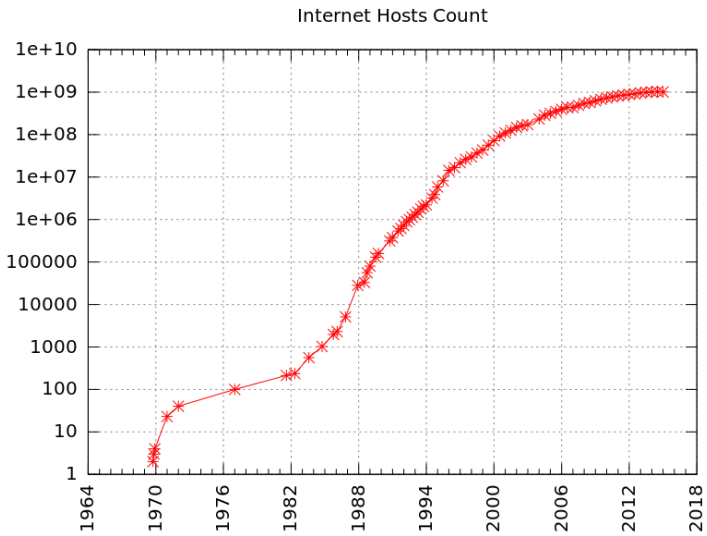
CS 428/528 Computer Networks – Lecture 01

Yan Wang

Motivation: Why bother?

- Explosive growth of networks
  - 1989, 100,000 hosts on the Internet
- Distributed Applications and Systems
  - E-mail, WWW, multimedia, distributed file systems and databases

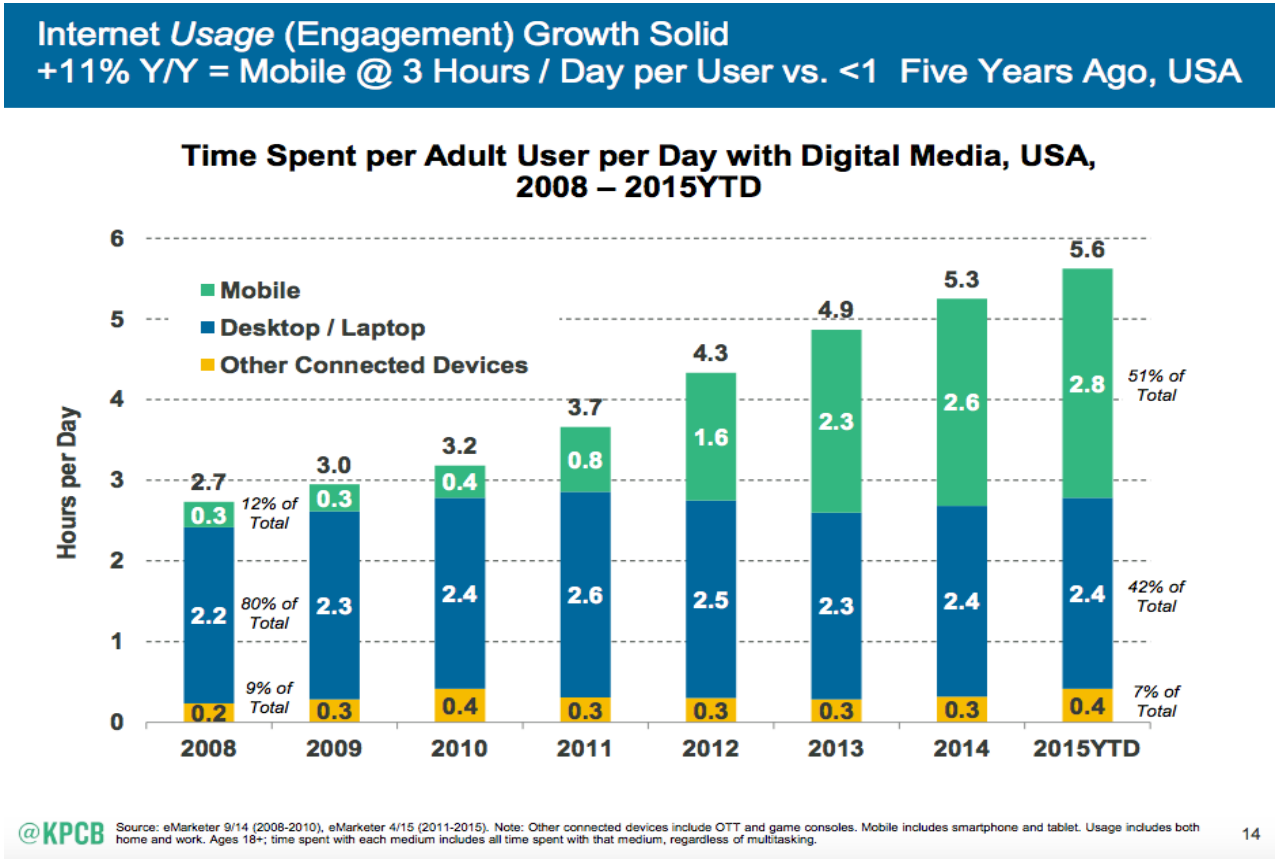


Source: Internet Systems Consortium.[9]

- Wireless/Mobile and Cellular telecommunications
  - Used for voice, data, multimedia
  - Ad hoc and sensor systems
- *Most computer systems of interest are now networked*

Pervasive Data Communications

- How much of your time is spent on the Internet every day?



Position of Computer Networks

Distributed Systems

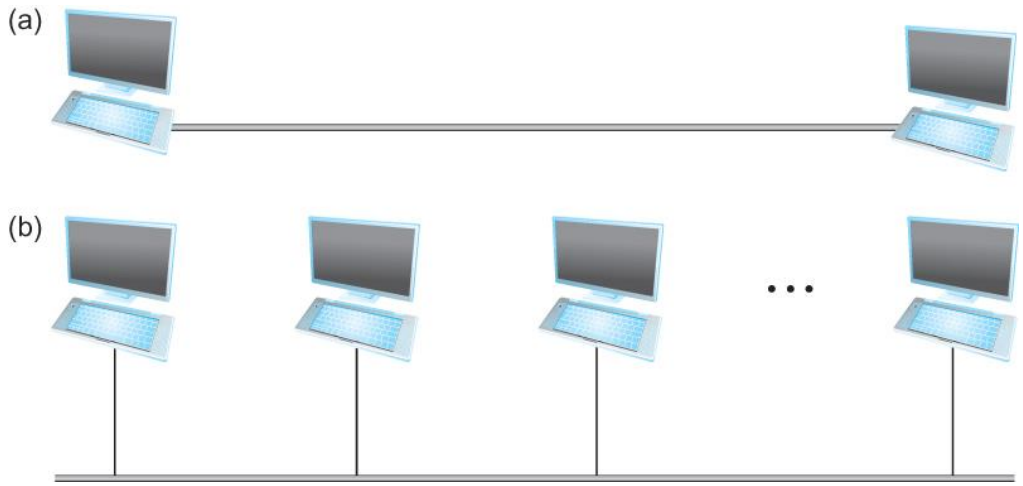
Networks

Communications

Building Blocks

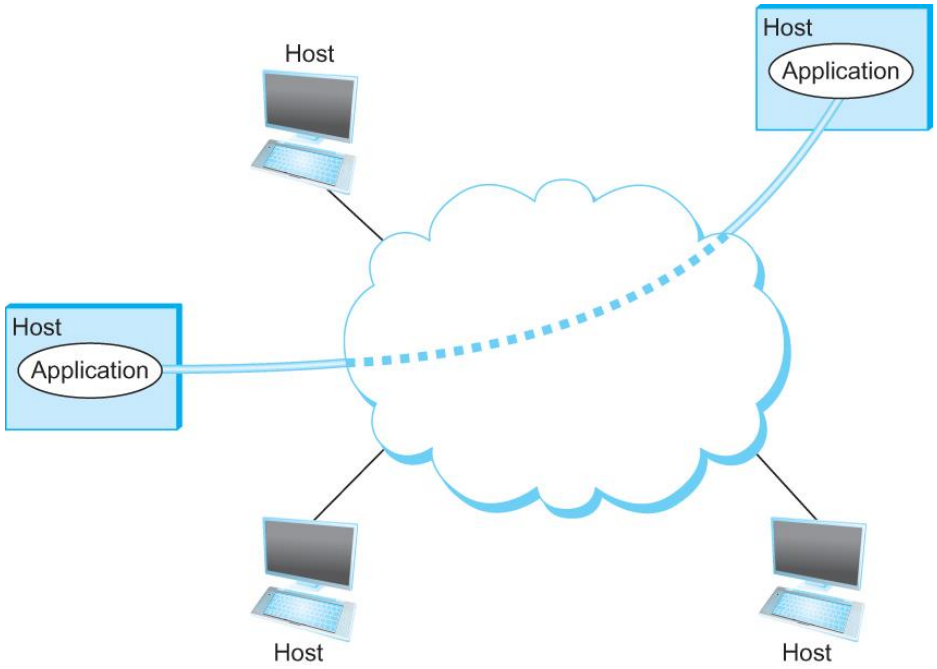
- Nodes: PC, special purpose hardware, …
  - Hosts
  - Switches

