# Lecture 1: Introduction

COMP 332, Spring 2018

Victoria Manfredi

WESLEYAN UNIVERSITY

**Acknowledgements**: materials adapted from Computer Networking: A Top Down Approach 7th edition: ©1996-2016, J.F Kurose and K.W. Ross, All Rights Reserved as well as from slides by Abraham Matta at Boston University and some material from Computer Networks by Tannenbaum and Wetherall.

# Today

1. Administrivia

2. Computer network
   - components
   - communication

3. Internet
   - some statistics
   - structure

# Administrivia

# Course webpage

## Everything posted here

- http://vumanfredi.web.wesleyan.edu/comp332-s18/

## Grade breakdown

- 40%: 2 exams
- 60%: 10 homework assignments, no scores dropped
  - mix of written and (multi-assignment) programming projects

## Late days

- 4 free days, use at most 2 for any assignment
- Once used, you will lose 15% of grade for each 24 hours late

## Python3

- we'll review as needed, see class resources webpage
  - please check you have python3 installed!
  - tutorials and other resources posted

## Computer Network

# SOME MOTIVATION

# Why do computer networks matter?

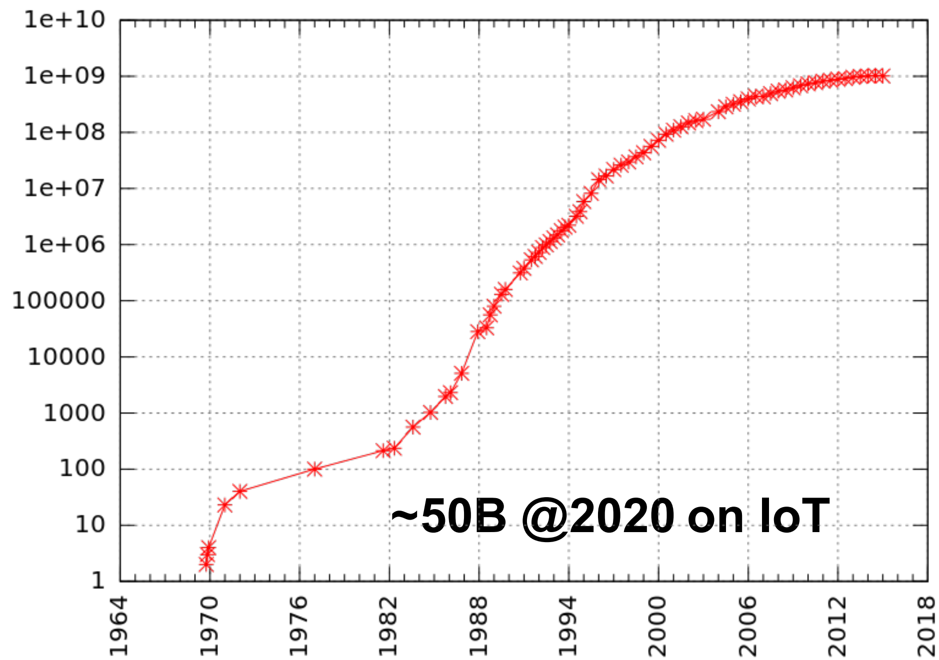
## Networks of processes are ubiquitous

- to support a myriad of distributed applications

## They are getting larger and more complex

- need experts in leveraging & managing them


**Number of hosts on Internet**



# Many “networking” firsts originated not too far away

## First optical (light) “one-if-by-land-and-two-if-by-sea” signals

- used to signal that the British are coming in 1775

**Paul Revere**

## First telegraph (Morse code)

- used by Boston Fire Alarm Telegraph System for reporting fires in 1852

## First transatlantic radio message

- from Nova Scotia to England in 1902

**Guglielmo Marconi**

## First switches and email message

- at BBN in 1967-1972

# Expectations

## This class **IS** about...

- concepts, principles, and protocols
- general-purpose computer networks
- Internet perspective
- network software
- designing and building a system

## This class **IS NOT** about...

- specialized networks (e.g., CATV, telephone)
- ISO/OSI perspective
- network hardware
- advanced theoretical analysis

# Why build a computer network?

## User view

### Sharing resources

- **hardware**: printers, compute servers, cloud computing
- **software**: word, Matlab
- **data**: customer records, inventory, financials, p2p file sharing
- **information**: web-browsing, Wikipedia, search

