Distributed Systems Design
COMP 6231

Communication

Lecture 4

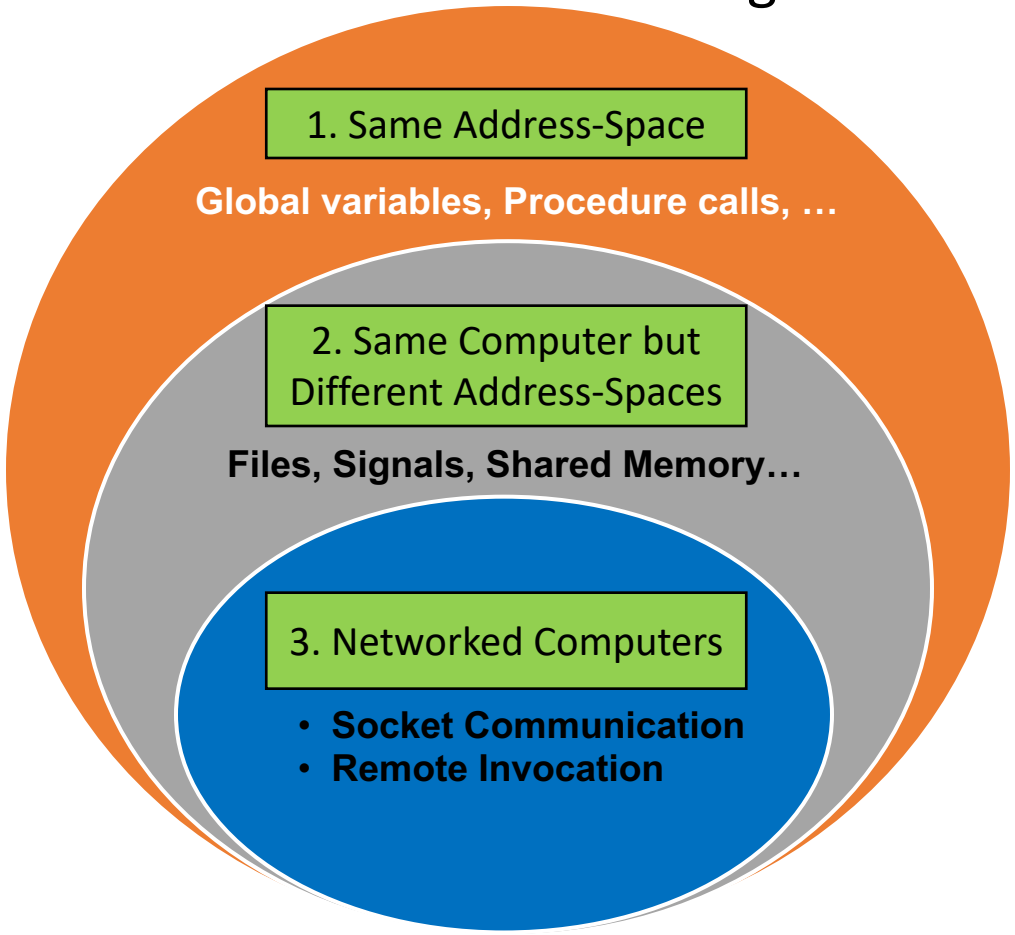
Essam Mansour

Today…

- Last Session:
  - Chapter 3
- Today's Session (Chapter 4.1):
  - Basic networking model
  - Layered Protocols

Classification of Communication Paradigms

- Communication paradigms can be categorized into *three* types based on where the entities reside. If entities are running on:



Today, we will study how entities that reside on networked computers communicate in distributed systems using socket communication and remote invocation

Networks in Distributed Systems

- A distributed system is simply a collection of components that communicate to solve a problem

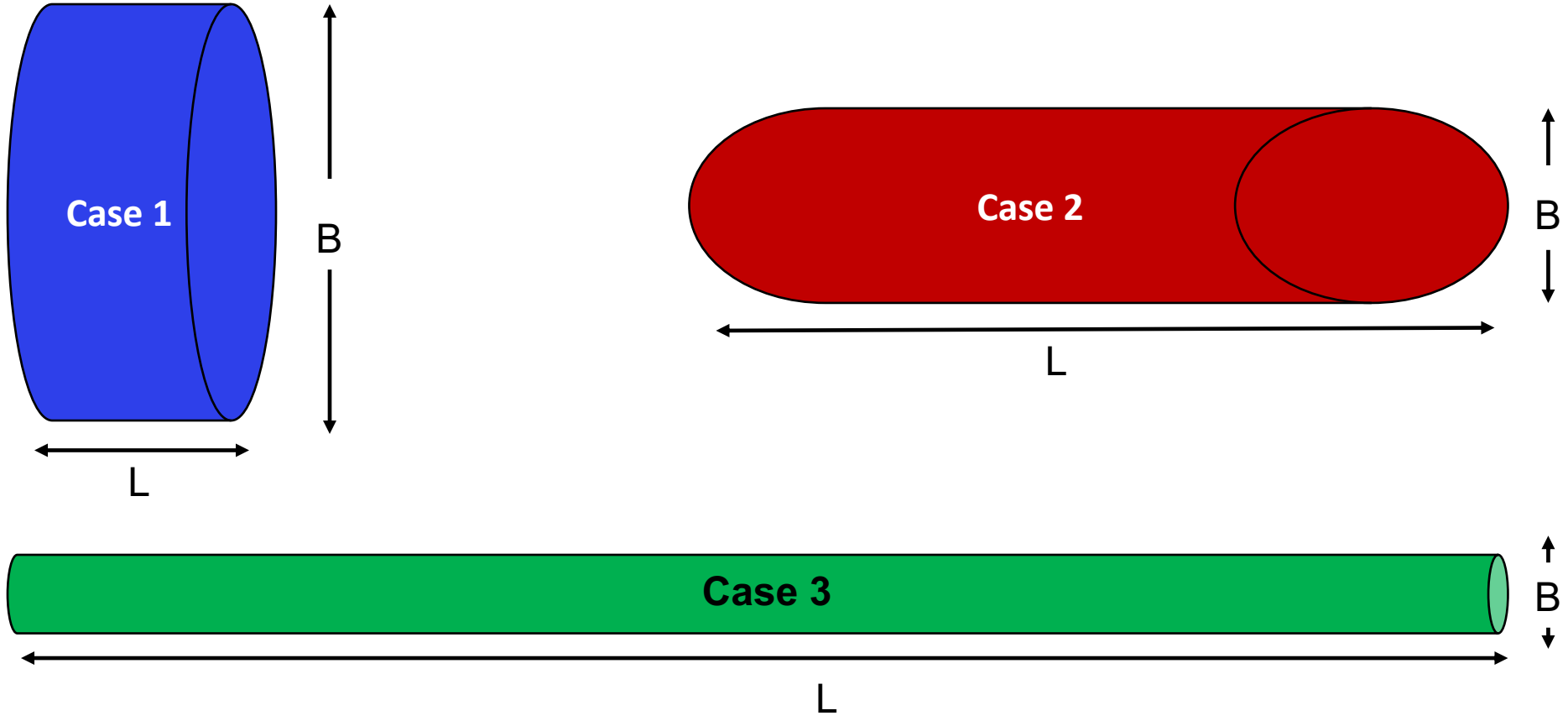
- Why should designers of distributed systems know about networks?
  - Networking issues severely affect performance, fault-tolerance, and security of distributed systems
  - E.g., Gmail outage on Sep 1, 2010 – Google Spokesman said "*we had slightly underestimated the load which some recent changes placed on the request routers. … . few of the request routers became overloaded… causing a few more of them to also become overloaded, and within minutes nearly all of the request routers were overloaded.*"