- Links: coax cable, optical fiber
  - Point-to-point
  - Multiple access



Design Issues

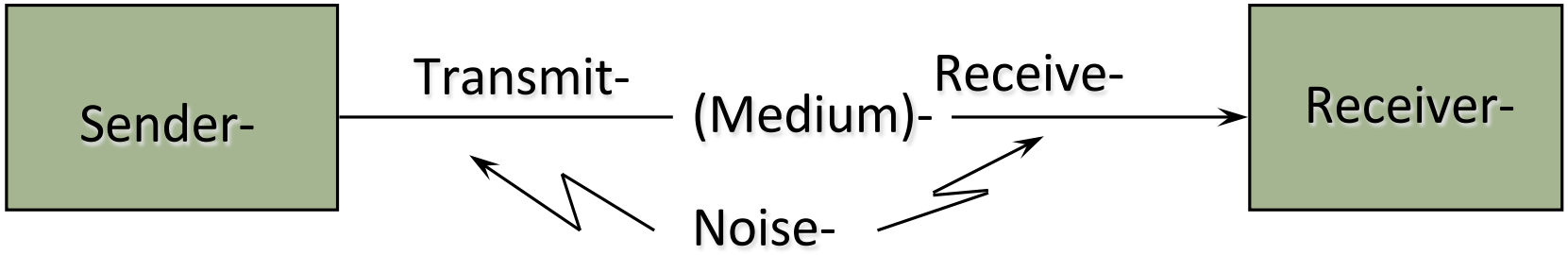
- Connectivity
- Resource sharing
- Reliability, make system design “simpler”



How do we study networks?

- Best way: build it
- Algorithmically
- Analytically
- By argument
- Through simulation

Model of Communication



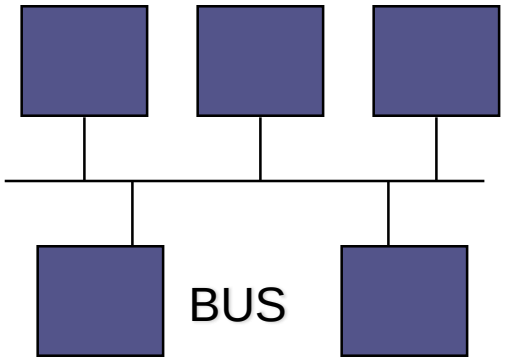
- Communications: transferring information over a distance
- Requires a shared symbol set that can be transmitted as a "**signal**" through a medium, received, and understood
- Encoding: the physical forms of signals
- Medium: anything capable of passing information: copper wire, glass fiber, radio waves, and so on
- Some noise always exists to corrupt the signal

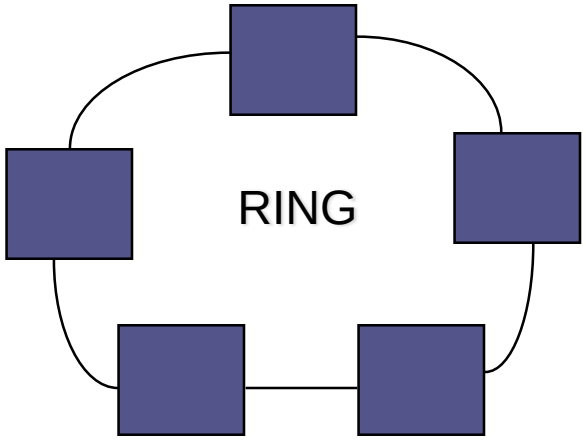
Network (Hardware) Connections

- Two types of communications connections:
  - **Broadcast networks** have a single communication channel that is shared by all machines on the network.
  - **Point-to-point networks** consist of many connections between individual machines.
- In general, small local networks use broadcasting, larger networks use point-to-point
  - There are many exceptions

Local Area Network

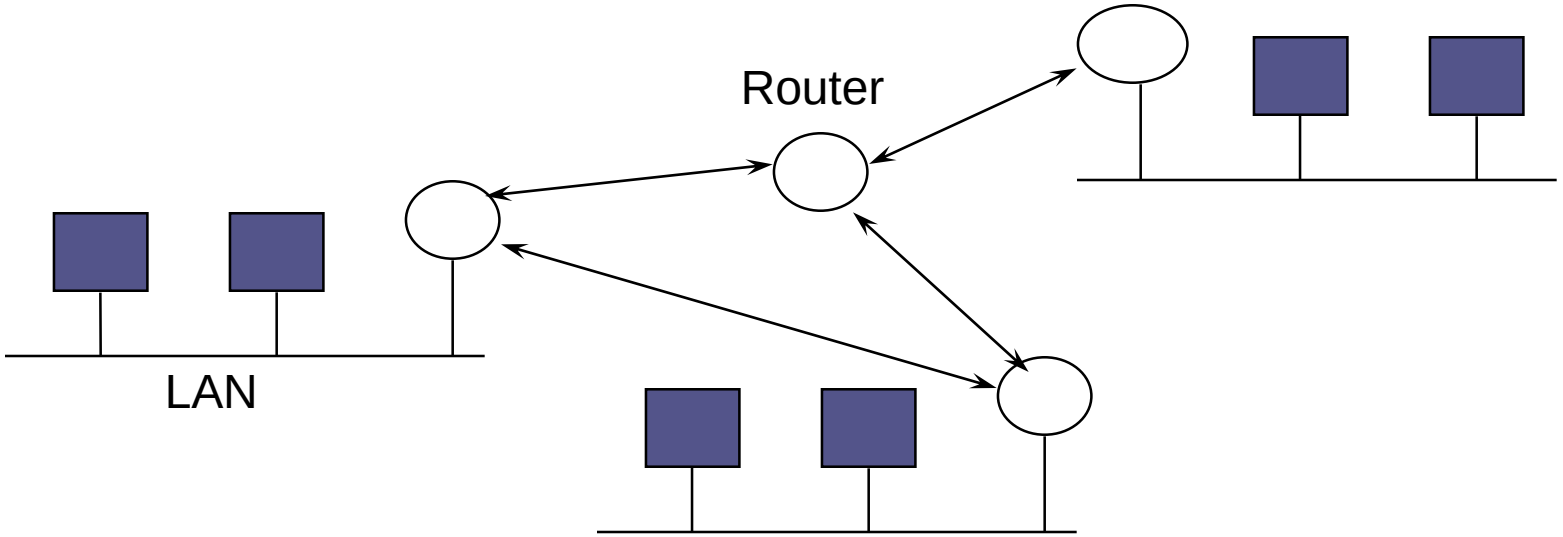
- Typically a single logical wire shared by multiple hosts
  - A popular arbitration protocol is Ethernet and IEEE 802.11
- Different topologies possible



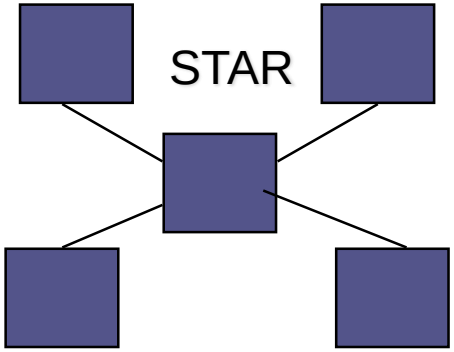


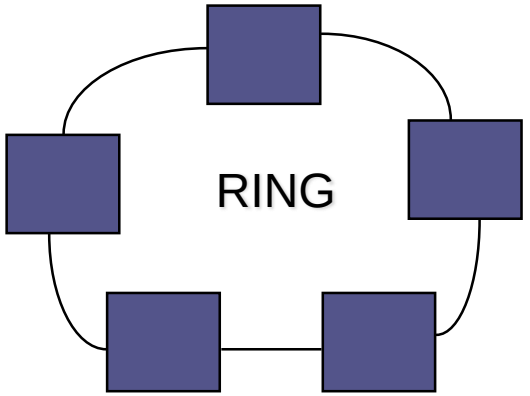
Wide Area Network

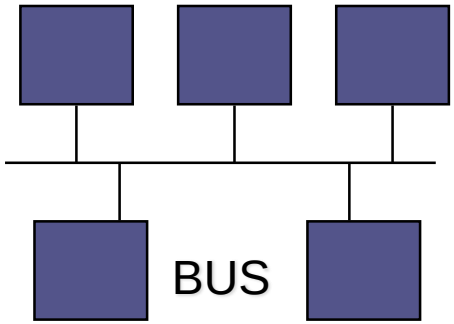
- Typically point to point links connecting specialized hardware (routers, switches)

