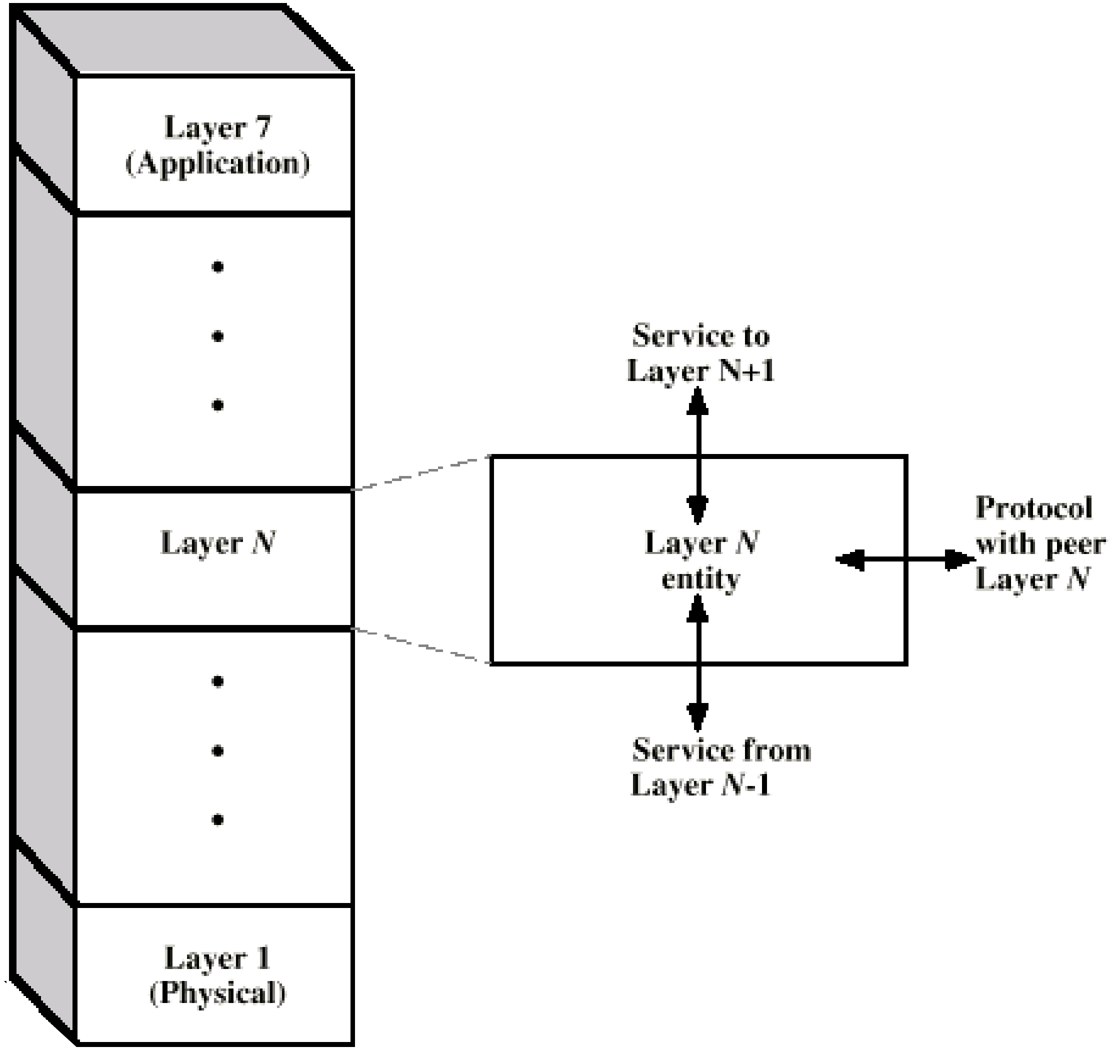


# Adjacent Layers (Cont.)

- Question: How do adjacent layers serve each other? Each layer provides a certain set of services to the layer above it
- When the node (router or end system) sends the data, the data goes from top to bottom layers passing through interfaces, and when the node receives data, it goes from bottom to top layers and passing through interfaces

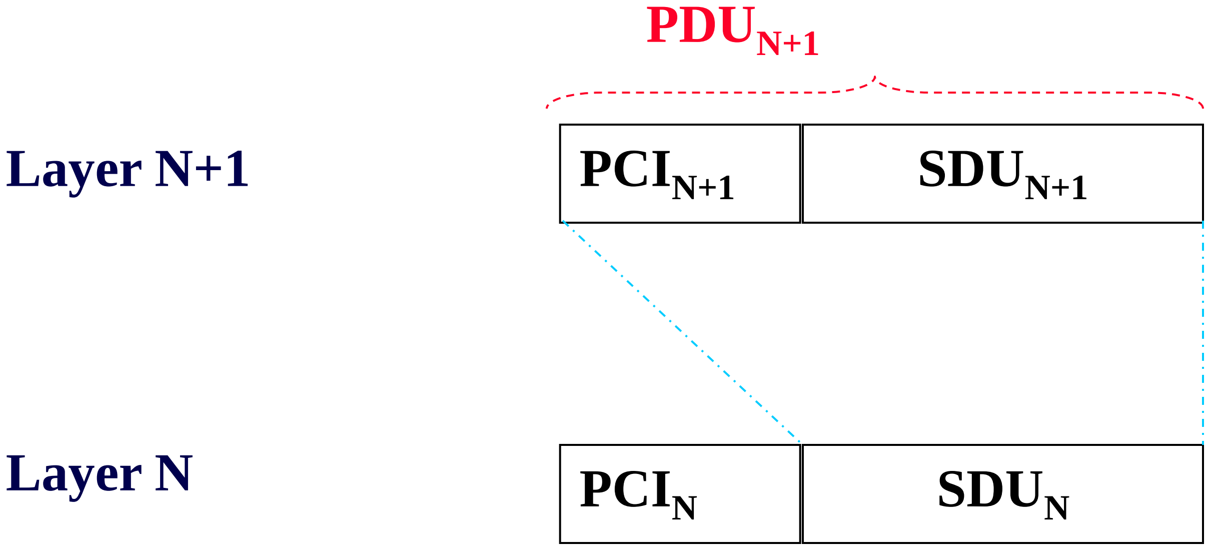


# Summary – OSI Adjacent and Peer Layers



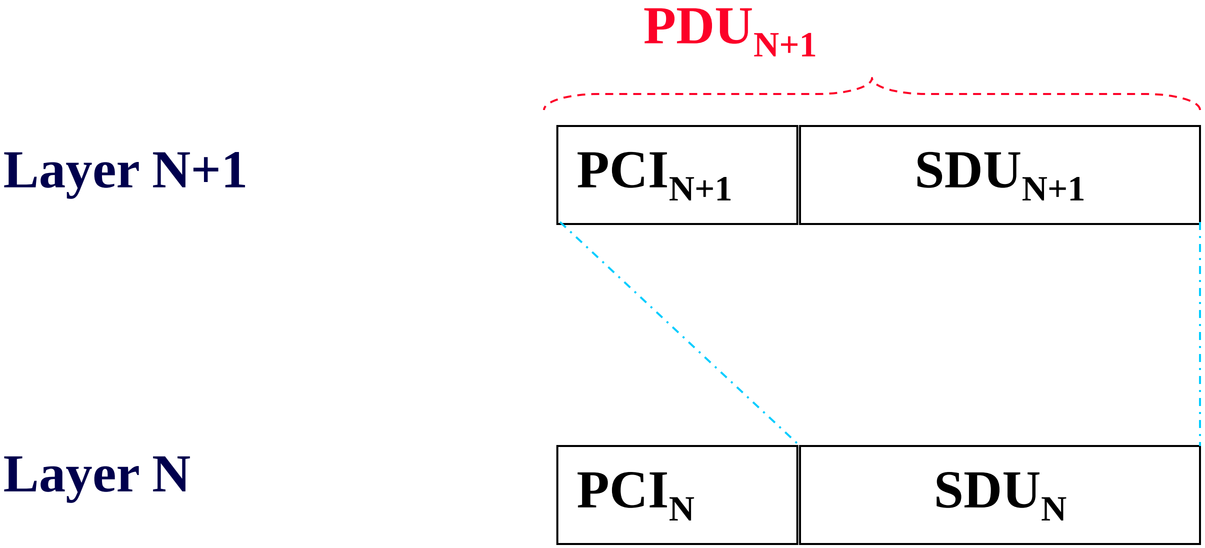
# SDU and PDU

- SDU (Service Data Unit) is the payload
- PCI (Protocol Control Information) the header which has the protocol information
- The whole SDU (the payload) + PCI (the header) is called the PDU (Protocol Data Unit) the name implies that PDU is the combination of protocol and data



# SDU and PDU (Cont.)

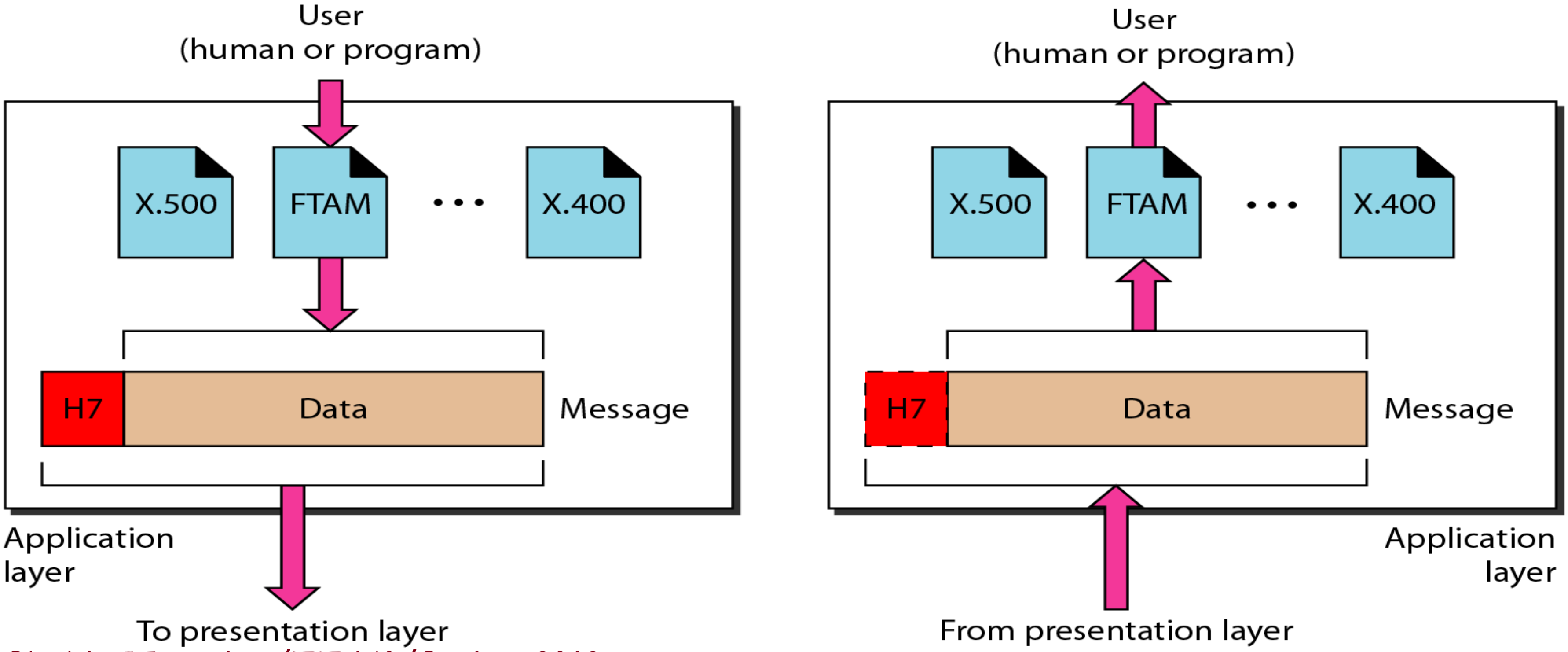
- Note that $PDU_{N+1} = SDU_N$
- Also remember that Layer N gives service to Layer N+1



$PDU_{N+1} = SDU_N$

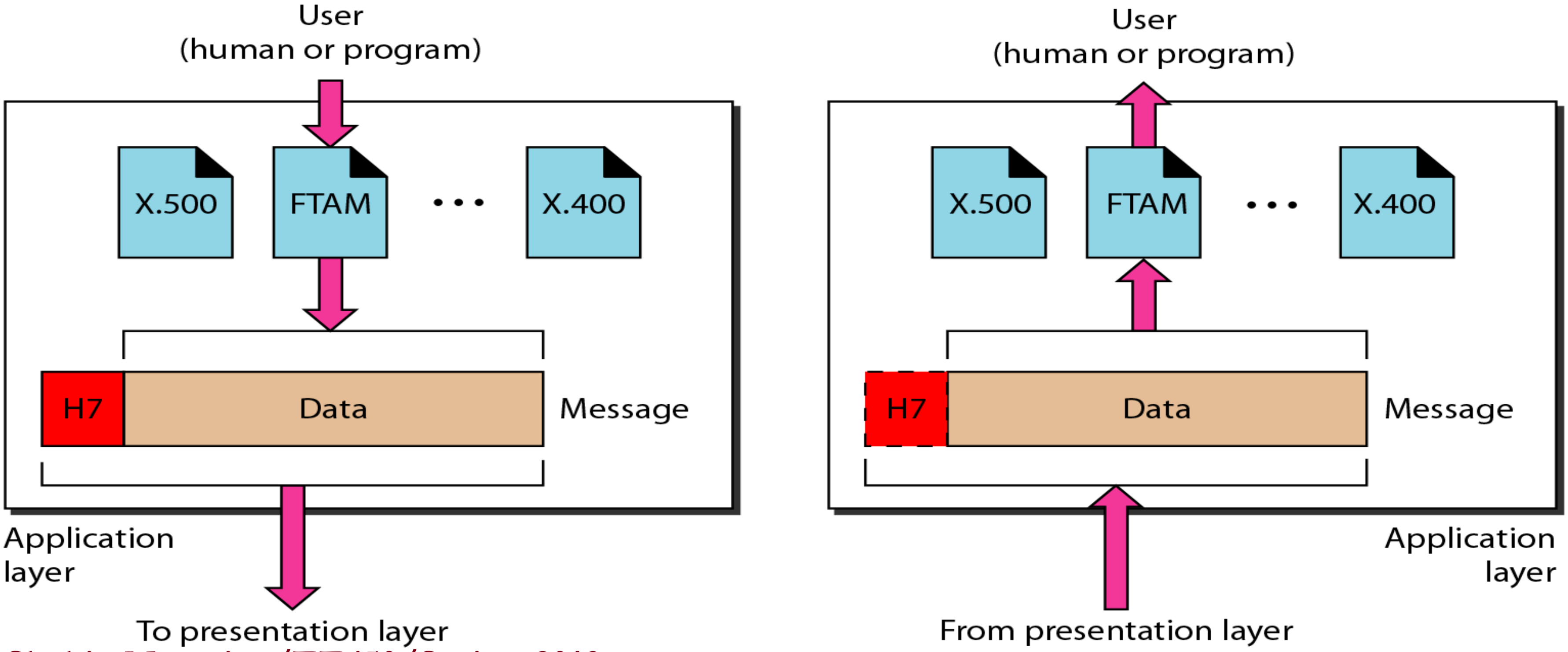
# Application Layer

- The application layer is responsible for providing services to the user
- It enables the user, whether human or software to access the network. It provides user interfaces and support for services such as electronic mail, etc.



