# CS 520 Networking in the TCP/IP enviornment


Instructors:

Dr. Esam Alwagait

*alwagait@KSU.EDU.SA*

Dr. Mishari Almishari

*mialmishari@ksu.edu.sa*


*Slide Set were original prepared by Dr. Tatsuya Susa*

# Contacts


- Dr. Esam Alwagait
  - *alwagait@ksu.edu.sa*
- Dr. Mishari Almishari
  - *mialmishari@ksu.edu.sa*
  - *4698903*

# Syllabus -- Tentative

- Introduction
- Data Link Layer
- Network Layer
  - IP, Routing Protocols, RIP, OSPF, BGP,…
- Transport Layer
  - TCP, UDP, Congestion Control,…
- Multicast
- QoS
  - Queueing ,Integrated Services, Differentiated Services
- Application Protocols
- P2P Networks
- Content-Centric Network
- Network Security

# Grading

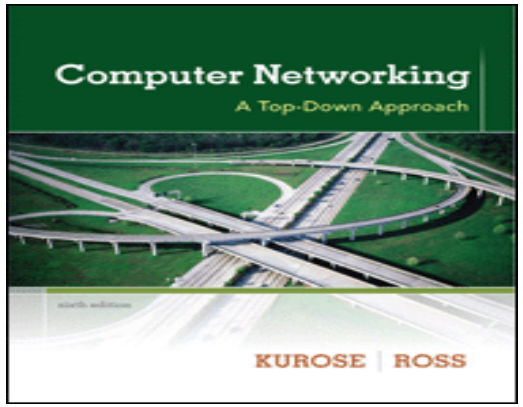
30 % Midterm

30% Project

40% Final

# Course Materials

- **Slides**
- **Problem Sets**
- **Book**
  - **Computer Networking: A Top-Down Approach, Kurose and Ross – Sixth edition**




- **Additional Readings**

# Overview of Computer Networks

# Overview of the Principles of Computer Networks

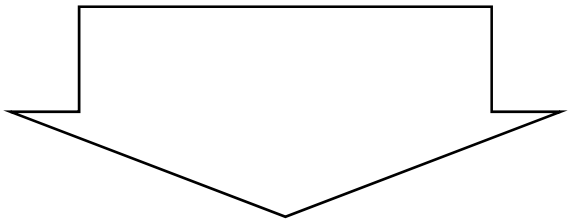
1. Physical communication
2. Error control
3. Flow control
4. Data switching
5. Routing
6. Congestion Control
7. Shared Medium Access
8. Protocols

# 1. How Do Computers Communicate?

- With 1's and 0's
  - Computers only deal with 1's and 0's
  - So do networks

# 2. Error Detection/Correction

Physical channels are not perfect



Can we detect, possibly correct errors?

# 2.1 Error Detection

Example: Parity bit

| Message | Parity Bit | |
|---|---|---|
| 0 0 1 1 | 1 | (odd) |
| 1 0 0 0 | 0 | (odd) |
| 0 1 0 1 | 0 | (even) |
| 1 0 0 0 | 1 | (even) |

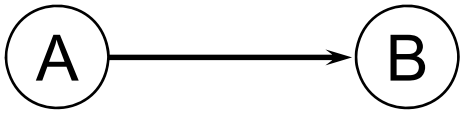
# 2.2 Error Correcting Codes

Detect and correct errors

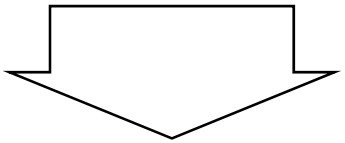
Example: Hamming Code

Capable of detecting and correcting a single bit error

# 3. Flow Control

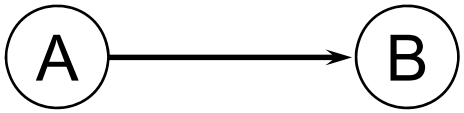



If A sends at a faster rate than B can receive, bits will be lost



We need flow control!

# 3.1 Stop-and-Wait

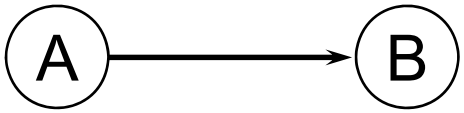



A sends data →

← B sends ACK

A sends more data →

# 3.2 Windowed Flow Control




A sends packets 1, 2, 3 →

← B sends ACK for 1, 2

A sends packets 4, 5 →

← B sends ACK for 3, 4, 5

# 4. Switching Schemes

(1) Circuit Switching

(2) Message/Packet Switching (Store-and-Forward)