A Primer: Latency and Bandwidth

Road 1

Road 2

A Primer: Latency and Bandwidth



- B = Bandwidth (or *Capacity*) and L = Latency (or *Delay*)
- B × L gives approximately the number of bits in flight
- As B × L increases, *uncertainty* increases (more bits might get lost)
- High value of B × L leads to *"Buffer Bloat"*

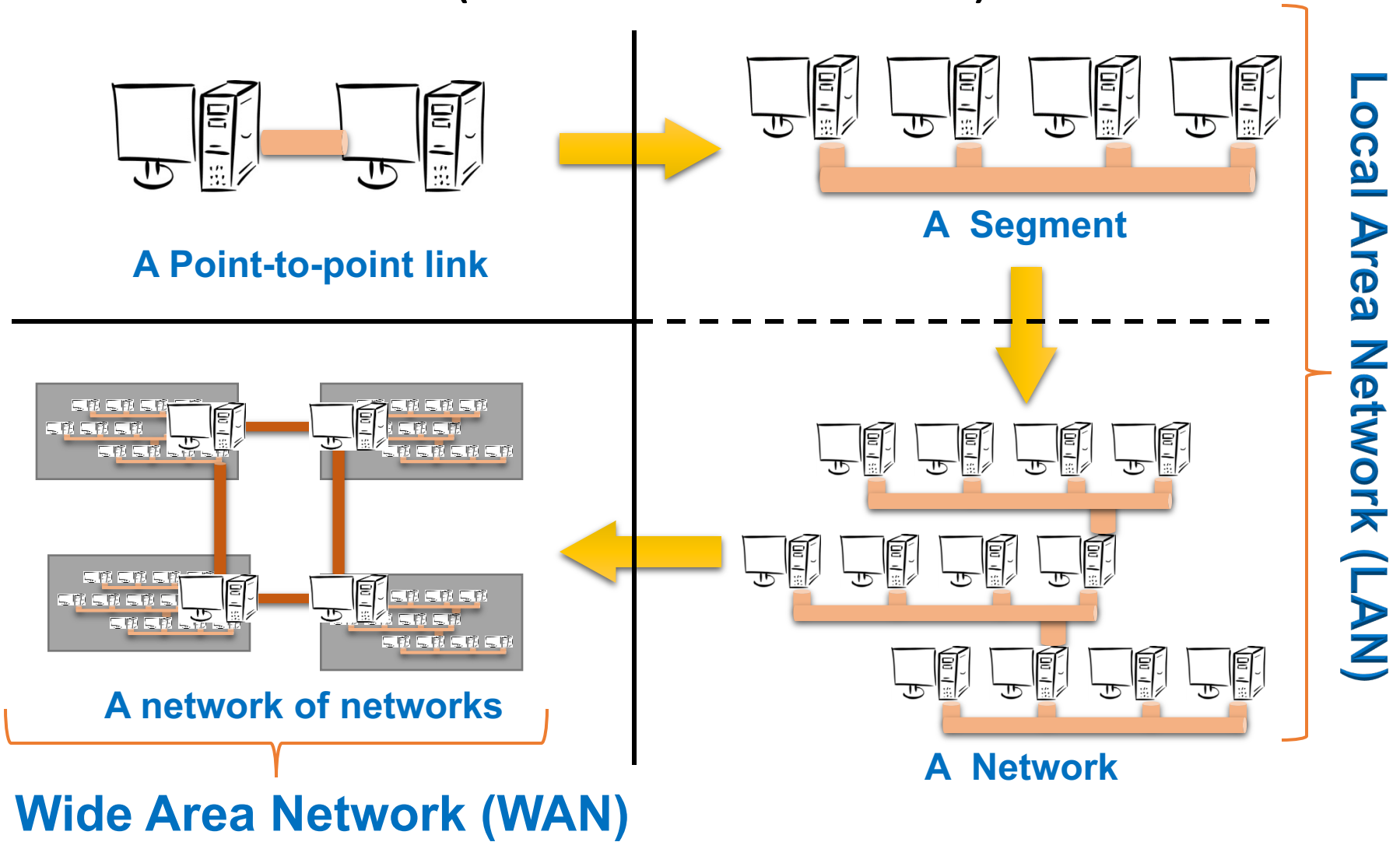
Networks in Distributed Systems

Networking Issue	Comments on a Distributed System Design
Performance	Affects choices of whether to optimize for network or other resources
Scalability	Size of Internet is increasing; expect greater traffic and latency in future
Reliability	Detect communication errors and perform error-checks at the application layer (*end-to-end argument!*)
Security	Install firewalls at gateways; deploy end-to-end authentication; employ encryption, etc.,
Mobility	Expect intermittent connection for mobile devices
Quality-of-service	Internet is best-effort. It is hard to ensure strict QoS guarantees for, say, multimedia data

Network Classification

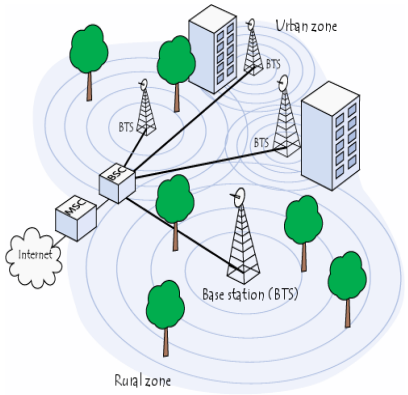
- Important ways to classify networks
  1. Based on size
     - Local Area Networks (LAN)
     - Wide Area Networks (WAN)
  2. Based on technology
     - Ethernet Networks
     - Wireless Networks
     - Cellular Networks

Brief Summary of Important Networks (Based on Size)



Types of Networks – Based on Technology

- Ethernet Networks
  - Predominantly used in the wired Internet
- Wireless LANs
  - Primarily designed to provide wireless access to the Internet
  - Low-range (100s of m), high-bandwidth
- Cellular networks (2G/3G/4G)
  - Initially, designed to carry voice
  - Large range (few kms)
  - Low-bandwidth



