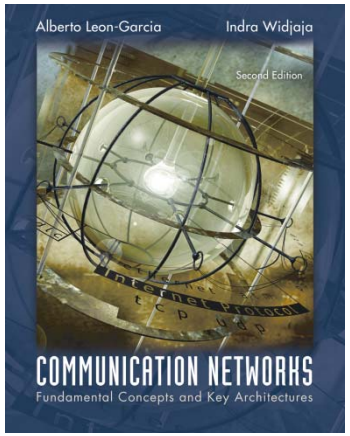
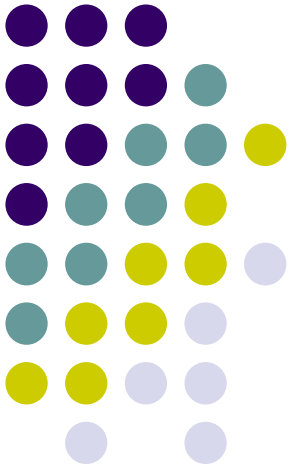
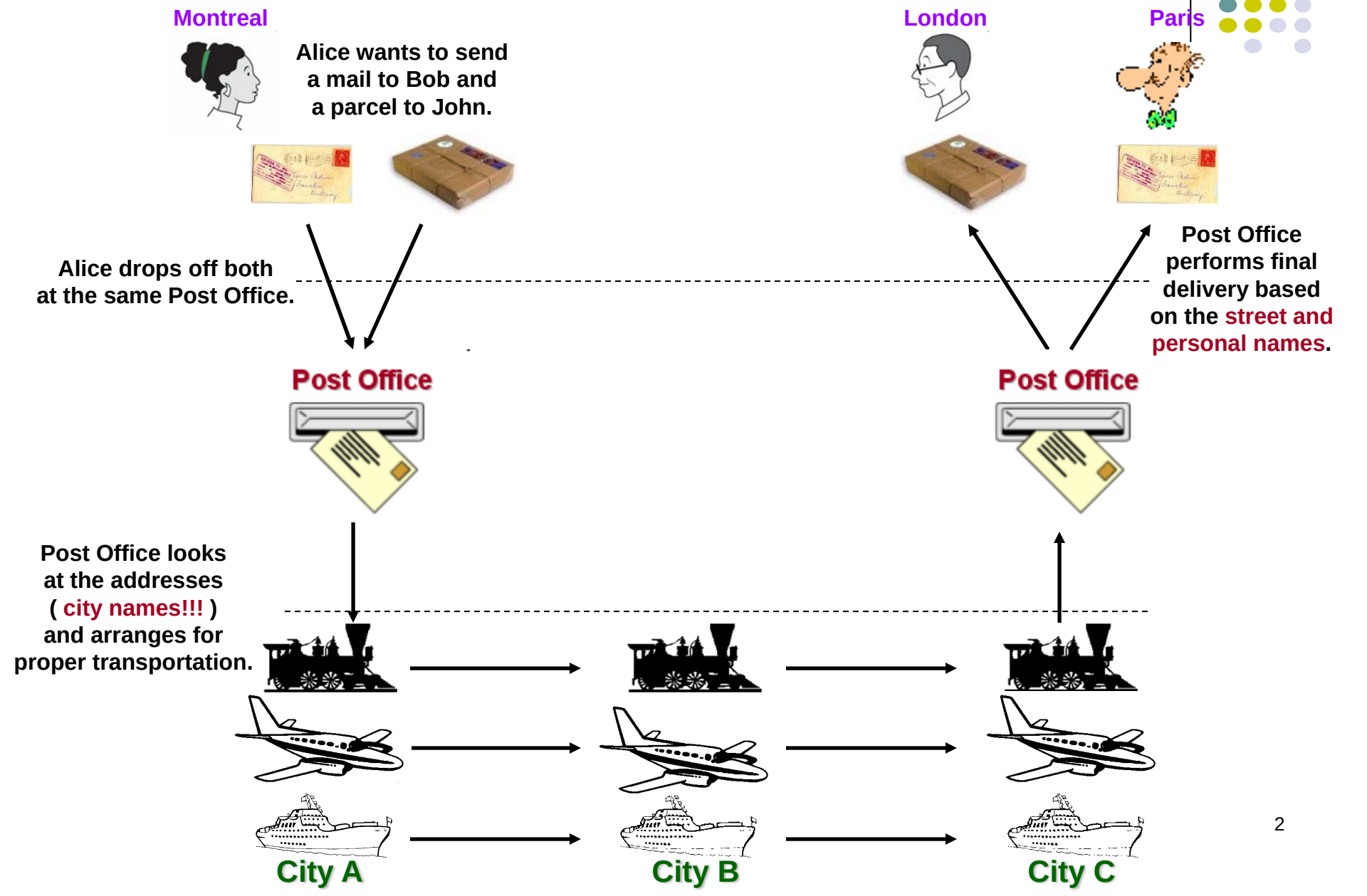
Chapter 2
Applications and Layered Architectures



Protocols, Services & Layering
OSI Reference Model
TCP/IP Architecture
How the Layers Work Together



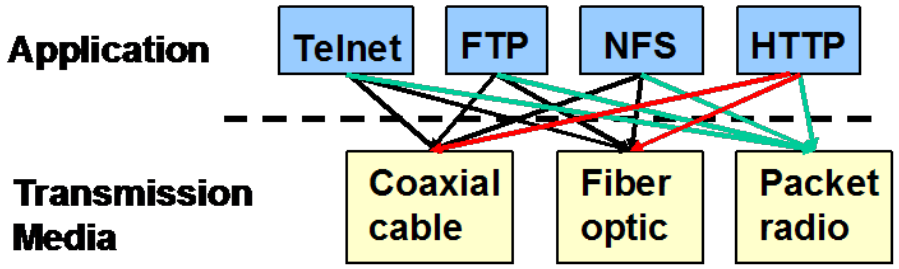
Why Layering?



Why Layering? (cont.)

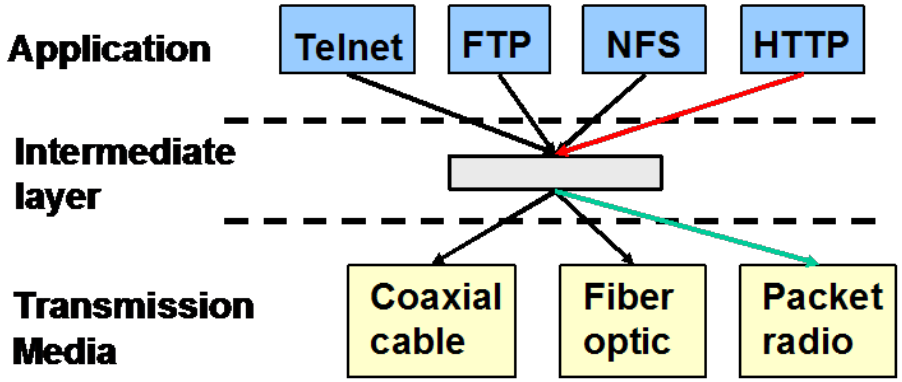
No Layering

- Each new application has to be *re*-implemented for every network technology!



Layering

- intermediate layer(s) provide a unique abstraction for various network technologies



Why Layering? (cont.)

Why Protocol Layering?