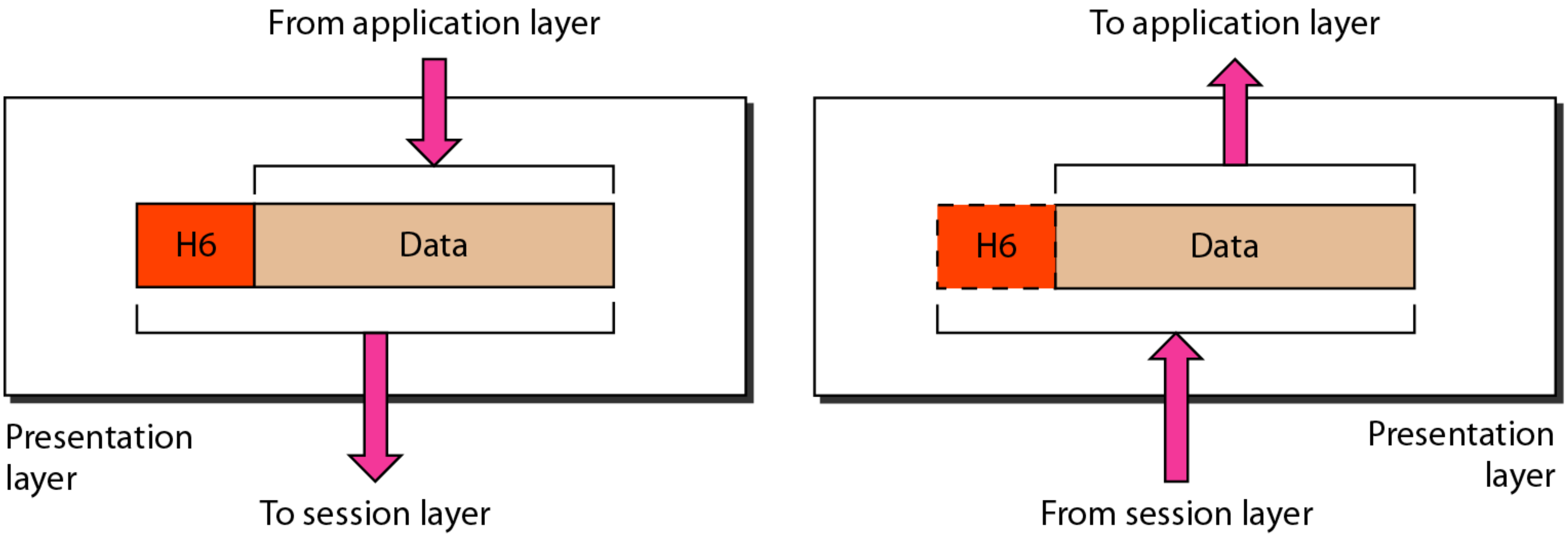
# Application Layer (Cont.)

- Among the many application services available, the following example shows X.400 (message-handling services), X.500 (directory services) which provide distributed database sources and FTAM (file transfer access and management)



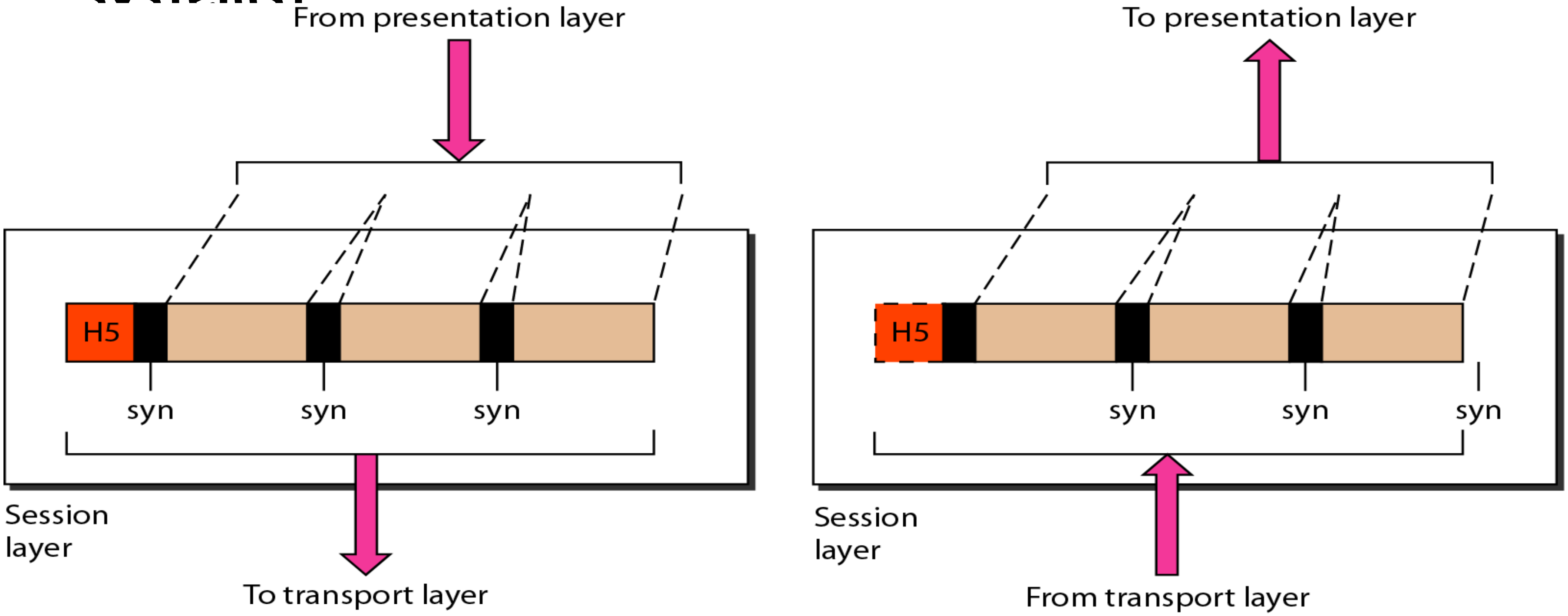
# Presentation Layer

- The presentation layer is responsible for **translation**, **compression**, and **encryption**
- This layer basically cares about how the data looks like, but does not understand the data (which is provided by the application layer)



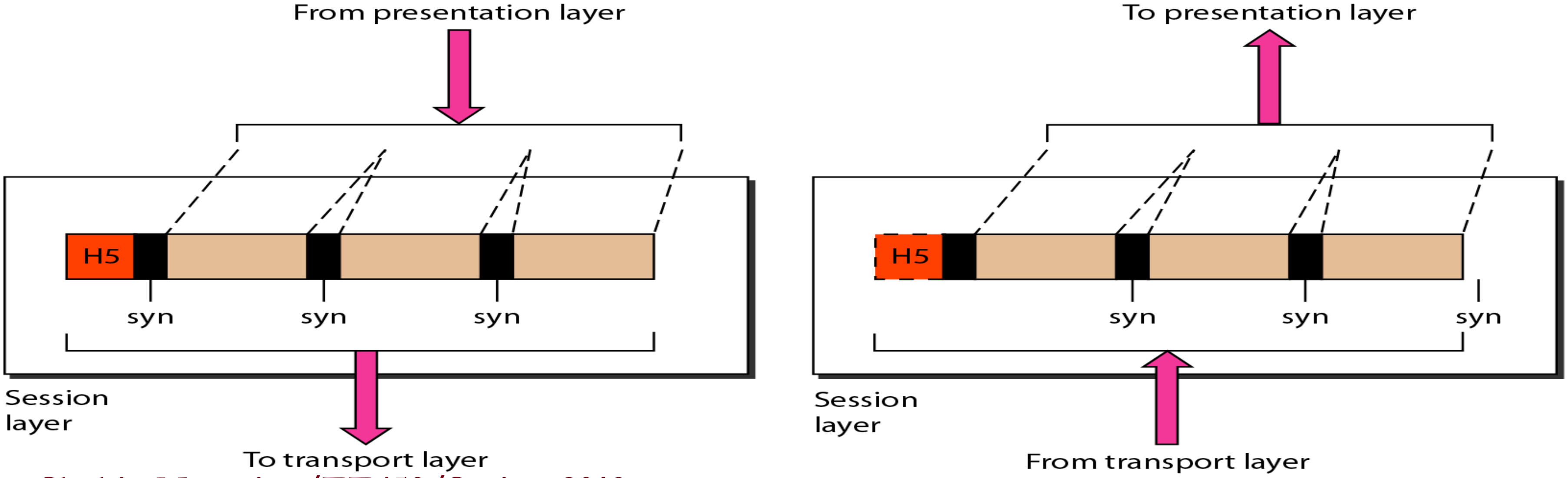
# Session Layer

- The services provided by the bottom-3 layers are not sufficient for some processes (applications)
- Session layer is responsible for **dialog control** (establishing and maintaining communication among systems) and **synchronization** [of communication among systems]



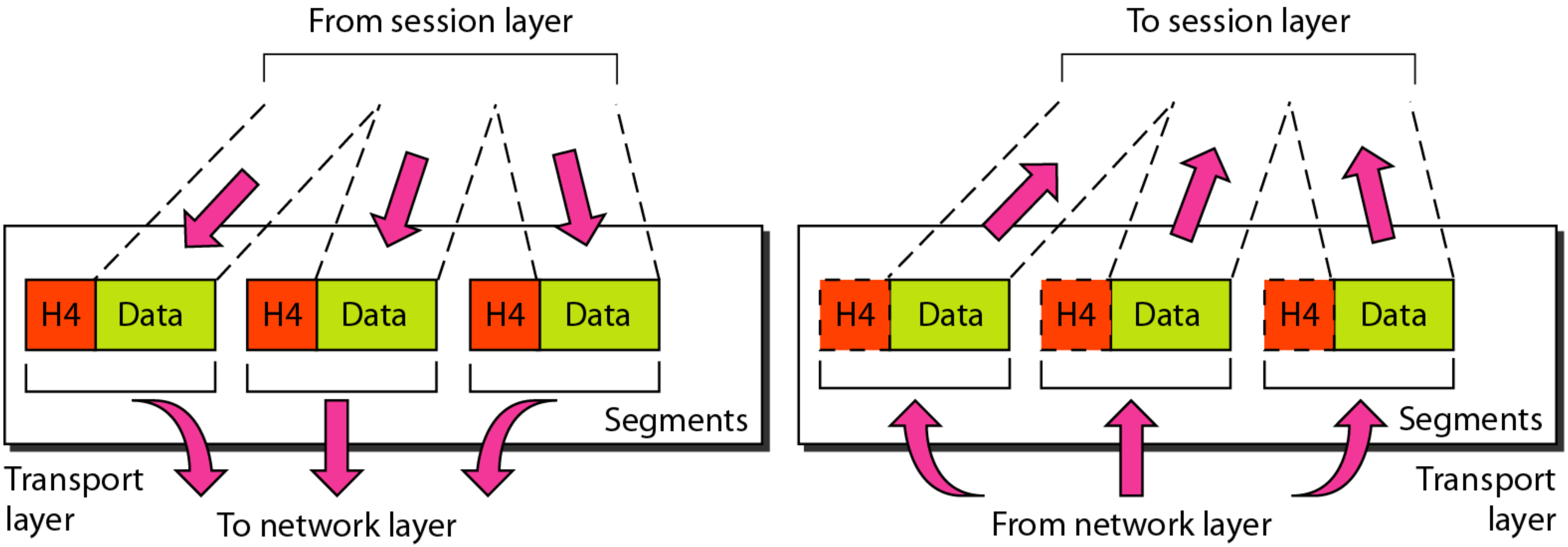
# Session Layer (Cont.)

- **Dialog Control:** Allows FDX or HDX mode communication btn two systems (another words, two processes)
- **Synch.:** Session layer allows a process to add checkpoints (synch. points) to a stream of data, e.g., if a sender is sending a 2000 page file, it may insert synch. points every 100 pages to ensure each 100 page is received and acknowledged independently



# Transport Layer

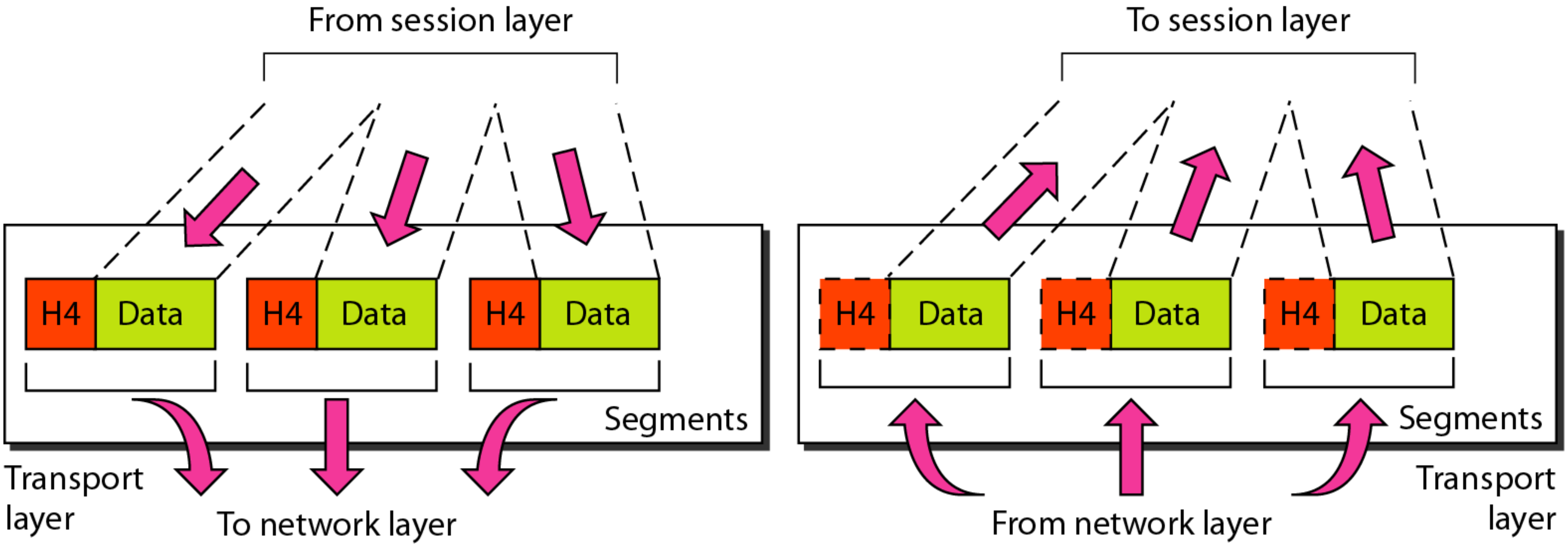
- The transport layer is responsible for the delivery of a message from one process to another
- A process is an application program running on a host



Transport Layer: Process-to-process delivery

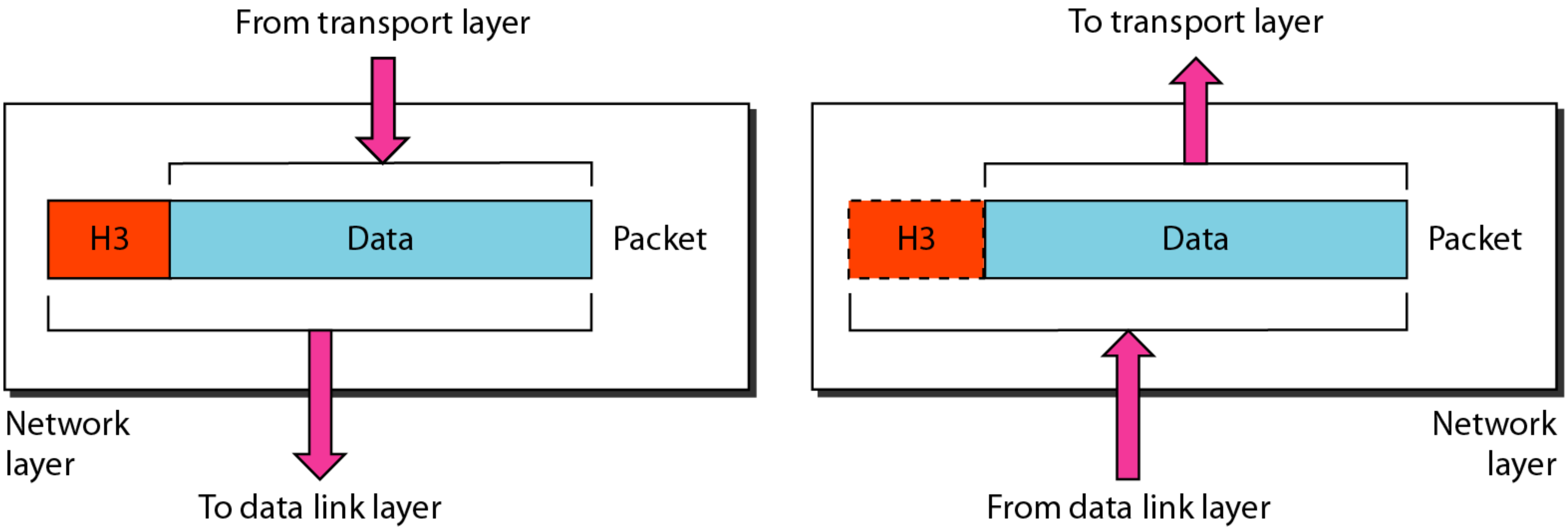
# Transport Layer

- Network layer oversees the source-to-destination delivery of individual packets and does not recognize the relationship among the packets. It treats each packet independently, as if it belonged to separate messages (whether or not it does)
- However, Transport layer ensures that the whole message arrives intact and in order, overseeing both error and flow control at source-to-destination level