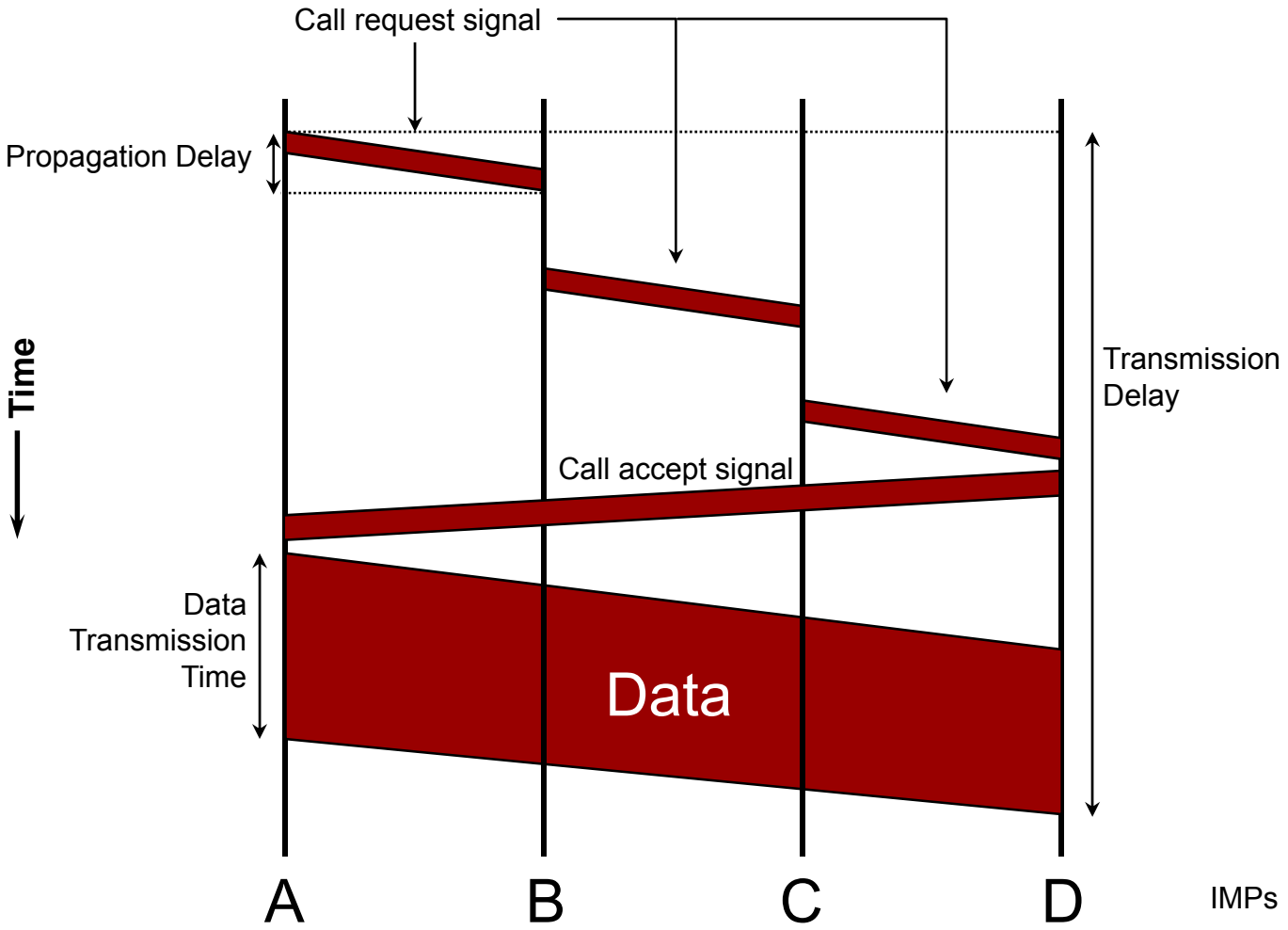
# 4.1 Circuit Switching

- Provides service by setting up the total path of connected lines from the origin to the destination
- Example: Telephone network

# Circuit Switching (cont'd)

1. Control message sets up a path from origin to destination
2. Return signal informs source that data transmission may proceed
3. Data transmission begins
4. Entire path remains allocated to the transmission (whether used or not)
5. When transmission is complete, source releases the circuit

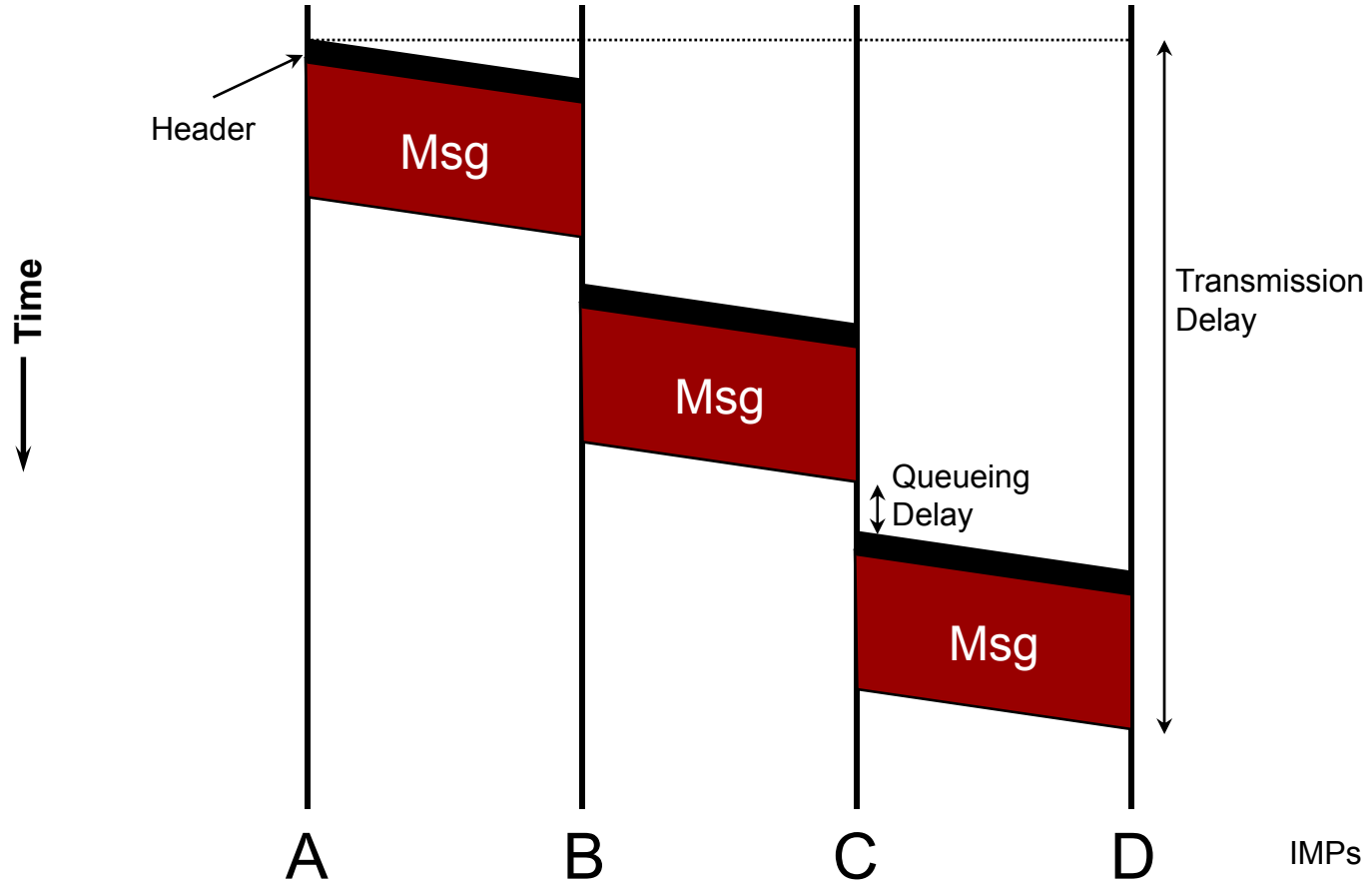
# Circuit Switching (cont'd)



# 4.2 Message Switching

- Each message is addressed to a destination
- When the entire message is received at an IMP "Interface Message Processor", the next step in its journey is selected; if this selected channel is busy, the message waits in a queue until the channel becomes free
- Thus, the message "hops" from node to node through a network while allocating only one channel at a time

# Message/Packet Switching (cont'd)



# 4.4 Comparisons

(1) **Header Overhead**

Circuit < Message

(2) **Transmission Delay**

Short Messages:

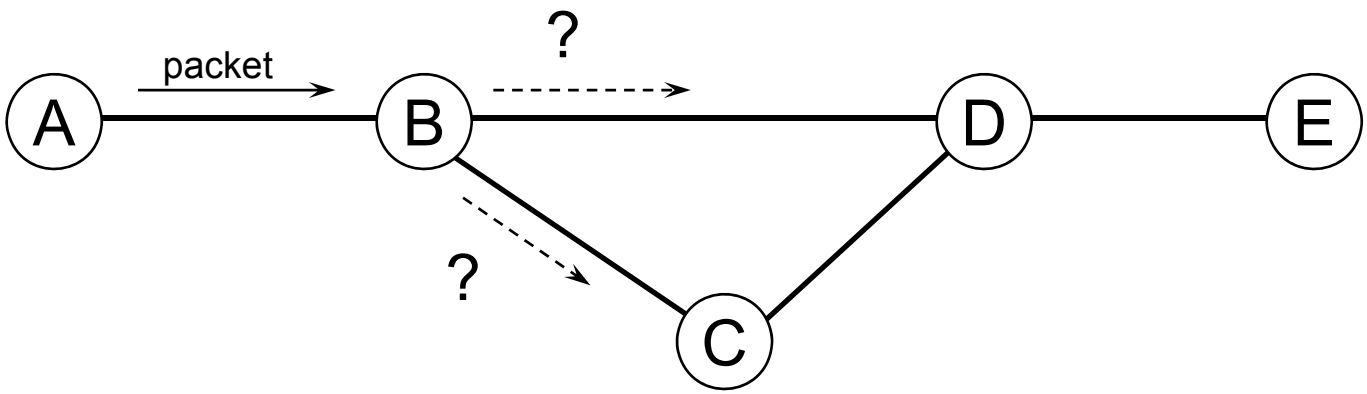
Message < Circuit

Long Messages:

Circuit < Message

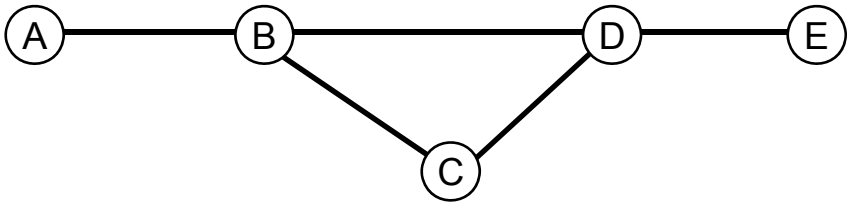
# 5. Routing

A slightly more complex network:




If A sends packets to E, B has to make a routing decision

# 5.1 Fixed Routing Schemes

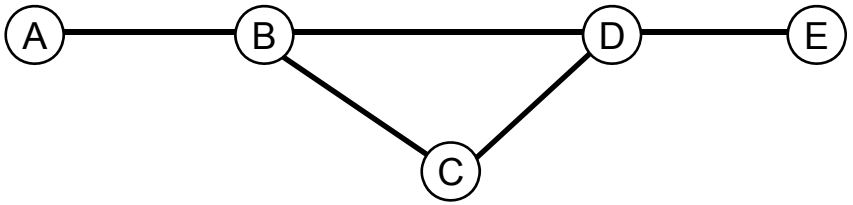



- For every source/destination pair, a fixed path is given
- Example: A to B to D to E
- Easy to implement, but

  (1) What if link BD fails?

  (2) What if link BD is congested?

# 5.2 Dynamic Routing Schemes




- Hosts and IMPs periodically exchange information and find the best paths for all source/destination pairs
- Can adjust to link failure or congestion, but

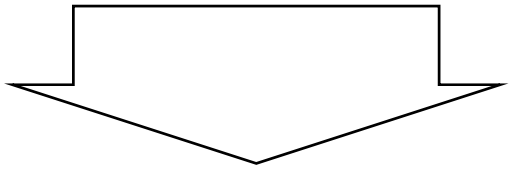
  (1) information exchange creates extra traffic

  (2) it takes time for information exchange to happen, so information could be outdated

  packet looping can happen

# 6. Congestion Control

What if too many packets flow through a network?



Congestion

Analogy: Two merging freeways

# 6.1 Packet discarding

- In case of congestion, discard some packets
- Require receiving host to send ACK when it receives a packet; this way the transmitter will know when a packet is lost and it can retransmit the packet
- Reactive

# 6.2 Choke Packets

- When the network becomes congested, a host or IMP sends a “choke packet” telling a transmitter to slow down
- Reactive

# 6.3 Isarithmic Flow Control

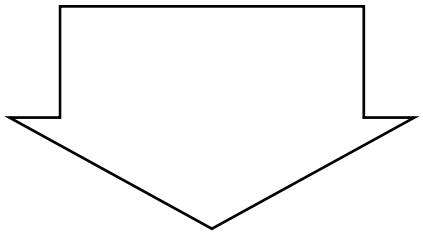
- Only allow one packet into the network for every packet that leaves the network
- In freeway terms: Only let a car in at the onramp if another car gets off the freeway somewhere else
- Preventive

# 6.4 Rate Control

- Only allow one packet into the network for every T time interval
- In freeway terms: Only let a car in per green light at a freeway entrance
- Preventive

# 7. Medium Access

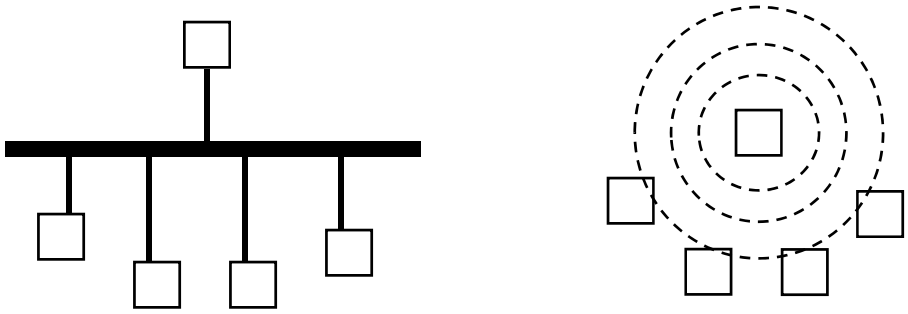
Many users typically share a single link or a single medium



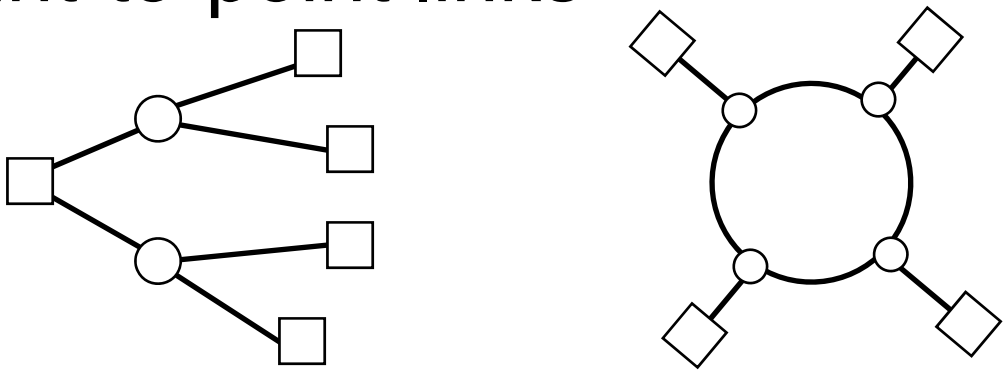
How do you give them all access?