1) modularity – one problem is decomposed into a number of smaller more manageable subproblems ⇒ more flexibility in designing, modifying and evolving computer networks

2) functionality reuse – a common functionality of a lower layer can be shared by many upper layers

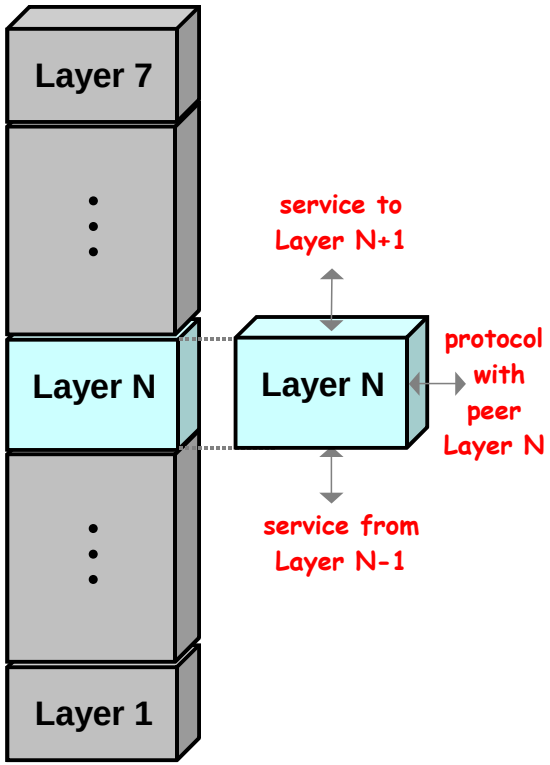
A monolithic network design that uses a single large body of hardware and software to meet all network requirements can quickly become obsolete and also is extremely difficult and expensive to modify.

Layered approach accommodates incremental changes much more rapidly.

Layered Architecture

Protocol Layering

grouping of related communication functions into hierarchical set of layers



- each layer:
  - (1) performs a related subset of functions required for communication with another system
  - (2) relies on next lower layer to perform more primitive functions
  - (3) provides service to next higher layer
  - (4) implements protocol for communication with peer layer in other systems
- vertical communication – communication between adjacent layers – requires mutual understanding of what services and/or information lower layer must provide to layer above
- horizontal communication – communication between software or hardware elements running at the same layer on different machines

Communication between peer processes is virtual, i.e. indirect.

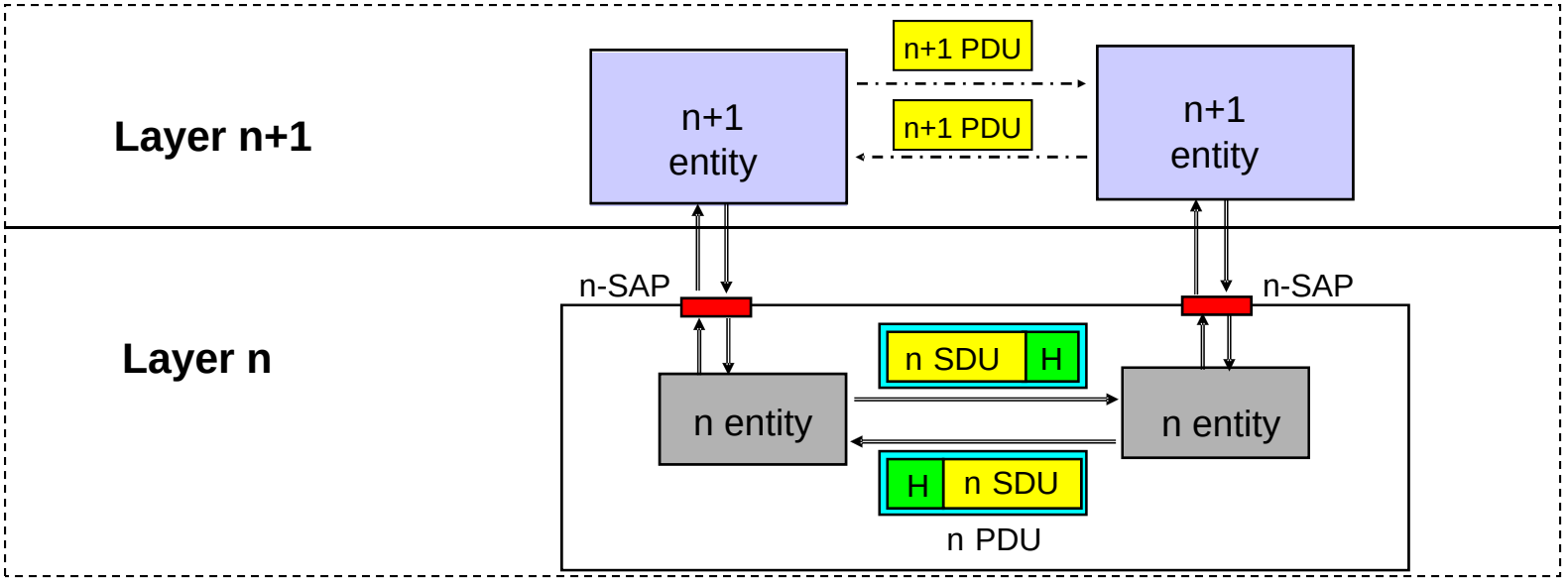
Layered Architecture (cont.)

Protocol – set of rules that governs data comm. between peer entities

- layer-n peer processes communicate by exchanging Protocol Data Units (PDUs)

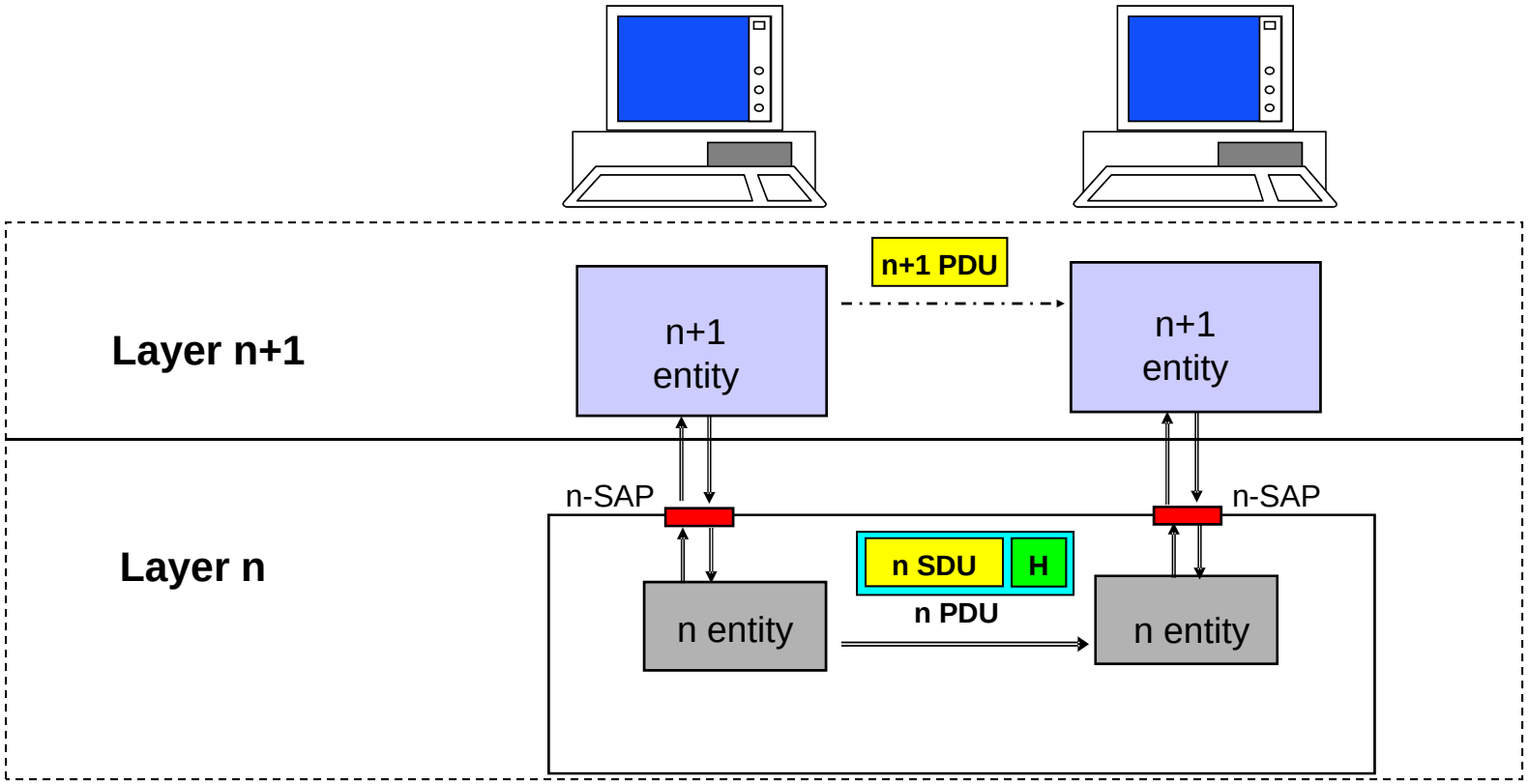
Service – can be accessed through Service Access Points (SAP's)