### Communication

- email, text, voIP, screen share, video conference, social network

### Electronic commerce

- online shopping, banking, business

### Entertainment

- multi-user network games, video streaming

## Programmer view

### To support distributed applications

- e.g., web, ftp, …

### Most functionality in software:

- many applications, easy to create

### General-purpose, increasingly faster computers

- can manage many processes

### New functionality easily added ``inside'' network

- e.g., Content Distribution Net

# Distributed system vs. computer network

## Distributed system

– software system built on top of a computer network

## Example

– World Wide Web is built on top of the Internet

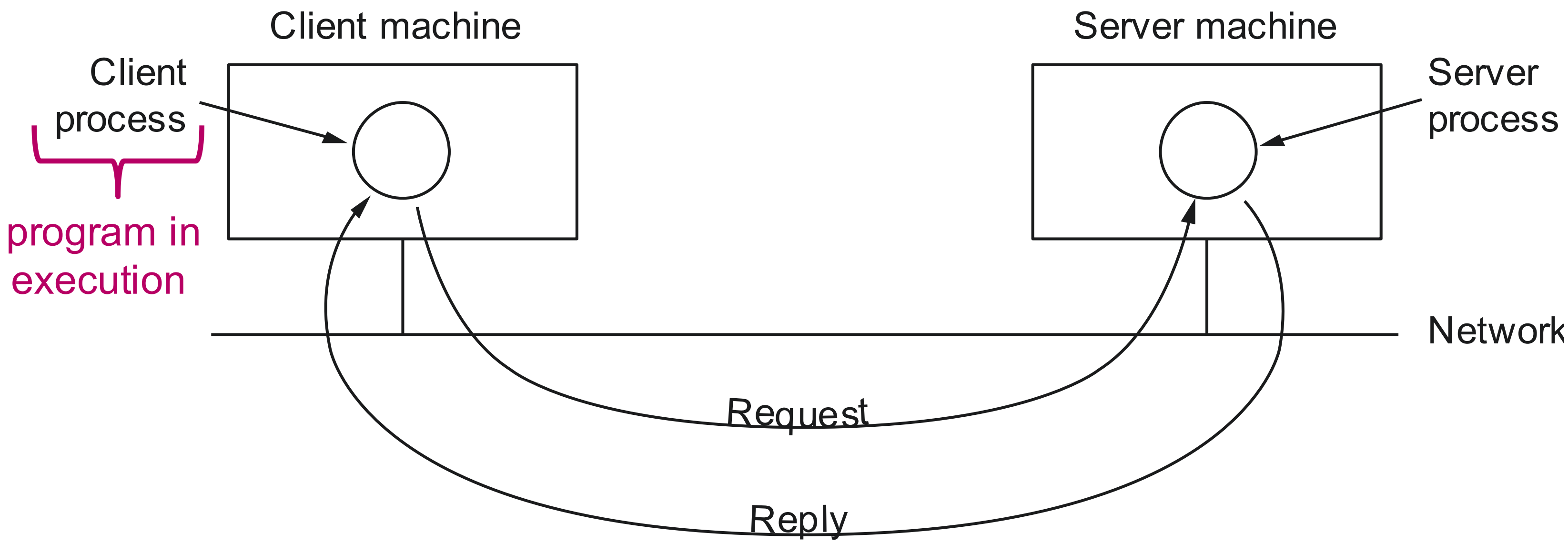
Distributed system — Computer network

## Computer Network

# COMPONENTS

# Common functionality for networks

2 or more computing devices able to exchange data




1. Specify remote machine
2. Connect to it (possibly some handshaking)
3. Transfer data
4. Close connection

# Computer network requirements

1. Connectivity

2. Cost-effective resource-sharing

3. Process-to-process channels

# 1. Connectivity

## Building blocks

- **nodes**: laptop, server, router, switch, cell phone, UAV, IoT device…
- **links:** copper wire, coaxial cable, optical fiber, radio, …

### Telephone lines

- Ethernet
- up to 10 Gbps

### Cable television infrastructure

- shared/broadcast medium
- more people use simultaneously, less bandwidth you get
- 10’s of Mbps