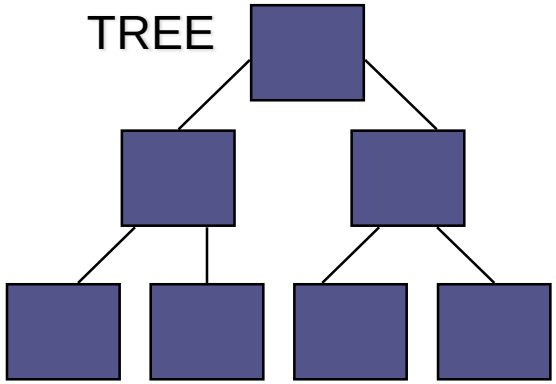


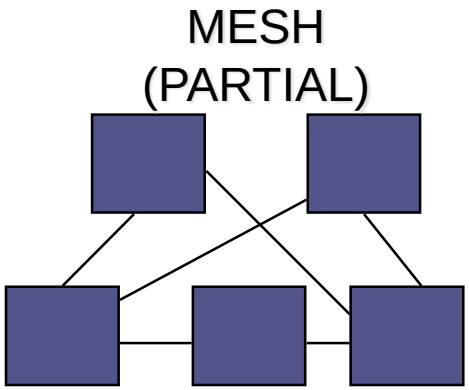
General Network Topologies





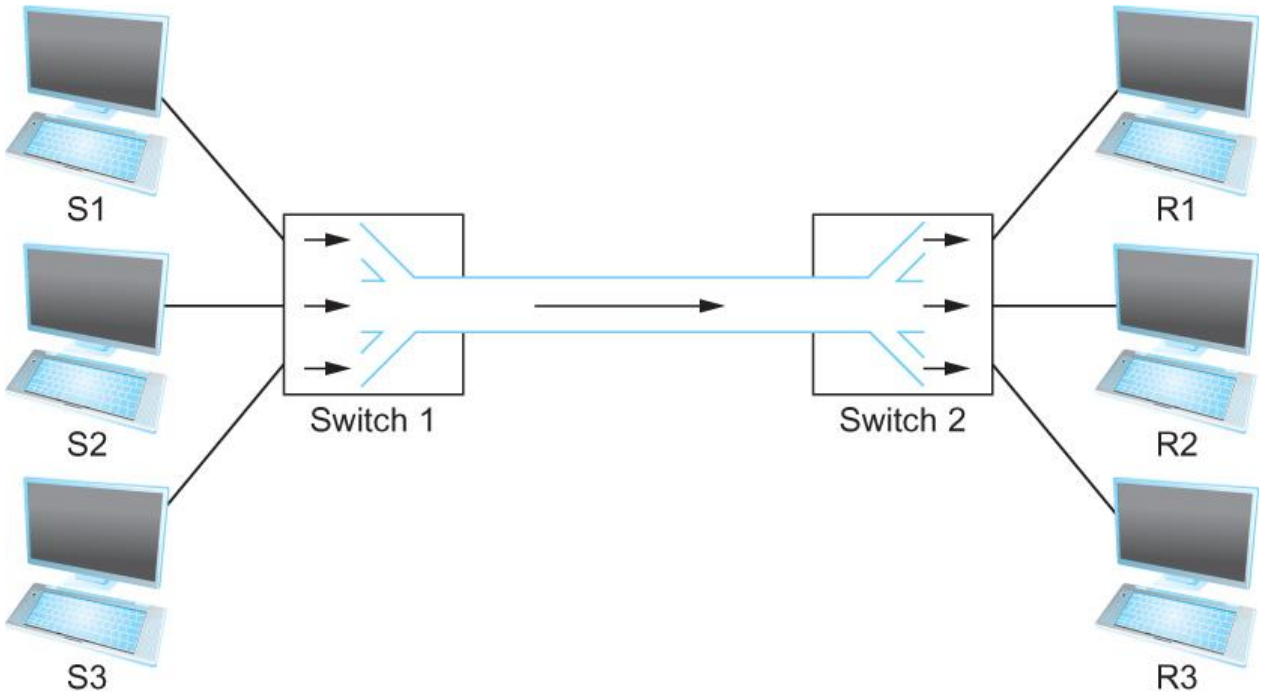






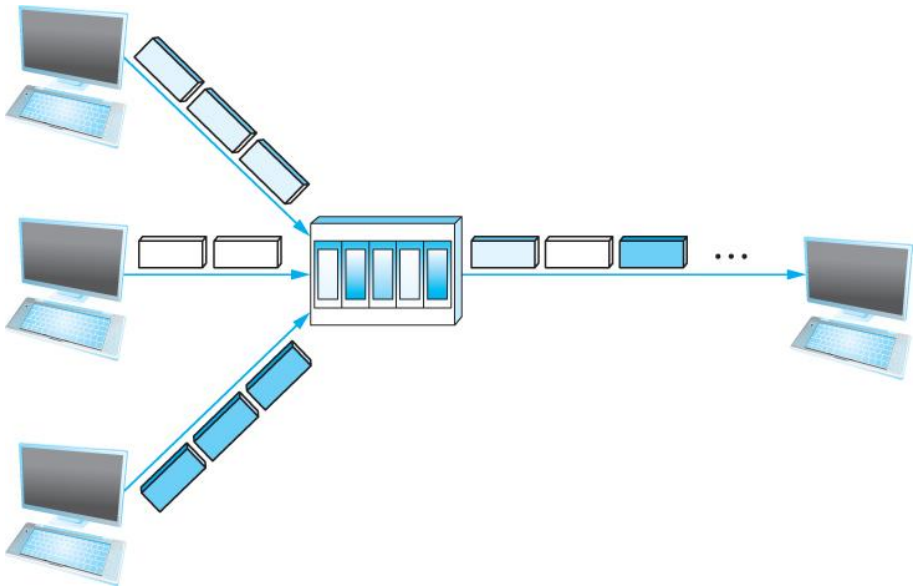
Multiplexing

- Time-Division Multiplexing (TDM)
- Frequency-Division Multiplexing (FDM)



Statistical Multiplexing

- On-demand time-division
- Schedule link on a per-*packet* basis
- Packets from different sources interleaved on link
- Buffer packets that are *contending* for the link
- Buffer (queue) overflow is called *congestion*



Network Software

- Networks are organized as a series of layers, in order to reduce complexity
  - Each layer builds upon the one below it and offers services to the one above it.
  - Between each layer is an interface
- A list of protocols is called a **network stack**
- A set of layers and protocols is called a Network Architecture
  - **Understanding a general purpose network architecture is the focus of the course**

Layer Design Issues

- Identify senders and receivers
- Error control
- Message ordering
- Control message rates
- Connection-oriented versus connectionless
- Rules for data transfer