# 7.1 Possible Media

- Broadcast or shared channels

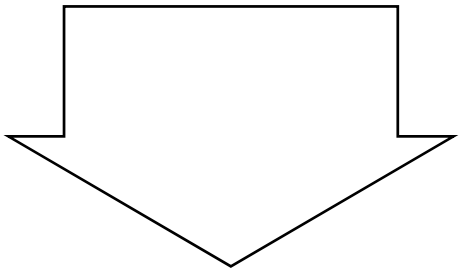


- Point-to-point links



# 7.2 Multiple Access Protocols

If more than one host sends at the same time, there is a collision

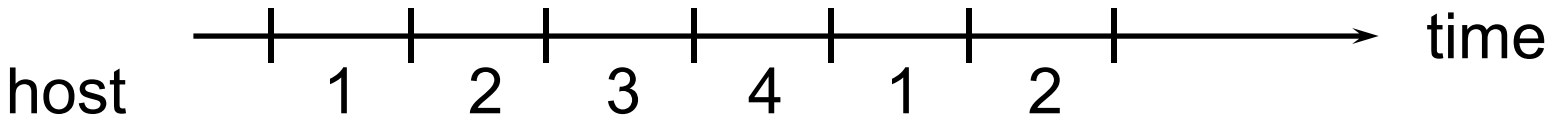


Need algorithm to share the channel:

**Multiple access protocol**

# 7.2.1 Fixed Assignment Schemes

Example: Time Division Multiple Access (TDMA)

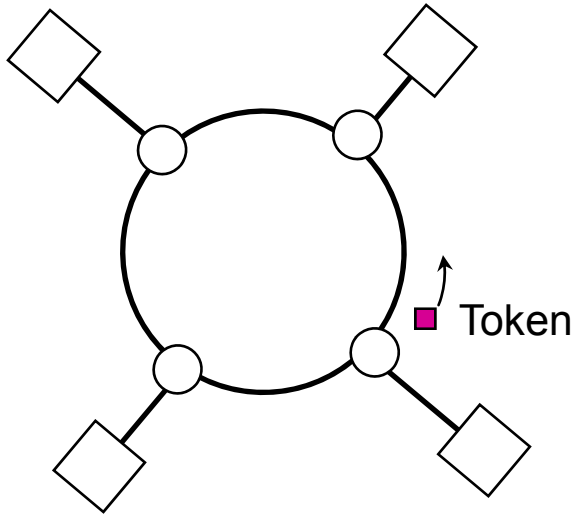



Channel capacity is assigned even to users who have nothing to send

# 7.2.2 Demand Assignment

Assign channel capacity only to those who have packets to send

Example: Token Passing Scheme

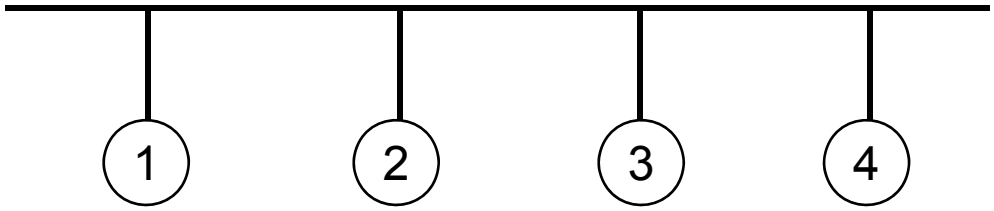



Hosts may send packets when they grab the token

# 7.2.3 Random Access

Send when you have a packet to send. If collision, retransmit

Example: Ethernet



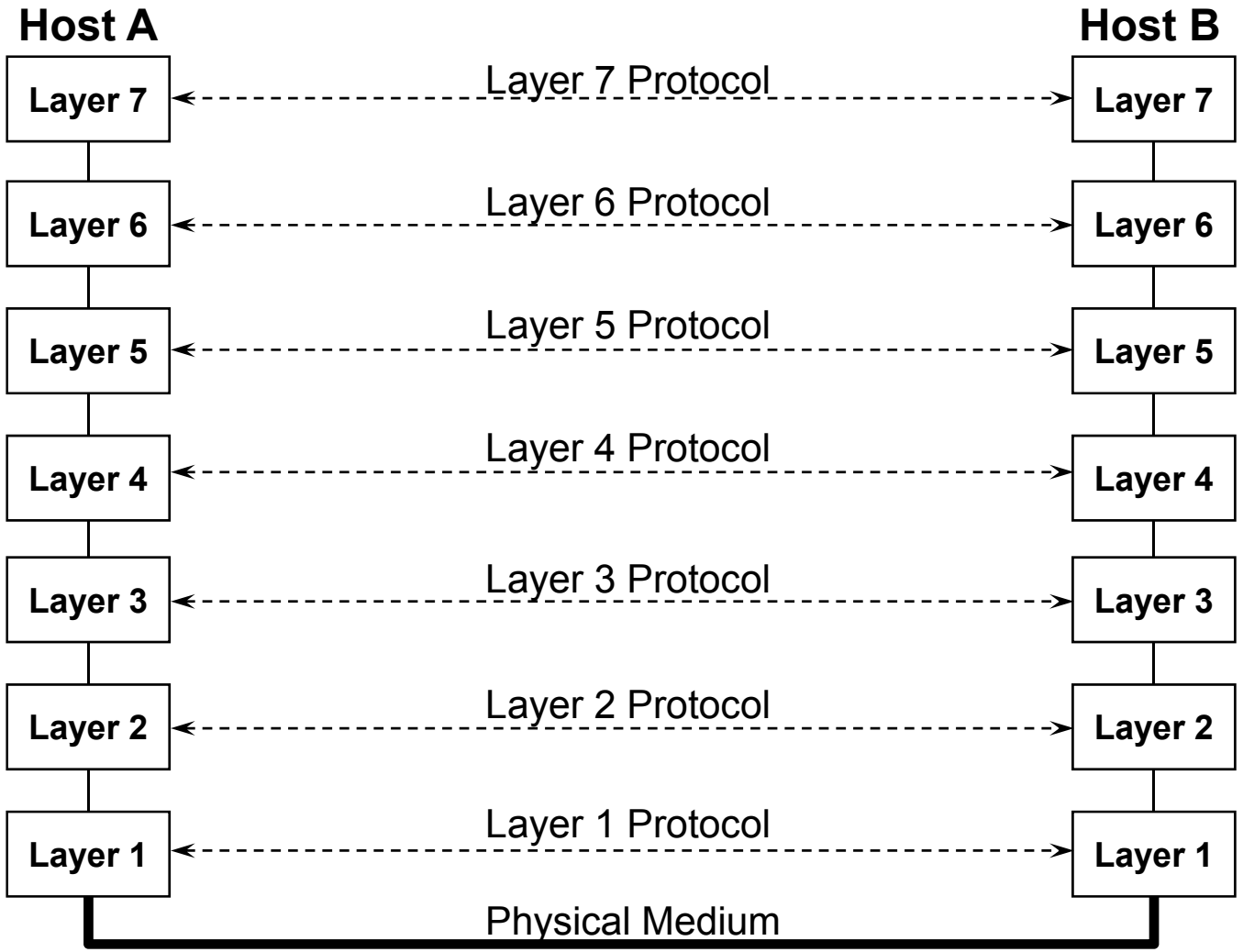

Listen before transmission
if busy, wait
if idle, transmit

Listen during transmission
if collision, abort
and retransmit

# 8. Protocol Hierarchy

- Use layers to hide complexity
  - Each layer implements a service
    - Layer N uses service provided by layer N-1
    - layer N-1 provides a service to layer N
  - Protocols
    - Each layer communicates with its peer by a set of rules
- Interface
  - A layers interface specifies the operations