- layer n+1 PDU = layer n SDU (SDU = Service Data Unit)
- layer n process adds control information (header) to its SDU to produce layer n PDU – encapsulation!
- layer n does not interpret or make use of information contained in its SDU



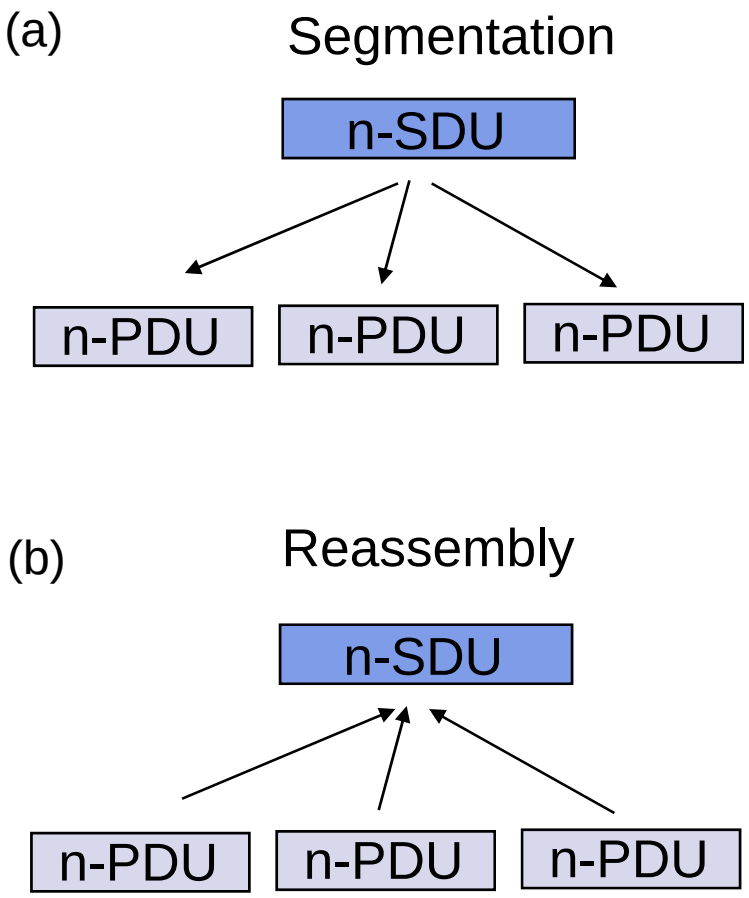
Layered Architecture (cont.)

Example [layering – vertical vs. horizontal flow of information]



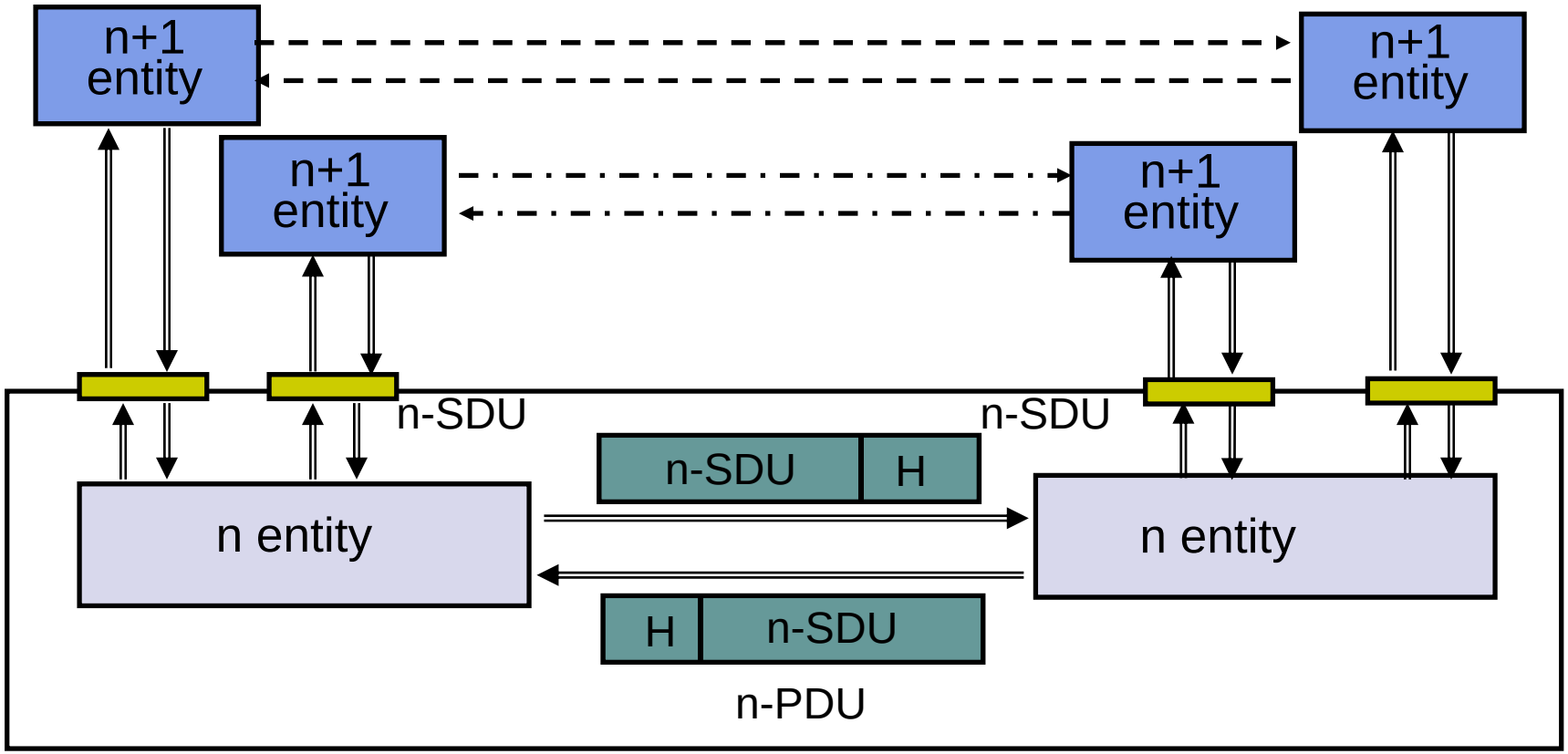
Segmentation & Reassembly

- A layer may impose a limit on the size of a data block that it can transfer for implementation or other reasons
- Thus a layer-n SDU may be too large to be handled as a single unit by layer-(n-1)
- Sender side: SDU is segmented into multiple PDUs
- Receiver side: SDU is reassembled from sequence of PDUs