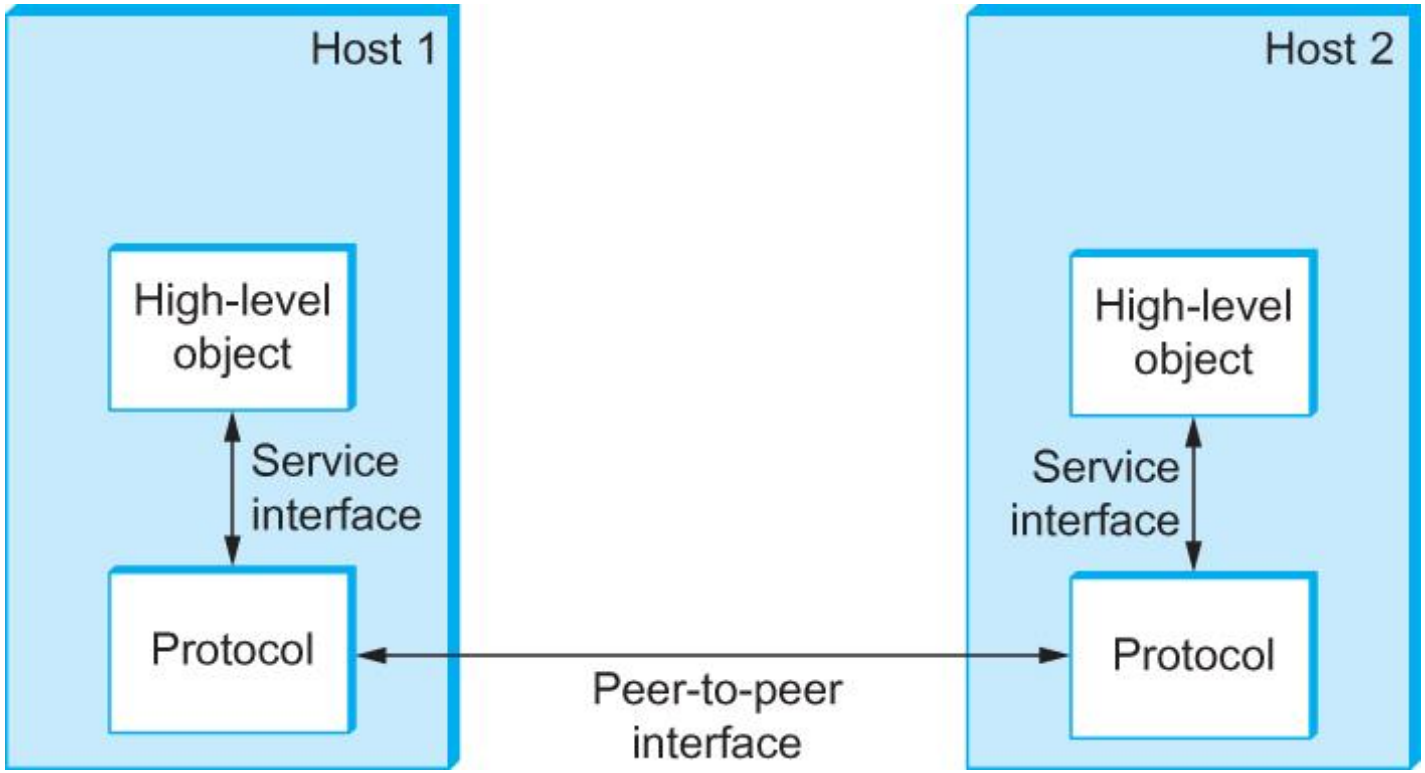
Network Services

- A **service** is a formally specified set of primitives
  - Examples are REQUEST, INDICATION, RESPONSE, CONFIRM
  - Purpose is to provide a way of invoking operations to the layer above it
- Different from a **protocol**
  - A protocol is a set of rules governing the format and meaning of the frames, packets or messages exchanged by the peer entities or layers

Protocols/Standards

- Building blocks of a network architecture
  - Typical protocols: FTP, HTTP, BGP, DHCP, POP, SMTP, TCP, UDP, IP, IPsec, ARP, MAC, PPP
- Each protocol object has two different interfaces
  - ***Service interface***: operations on this protocol
  - ***Peer-to-peer interface***: messages exchanged with peer
- Term "protocol" is overloaded
  - Specification of peer-to-peer interface
  - Module that implements this interface

Interfaces

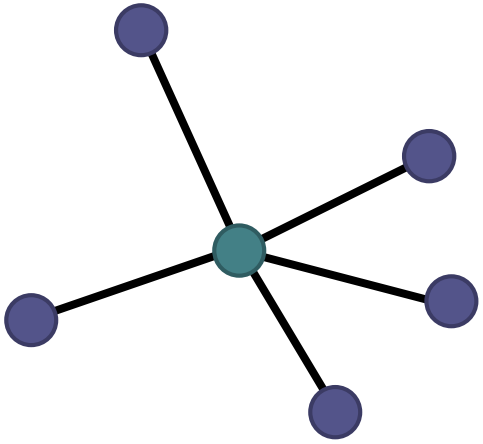

Host 1
High-level object
Service interface
Protocol
Peer-to-peer interface
Host 2
High-level object
Service interface
Protocol

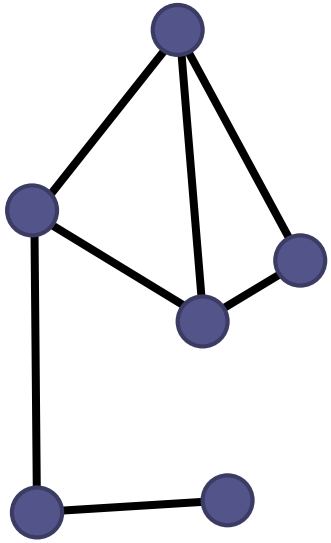
Protocols/Standards

- Reasons for networking standards
  - Allow different computers to communicate
  - Decrease price through economies of scale
- Disadvantages of standards
  - Tend to freeze technology: by the time the standard is developed, reviewed, agreed upon, and distributed, better technologies become available.
  - Often multiple, conflicting standards for the same thing.

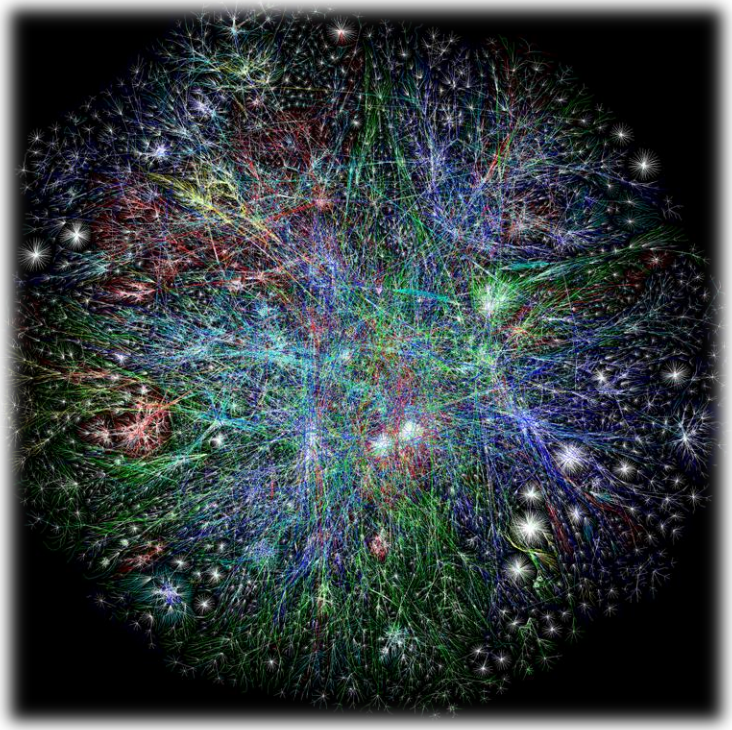
Discussion 1

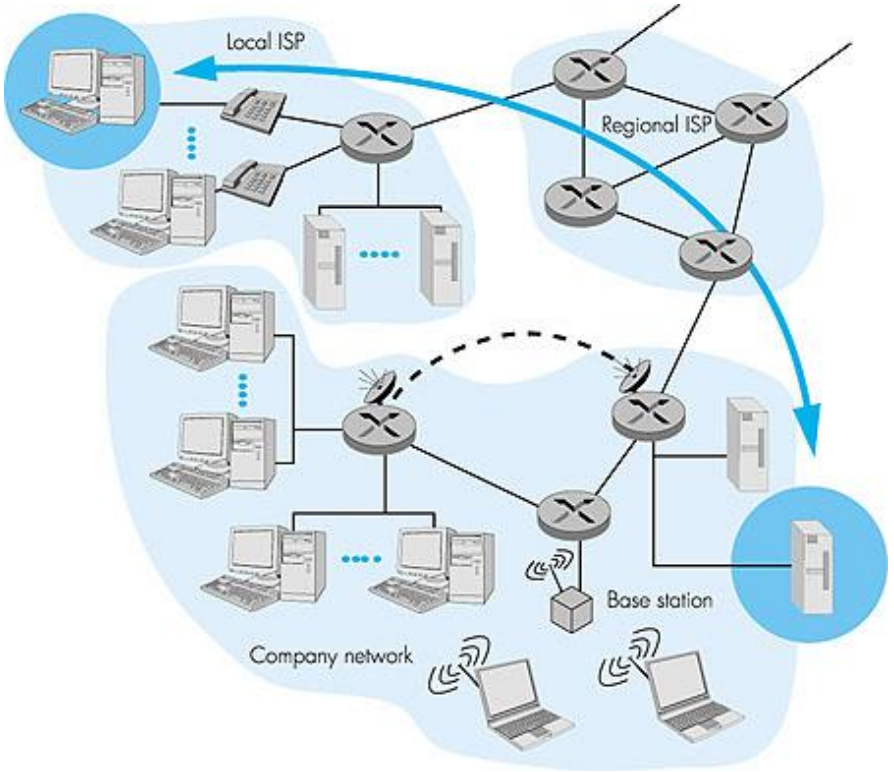
- What does the topology of the INTERNET look like ??