# Protocol Hierarchy *(cont'd)*



# Why Layering?

- Simplicity
  - Network communication is very complex
  - Testing and maintenance is simplified
  - Easy to replace a single layer with a different version

# 9. Types of Networks in an Internet

- Local area networks
  - Privately owned, within building
  - High speed, broadcast, Ethernet
    - 2 to 100 Mbps
- Wide area networks
  - Spans a large area
  - Point-to-point, high speed fiber or trunk lines
    - Long delays but very high speed links

# Types of Networks *(cont'd)*

- Wireless networks
  - Hosts connected by infrared or radio links
  - Local area and wide area
  - Satellite networks
  - Others

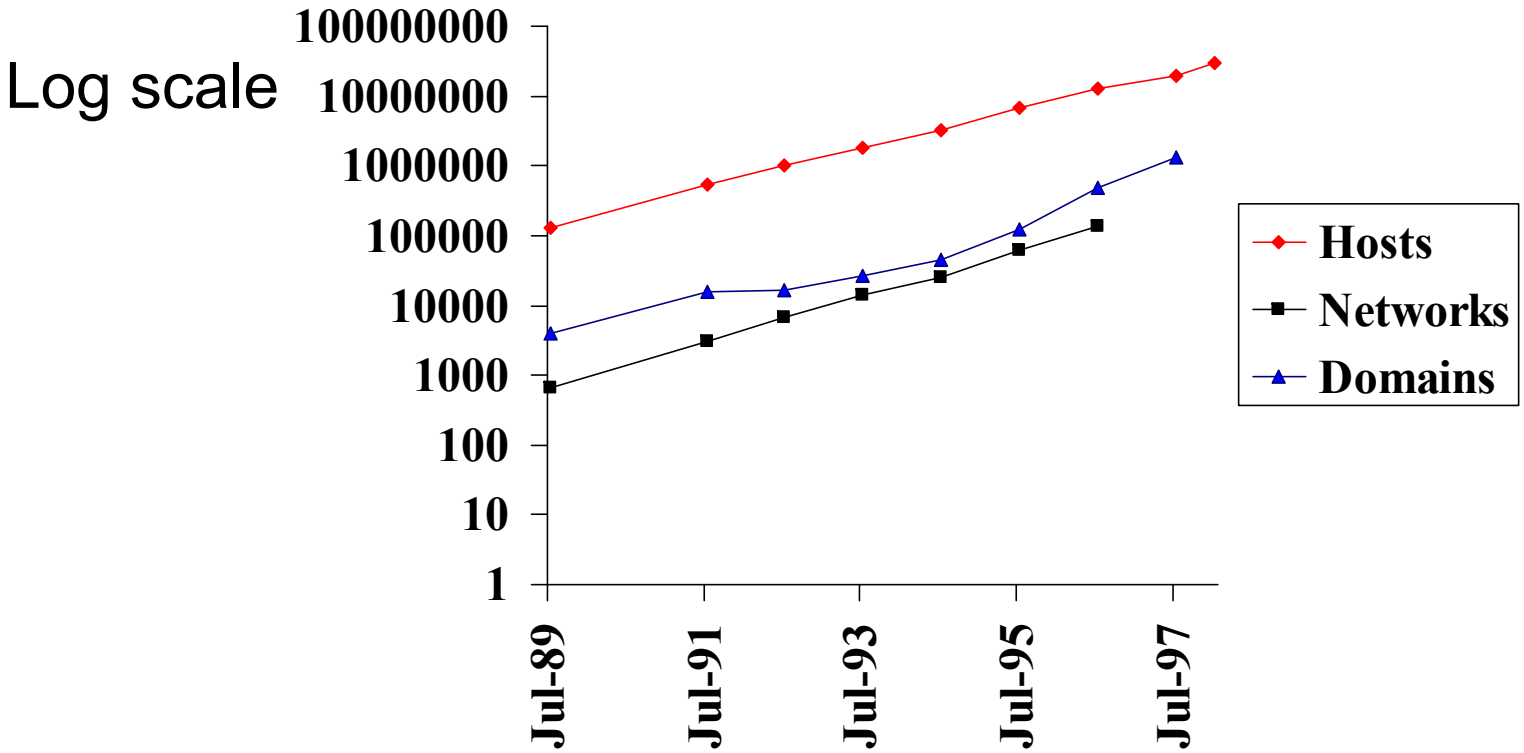
# 10. Historical perspective

- 1960's: ARPAnet (4 nodes)
- 1970's: Aloha net, Ethernet, multiple access protocols
- 1980's: early internet growth, e-mail & ftp still dominant, CSNET, NSFNET
- Early 1990s: 100,000 hosts, gopher, newsgroups