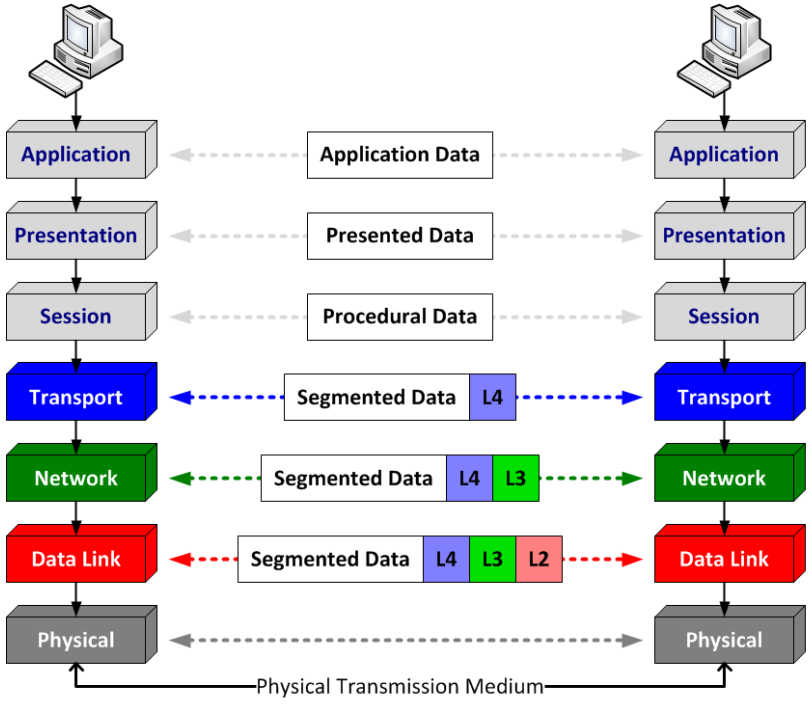
Typical Performance for Different Types of Networks

Network	Example	Range	Bandwidth (Mbps)	Latency (ms)
Wired LAN	Ethernet	1-2 km	10 – 10,000	1 – 10
Wired WAN	Internet	Worldwide	0.5 – 600	100 – 500
Wireless PAN	Bluetooth	10 – 30 m	0.5 – 2	5 – 20
Wireless LAN	WiFi	0.15 – 1.5 km	11 – 108	5 – 20
Cellular	2G – GSM	100m – 20 km	0.270 – 1.5	5
Modern Cellular	3G	1 – 5 km	348 – 14.4	100 – 500

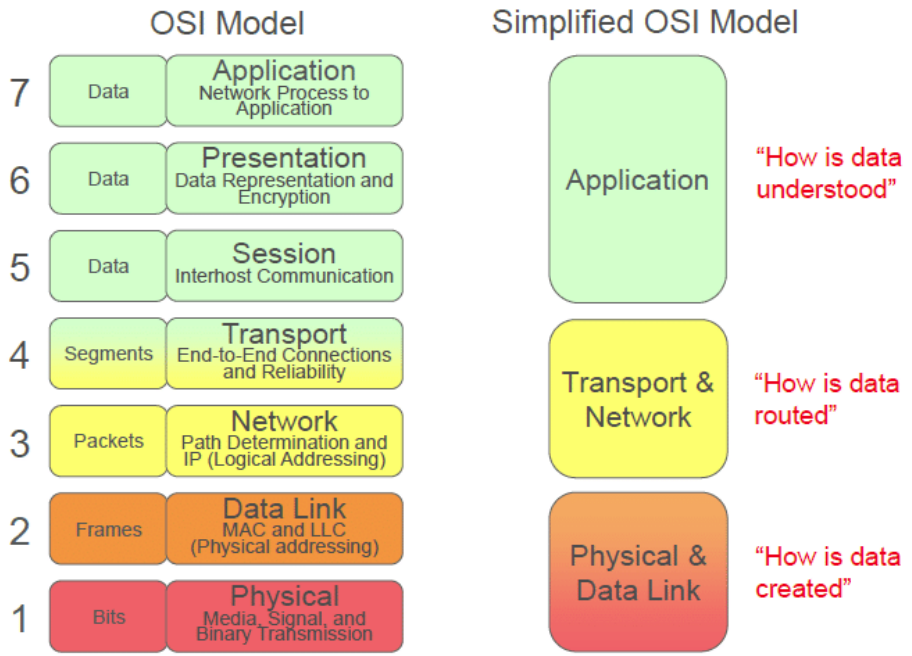
Networking Principles

- Network Protocols
- Packet Transmission
- Network Layers
  - Physical layer
  - Data-link layer
  - Network layer and routing
  - Transport layer and congestion control

Layered Protocols



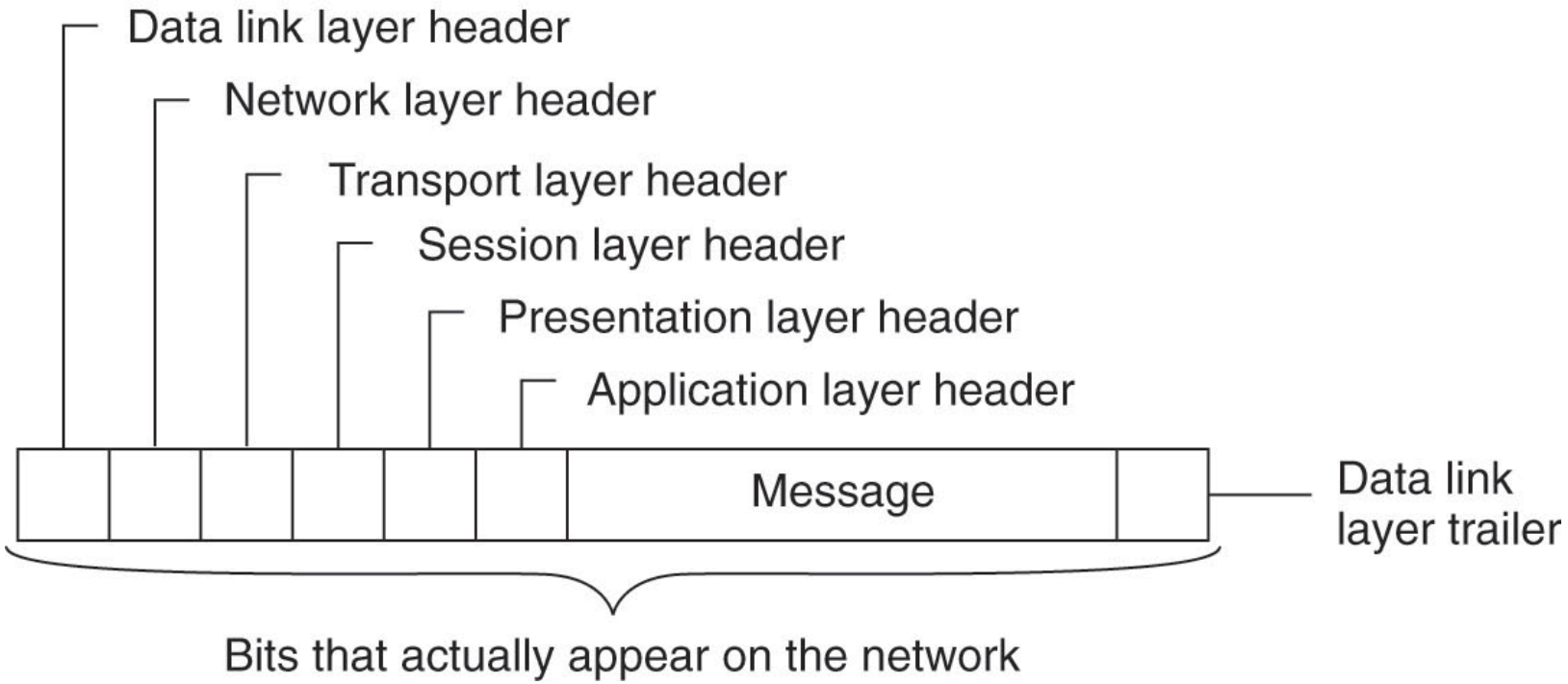
LAYERS



Networking Protocols

- If two entities want to communicate on a network, pre-defined agreements are necessary
  - How a message will be formatted?
  - How does the receiver know the last bit in the message?
  - How can a receiver detect if the message is damaged?
- "Protocol" is a well-known set of rules and formats to be used for communication between the entities
- Standardizing a well-known set of protocols supports communication among *heterogeneous* entities

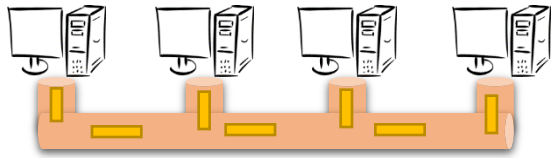
Networking Principles



A typical message as it appears on the network.