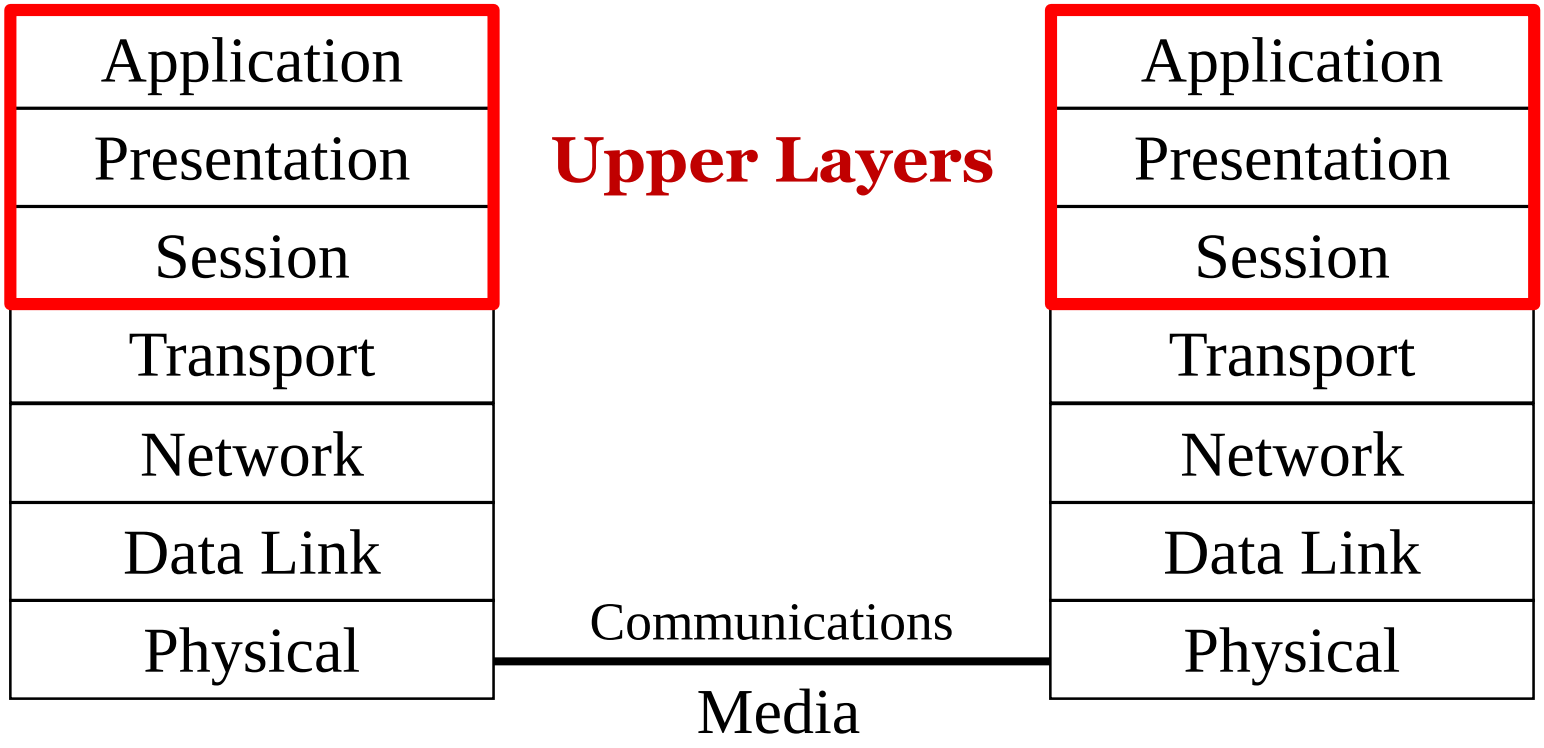
The Topology of the INTERNET





OSI Reference Model

- International Organization for Standardization (ISO) has defined a seven layer model: **Open Systems Interconnection** (**OSI**) reference model.



Physical and Data Link layers

- **Physical Layer:** Transmit and receive bits on physical media
  - Analog and digital transmission
  - A definition of the 0 and 1 bits
  - Bit rates (bandwidth)
- **Data Link Layer:** Provide error-free bit streams across physical media
  - Error detection/correction:
  - Reliability
  - Flow control

Network Layer

Controls the operations of the network

- **Routing:** determining the path from the source of a message to its destination
- **Congestion Control:** handling traffic jams
- **Internetworking** of both homogeneous and heterogeneous networks
  - Homogeneous networks: similar configuration and protocols
  - Heterogeneous networks: different …

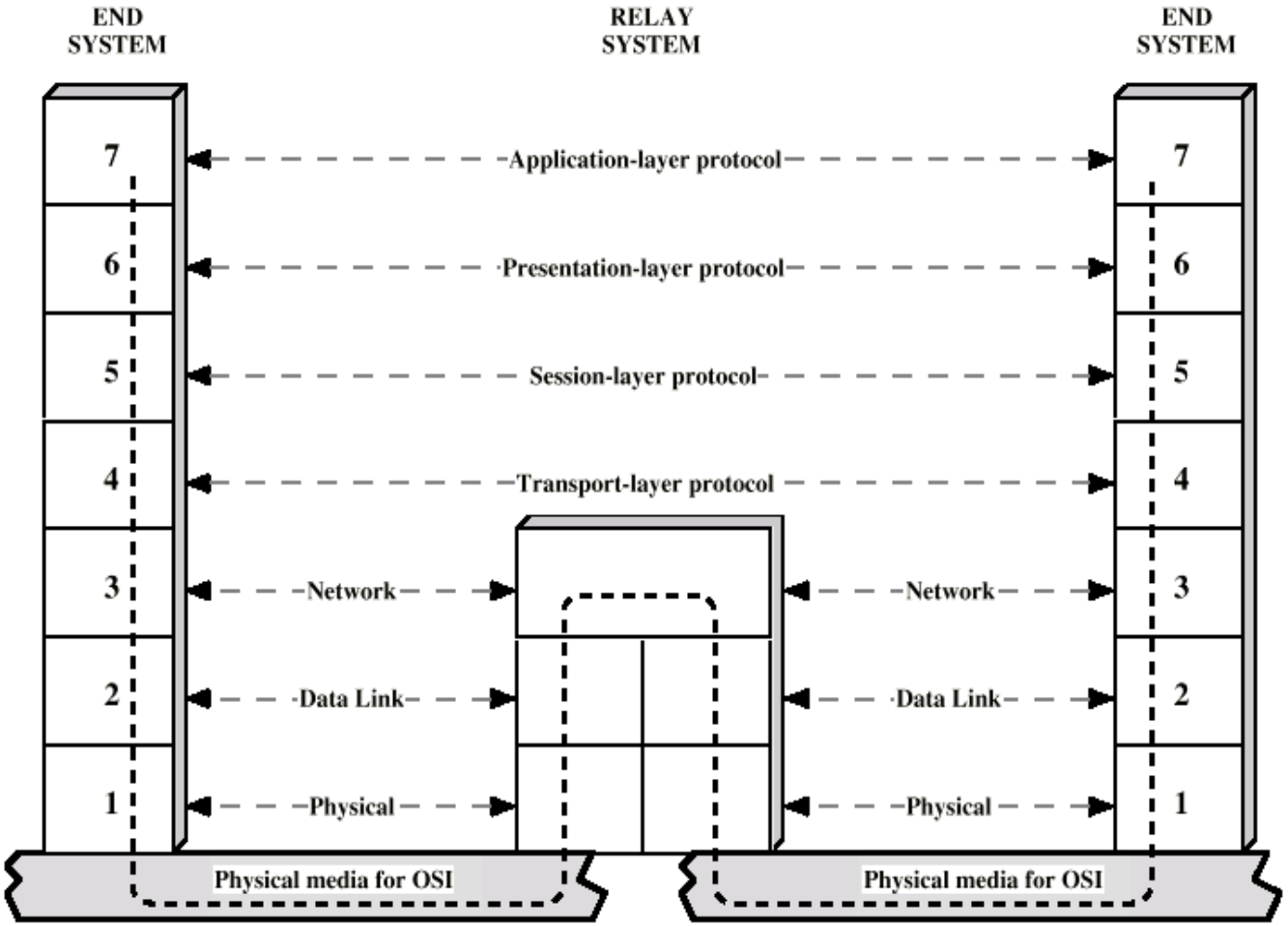
Transport Layer

- Provides **end–to–end (host–to–host)** connections
- **Packetization**: cut the messages into smaller chunks (packets)
- An ensuing issue is **ordering**: the receiving end must make sure that the user receives the packets in the right order
- Host–to–host **flow control**