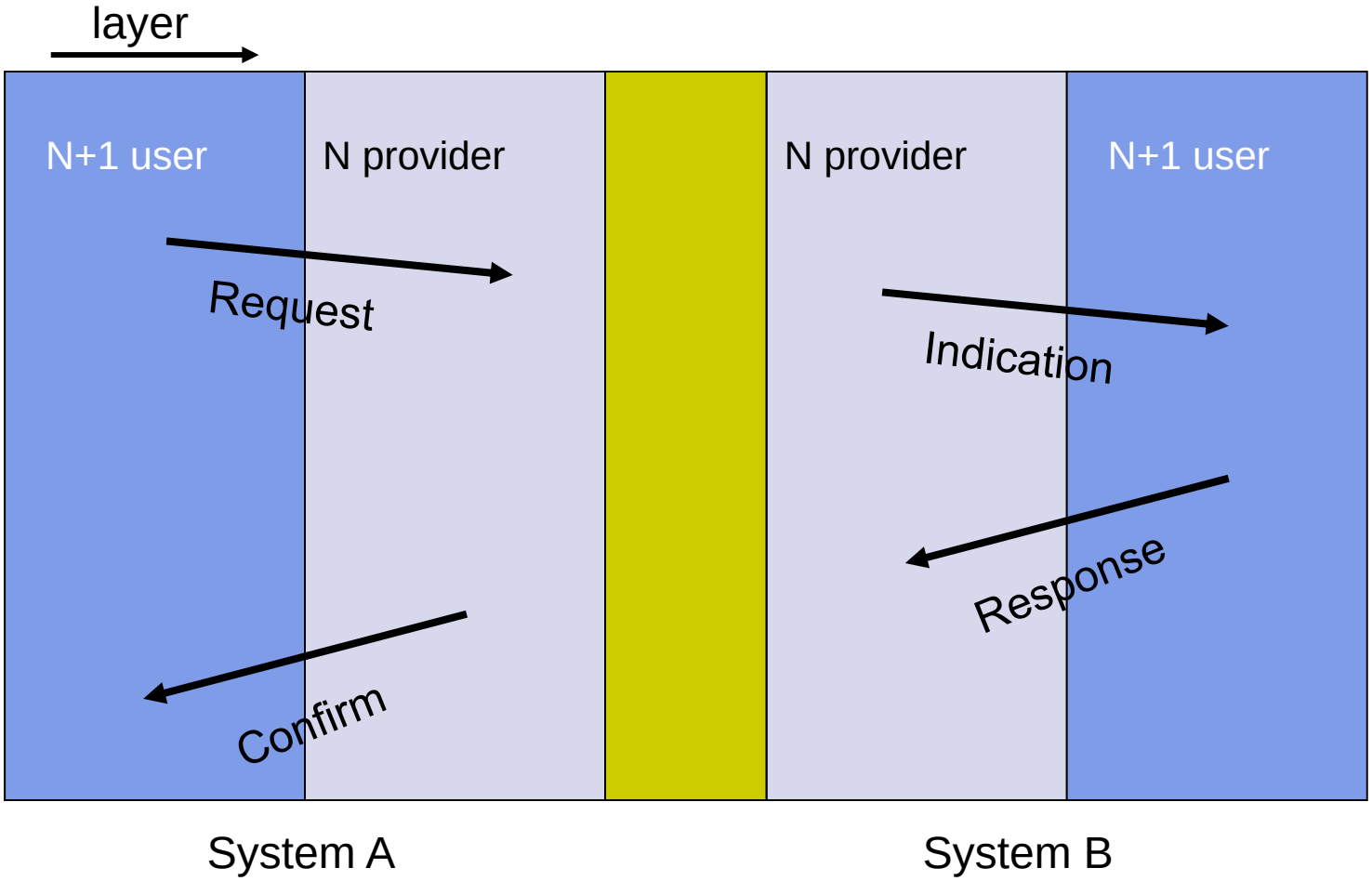


Multiplexing

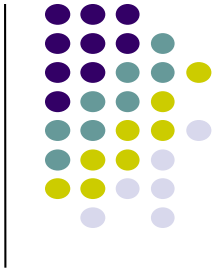
- Sharing of layer n service by *multiple* layer n+1 users
- Multiplexing tag or ID required in each PDU to determine which users an SDU belongs to



Interlayer Interaction

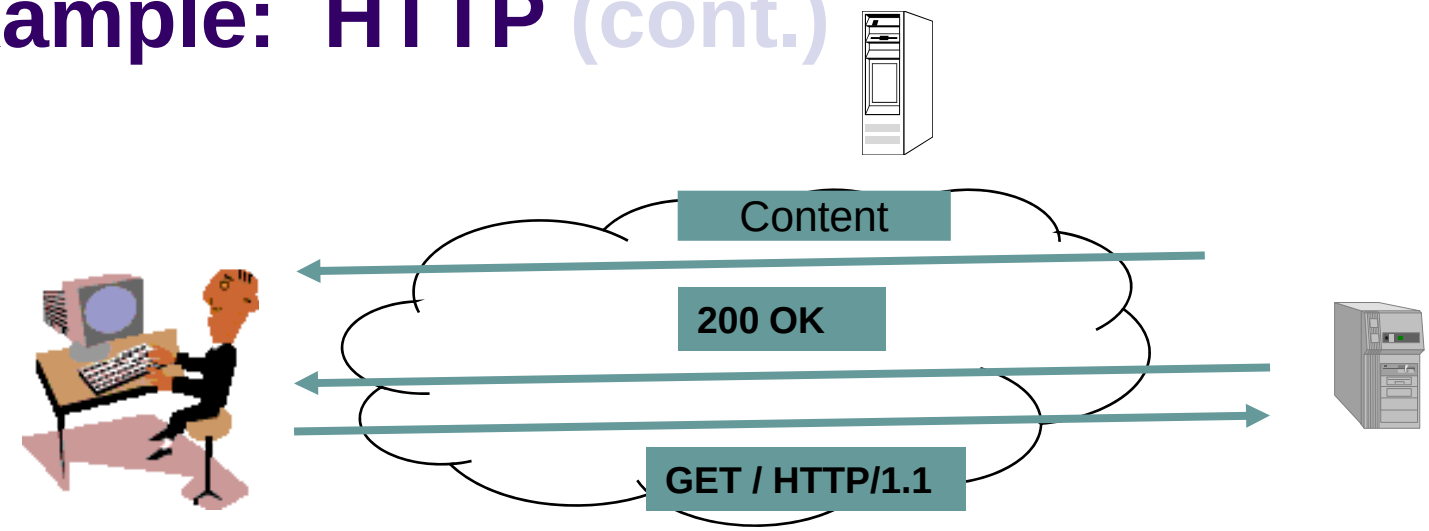


Example: HTTP



- HTTP is an application layer protocol
- Retrieves documents on behalf of a browser application program
- HTTP specifies fields in request messages and response messages
  - Request types; Response codes
  - Content type, options, cookies, …
- HTTP specifies actions to be taken upon receipt of certain messages

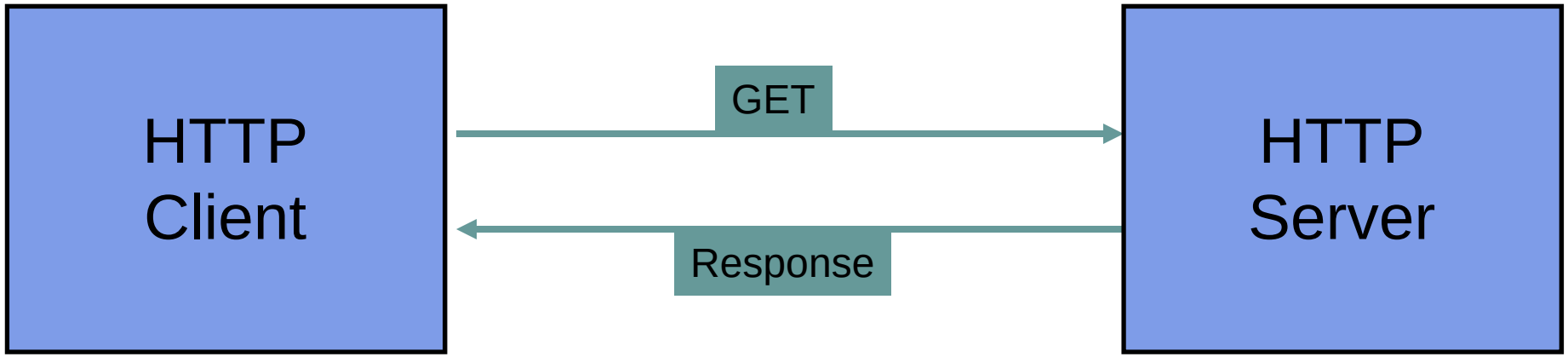
Example: HTTP (cont.)