Packet Transmission

- Messages are broken up into packets
  - A packet is the unit of data that is transmitted between an origin and a destination
  - Packets can be of arbitrary lengths

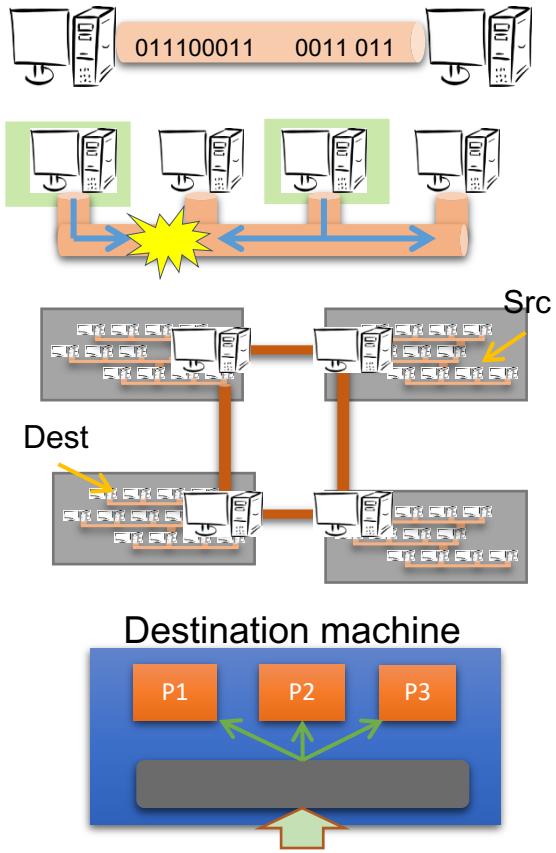


- Maximum size of the packet is known as Maximum Transmission Unit (MTU)
  - MTU prevents one host from sending a very long message
- Each packet has two main fields
  - Header: Contains meta-information about the packet
    - e.g., Length of the packet, receiver ID
  - Data

Network Layers

- Network software is arranged into a hierarchy of layers
  - Protocols in one layer perform one specific functionality
  - Layering is a scalable and modular design for complex software
- Typical functionalities in a network software:

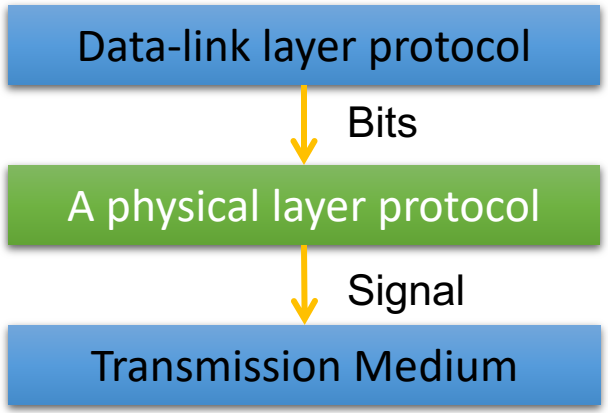
Functionality	Layer
contains the specification and implementation of bits, and their transmission between sender and receiver	Physical
prescribes the transmission of a series of bits into a frame to allow for error and flow control	Data link
describes how packets in a network of computers are to be routed.	Network
provides the actual communication facilities for most distributed systems.	Transport
Satisfies communication requirements for specific applications	Application



Physical Layer

- Physical layer protocols transmit a sequence of bits over a transmission medium
  - Modulate the bits into signals that can be transmitted over the medium

Transmission Medium	Type of signal transmitted
Twisted-pair (Ethernet cable)	Electrical signal
Fiber Optic Circuits	Light signal
Wireless channel	Electro-magnetic signal



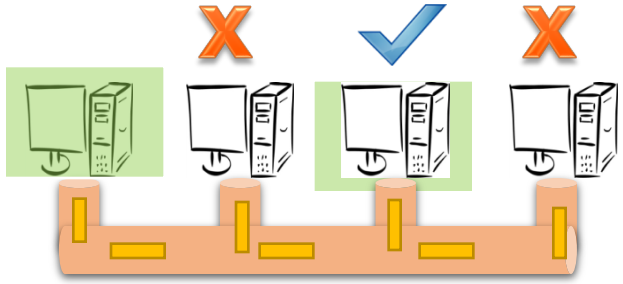
Data-link Layer

- Protocols in data-link layer ensure that the packets are delivered from one host to another within a local network
- Data-link layer protocols provide two main functionalities:
  - How to coordinate between the transmitters such that packets are successfully received?
    - Coordination
  - How to identify another host on the local network?
    - Addressing over local networks

Addressing over Local Networks

- Each device that is connected to a network has a unique address called Medium Access Control (MAC) address
  - MAC addresses are six bytes long
    - e.g., 2A:D4:AB:FD:EF:8D

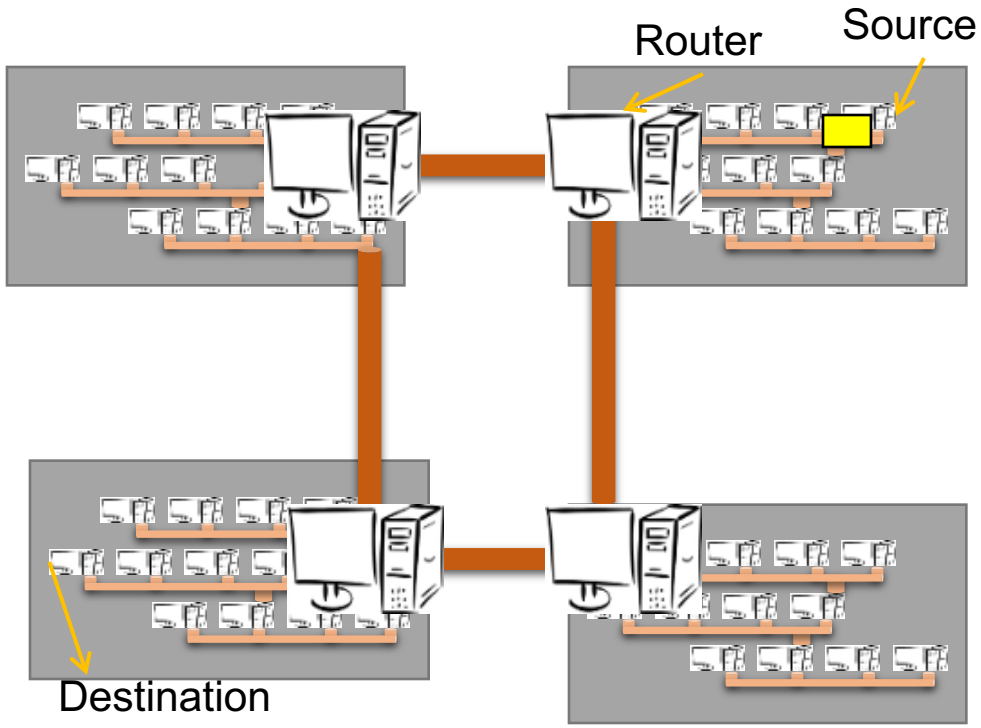
- Approach:
  - Data-link layer *broadcasts* the packet over the medium
  - Receiver reads the packet header and checks if the packet is addressed to it