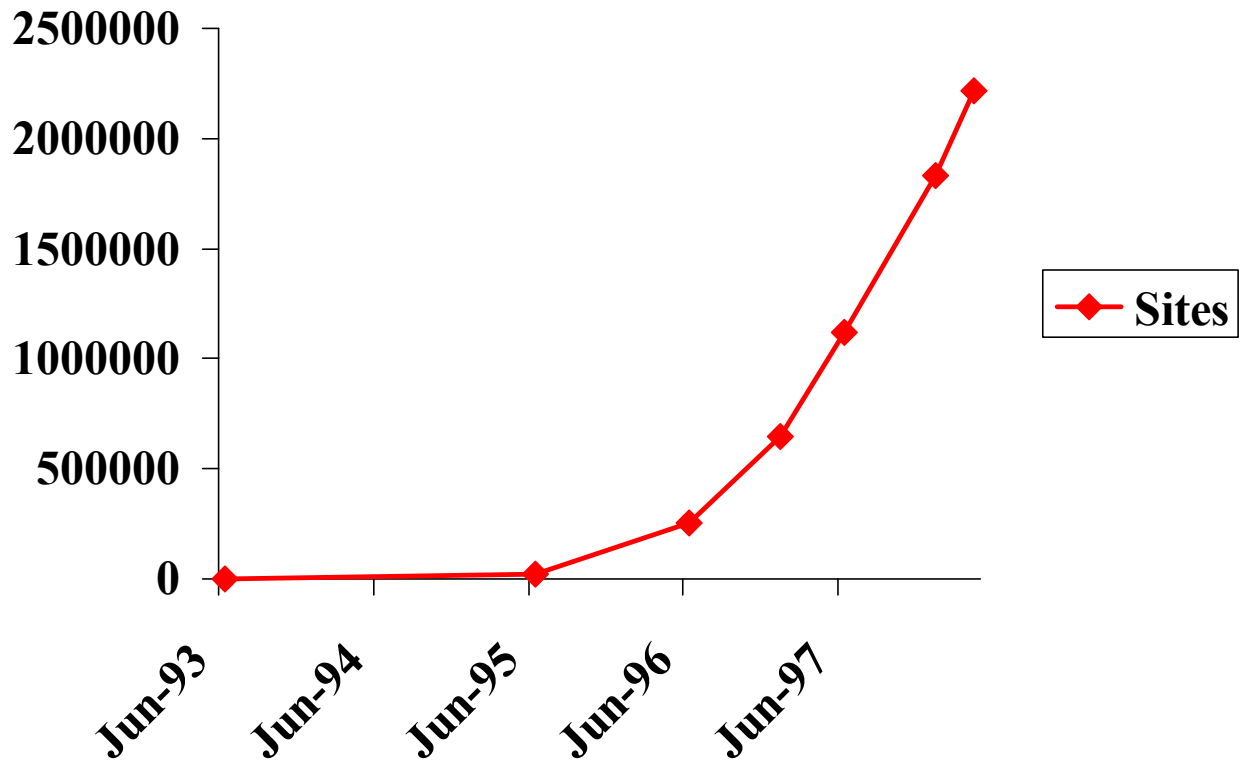
# Historical perspective

- 1991: http proposed by Tim Berners Lee at CERN, Switzerland
  - Changed everything, Information highway, e-commerce, alternate medium, part of popular culture, net surfing
- Peer-To-Peer
  - bittorrent
- Now…Web 2
  - User generated content
  - Social media

# 11. Internet Growth



# WWW server Growth



# Overview of Network Protocols

# Standards Making Organizations

ISO = International Standards Organization

ANSI = American National Standards Institute

IEEE = Institute of Electrical  and Electronic Engineers

IETF = Internet Engineering Task Force

ATM Forum = ATM standards-making body

...and many more

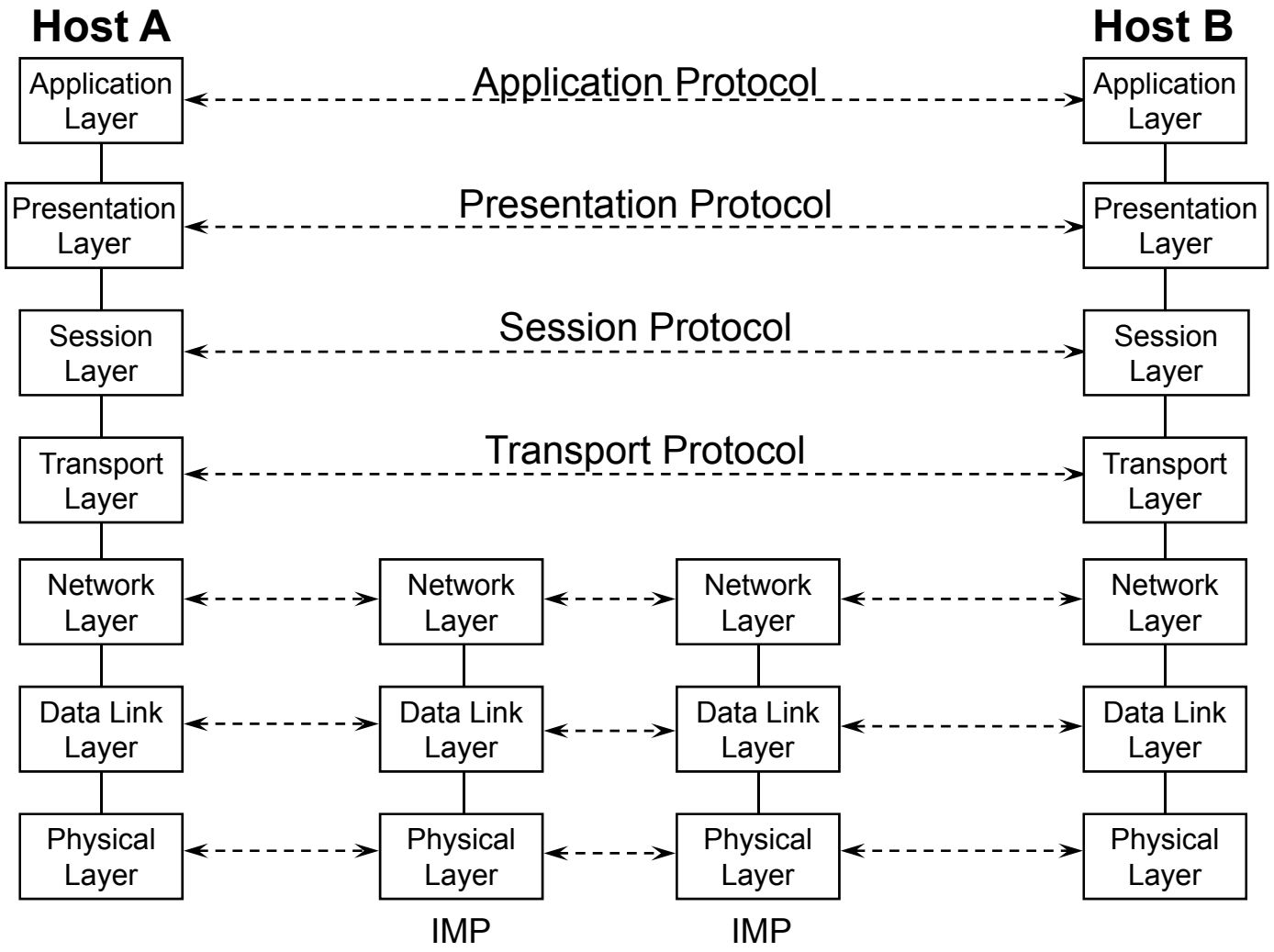
# Why So Many Standards Organizations?