# Network Layer

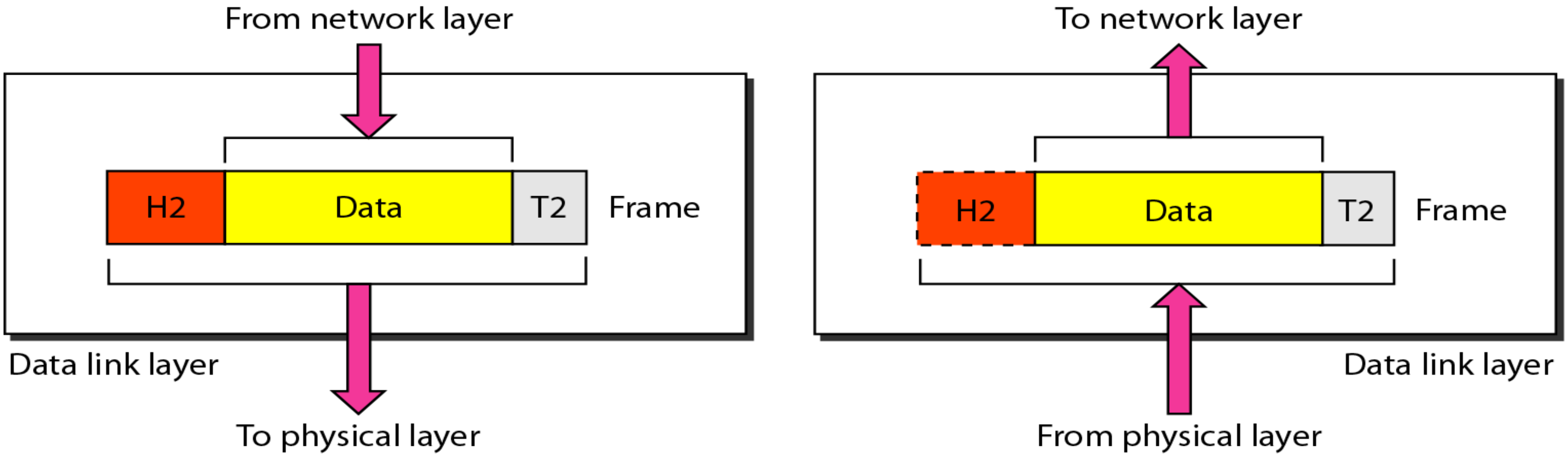
- The network layer is responsible for the delivery of individual **packets**
- Two main responsibilities: **Logical addressing** (to specify the sender, receiver address), and **routing**



**Network Layer: oversees source-to-destination delivery**

# Data Link Layer

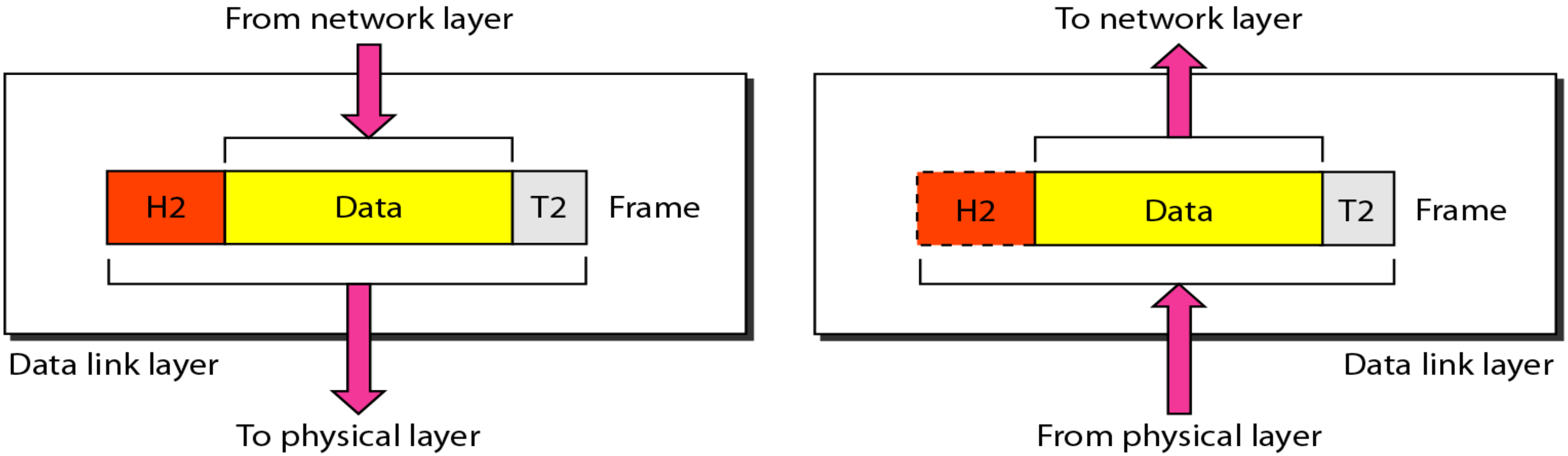
- The data link layer is responsible for moving frames from one hop (node) to the next, i.e., two systems on the same network, therefore delivery is local and not remote
- Data link layer essentially makes the physical layer (a raw transmission facility) appear error-free to the (upper) network layer



Data Link Layer: hop-to-hop (node-to-node) delivery

# Data Link Layer (Cont.)

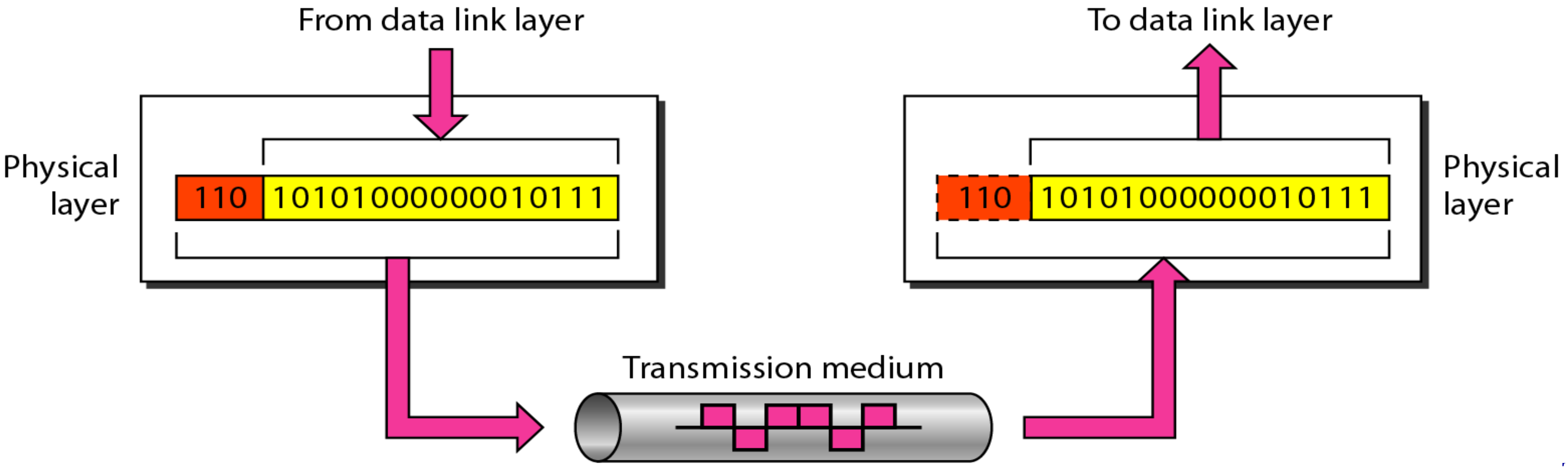
- **Main responsibilities:**
- **Framing, physical addressing (for local delivery), flow control, error control, access control**



**Data Link Layer: hop-to-hop (node-to-node) delivery**

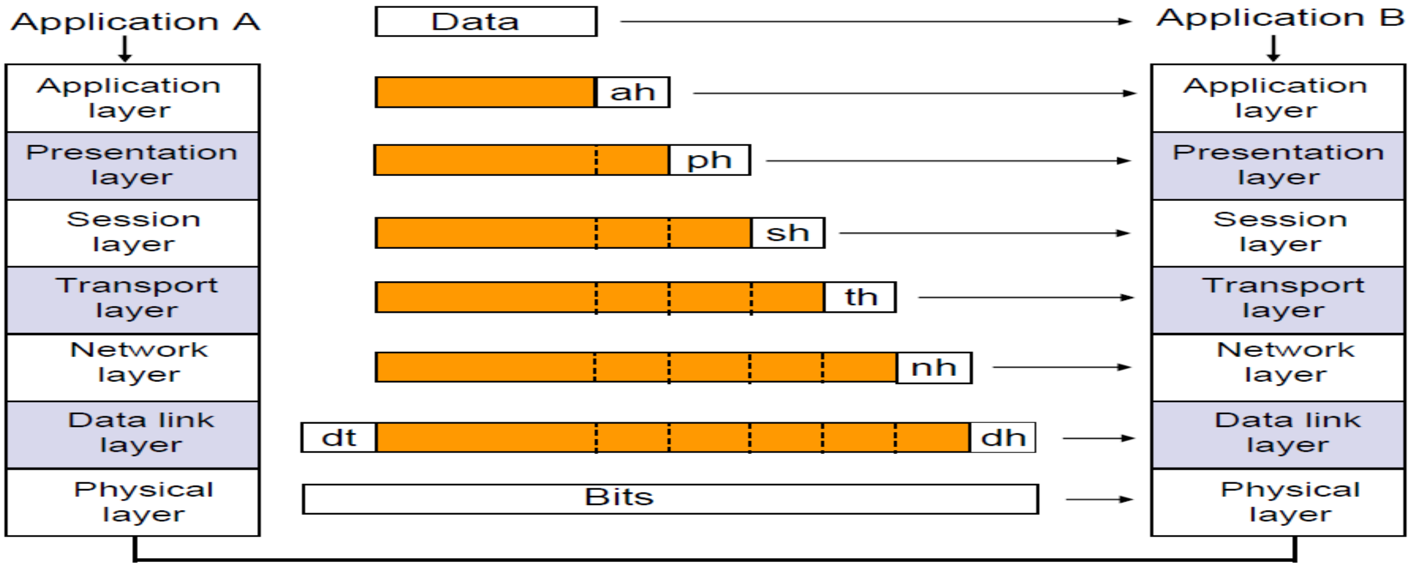
# Physical Layer

- The physical layer is responsible for movement of individual bits from one hop (node) to the next
- It is concerned with the following:
  - Physical characteristics of the interface and medium, representation of bits, data rate, synchronization of bits, line configuration (point-to-point, or multi-point), physical topology, transmission mode



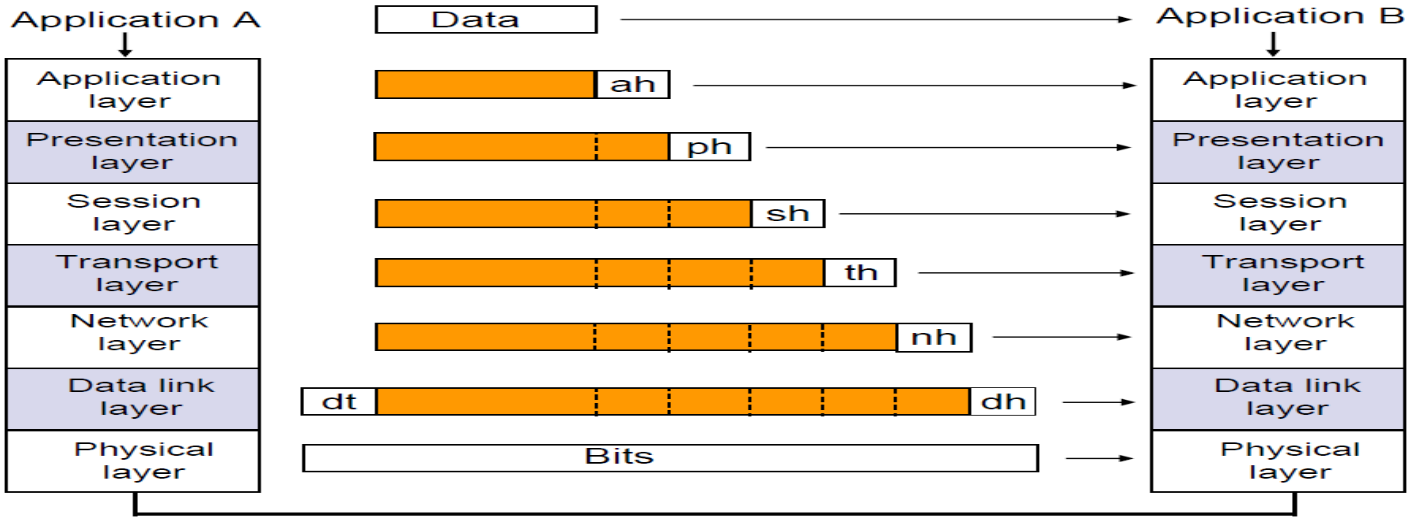
# OSI – Data Transfer

- First the actual data is generated by host A to be passed to the receiving host, B. Data is going to pass through each layer across the interfaces
- When it goes through the application layer, that layer takes the data and adds an application header (ah)
- A header has the details of its respective protocol
- Note that in this diagram the intermediate nodes (routers) have been removed for simplicity



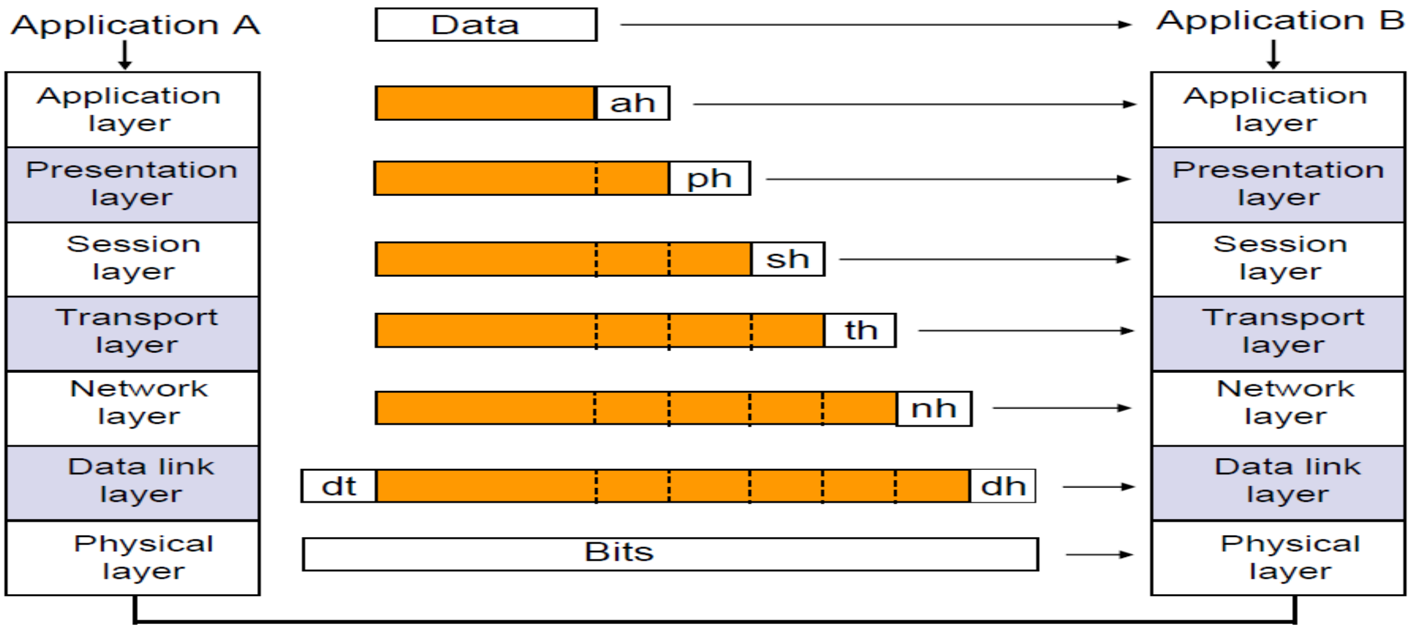
# OSI – Data Transfer (Cont.)