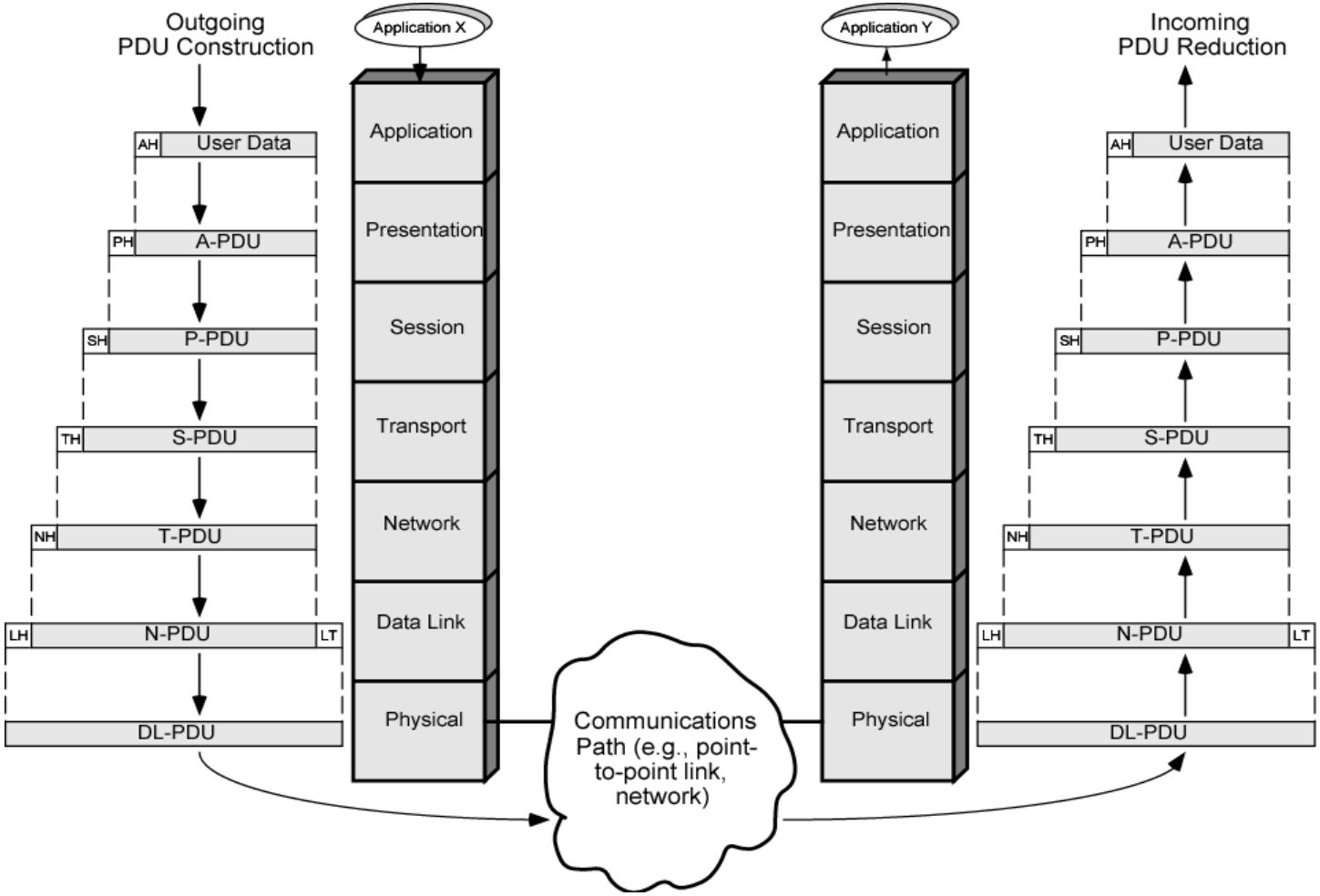
Upper Layers

- **Session Layer**
  - User–to–user connection
  - Synchronization, checkpoint, and error recovery
- **Presentation Layer**
  - Data representation/compression
  - Cryptography and authentication
- **Application Layer**
  - File transfer, email, WWW, and so on

Data Communication based on OSI



Data Encapsulation in OSI

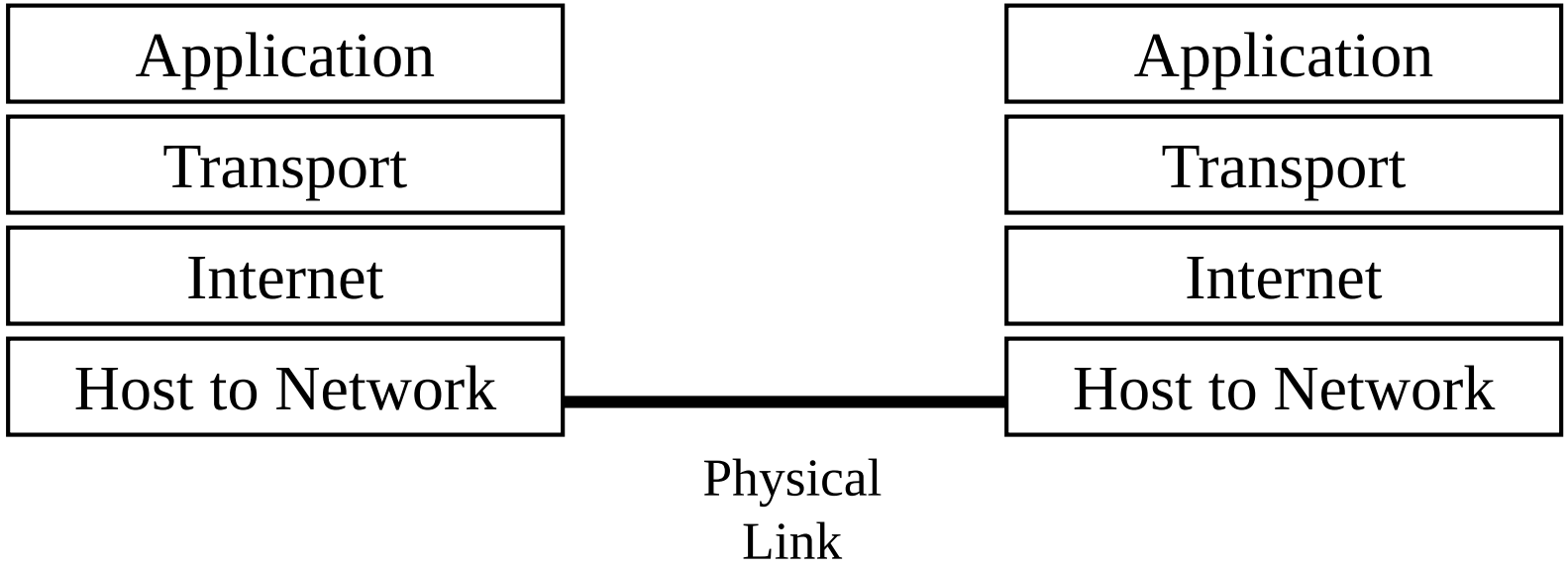


Shortcomings of the OSI Model

Just because someone says it is a model/standard does not mean you have to follow it

- Layers do not have the same size and importance
  - Session and presentation layers seldom present
  - Data link, network, and transport layers often very full
- Little agreement on where to place various features
  - Encryption, network management
- Large number of layers increases overheads