### Glass fiber carrying light pulses (bits)

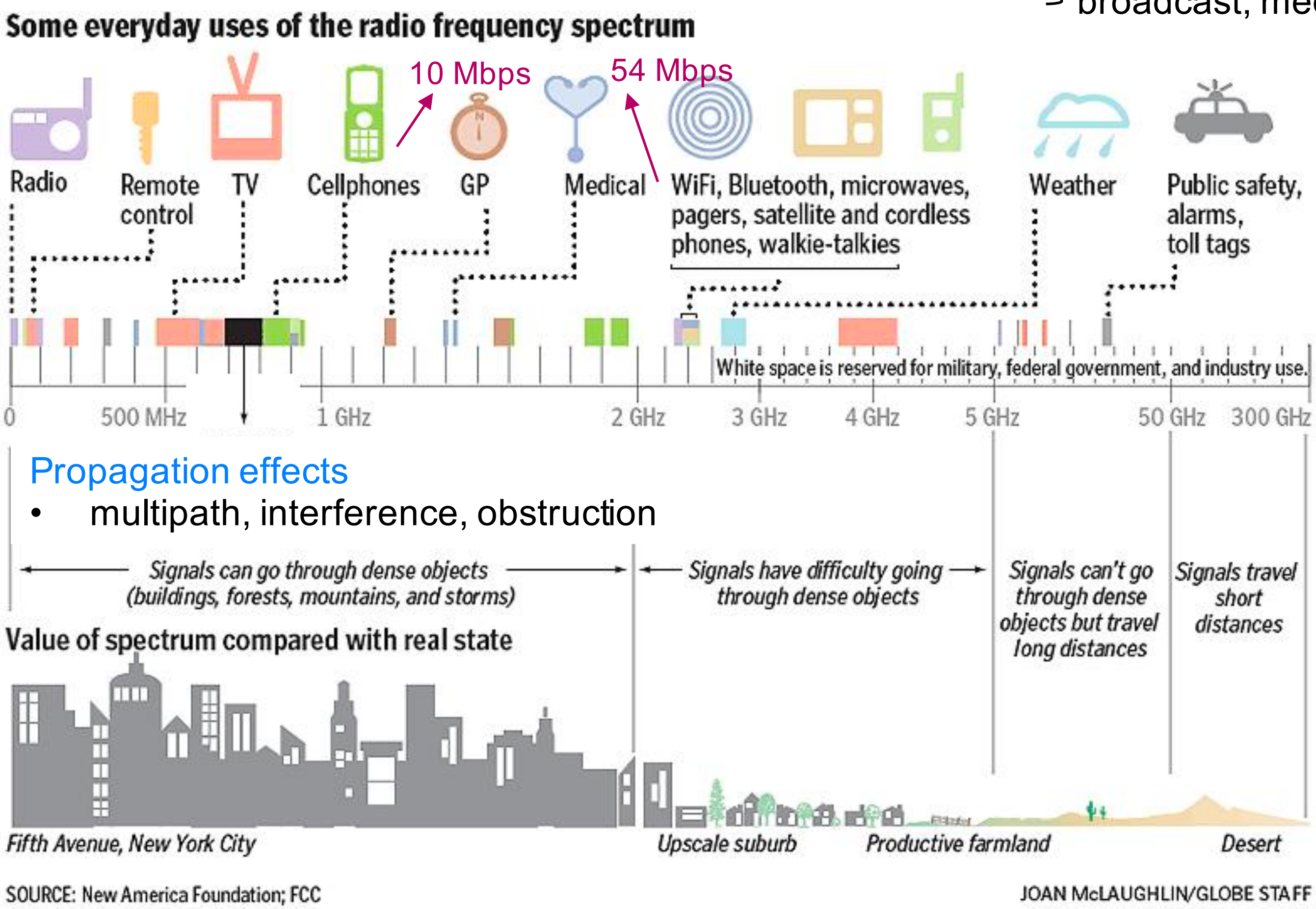
- forms Internet core: carries lots of traffic
- low bit error rate: unaffected by electromagnetic noise
- up to 100s of Gbps

# 1. Connectivity

Building blocks

- **nodes**: laptop, server, router, switch, cell phone, UAV, IoT device…
- **links:** copper wire, coaxial cable, optical fiber, radio, …