- Question: What layer is going to interpret the application header (ah)? A's Presentation Layer? A's Application layer? Or B's Application layer?
- Answer: Its peer layer, i.e., B's Application layer, which is the only layer that understands it. Other layers do not. The presentation layer looks at the package (Data+ah) as its data and adds the presentation header (ph)
- Note that the Presentation layer does not understand the application data, what the application layer meant by that, it just treats it as data and adds its own header, i.e., ph to that



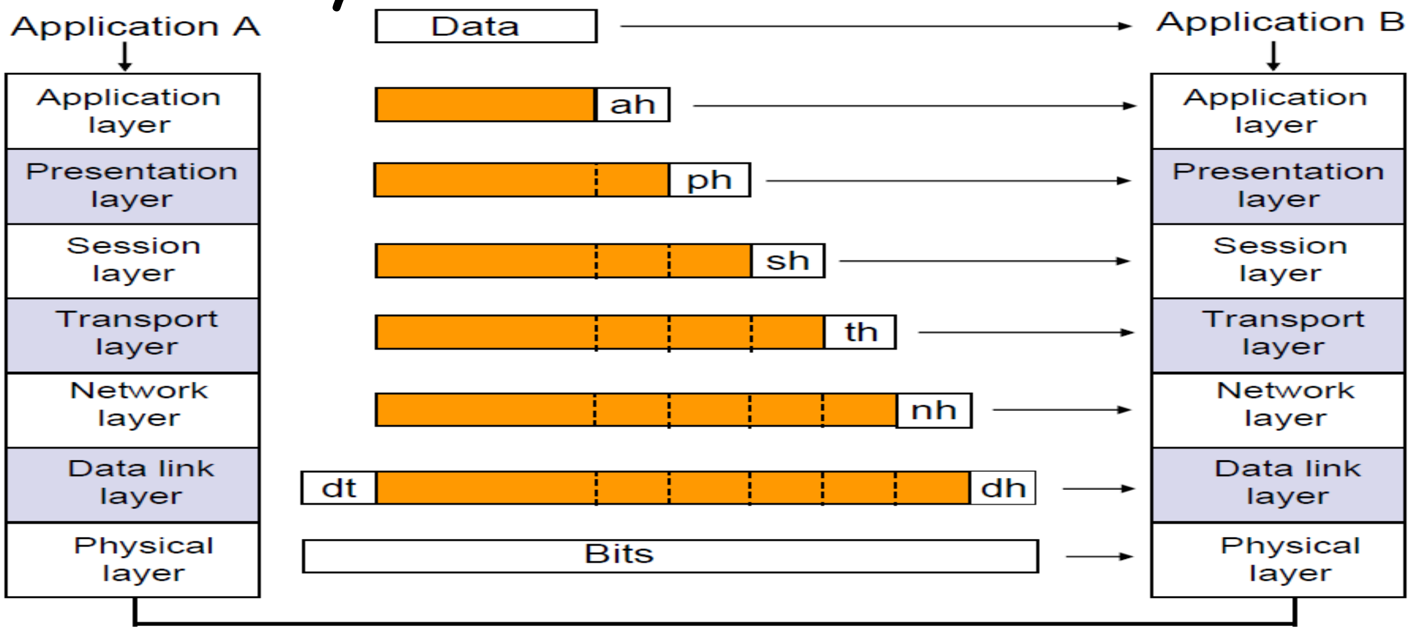
# OSI – Data Transfer (Cont.)

- Question: What layer is going to interpret the ph)? Its peer header, i.e., B's Presentation layer
- Similarly the entire thing (PDU) from the Presentation layer is taken by Session layer as the data (SDU) and session header (sh) is added to it
- Similarly for Transport layer; th is added and its peer transport layer is going to interpret th



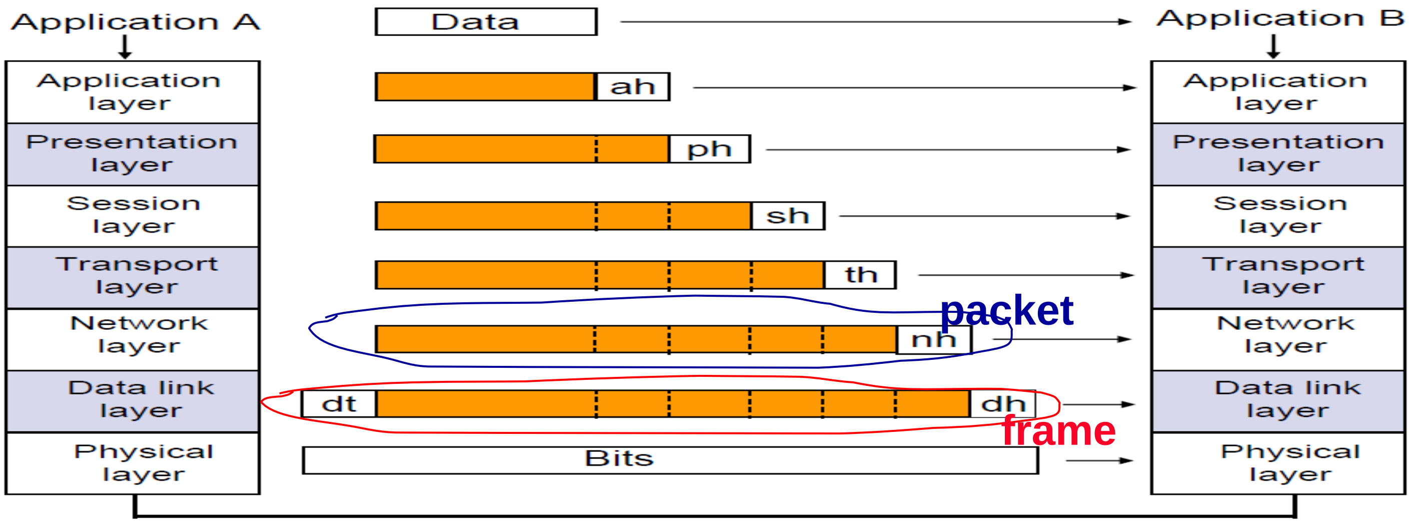
# OSI - Data Transfer (Cont.)

- Note: The receiving process (application) is not going to receive any of these headers, however they are necessary as they have the details of the protocol
- This is similar to mailing letter by putting it into an envelope that has the addresses on it. It is part of the postage protocol. Envelope is used to send the mail to the right recipient but when he or she receives the mail, the envelope is thrown away



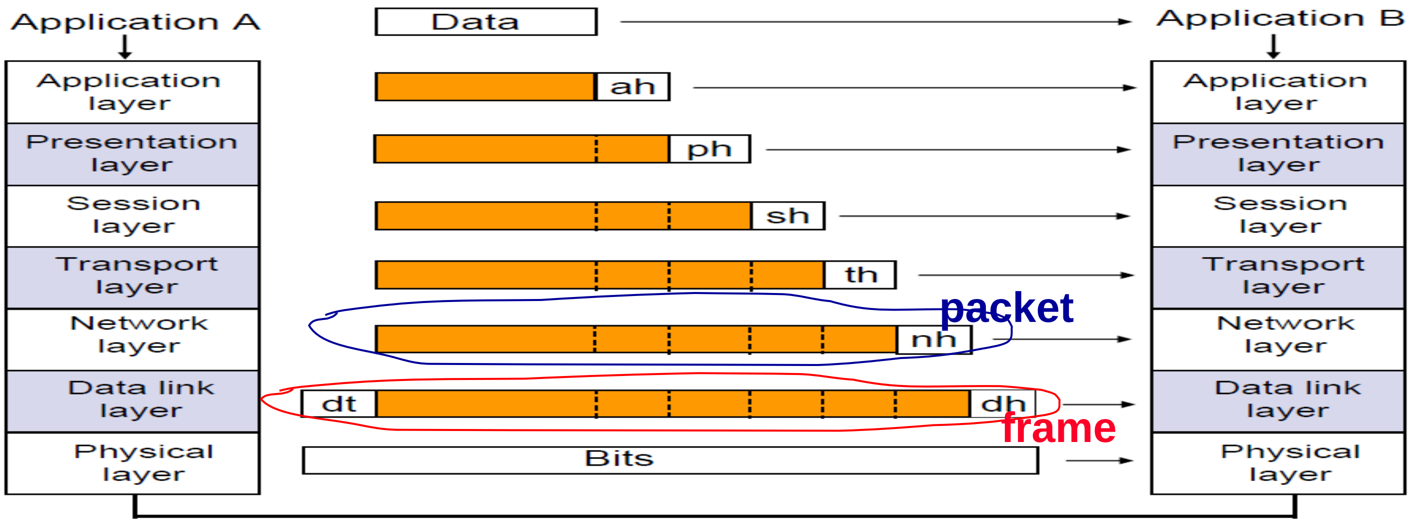
# Packet

- Data or SDU is also referred to as the payload
- Note that the payload in Network layer, is not pure payload (it has headers of top layers,) however as long as the Network layer is concerned it's purely a payload for it and it adds a nh to that payload to create a packet
- The peer layer (i.e., the network layer in the next intermediate node) is going to interpret nh



# Frame

- The network header (nh) has IP address of the host that sent the data and the IP address of the destination host where the packet is destined to
- Note that the Data link Layer does not worry about the IP addresses and does not interpret them; it looks at the whole packet as the data, adds dh and dt to it, the whole PDU is called a frame. Reminder: a packet is encapsulated inside a frame



# Physical Layer (Cont.)

- The Data Link layer passes frame to the physical layer and physical layer puts the frame bits on the line

- Notes:
  - In each layer the whole thing, (i.e., PDU or data+header/trailer) consists of 0s and 1s, however there is a structure for each, meaning they have a beginning and an end
  - In other words, a 1 in data header, a 1 in packet and a 1 in the network payload look the same, but they have different meanings

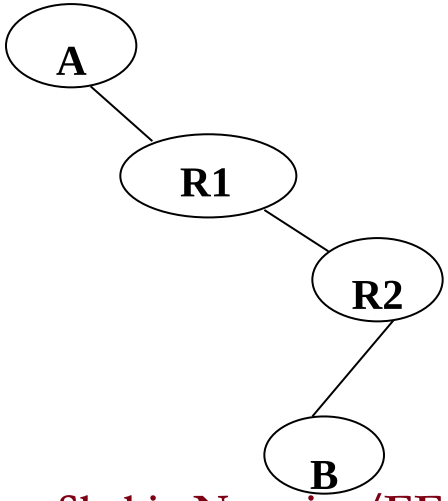
# Physical Layer (Cont.)

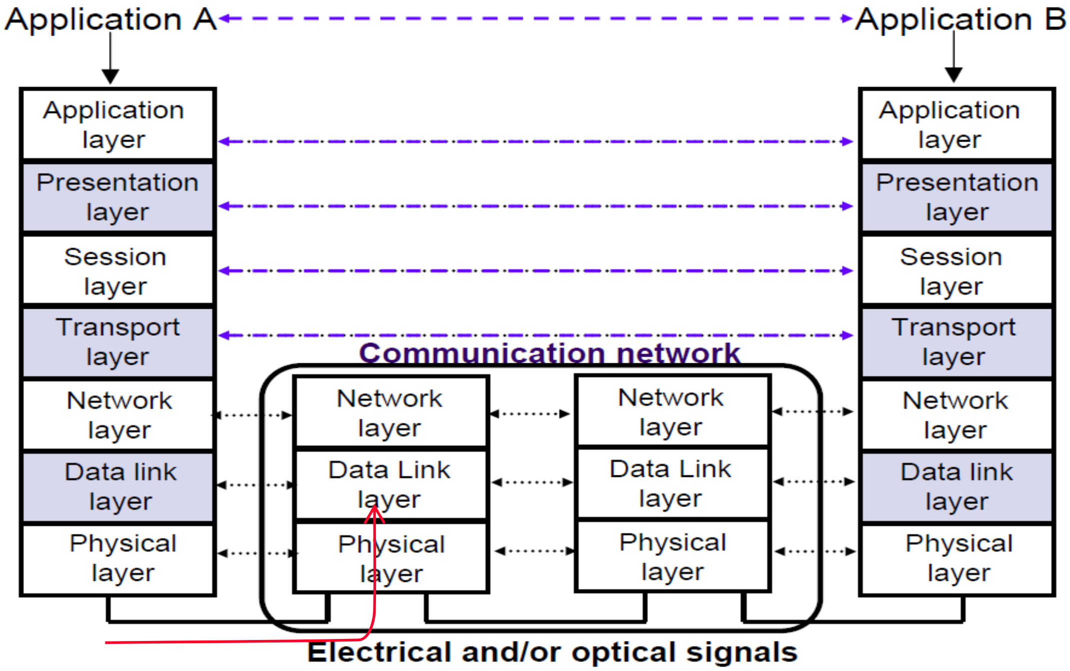
- When the whole thing goes to the physical layer, the 1s not only look the same, but also mean the same from Physical Layer's point of view, as it does not mind if the 1s came from header, data, or trailer

- In industry the terms "packet" and "frame" may be used all over the place and on the wrong layers. Q: is the packet the payload of the frame? Yes [frame is the 2nd layer PDU]

# Data Transfer (Cont.)

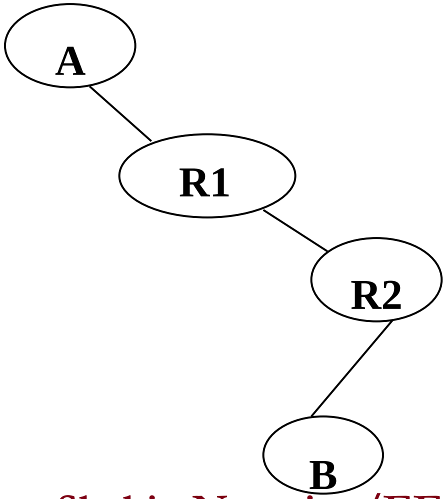
- After the physical layer of the sending host puts the frame on the line, the physical layer at the router (R1) picks it up and passes it to its Data Link layer, while it doesn't worry what the data means
- The first thing that the Data Link layer has to do to recreate the frame is to understand the start and the end of the frame. The goal is that after recreation, the frame is exactly the same as the one in the data link layer of the sending host, so there will be no error