Internet Protocol Suite Reference Model

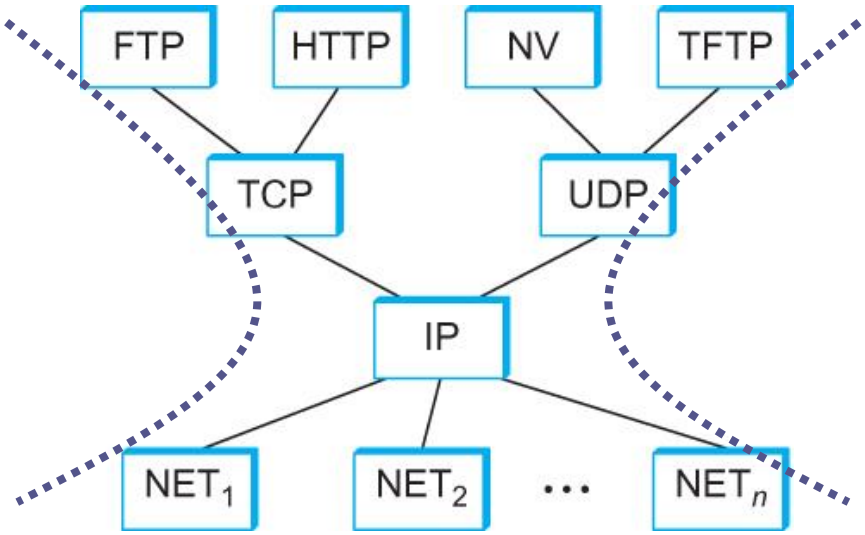


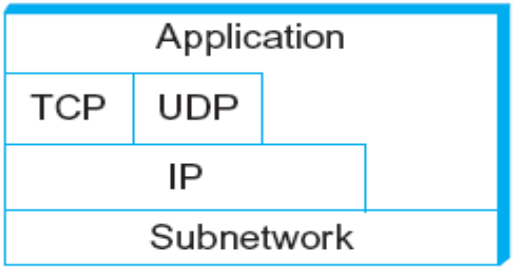
Internet Protocol Suite Reference Model

- Also called **TCP/IP** Reference Model
- There is **NO** *presentation* and *session* layers in the Internet model
- Internet layer = network layer (OSI model)
- The physical and data link layers in the OSI model are merged to the “Host to Network” layer

Internet Architecture

- Defined by Internet Engineering Task Force (IETF)
- Hourglass Design
- Application vs. Application Protocol (FTP, HTTP)





Application

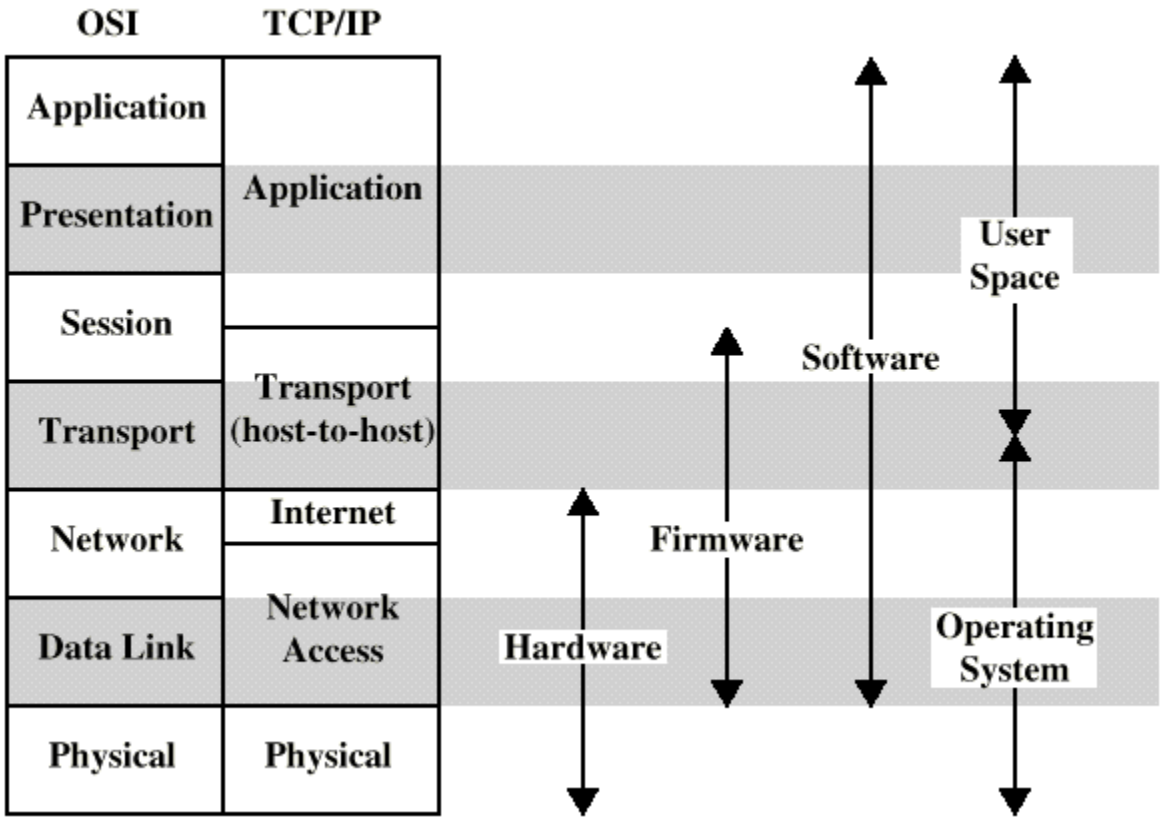
Transport

Internet

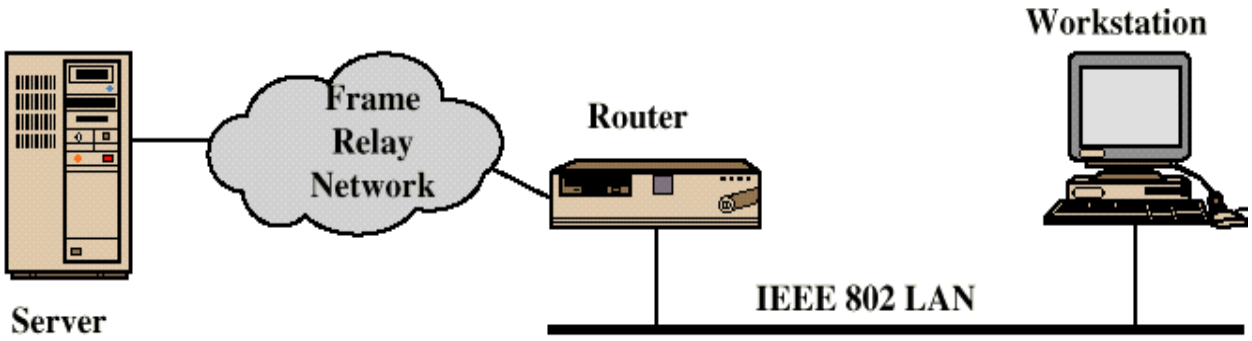
Host to Network

Does not imply strict layering. The application is free to bypass the defined transport layers and to directly use IP or other underlying networks