Signal carried in electromagnetic spectrum } Shared, typically broadcast, medium




10 Mbps

54 Mbps

Low bandwidth
Long distance

High bandwidth
Short distance

Propagation effects

- multipath, interference, obstruction

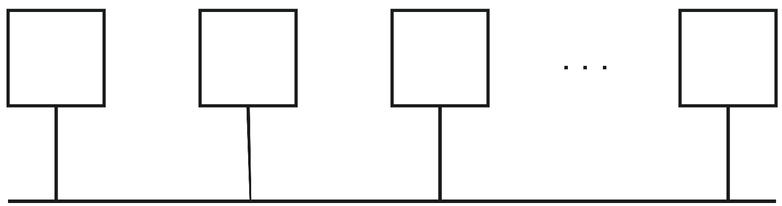
# 1. Connectivity

## Direct links

– point-to-point (e.g., dial-up, Digital Subscriber Line (DSL))

point-to-point network

– multiple access (LAN environment)



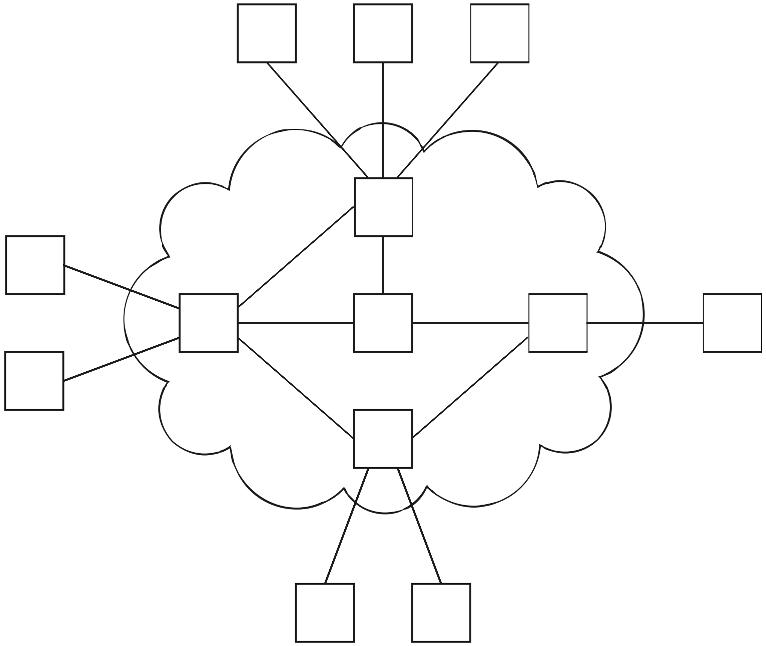
multiple access network

Need MAC (Medium Access Control) protocol to control access to shared medium (e.g., shared Ethernet, Hybrid Fiber Coaxial (HFC) upstream channel, wireless)

# 1. Connectivity

## Indirect connectivity

– switched networks (WAN environment)



**Intermediate nodes**

- called switches (net's core)

**End nodes**

- called hosts or end-systems (net's edge)