- HTTP client sends its request message: “GET …”
- HTTP server sends a status response: “200 OK”
- HTTP server sends requested file
- Browser displays document

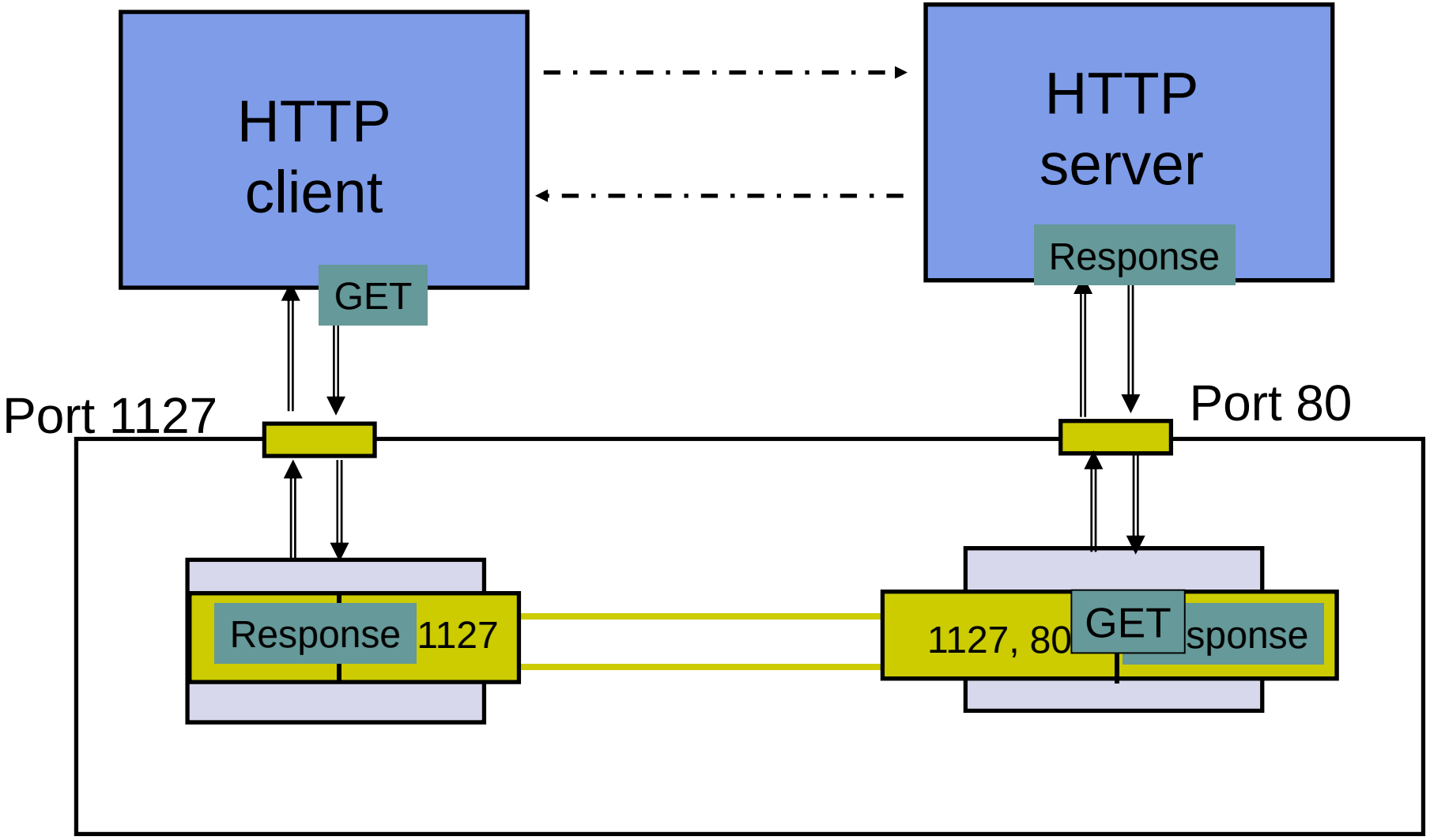
- Clicking a link sets off a chain of events across the Internet!
- Let’s see how protocols & layers come into play…

HTTP Protocol

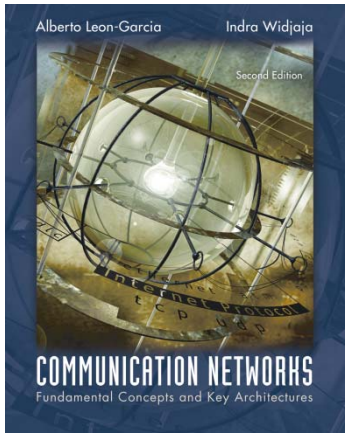


- HTTP assumes messages can be exchanged directly between HTTP client and HTTP server
- In fact, HTTP client and server are processes running in two different machines across the Internet
- HTTP uses the reliable stream transfer service provided by TCP

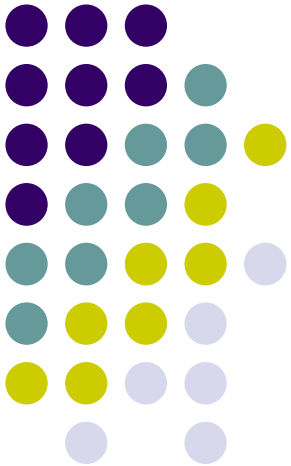
HTTP uses service of TCP



Chapter 2
Applications and Layered Architectures



OSI Reference Model



OSI Reference Model

- Describes a seven-layer abstract reference model for a network architecture
- Purpose of the reference model was to provide a framework for the development of protocols
- OSI also provided a unified view of layers, protocols, and services which is still in use in the development of new protocols
- Detailed standards were developed for each layer, but most of these are not in use
- TCP/IP protocols preempted deployment of OSI protocols

OSI Reference Model (cont.)

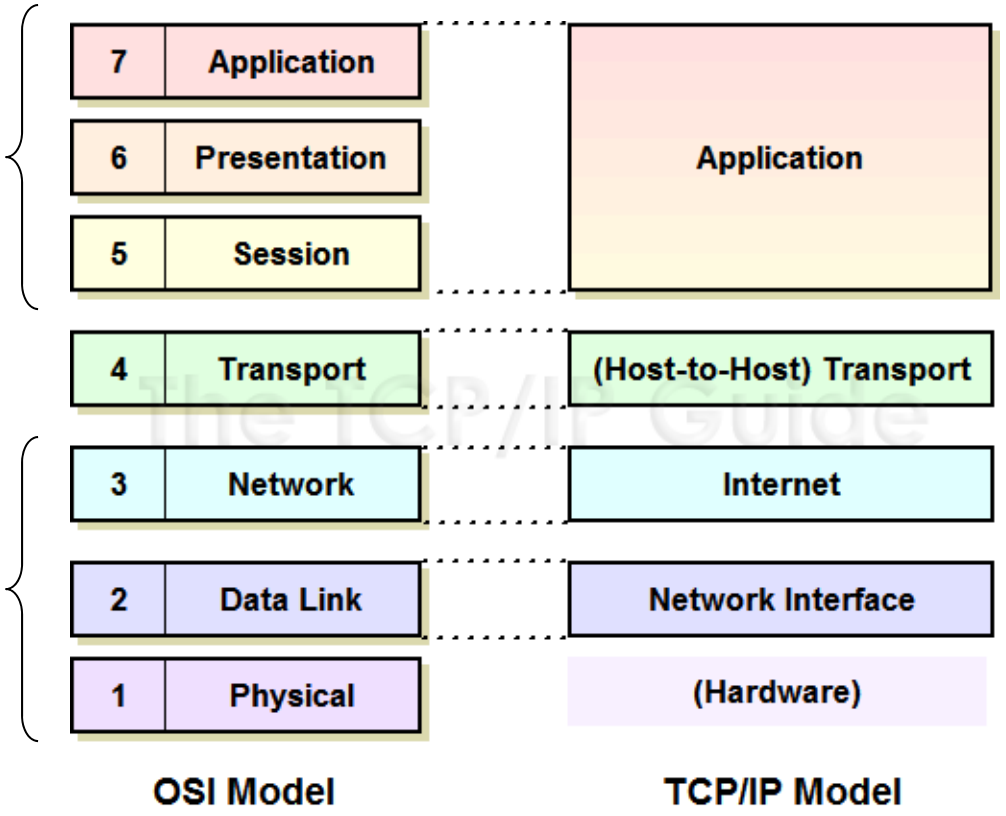
Layered OSI Architecture

- composed of 7 ordered layers
- there is fairly natural correspondence between TCP/IP and OSI layers ⇒ TCP/IP architecture is commonly explained in terms of corresponding OSI layers

application support layers – allows communication with end-user and interoperability among unrelated software systems

transport layer - links upper and lower group - ensures that what lower layers have transmitted is in a form that upper layers can use