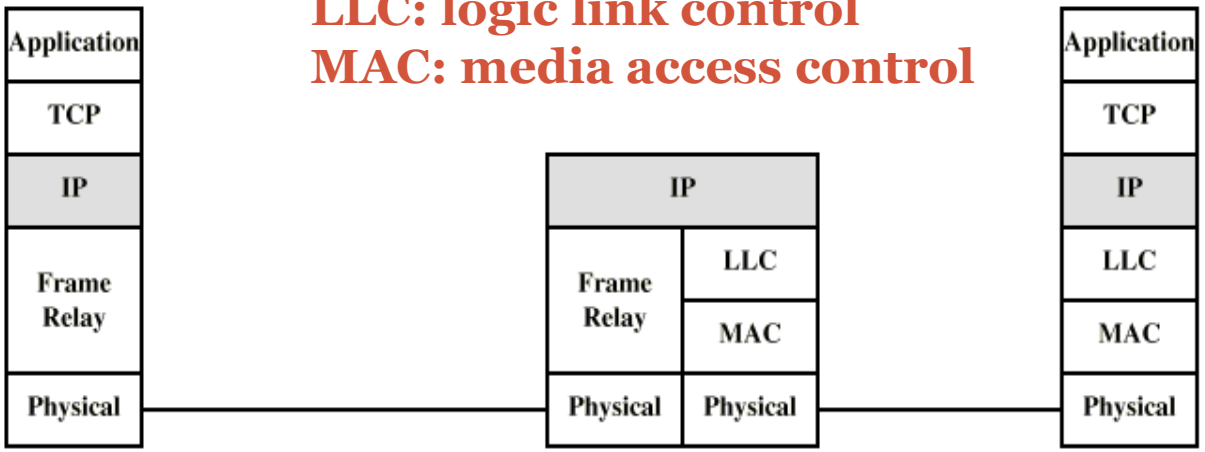
OSI/TCP-IP Comparison



Another View

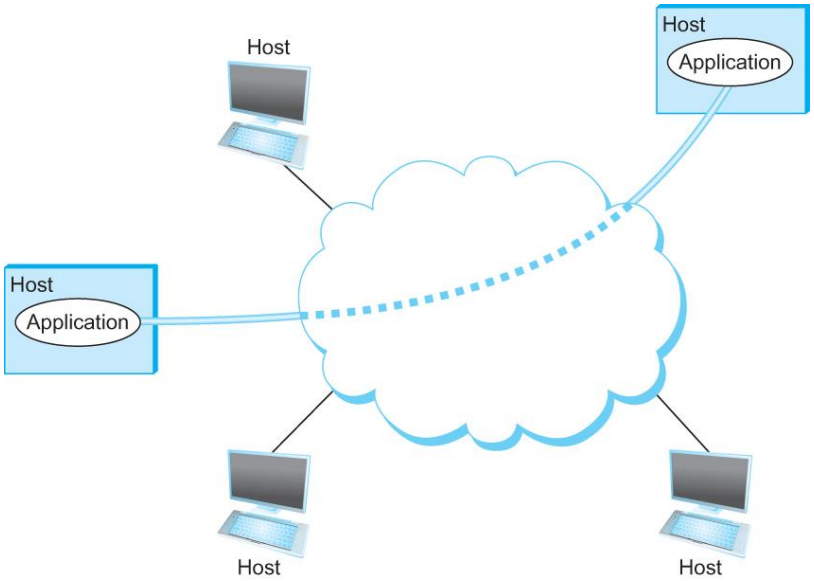


LLC: logic link control
MAC: media access control



Inter-Process Communication

- Turn host-to-host connectivity into **process-to-process** communication
- Fills the gap between what applications expect and what the underlying technology provides



Socket

- What is a socket?
  - The point where a local application process attaches to the network
  - An interface between an application and the network
  - An application creates the socket
- The interface defines operations for
  - Creating a socket
  - Attaching a socket to the network
  - Sending and receiving messages through the socket
  - Closing the socket

Socket API

```
int sockfd = socket(address_family, type, protocol);
```

- The socket number returned is the socket descriptor for the newly created socket

```
int sockfd = socket (PF_INET, SOCK_STREAM, 0);
int sockfd = socket (PF_INET, SOCK_DGRAM, 0);
```

The combination of PF_INET and SOCK_STREAM implies TCP

Socket

- Socket Family (denote the address domain)
  - PF_INET denotes the Internet family – IP + port
  - PF_UNIX denotes the Unix pipe facility – name + port
  - PF_PACKET denotes direct access to the network interface (i.e., it bypasses the TCP/IP protocol stack)

- Socket Type (protocol related)
  - SOCK_STREAM is used to denote a byte stream – reliable TCP
  - SOCK_DGRAM is an alternative that denotes a message oriented service – unreliable UDP
  - SOCK_RAW bypasses the transport layer, and packet headers are made accessible to the application

Socket API

- Passive Open (on server)

```
int bind(int socket, struct sockaddr *addr, int addr_len)
int listen(int socket, int backlog)
int accept(int socket, struct sockaddr *addr, int addr_len)
```

- Active Open (on client)

```
int connect(int socket, struct sockaddr *addr, int addr_len)
```

- Sending/Receiving Messages

```
int send(int socket, char *msg, int mlen, int flags)
int recv(int socket, char *buf, int blen, int flags)
```

Network Performance

- Communication overhead
- Throughput, e.g., 45 Mbps
- Latency, e.g., 50 ms coast-to-coast, round-trip ~100 ms.
- Depends on the application:
  - A 1 byte message is dominated by Round-trip Time
  - Loading a 25 MB file dominated by throughput (10 Mbps link takes 20 seconds.

Performance Metrics

- **Bandwidth**
  - Meaning changes from case by case:
    - The "*real*" definition is the range of frequencies in a signal.
    - The commonly used definition in networking: data transmitted per time unit, e.g., throughput

You must distinguish between these two!

- **Throughput -** Link versus end-to-end
  - Notation
    - KB = 2^{10} bytes
    - Mbps = 10^6 bits per second

Performance Metrics

- **Latency (delay)**
  - Time to send message from point A to point B
  - *One-way* versus *round-trip time* (RTT)

```
Latency = Propagation + Transmit + Queue
Propagation = Distance / c
Transmit = Size / Bandwidth
```

Queue??

Bandwidth versus Latency

- Relative importance
  - **1-byte:**

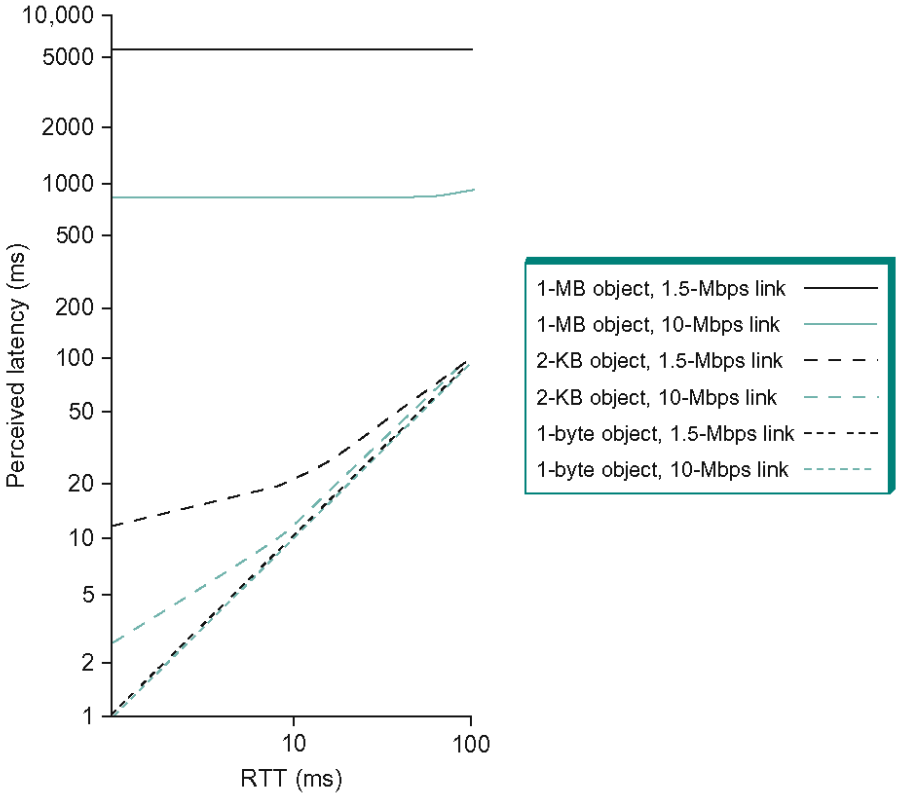
1ms vs. 100ms (latency) dominates

1Mbps vs. 100Mbps (bandwidth)

  - **25MB:**

1Mbps vs. 100Mbps dominates

1ms vs. 100ms



Delay x Bandwidth Product

- Amount of data "in flight" or "in the pipe"
- Usually relative to RTT
- Example: 100ms x 45Mbps = 550KB

  - If the sender does not fill the pipe (send a whole delay × bandwidth product's worth of data before it stops to wait for a signal), the sender will not fully utilize the network

Delay x Bandwidth Product

- Relative importance of bandwidth and latency depends on application
  - For large file transfer, bandwidth is critical
  - For small messages (HTTP, NFS, etc.), latency is critical
  - Variance in latency (jitter) can also affect some applications (*e.g.*, audio/video conferencing)

Summary

- We have defined a layered architecture for computer networks that will serve as a blueprint for our study
- We have discussed the socket interface which will be used by applications for invoking the services of the network subsystem
- We have discussed two performance metrics using which we can analyze the performance of computer networks

Reading

- Chapter 1 of textbook