**Packet switching**

- send/receive messages (packets)
- may need fragmentation or reassembly

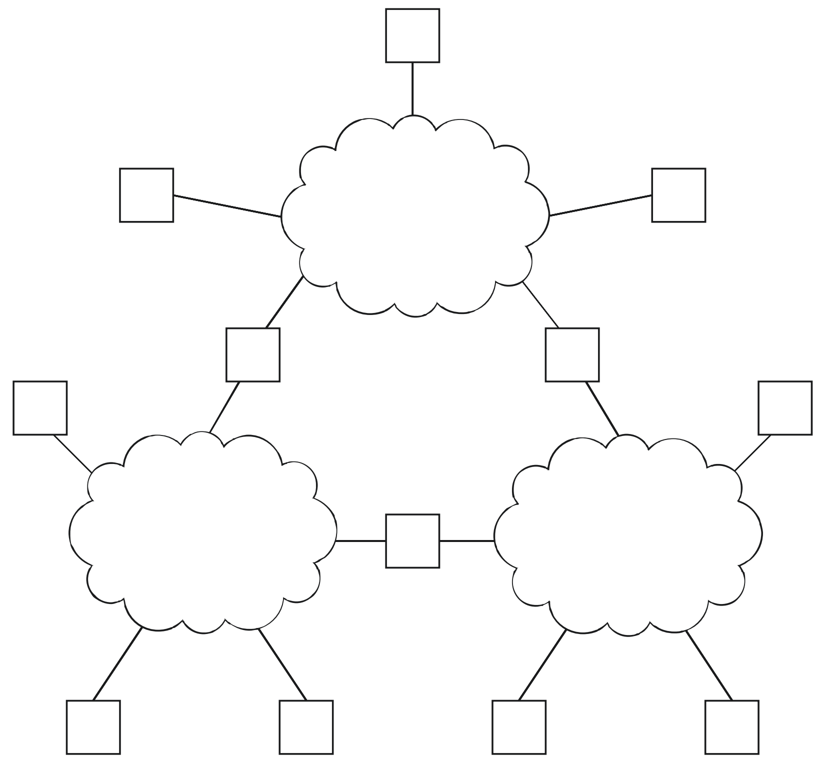
**Store-and-forward**

- switch must receive all bits of packet before forwarding

# 1. Connectivity

Internetworks

- nodes that connect networks are called routers or gateways



A network can be defined recursively

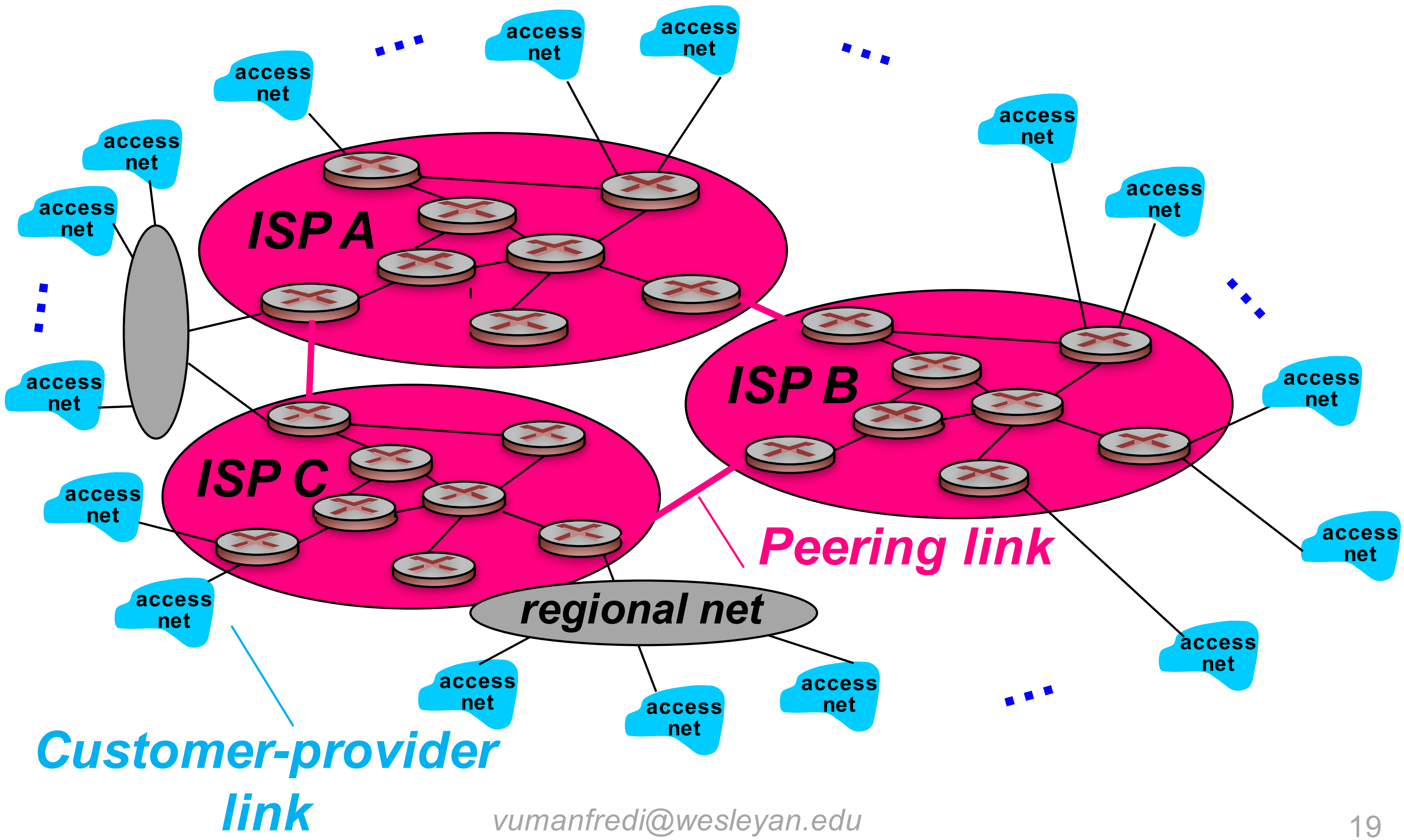
- 2 or more nodes connected by a physical link