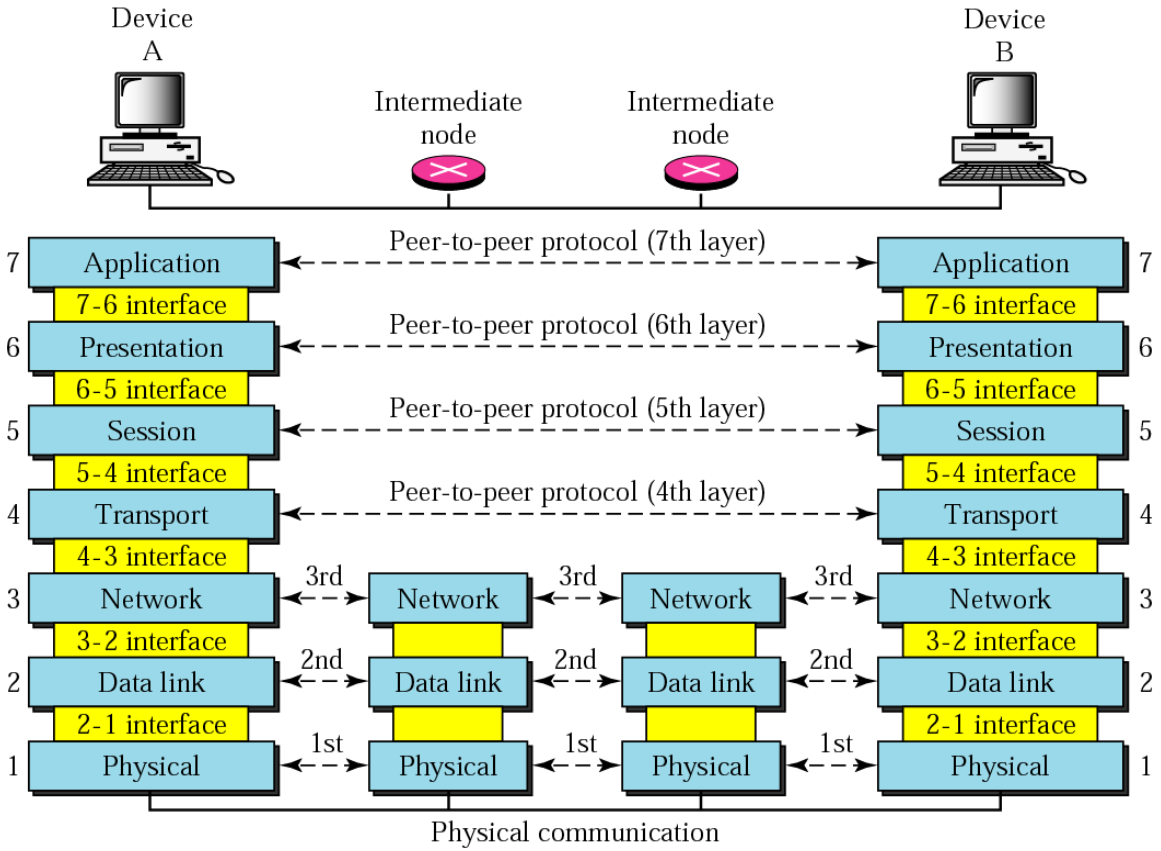
network support layers – deal with physical aspects of moving data from one device to another – across one link and across the whole network



OSI Reference Model (cont.)

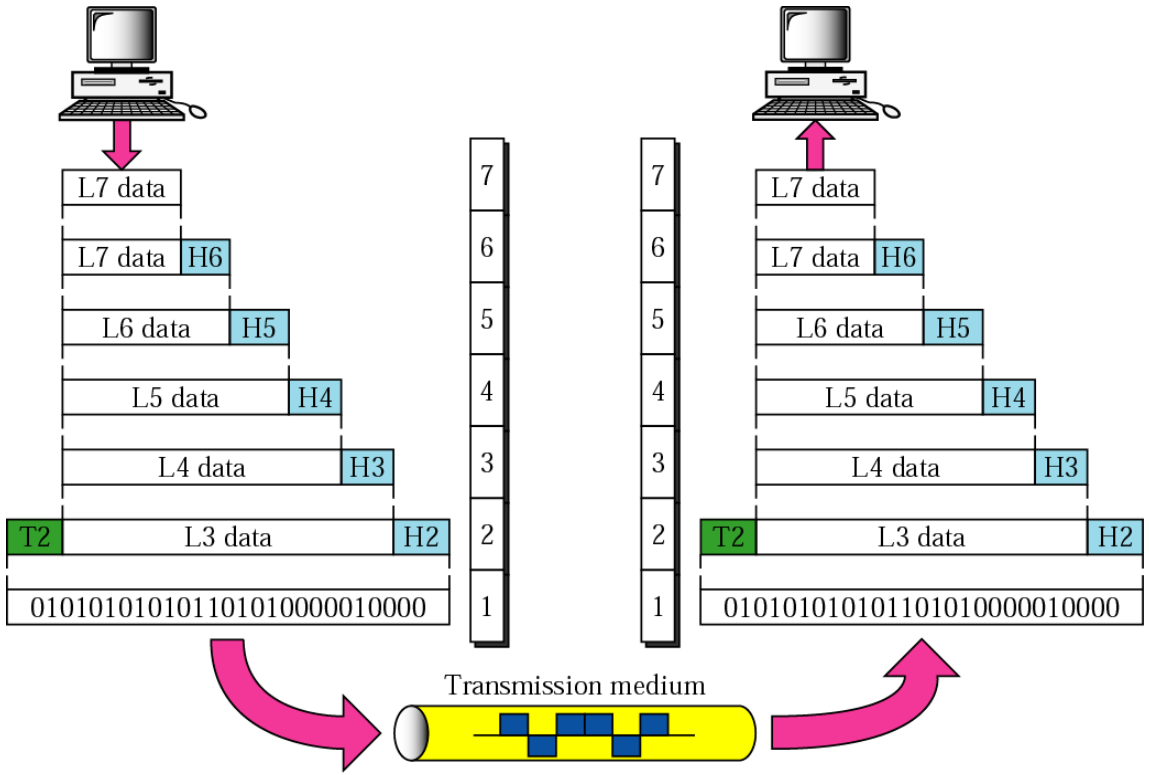
Peer-to-Peer Communication over 7 OSI Layers

- message moves down through layers on sending device, over intermediate nodes, to receiving station, and then back up through layers
- at intermediate nodes (routers), data is pulled only up to network layer, so that next hop could be determined



OSI Reference Model (cont.)