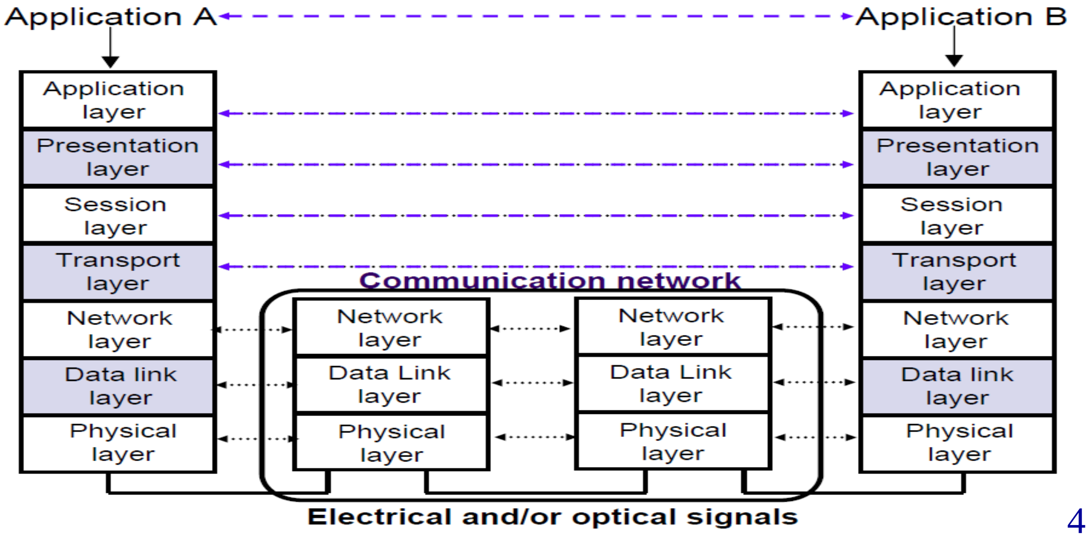




# Data Transfer (Cont.)

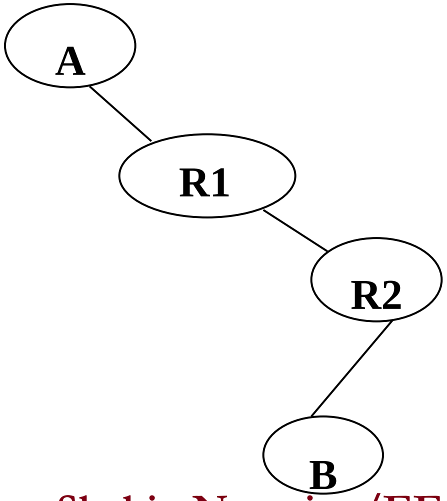
- After R1's Data Link layer performs its functionality, it removes the header and the trailer, because it needed them to do the functionality, so after that it does not need them any longer
- Data layer passes the remaining part (its payload or the packet) to R1's Network layer
- R1's Network layer checks: error, and local or remote delivery

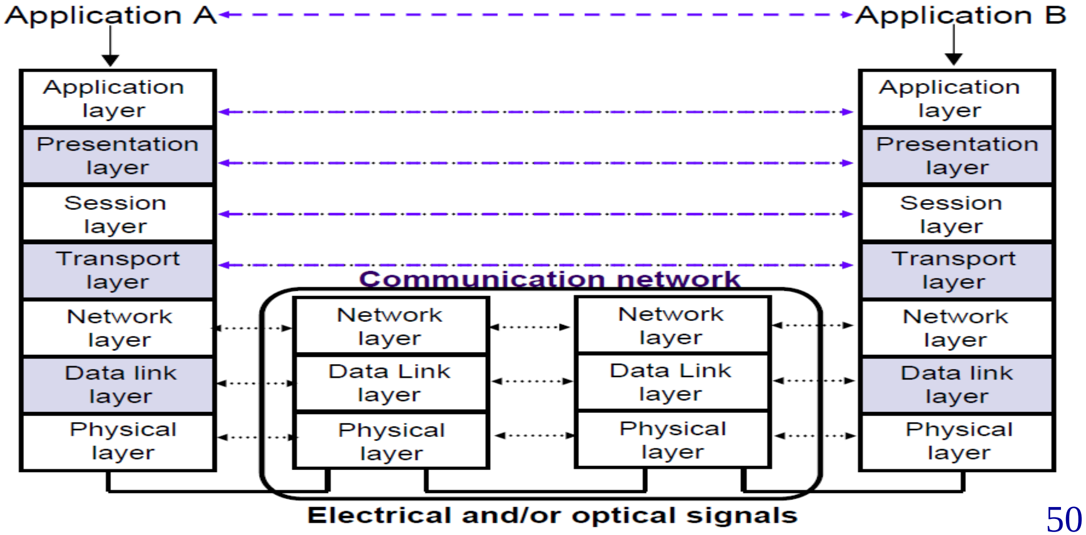




# Data Transfer (Cont.)

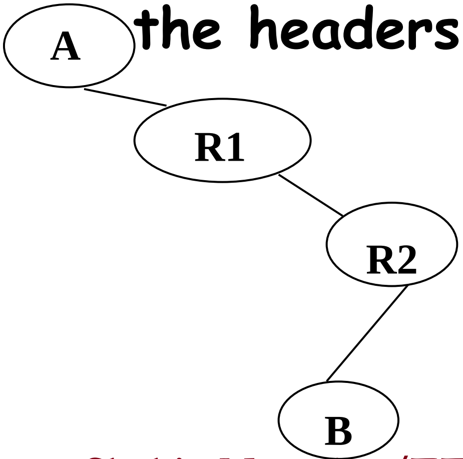
- Here the delivery is remote, because it has to go to R2
- In the network layer of R1 the packet goes up and down and the part of the packet header that contains the IP addresses doesn't change (because the IP addresses are for the source and the ultimate destination)
- The Data Link layer of R1, inserts the packet to a new frame (note: not the same as the one in the sending host A.) This new frame header and trailer would later be interpreted and understood by the Data Link layer of R2
- The new frame is then put on the physical line to get to R2

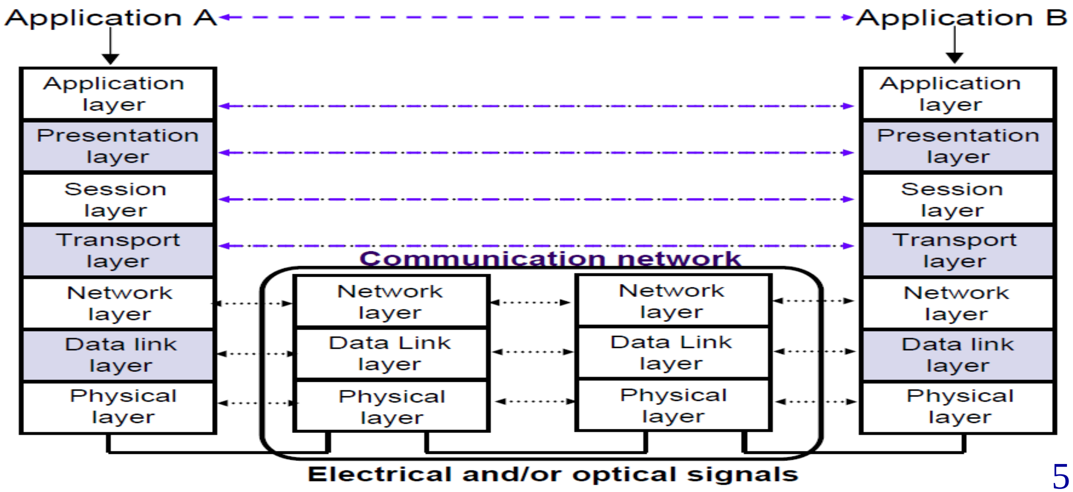




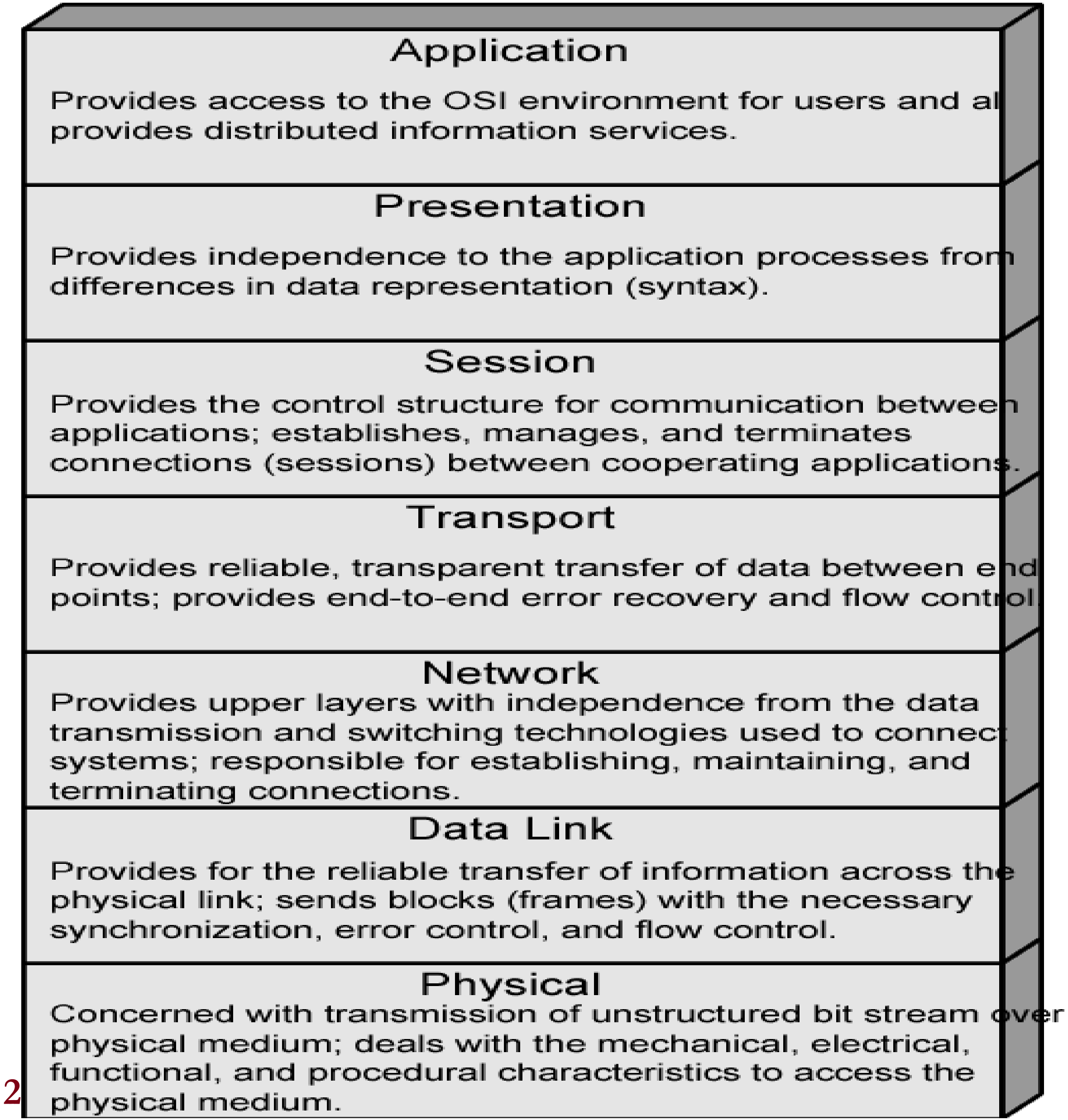
# Data Transfer (Cont.)

- Steps in R2 are similar, however in R2's network layer, when it looks at the routing table, R2 finds out that the delivery is local
- The IP addresses in the packet header do not change and R2's Data Link layer creates a new frame
- In B, going from bottom to top layer, in each layer the functionality is first done and the headers (and also trailer in 2nd Layer) are then removed and the rest is given to the upper layer
- The end host is going to eventually receive the data with all the headers removed





# OSI Layer Functionality – Summary



Application

Provides access to the OSI environment for users and al provides distributed information services.

Presentation

Provides independence to the application processes from differences in data representation (syntax).

Session

Provides the control structure for communication between applications; establishes, manages, and terminates connections (sessions) between cooperating applications.

Transport

Provides reliable, transparent transfer of data between end points; provides end-to-end error recovery and flow control.

Network

Provides upper layers with independence from the data transmission and switching technologies used to connect systems; responsible for establishing, maintaining, and terminating connections.

Data Link

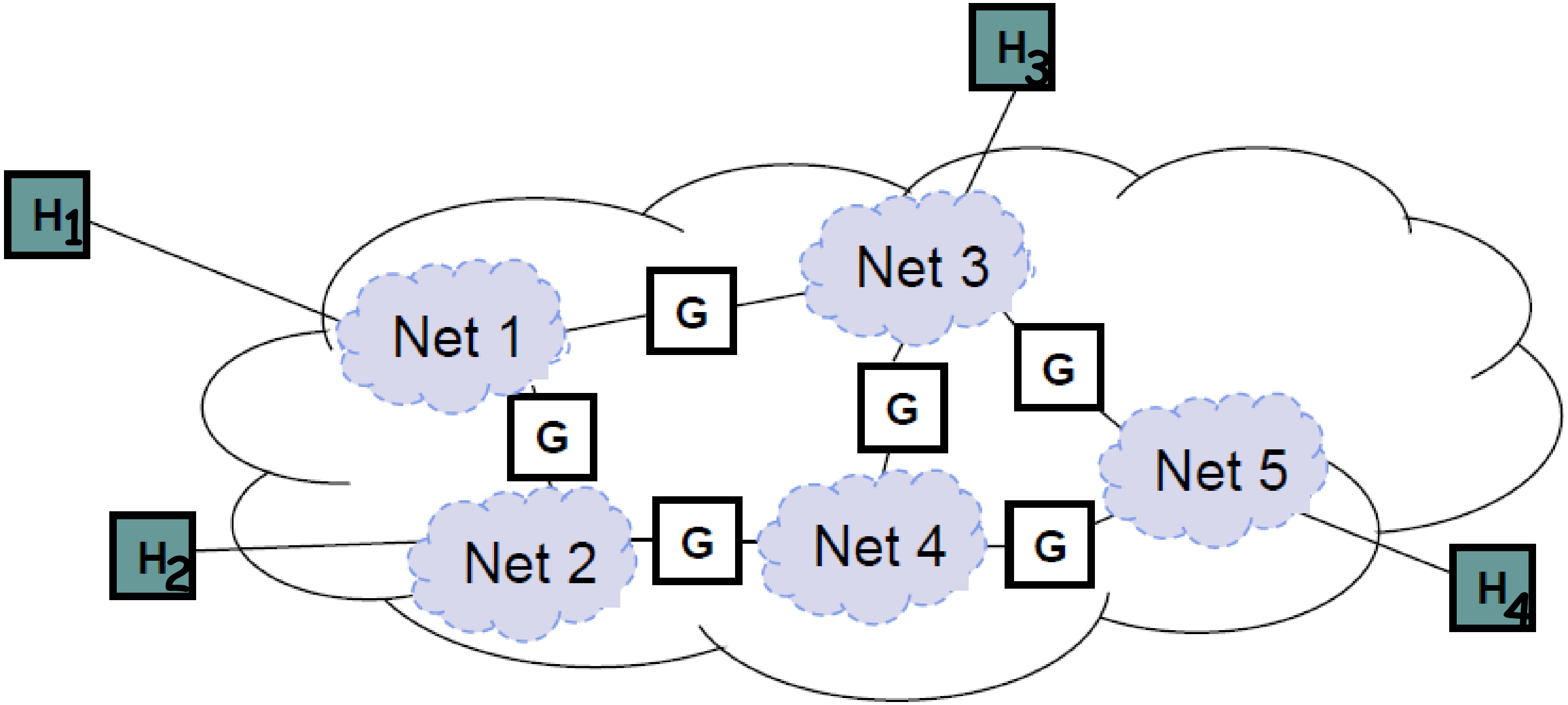
Provides for the reliable transfer of information across the physical link; sends blocks (frames) with the necessary synchronization, error control, and flow control.

Physical

Concerned with transmission of unstructured bit stream over physical medium; deals with the mechanical, electrical, functional, and procedural characteristics to access the physical medium.