Network Layer

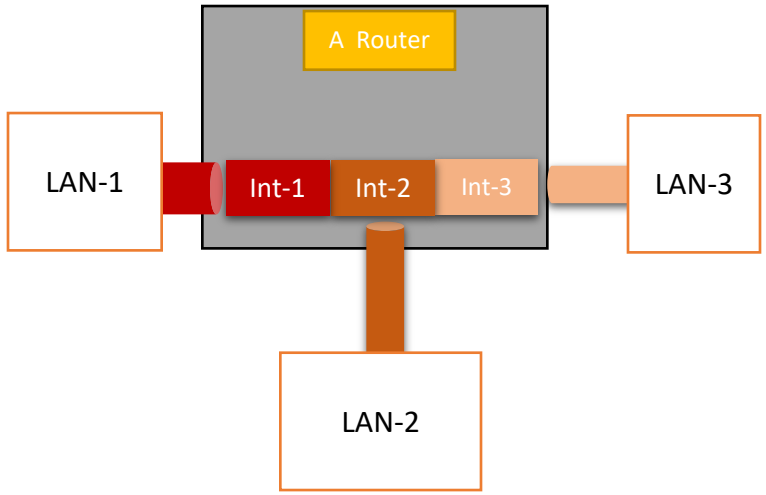
- Network layer protocols perform the role of routing
  - They ensure that a packet is routed from the source machine to the destination machine
  - Packets may traverse different LANs to reach the destination

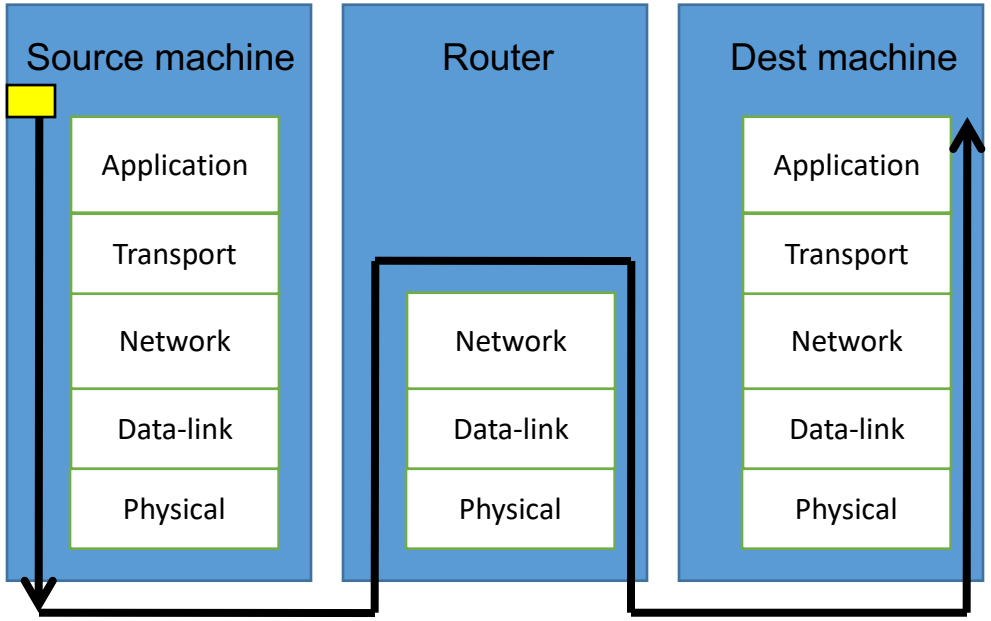
- Internet Protocol (IP) is a widely-used network layer protocol
  - IP addresses are typically used to identify machines



Router

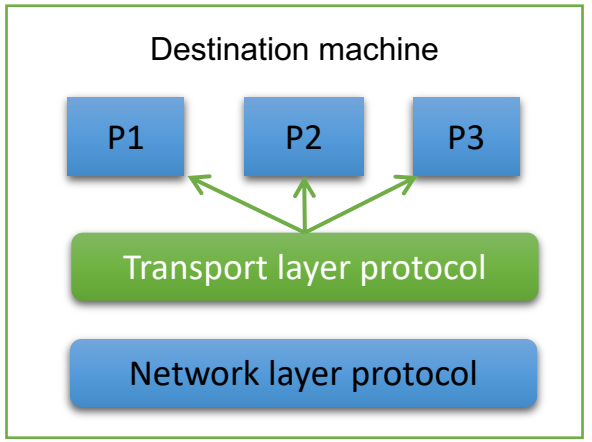
- A router is a device that forwards the packets between multiple networks
- Routers are connected to two or more networks
  - Each network *interface* is connected to a LAN or a host
- Packet travels up until the network layer on the router





Transport Layer

- The transport layer provides the actual communication facilities for most distributed systems.



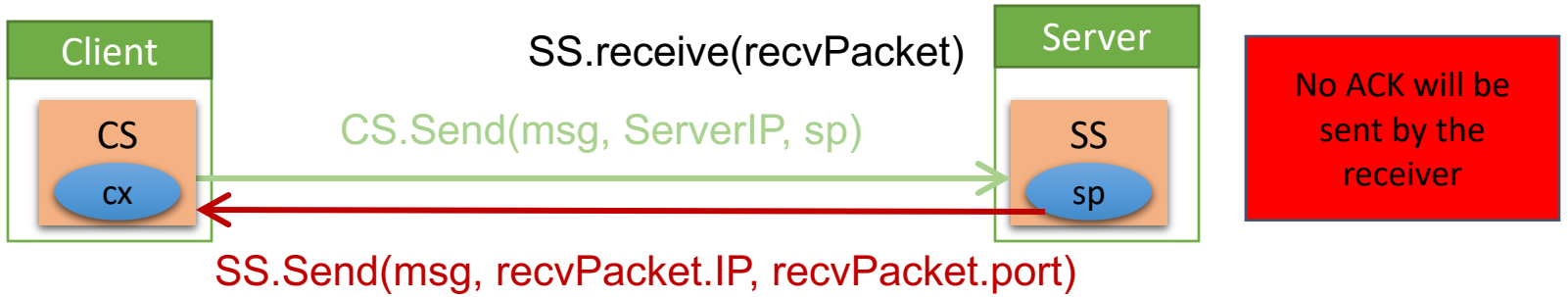
Standard Internet protocols

- TCP: connection-oriented, reliable, stream-oriented communication
- UDP: unreliable (best-effort) datagram communication

UDP Sockets

- User Datagram Protocol (UDP) provides *connectionless* communication, with no acknowledgements or message retransmissions

- Communication mechanism:
  - Server opens a UDP socket *SS* at a known port *sp*,
  - Socket *SS* waits to receive a request
  - Client opens a UDP socket *CS* at a random port *cx*
  - Client socket *CS* sends a message to ServerIP and port *sp*
  - Server socket SS may send back data to *CS*