<u>*OR*</u>

- 2 or more networks connected by 2 or more nodes

# 1. Connectivity

## Internet's core managed by a hierarchy of ISPs



# 1. Connectivity

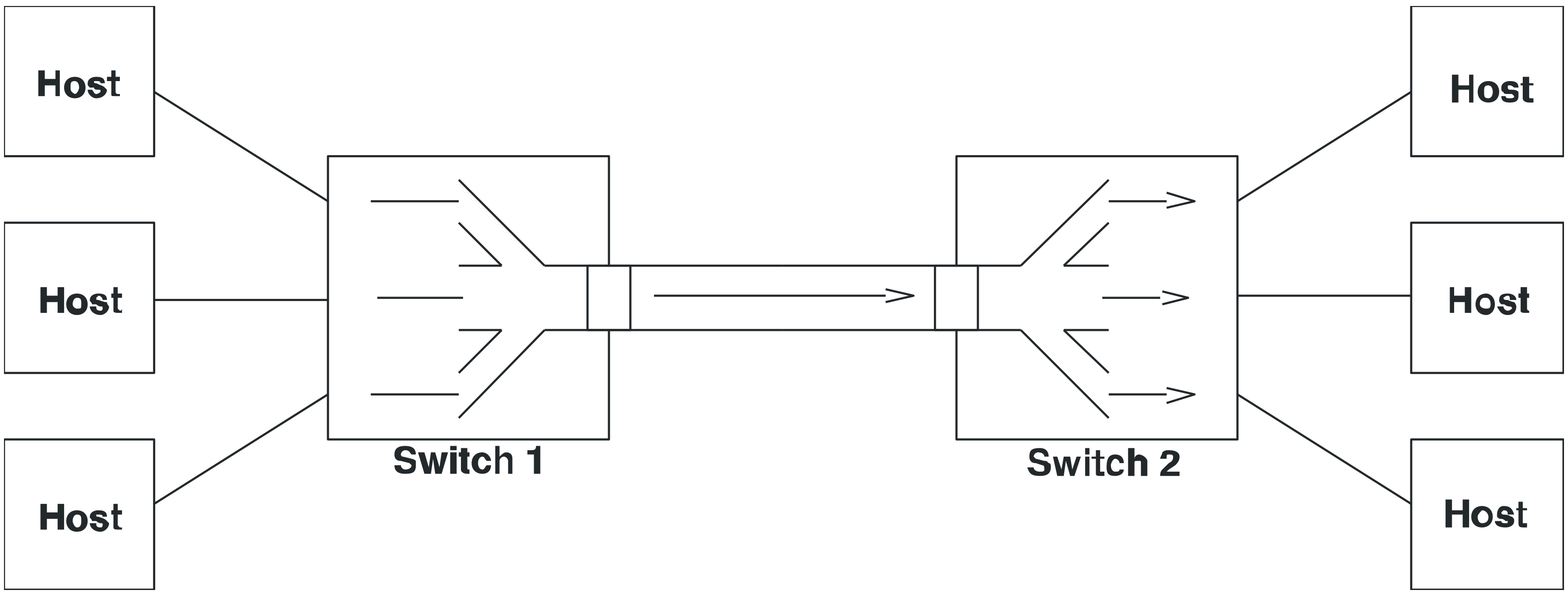
## Addressing and routing

- address
  - byte-string that identifies a node; usually unique
- routing
  - process of determining how to forward messages toward the destination node based on its address
- types of addresses
  - **unicast**: node-specific
  - **broadcast**: all nodes on the network
  - **multicast**: some subset of nodes on the network