- Multiple standards
- Different areas of emphasis

# Different Layering Architectures

- ISO OSI 7-Layer Architecture
  - ISO: International Standard Organization
  - OSI: Open Systems Internconnection
- TCP/IP 4-Layer Architecture
- Novell NetWare IPX/SPX 4-Layer Architecture

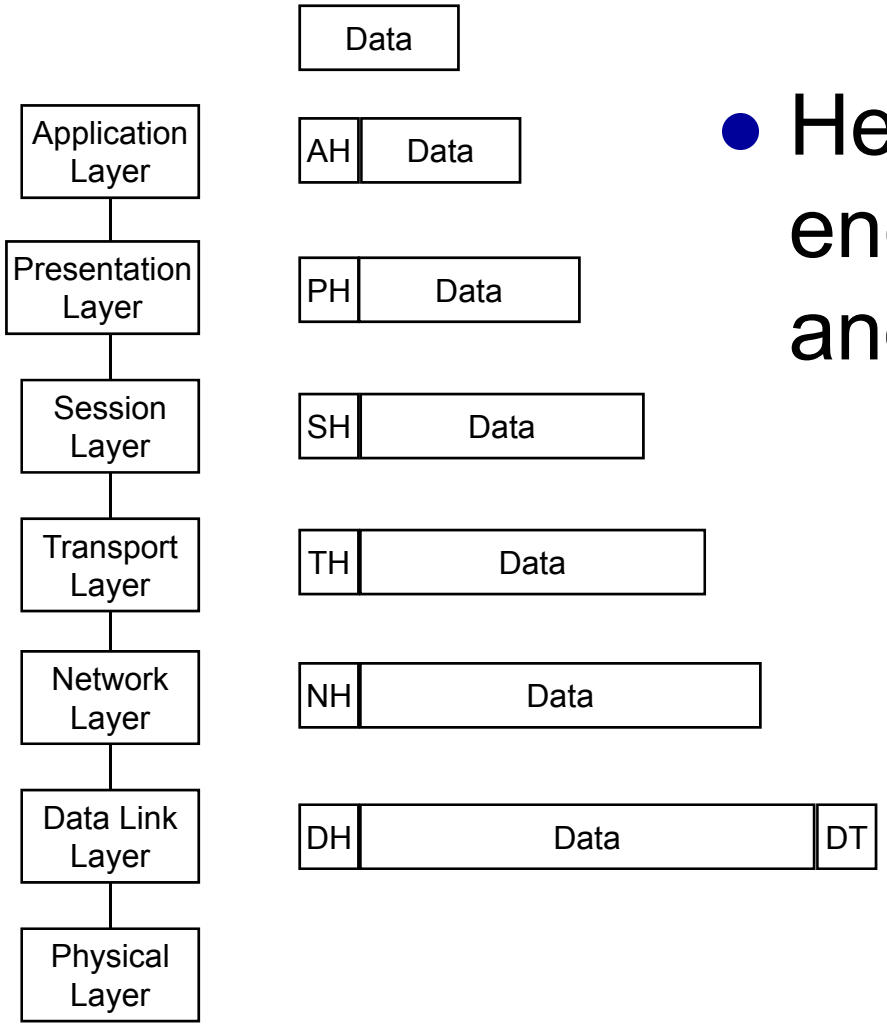
# ISO OSI Layering Architecture



# ISO's Design Principles

- A layer should be created where a different level of abstraction is needed
- Each layer should perform a well-defined function
- The layer boundaries should be chosen to minimize information flow across the interfaces
- The number of layers should be large enough that distinct functions need not be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy

# General Protocol Functions




- Header encapsulation and stripping

# Layer 1: Physical Layer

- Functions:
  - Transmission of a raw bit stream
  - Forms the physical interface between devices
- Issues:
  - Which modulation technique (bits to pulse)?
  - How long will a bit last?
  - Bit-serial or parallel transmission?
  - Half- or Full-duplex transmission?
  - How many pins does the network connector have?

# Layer 2: Data Link Layer

- Functions:
  - Provides reliable transfer of information between two adjacent nodes
  - Creates frames, from bits and vice versa
  - Provides frame-level error control
  - Provides flow control
- In summary, the data link layer provides the network layer with what appears to be an error-free link for packets

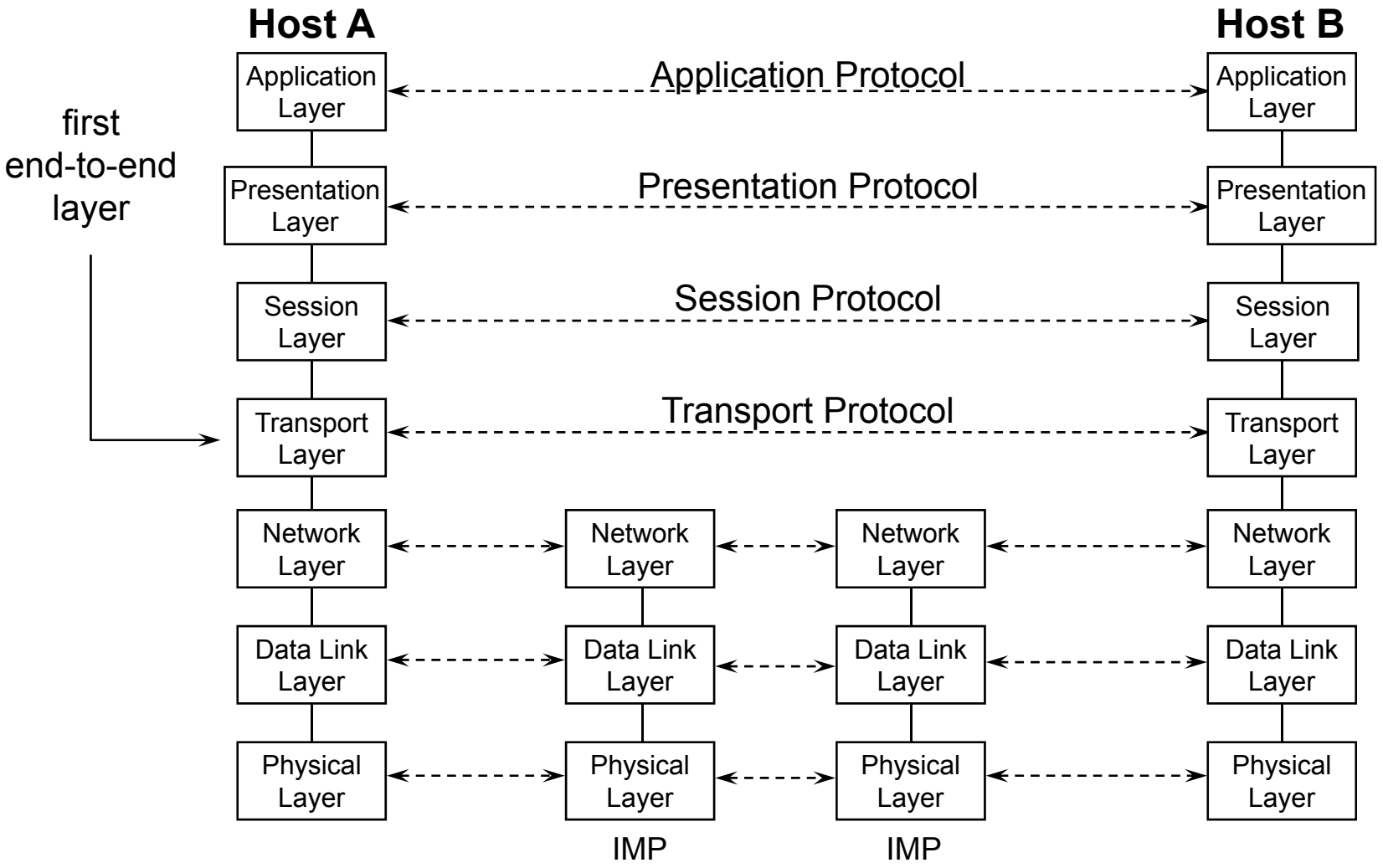
# Layer 3: Network Layer

- Functions:
  - Delivering message for destination
  - Responsible for routing decisions
    - Dynamic routing
    - Fixed routing
  - Performs congestion control