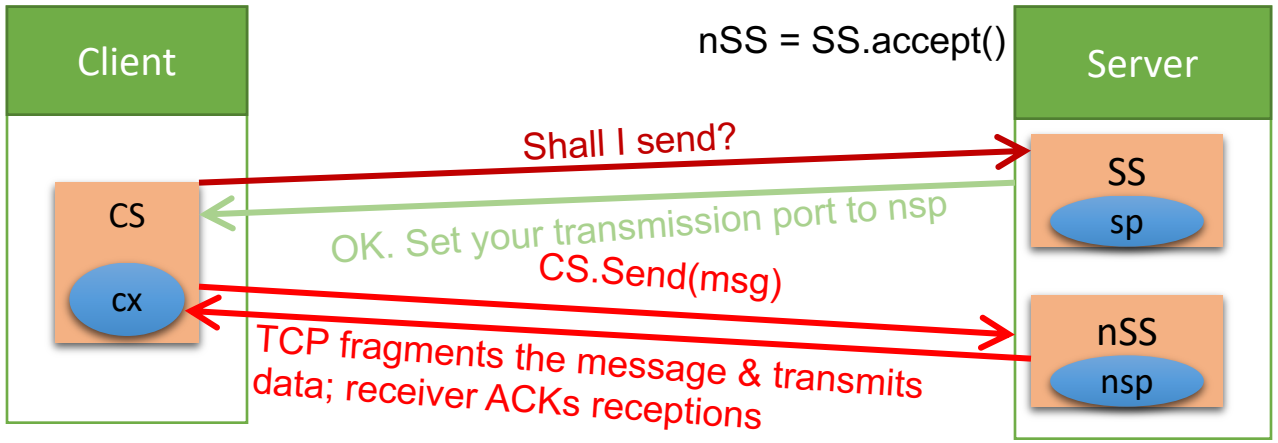
UDP– Design Considerations

- Sender must explicitly fragment a long message into smaller chunks before transmission
  - A maximum size of 548 bytes is suggested for transmission
- Messages may be delivered out-of-order
  - If necessary, programmer must re-order packets
- Communication is not reliable
  - Messages might be dropped due to check-sum errors or buffer overflows at routers
- Receiver should allocate a buffer that is big enough to fit the sender's message
  - Otherwise the message will be truncated

TCP Sockets

- Transmission Control Protocol (TCP) provides *in-order* delivery, *reliability,* and *congestion control*

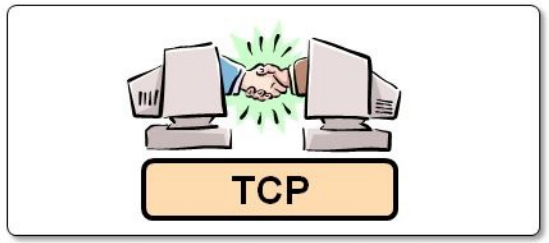
- Communication mechanism:
  - Server opens a TCP server socket *SS* at a known port *sp*
  - Server waits to receive a request (using *accept* call)
  - Client opens a TCP socket *CS* at a random port *cx*
  - *CS* initiates a connection initiation message to ServerIP and port *sp*
  - Server socket SS allocates a new socket NSS on random port *nsp* for the client
  - *CS* can send data to *NSS*



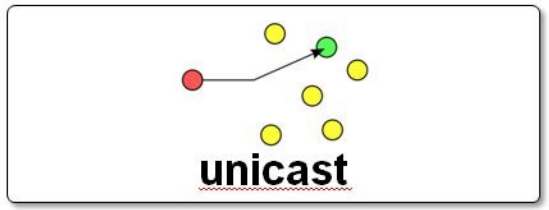
Main Advantages of TCP

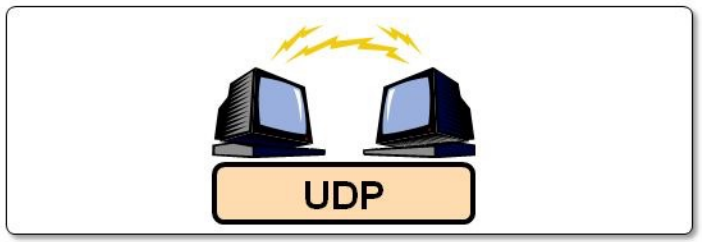
- TCP ensures in-order delivery of messages
- Applications can send messages of any size
- TCP ensures *reliable communication* via using acknowledgements and retransmissions
- Congestion control of TCP regulates sender rate, and thus prevents network overload

TCP vs UDP

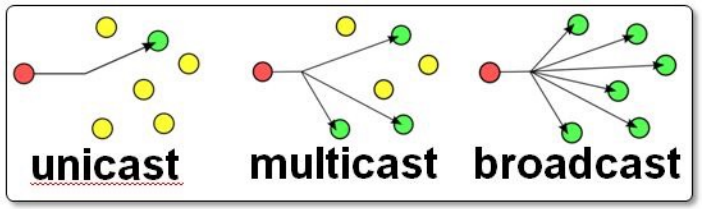


- **Slower but reliable transfers**
- **Typical applications:**
  - Email
  - Web browsing





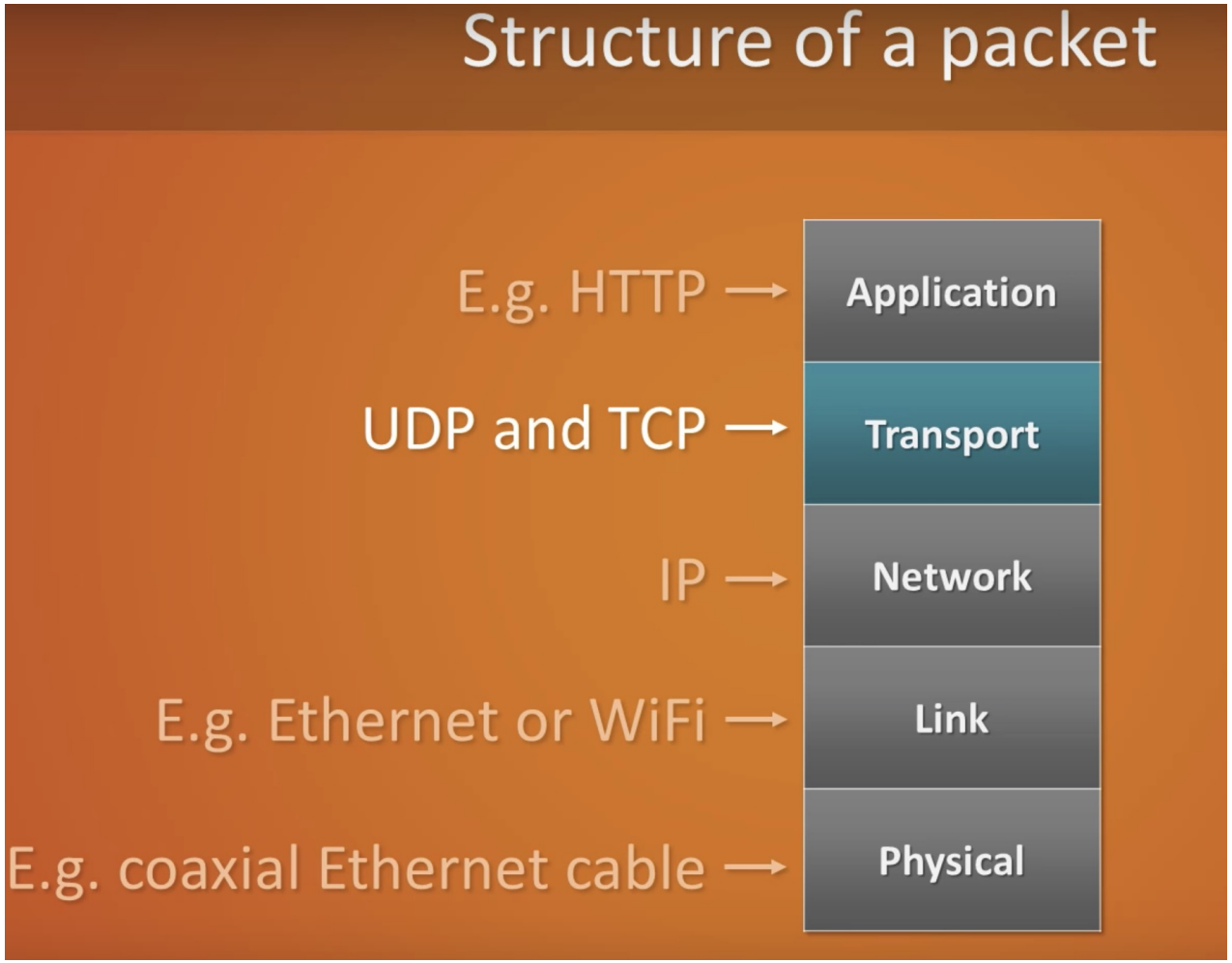
- **Fast but non-guaranteed transfers ("best effort")**
- **Typical applications:**
  - VoIP
  - Music streaming