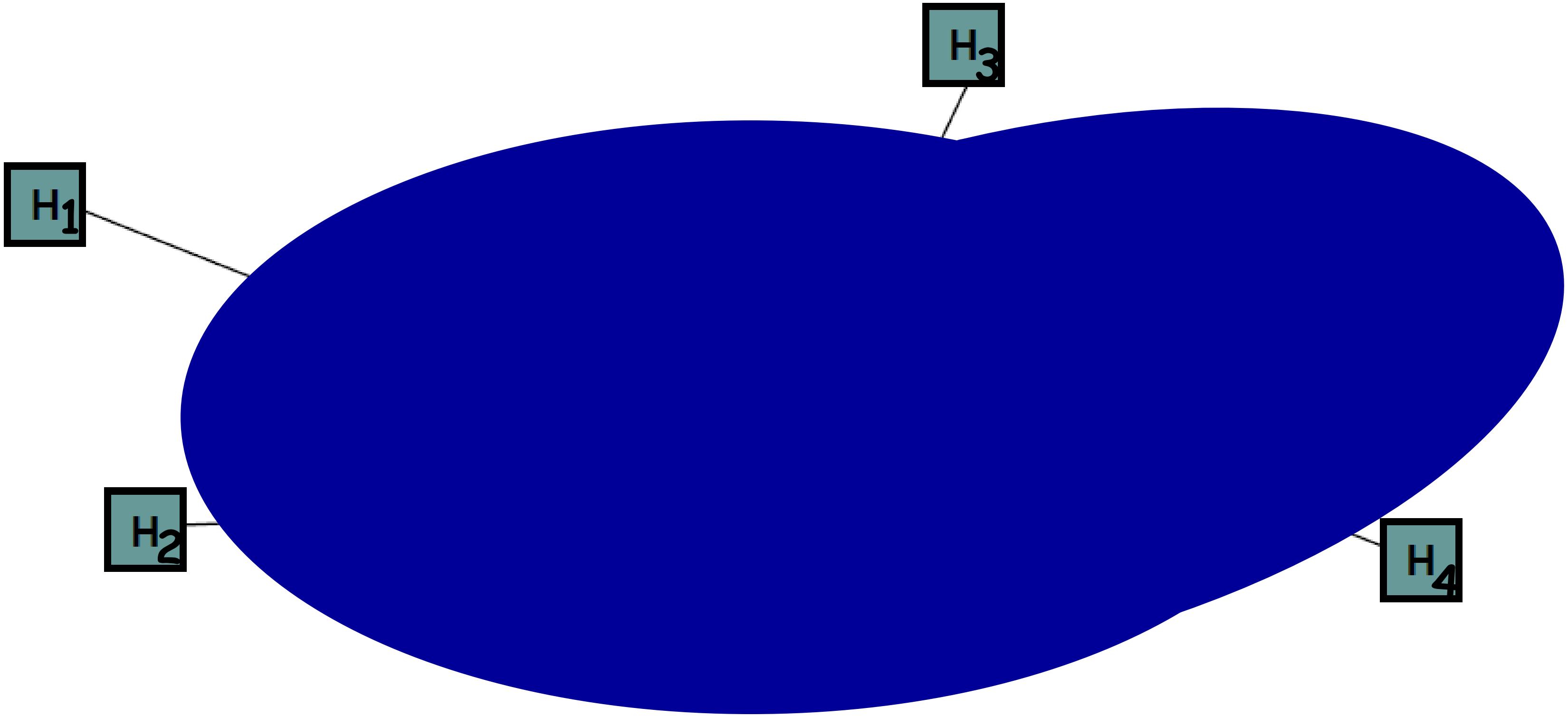
# Internetworking

- Internet is a network of networks, each with different technologies, then how can H1 and H2 communicate when they are not in the same network?



# Internetworking – TCP/IP View

- Networks may have different technologies, but they all agree on TCP/IP; therefore the whole network is transparent to the outside world

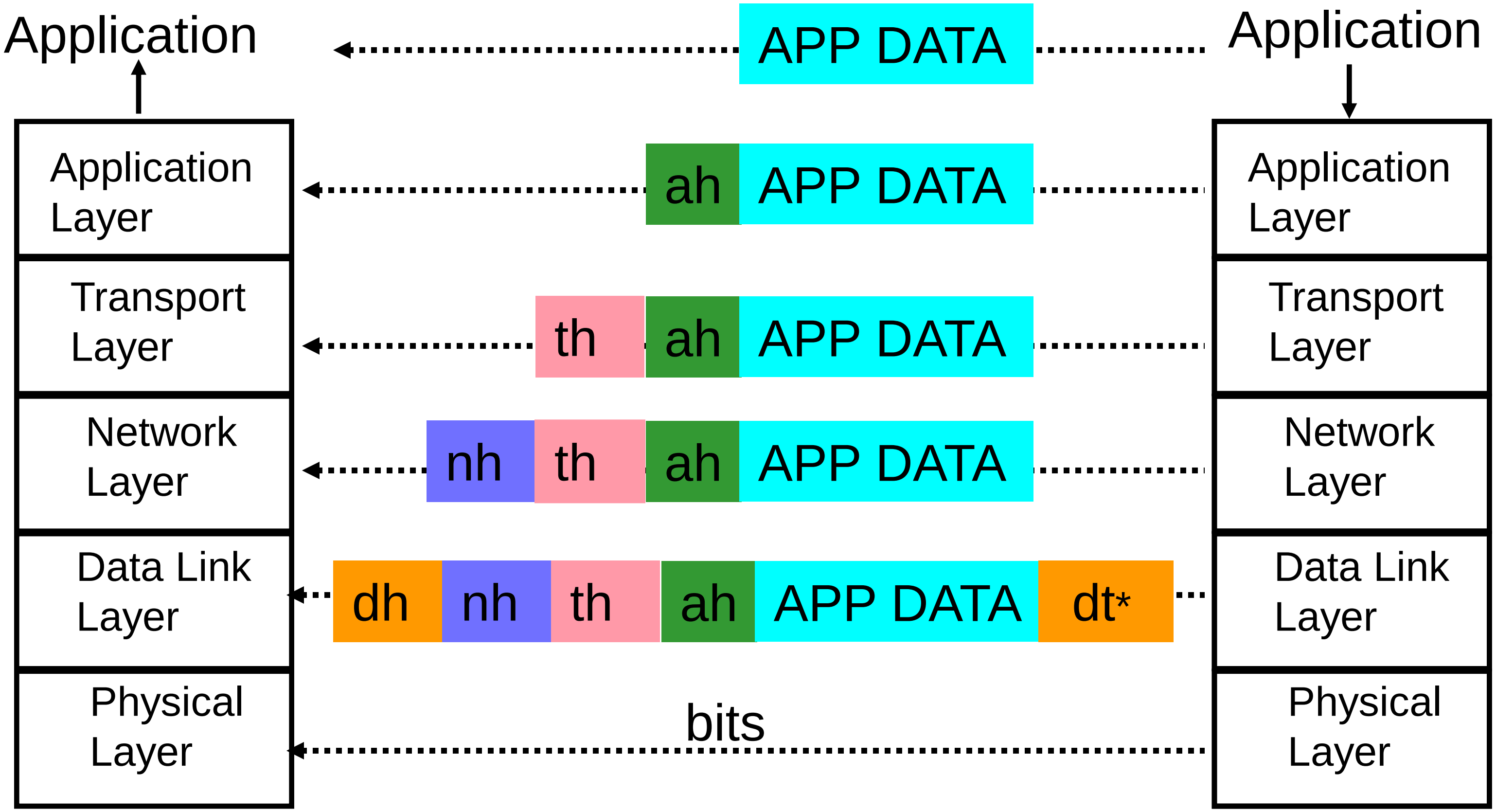


# TCP/IP Open Suite Protocol

- Our interest in this course is the TCP/IP open suite protocol that is used in global Internet and private networks (Intranet)
  - Developed by the US Defense Advanced Research Project Agency (DARPA) for its packet switched network (ARPANET)
- TCP/IP protocol suite layers do not exactly match those in the OSI model
  - The original TCP/IP protocol suite was defined with four layers: host-to-network, internet, transport, and application
- However, when TCP/IP is compared to OSI, we can say that the non-official, working TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application
  - OSI concepts and those of the 5 layer TCP/IP are similar. Only the functionalities of the Presentation and Session layers are incorporated into the Application layer

# TCP/IP Model

- There are 2 end-to-end layers (i.e., Application and Transport) and the bottom 3 layers are the network-related layers



* dt = FCS (Frame Check Sequence) = CRC (Cyclic Redundancy Check)

# Layer Functionality – Physical Layer

- The responsibility of the physical layer is that for each bit the Physical layer of the sender of the bit transmits, the receiver physical layer receives a bit. Note: the bit that is transmitted may be different from what is received due to error
- Physical layer does not do error detection or error correction, e.g., if the physical layer of A sends a 0 and physical layer of router R1 receives a 1 due to noise in the medium, R1's physical layer has not failed in its job. Its job was to receive the bits and deliver them to the Data Link layer of router R1
- Also, Physical layer needs to change the electrical signal generated by the computer to the proper form for the medium, e.g., in case of fiber optics to optical signal. Therefore the electrical to optical interface is part of the physical layer

# Layer Functionality – Data Link and Network Layers

- The first function Data Link layer needs to perform, is to look at what it receives from the Physical layer and find out where the frame starts and where it ends
- Data Link layer also does node-to-node (or link-to-link) error detection by checking the frame trailer (dt). dh has some bits for the purpose of error detection so that the receiver side can detect possible errors by checking them
- Data Link layer also needs to know the Data Link layer of the sender node (i.e., its MAC address) to re-transmit if the protocol necessitates retransmission after error detection
- Main functionality of Network layer is addressing and routing, meaning which router should be chosen next

# Layer Functionality – Transport and Application Layers

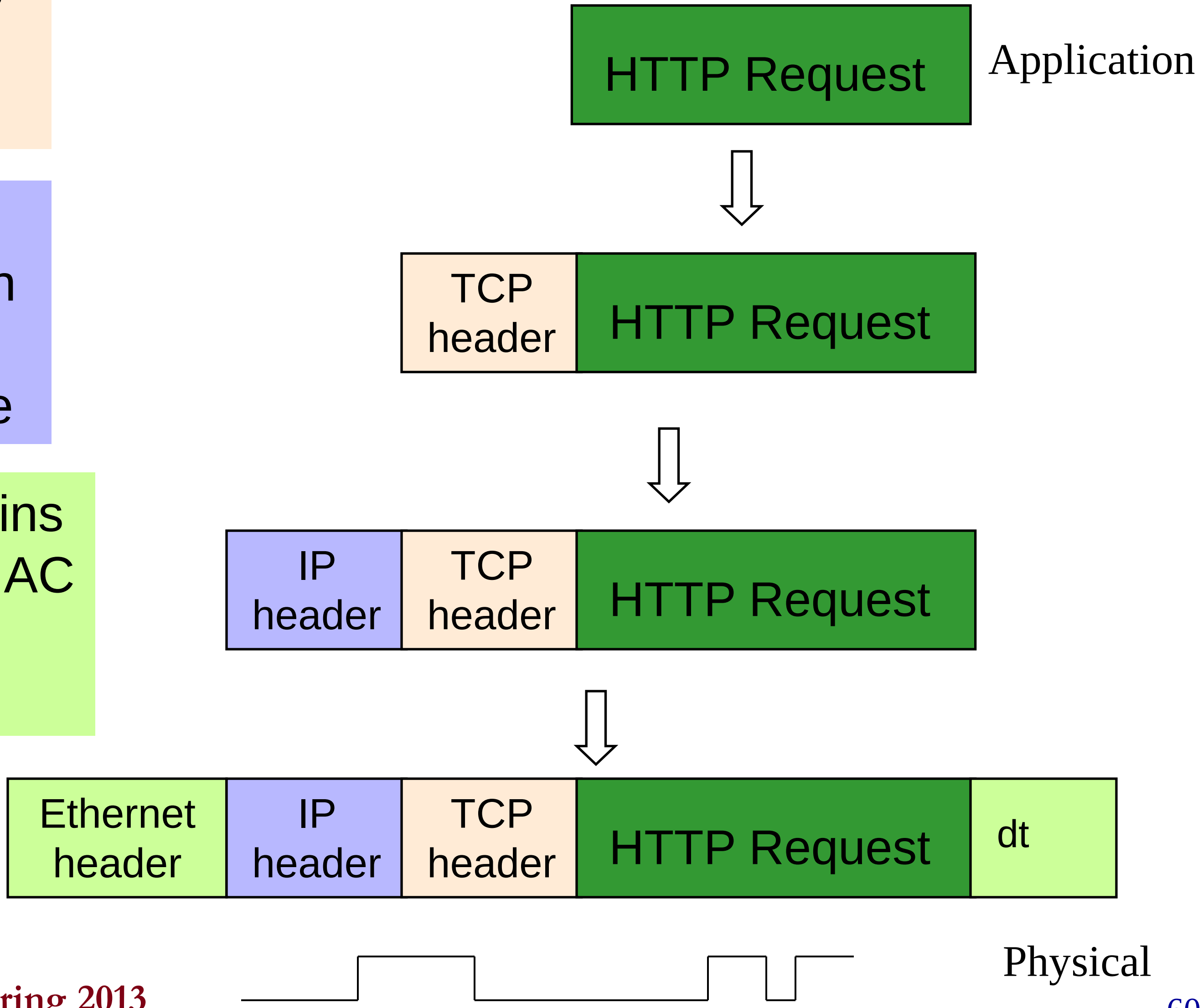
- Very important: In the process of data transfer in TCP/IP or OSI model, once we get to the Transport layer we are outside of the network
- Transport layer deals with end-to-end message delivery and error recovery, because some errors may be able to pass through the Data Link layer, so the Transport layer is considered as another layer to provide recovery, however Transport layer recovery is end-to-end whereas Data Link does node-to-node recovery
- Application layer provides the entire services to the end user and allows the end user to access the network resources

# Encapsulation

TCP Header contains source & destination port numbers

IP Header contains source and destination IP addresses; and transport protocol type

Ethernet Header contains source & destination MAC addresses; and network protocol type