# 2. Cost-effective resource sharing

## Network resources (nodes and links)

– must be shared (multiplexed) among multiple users



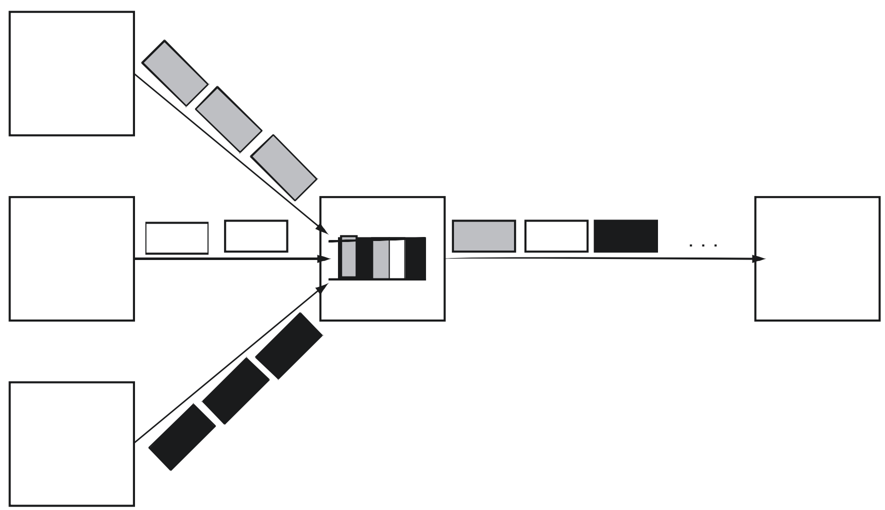

## Common Multiplexing Strategies

– Frequency-Division Multiplexing (FDM): pre-assign frequencies
– Time-Division Multiplexing (TDM): pre-assign time slots

# 2. Cost-effective resource sharing

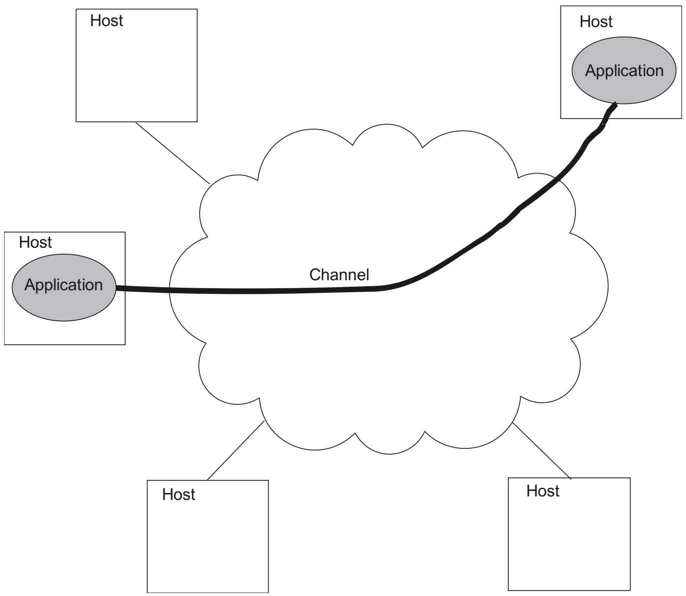
## Statistical Multiplexing

- time-division, but on demand rather than fixed (no waste)
- reschedule link on a per-packet basis
- packets from different sources interleaved on the link
- buffer packets that are contending for the link
- packet queue may be processed FIFO, but not necessarily
- buffer overflow, causing packet drop (loss), is called congestion



# 3. Process-to-process channels

Application programs running on hosts connected to network must be able to communicate in meaningful and efficient way




Bit

- propagates over links between src/dest

Packet

- sequence of bits
- 01011110101000000110…

Header Data

Network supports common process-to-process channels

- reliable: no loss, no errors, no duplication, in-order
  - for file access and digital libraries
- secure: privacy, authentication, message integrity
- delay-bounded: for real-time voice and video

# 3. End-to-end (e2e) channels

What goes wrong in the network?