TCP vs UDP

TRANSMISSION CONTROL PROTOCOL (TCP)	USER DATAGRAM PROTOCOL (UDP)
a connection-oriented protocol. Connection-orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.	a datagram oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, and terminating a connection. UDP is efficient for broadcast and multicast type of network transmission.
TCP is reliable as it guarantees delivery of data to the destination router.	The delivery of data to the destination cannot be guaranteed in UDP.
TCP provides extensive error checking mechanisms. It is because it provides flow control and acknowledgment of data.	UDP has only the basic error checking mechanism using checksums.
Sequencing of data is a feature of Transmission Control Protocol (TCP). this means that packets arrive in-order at the receiver.	There is no sequencing of data in UDP. If ordering is required, it has to be managed by the application layer.
TCP is comparatively slower than UDP.	UDP is faster, simpler and more efficient than TCP.
Retransmission of lost packets is possible in TCP, but not in UDP.	There is no retransmission of lost packets in User Datagram Protocol (UDP).
TCP has a (20-80) bytes variable length header.	UDP has a 8 bytes fixed length header.
TCP is heavy-weight.	UDP is lightweight.
TCP doesn't supports Broadcasting.	UDP supports Broadcasting.
TCP is used by **HTTP**, **HTTPs**, FTP, SMTP and Telnet.	UDP is used by DNS, DHCP, TFTP, SNMP, RIP, and **VoIP**.

Layered Protocols with Examples



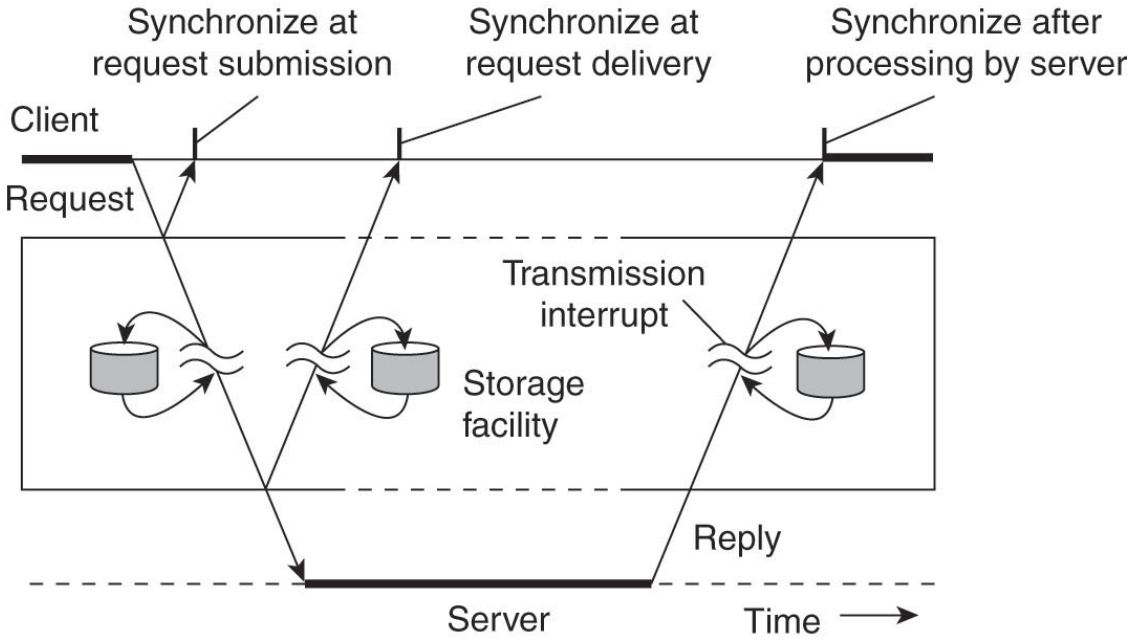
Types of communication

Transient communication: Comm. server discards message when it cannot be delivered at the next server, or at the receiver.

Persistent communication: A message is stored at a communication server as long as it takes to deliver it.

Places for synchronization

- At request submission
- At request delivery
- After request processing



Viewing middleware as an intermediate (distributed) service in application-level communication.

Transient communication

Some observations

Client/Server computing is generally based on a model of transient synchronous communication:

- Client and server have to be active at time of communication
- Client issues request and blocks until it receives reply
- Server essentially waits only for incoming requests, and subsequently processes them

Drawbacks synchronous communication

- Client cannot do any other work while waiting for reply
- Failures have to be handled immediately: the client is waiting
- The model may simply not be appropriate (mail, news)