HTTP Request

Application

TCP header | HTTP Request

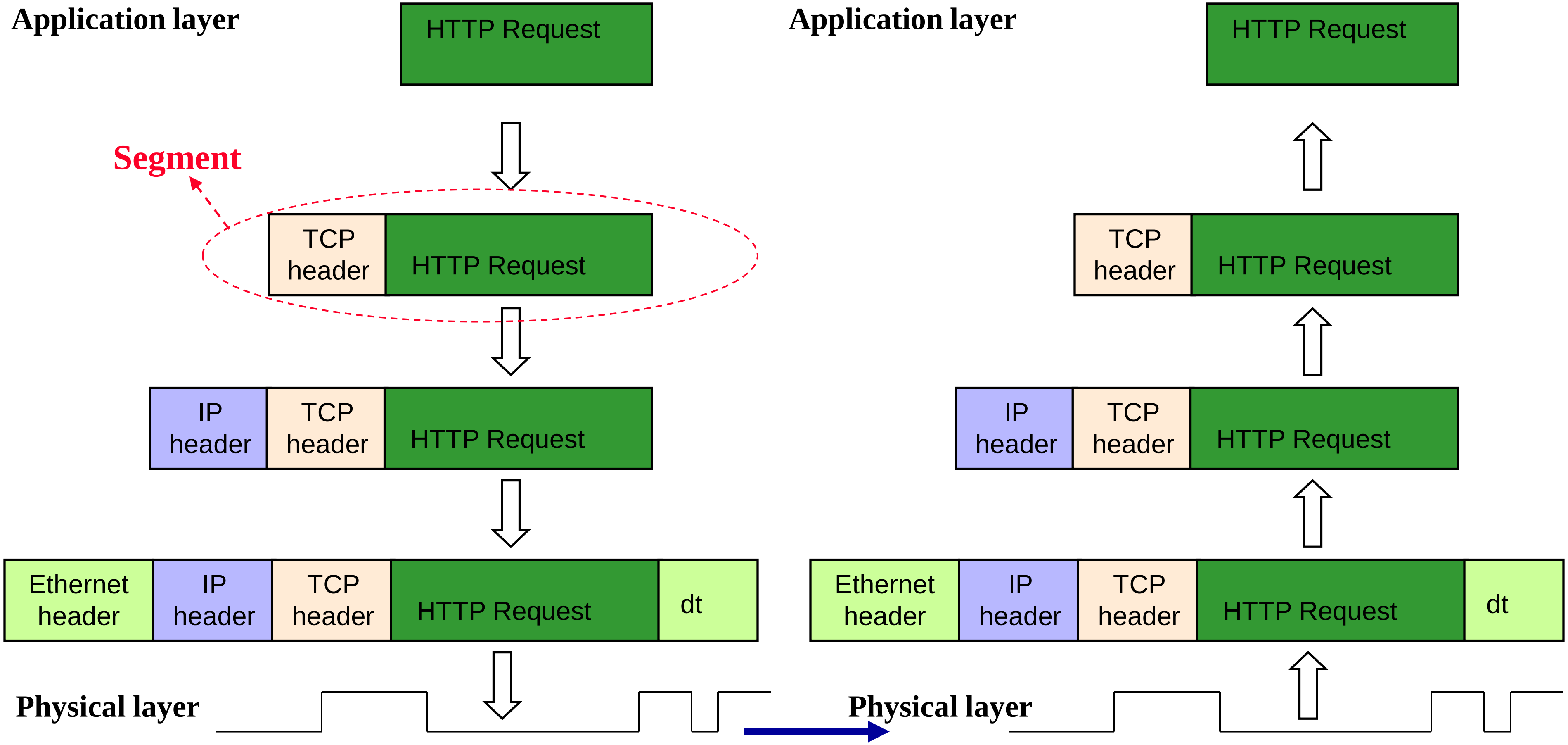
IP header | TCP header | HTTP Request

Ethernet header | IP header | TCP header | HTTP Request | dt

Physical

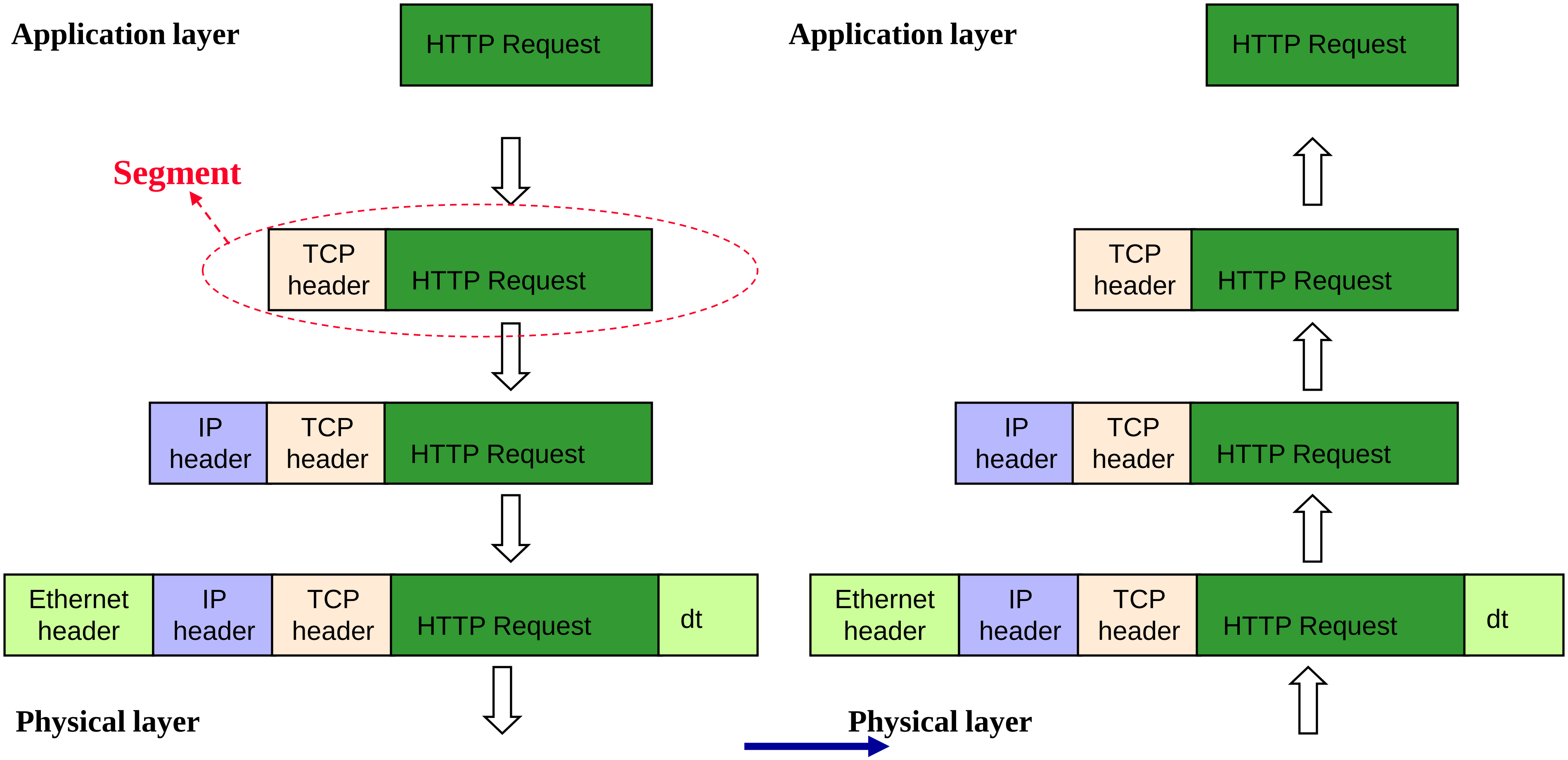
# Encapsulation

- What is HTTP? It is an application protocol (not an application.) HTTP is an application protocol that facilitates the web browsing application
- Question: Name a protocol used in the Data Link layer? Ethernet

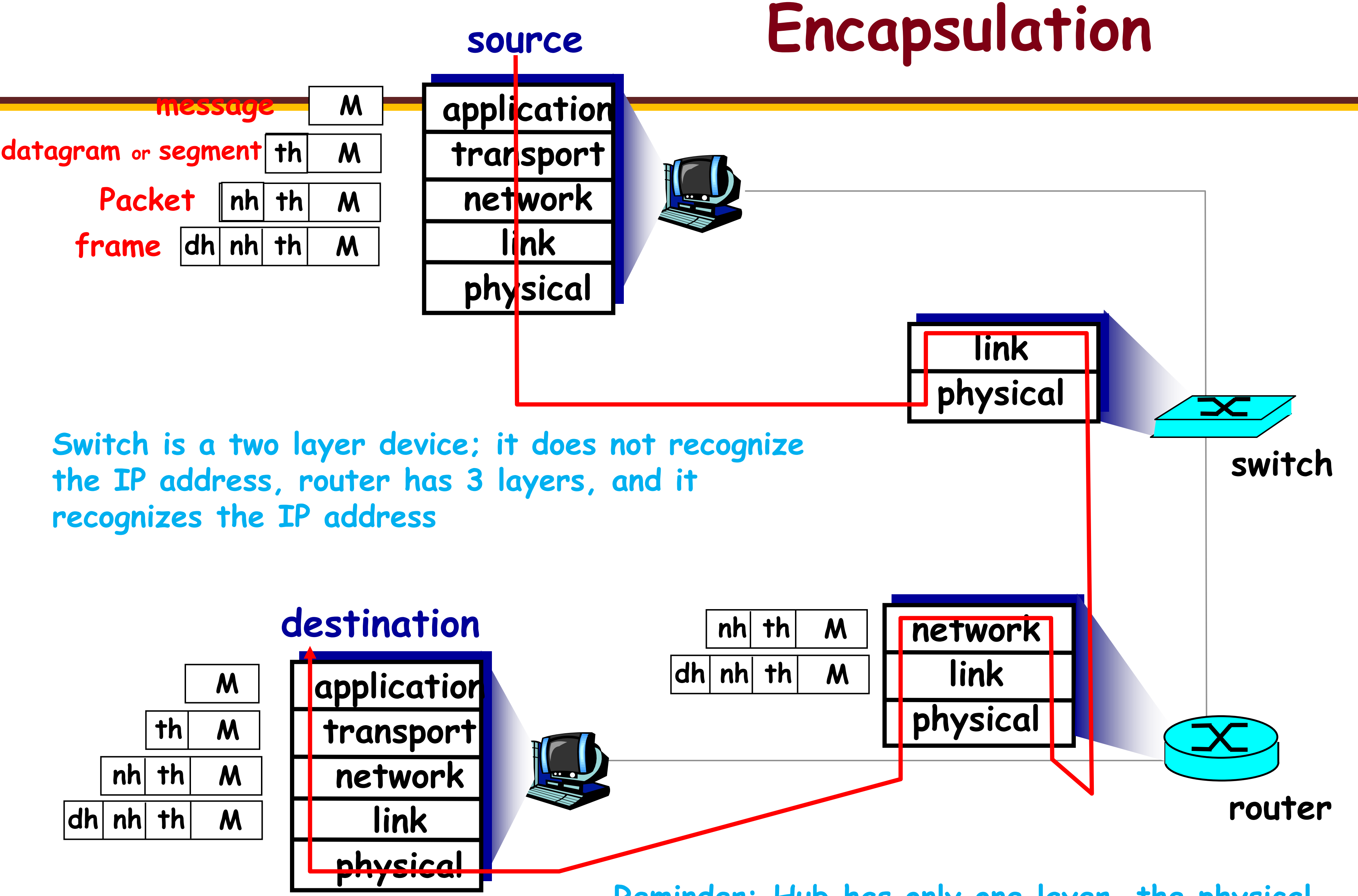


# Encapsulation

- HTTP runs over TCP and so the whole PDU (payload + header) is called *segment*



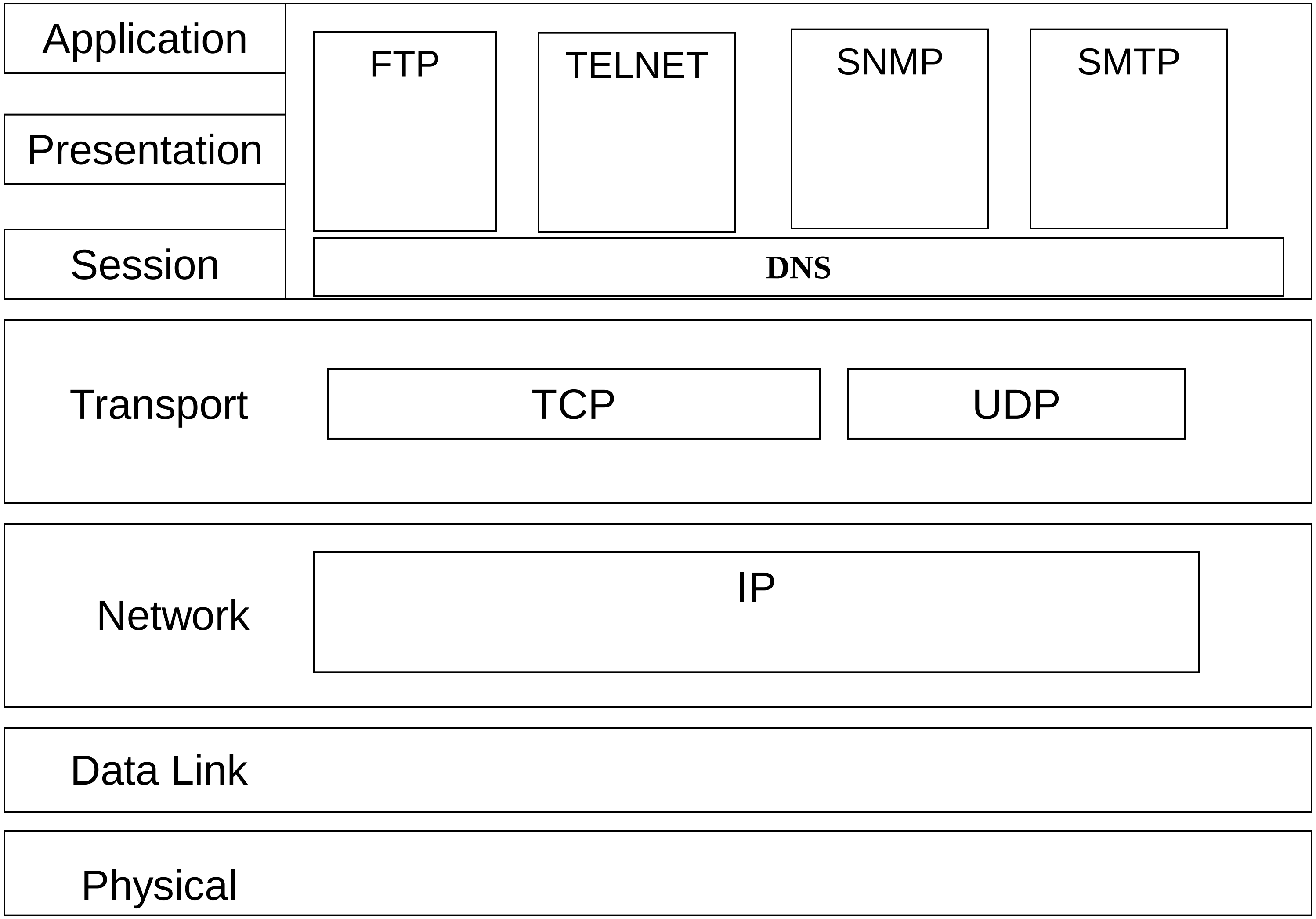
# Encapsulation



Switch is a two layer device; it does not recognize the IP address, router has 3 layers, and it recognizes the IP address

Reminder: Hub has only one layer, the physical layer, which deals with raw stream of bits; hub does not recognize any address

# OSI and TCP/IP Comparison



Application

Presentation

Session

FTP

TELNET

SNMP

SMTP

DNS

Transport

TCP

UDP

Network

IP

Data Link

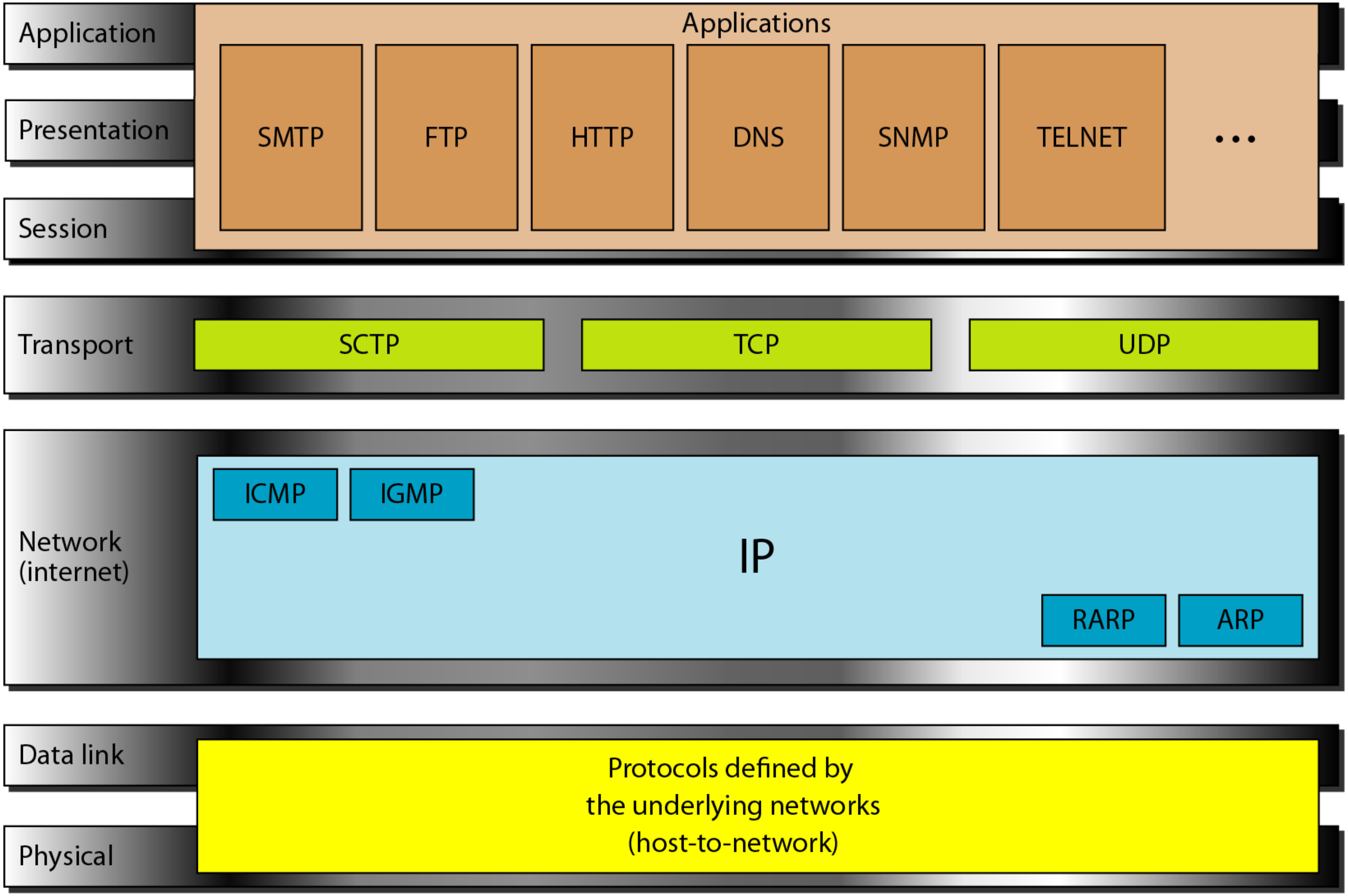
Physical

# OSI and TCP/IP Comparison (Cont.)

- DNS (Domain Name System) resides in the Session Layer. It does not interact with the end user. However, all other applications, can interact with end user and their protocols (e.g., FTP) interact with the DNS
- We said Internet is a network of networks with different technologies. By different technologies we mean that their first two layers are different, e.g., Ethernet, Cable Modem, T1; however they all agree on TCP/IP