- each layer in sending device adds its own information to message it receives from layer above it and passes whole package to layer just below it – reverse process occurs at receiving device
- when data reaches physical layer, it is changed into electromagnetic signal and sent along a physical link

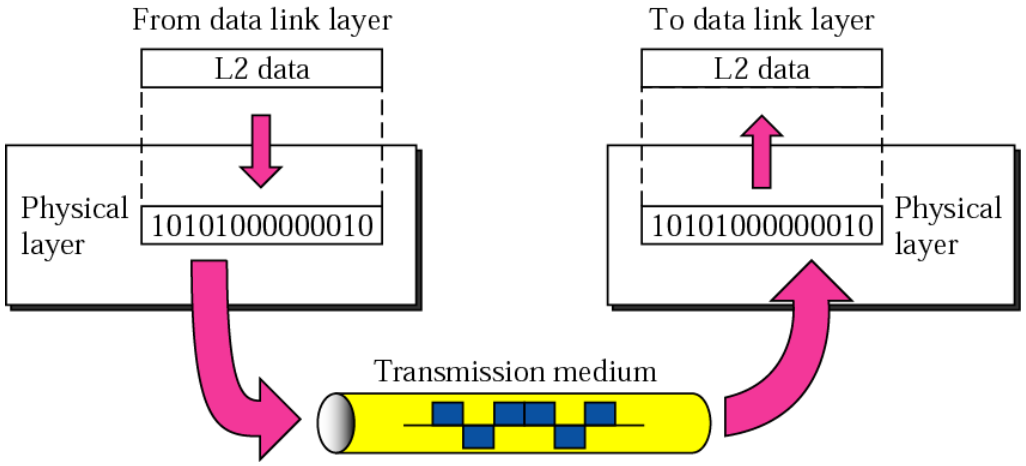


OSI Model (Physical Layer)

1. Physical Layer

- coordinates transmission of bit-stream over physical medium, including
  - representation of bits: to be transmitted, bits must be encoded into signals – electrical or optical; P.L. defines type of encoding – how 0s and 1s are changed to signals (e.g. 1 = +1V, 0 = -1V)
  - bit length – data rate: P.L. defines how long a bit lasts and, accordingly, number of bits sent each second

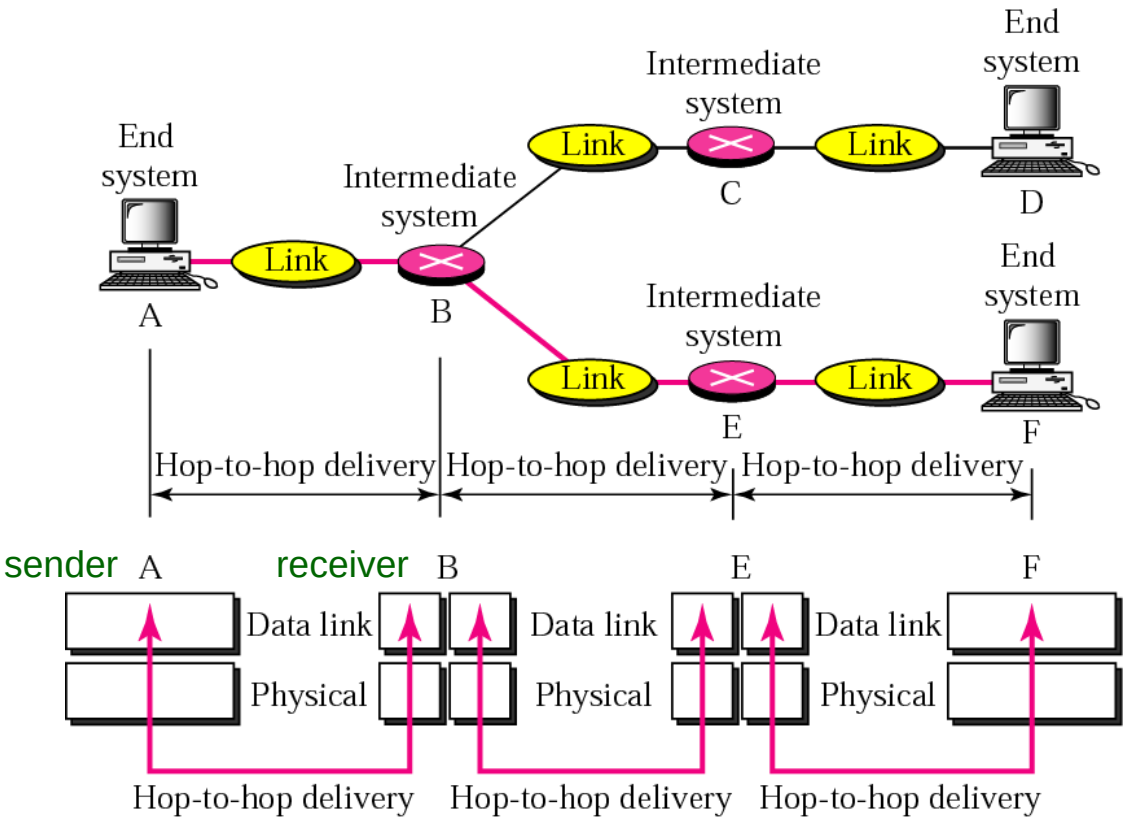
    (different values for copper wire, coaxial cable, fiber-optics, …)



OSI Model (Data-Link Layer)

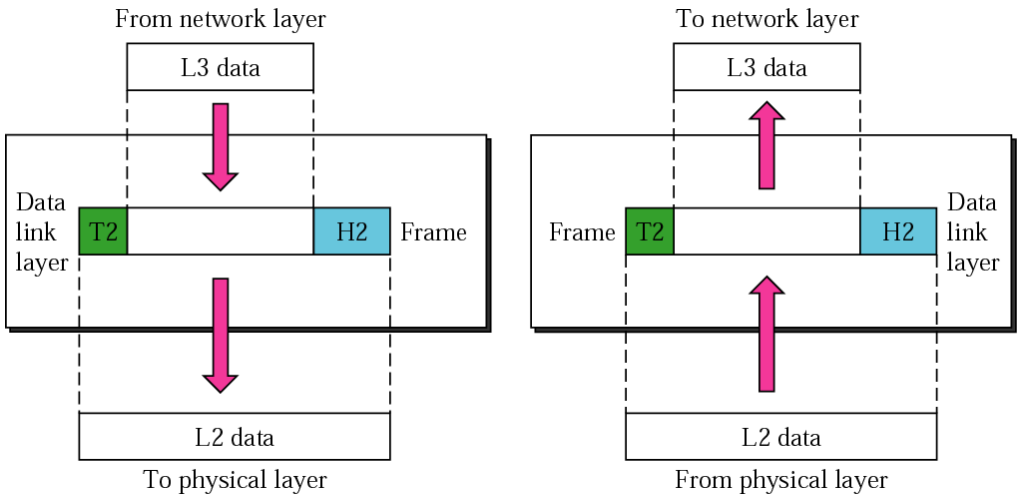
2. Data-Link Layer

The data link layer transforms the physical layer, a raw stream of bits, to a reliable link between two devices on the same network.
It makes the physical layer appear error-free to the upper layer.



OSI Model (Data-Link Layer)

- framing: The D.L.L divides the stream of bits received from the network layer into manageable data units called frames.
- physical addressing: The D.L.L adds a header to the frame to specify the NIC address of appropriate receiver on the other side (of wire).
- error control: The D.L.L adds reliability to the physical layer by adding a trailer with information necessary to detect / recover damaged or lost frames.
- access control. When two or more devices are connected to the same link, the D.L.L determines which device has control over the link at any given time.