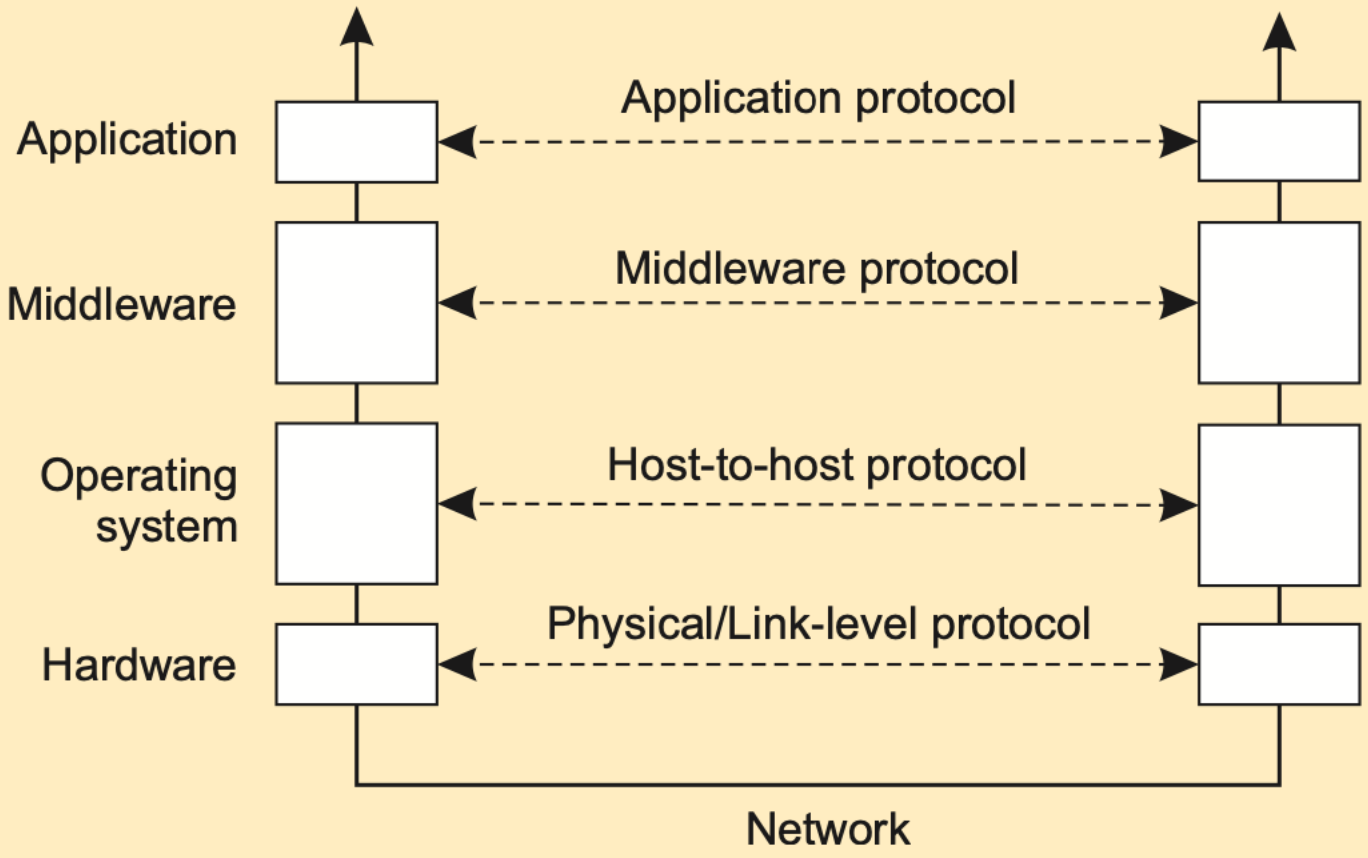
Persistent communication

Message-oriented middleware

Aims at high-level persistent asynchronous communication:

- Processes send each other messages, which are queued
- Sender need not to wait for immediate reply, but can do other things
- Middleware often ensures fault tolerance

An adapted layering scheme



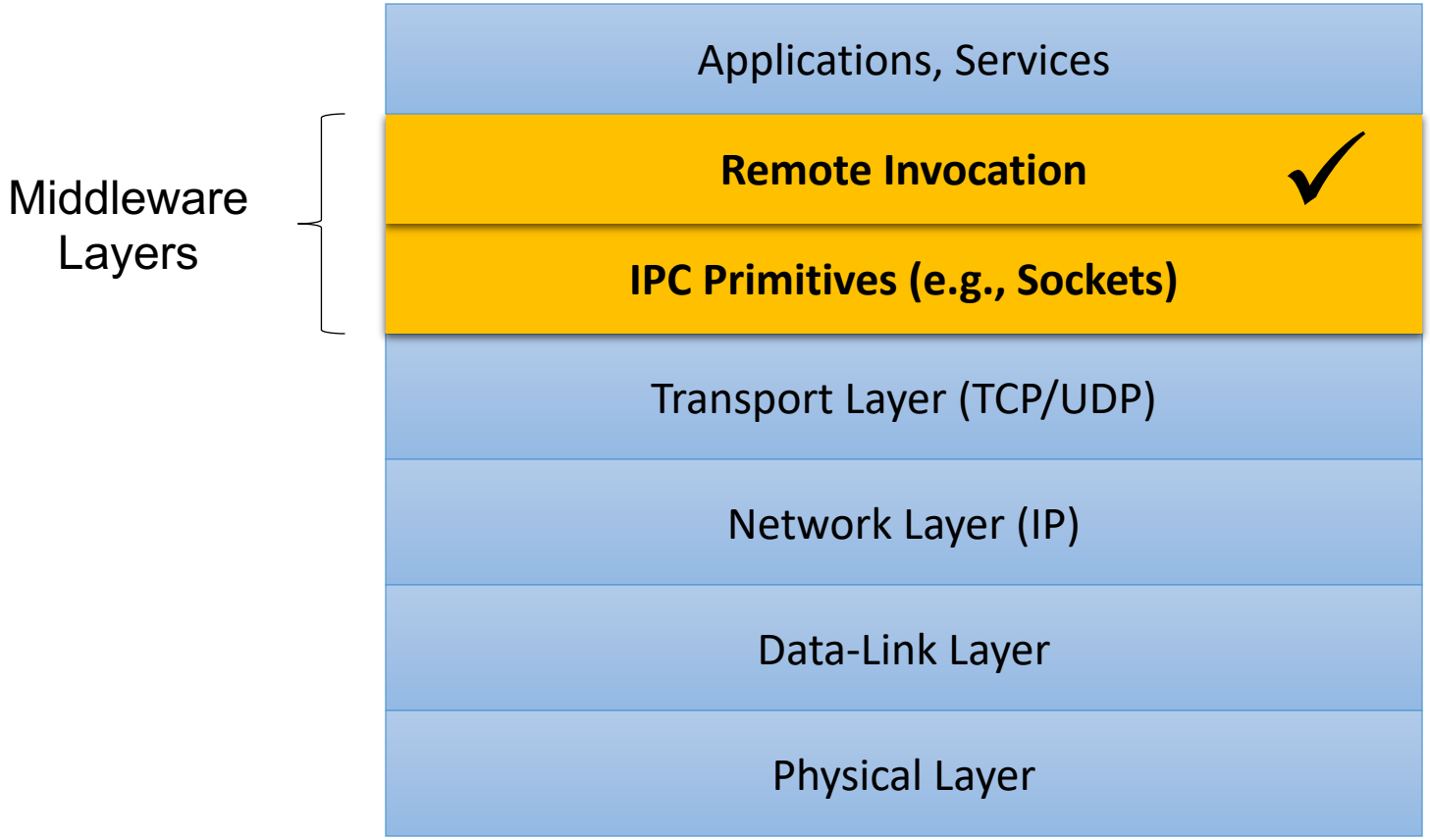
Middleware Layers

Observation

Middleware is invented to provide common services and protocols that can be used by many different applications:

- A rich set of communication protocols
- (Un)marshaling of data, necessary for integrated systems
- Naming protocols, to allow easy sharing of resources
- Security protocols for secure communication
- Scaling mechanisms, such as for replication and caching

Middleware Layers



A To-Do List

- Read Chapters 4.1
- Prepare for the mid-term exam on Oct 23