# OSI and TCP/IP Comparison (Cont.)

## Which block is misplaced in the following figure?



# TCP/IP Application Layer Communication

- Reminder: the end-to-end Application layer connections are not physical, they are just agreements

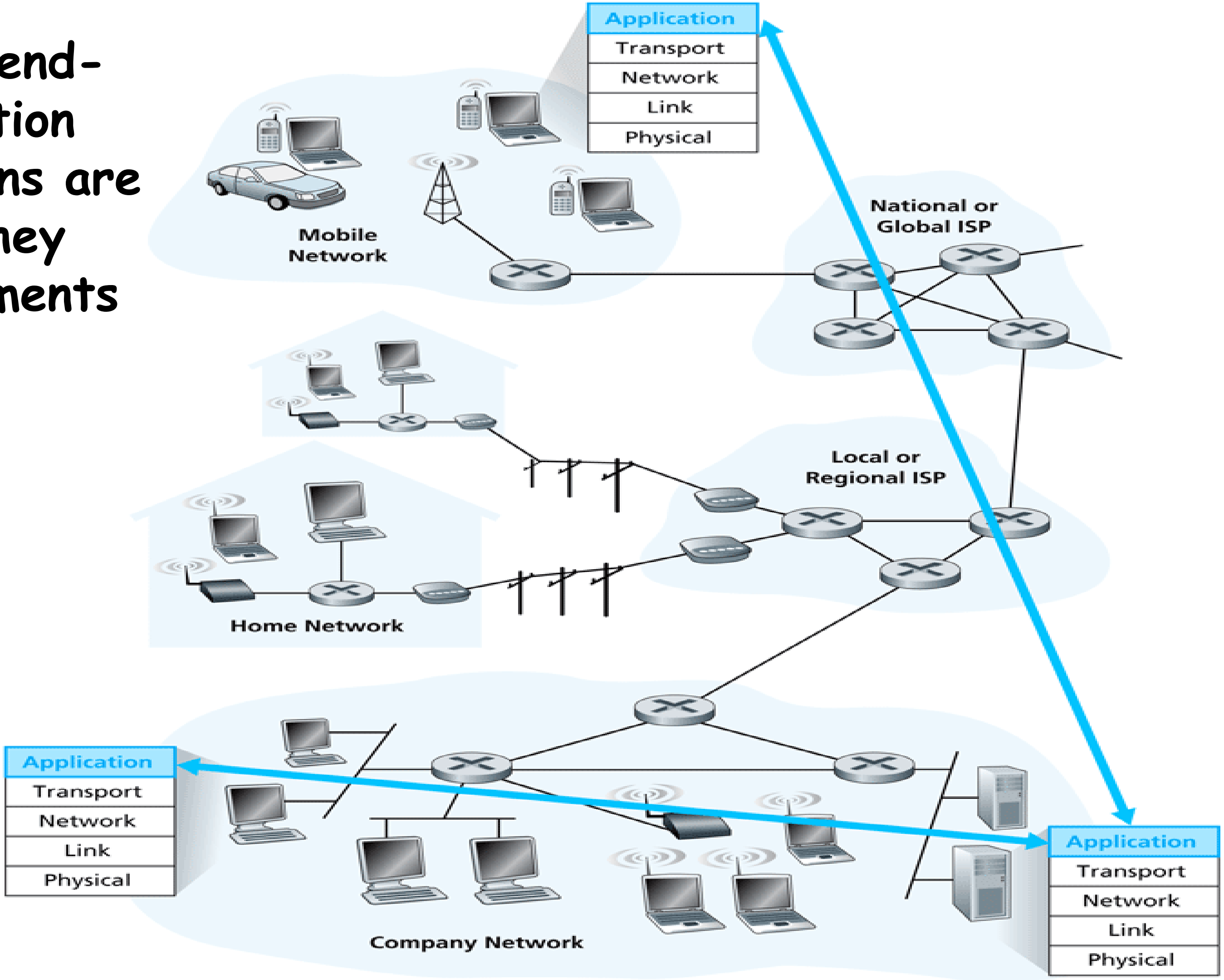


**Figure 2.1** ♦ Communication for a network application takes place between end systems at the application layer.

# TCP/IP Transport Layer Communication

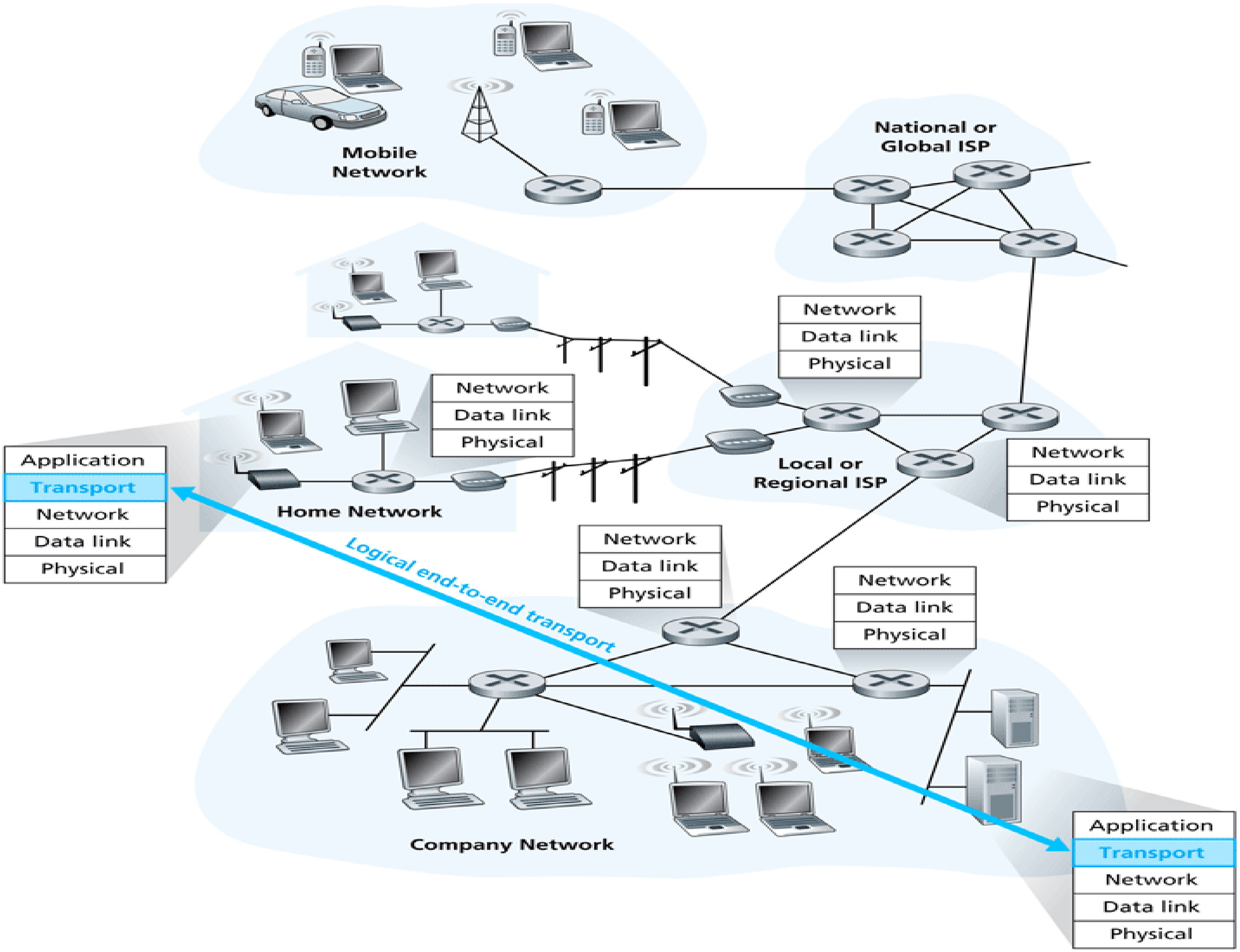


**Figure 3.1** ◆ The transport layer provides logical rather than physical communication between application processes.

# TCP/IP Network Layer Communication

- Reminder: Network layer connections are NOT end-to-end

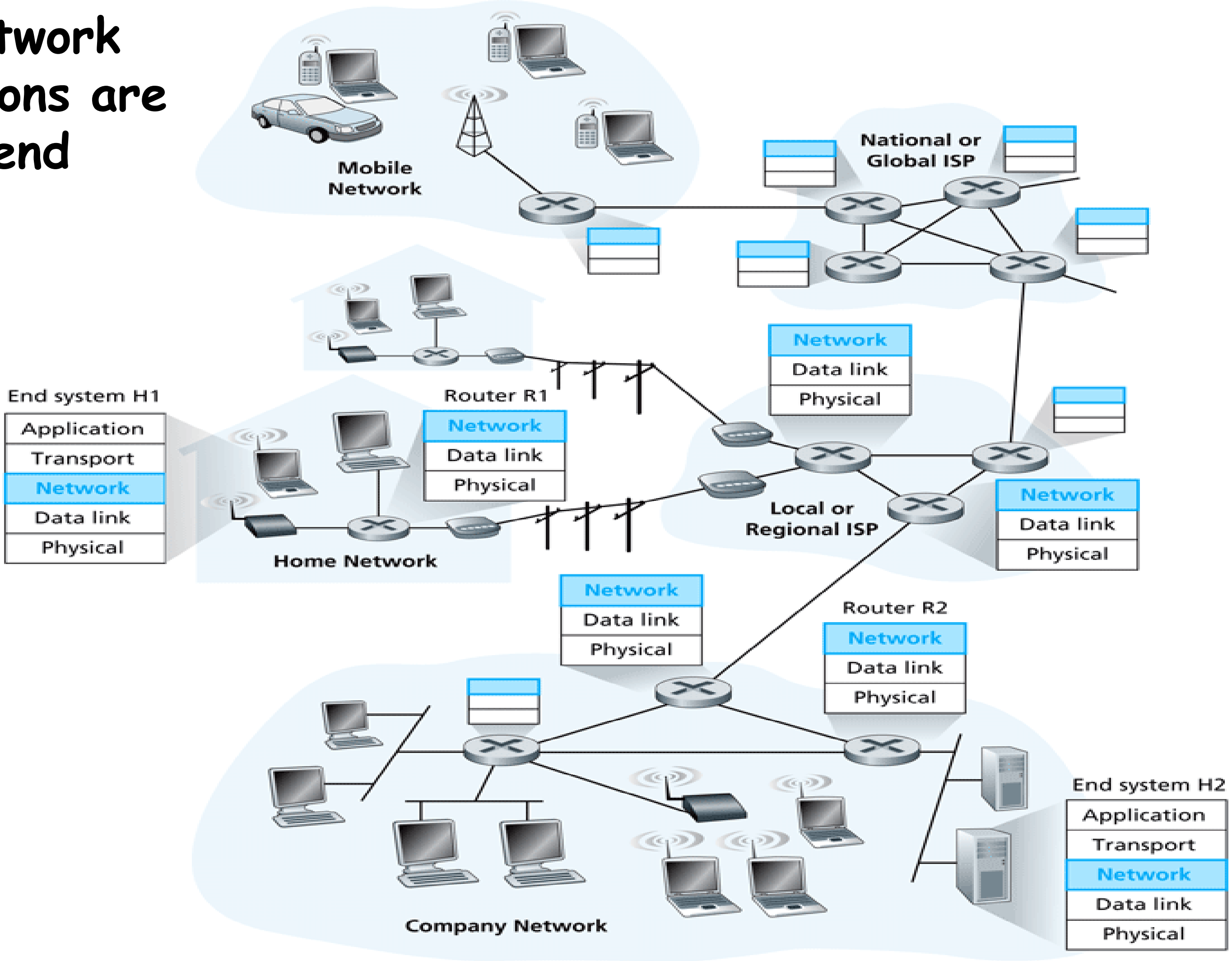


Figure 4.1 ♦ The network layer

# TCP/IP Data Link Layer Communication

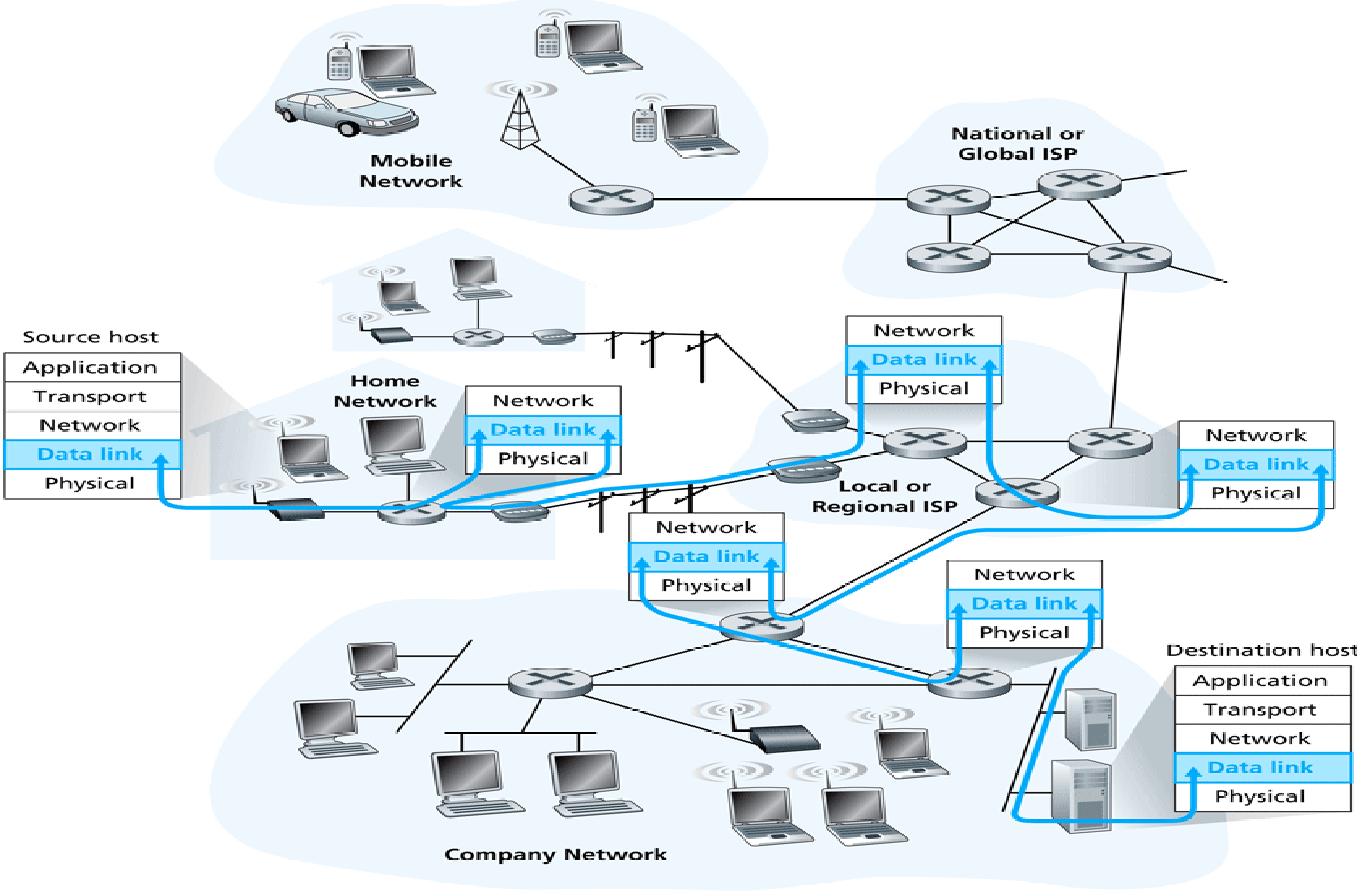


**Figure 5.1** ◆ The link layer