# Layer 4: Transport Layer

- Functions:
  - Hide the details of the network from the session layer
  - Provides reliable end-to-end communication

# Transport Layer (cont'd)



# Transport Layer (cont'd)

- Functions (cont'd):
  - Perform end-to-end flow control
  - Perform packet retransmission when packets are lost by the network

# Layer 5: Session Layer

- Mainly, groups several user-level connections into a single “session”

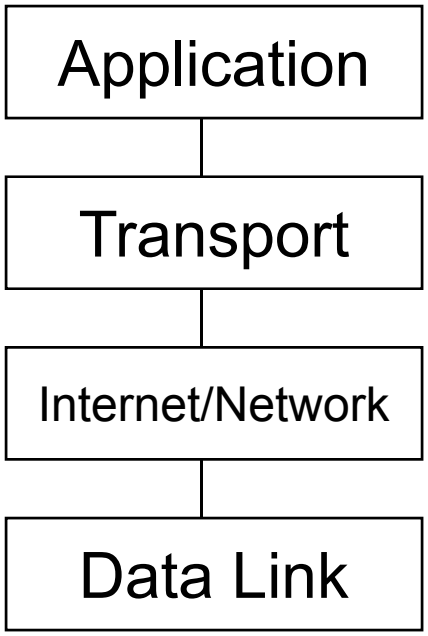
# Layer 6: Presentation Layer

- Performs specific functions that are requested regularly by applications
- Examples:
  - encryption
  - ASCII to Unicode, Unicode to ASCII
  - LSB-first representations to MSB-first representations

# Layer 7: Application Layer

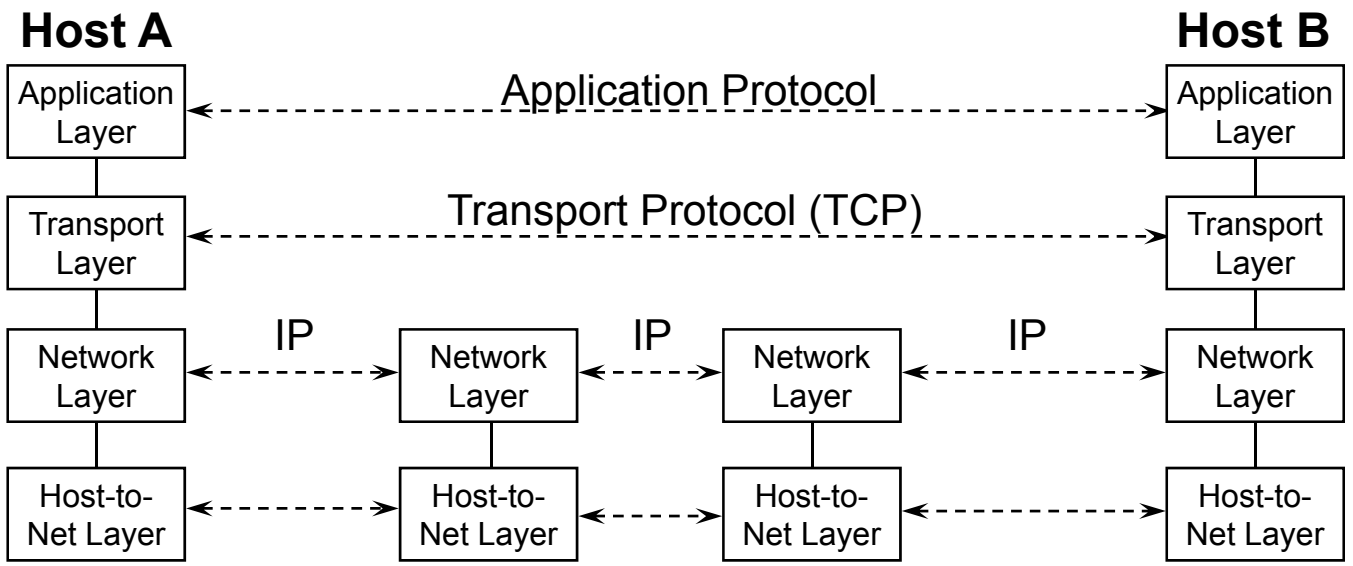
- Application layer protocols are application-dependent
- Implements communication between two applications of the same type
- Examples:
  - FTP
  - Netrek
  - SMTP (email)

# TCP/IP Layering Architecture




- A simplified model

# TCP/IP Layering Architecture *(cont'd)*



# The Internet's Design Principles

- Ad hoc: the protocols came first, and the layering model came later
- TCP/IP specifically designed for the Internet
- TCP/IP model doesn't describe other protocols well

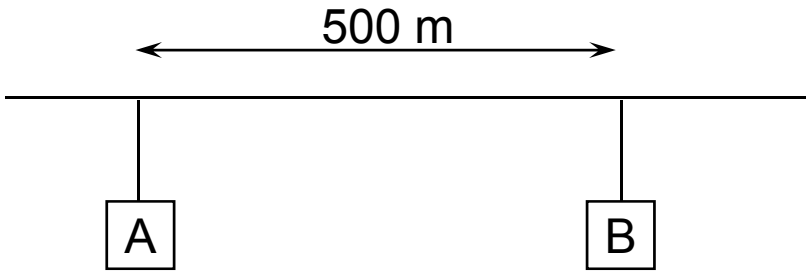
# Internet Design Issues

- Scale
- Incremental deployment
  - New protocols need to be deployed in pockets
- Heterogeneity
  - Different technologies, autonomous organizations
- End-to-end argument
  - Networking functions should be delegated to the edges; application knows best

# More Definitions

- **Packet length**: size of a packet (units = bits)
- **Channel speed**: How fast the channel can transmit bits (units = bits/second, Mbits/sec, Gbits/sec)
- **Packet transmission time**: amount of time to transmit an entire packet (units = seconds, msec, μsec)
- **Propagation delay**: Delay imposed by the transmission medium, depends on distance (units = μsec/km)

# Some Math




packet length - 1500 bytes

channel speed - 10 Mbits/sec

propagation delay - 5 μsec/km

---

packet transmission time = packet length / channel speed

= (1500 bytes x 8 bits/byte) / (10 x $10^6$ bits/sec)

= 1.2 x $10^{-3}$ sec or 1.2 msec

prop. delay between A & B = (500 m x 1km/1000 m) x (5 x $10^{-6}$ sec/km)

= 2.5 x $10^{-6}$ sec or 2.5 μsec