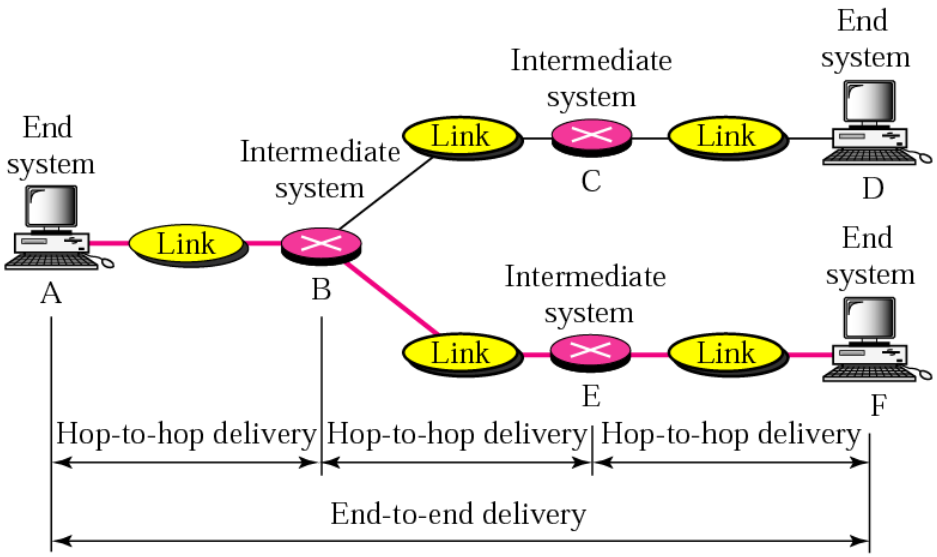


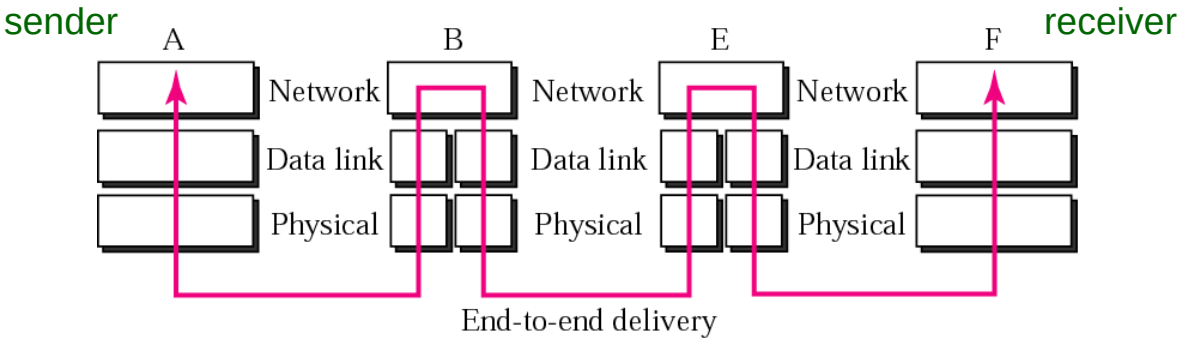
OSI Model (Network Layer)

3. Network Layer

While the data link layer oversees the delivery of packets between two devices on the same network, the network layer is responsible for the source-to-destination delivery of packet across multiple networks / links.

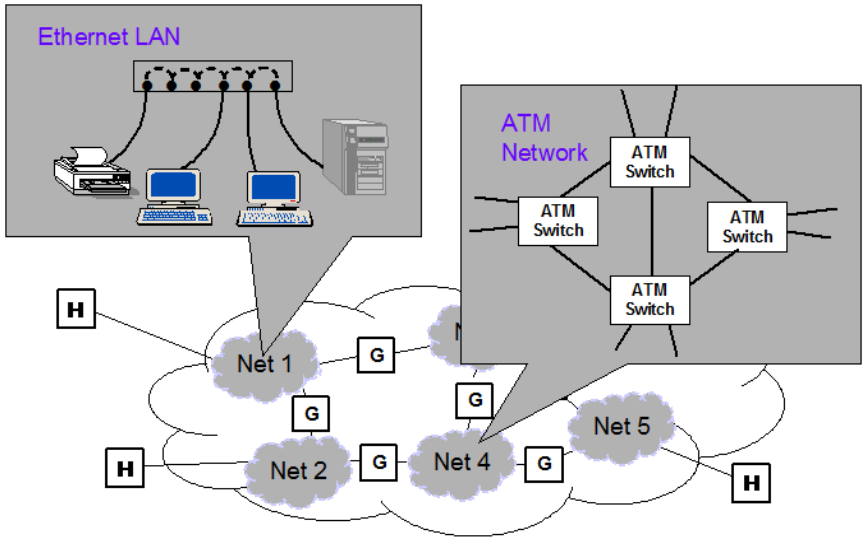
Routing over multiple networks:
1) in min time, AND
2) with min overhead.





OSI Model (Network Layer)

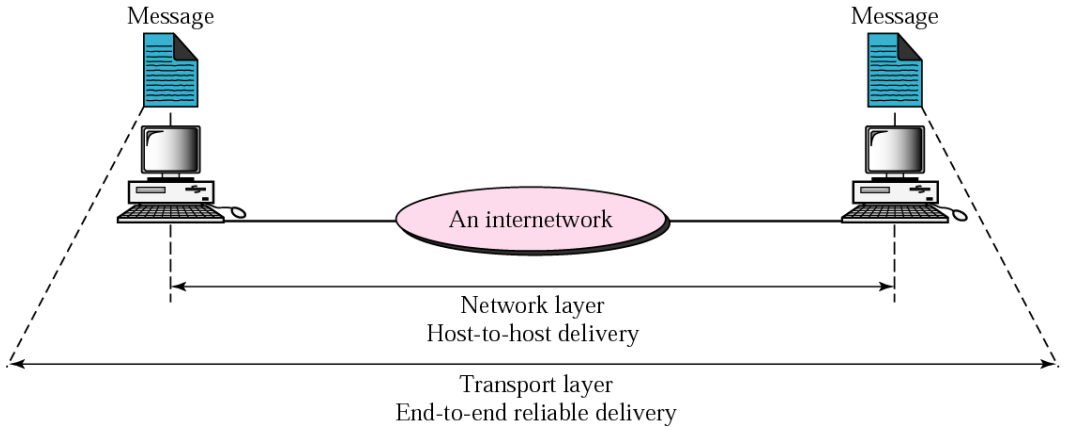
- logical addressing: The physical addressing implemented by the data link layer handles the addressing / delivery problem locally – over a single wire. If a packet passes the network boundary another addressing system is needed to help distinguish between the source and destination network.
- routing: The N.L. provides the mechanism for routing/switching packets to their final destination, along the optimal path – across a large internetwork.
- fragmentation and reassembly: The N.L. sends messages down to the D.L.L. for transmission. Some D.L.L. technologies have limits on the length of messages that can be sent. If the packet that the N.L. wants to send is too large, the N.L. must split the packet up, send each piece to the D.L.L, and then have pieces reassembled once they arrive at the N.L. on the destination machine.

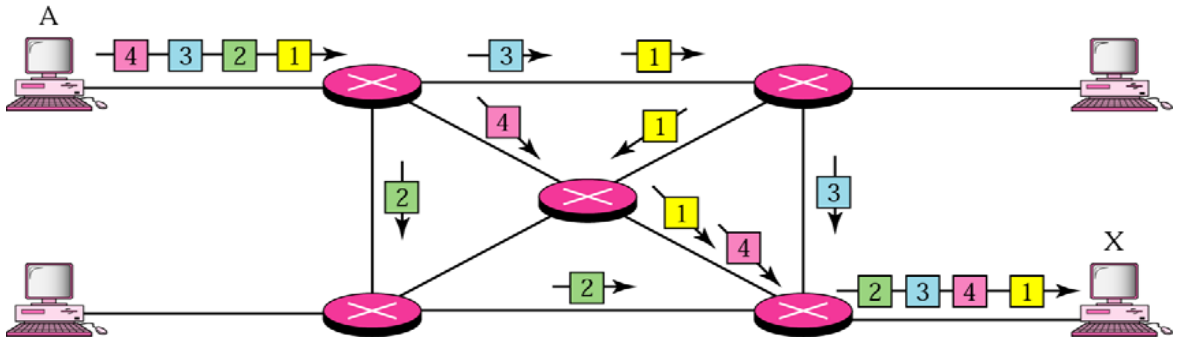


OSI Model (Transport Layer)

4. Transport Layer

The transport layer is responsible for process-to-process delivery of entire message. While the network layer gets each packet to the correct computer, the transport layer gets the entire message to the correct process on that computer.





OSI Model (Transport Layer)

- port addressing: Computers often run several processes at the same time. For this reason, process-to-process delivery means delivery not only from one computer to the other but also from a specific process on one computer to a specific process on the other. The transport layer header therefore must include a type of address called a port address.
- segmentation and reassembly: A message is divided into segments, each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arrival at the destination, and to identify and replace packets that were lost in the transmission.
- flow & error control: Flow & error control at this layer are performed end-to-end rather than across a single link.