- bit-level errors (electrical interference)
- packet-level errors (bit errors, congestion)
- link and node failures
- packets are delayed
- packets are delivered out-of-order
- third parties eavesdrop

The key problem is to fill in gap between what applications expect and what underlying technology provides

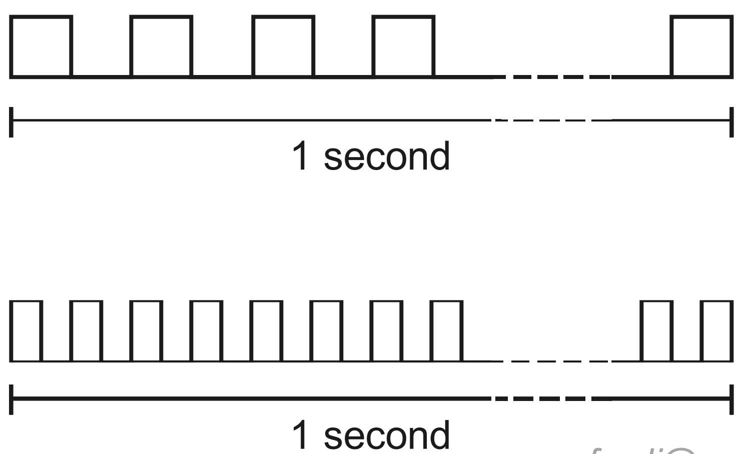
# 3. End-to-end (e2e) channels

## Performance: Bit Rate (throughput)

- amount of data that can be transmitted per time unit
  - example: 10 Mbps
  - link versus end-to-end

- notation
  - KB = $2^{10}$ bytes, MB = $2^{20}$ bytes
  - Kbps = $10^3$, Mbps = $10^6$ bits per second

- bit rate (aka capacity) related to ``bit width''




1Mbps
(each bit 1 microseconds wide)

2 Mbps
(each bit 0.5 microseconds wide)

# 3. End-to-end (e2e) channels

## Performance: Delay

- time it takes to send message from point A to point B
  - example: 24 milliseconds (ms)
  - sometimes interested in round-trip time (RTT)

- components of delay
  - Total Delay = **Processing** + **Propagation** + **Transmit** + **Queue**
  - **Propagation** Delay = Distance / SpeedOfLight
  - **Transmit** Delay = Size / Bit Rate

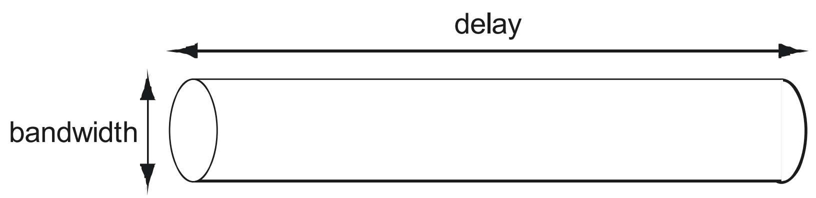
- speed of light
  - $3.0 \times 10^8$ meters/second in a vacuum
  - $2.3 \times 10^8$ meters/second in a cable
  - $2.0 \times 10^8$ meters/second in a fiber

# 3. End-to-end (e2e) channels

## Relative importance of bit rate and delay

- small message (e.g., 1 byte)
  - 1ms vs 100ms (**delay**) dominates 1Mbps vs 100Mbps (**bit rate**)
- large message (e.g., 25 MB)
  - 1Mbps vs 100Mbps (**bit rate**) dominates 1ms vs 100ms (**delay**)

## Bandwidth (Bit Rate/thruput) x Delay Product (BxD)




## Example

- 100ms RTT and 45Mbps Bit Rate = 4,500,000 bits ~ 550 KB of data

# Where do we go from here?

"The secret of getting ahead is getting started. The secret to getting started is breaking your complex overwhelming tasks into  small manageable tasks and then starting on the first one.”

--Mark Twain

## Internet

# PROTOCOL STACK

# Many, many things happening in a network

Networks are complex,
with many pieces

- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of organizing structure and managing a network?

# Standards

By having computers comply to the same standards, they can ``interoperate'' even if they are of different type or connected to different types of networks

Standards organizations

- In Europe:
  - ITU-T (formerly CCITT), e.g. publications X.25, V.24, etc.
    - X-series define how to connect a host to PSDN (Data)
    - V-series define how to connect a host to PSTN (Telephone)
    - I-series define how to connect a host to ISDN (Integrated)
  - ISO, developed OSI architecture
- In US: IETF, EIA, IEEE, ANSI, NIST, …
  - IETF RFCs define Internet standards
  - IEEE 802 define standards for links, e.g. Ethernet, WiFi

# How to determine what to send and when?

Protocols define format, type, order of messages sent and received among network entities, and actions taken on message transmission, receipt

Human protocols
- "What's the time?"
- "I have a question"
- introductions

Network protocols
- machines rather than humans
- all communication activity in Internet governed by protocols

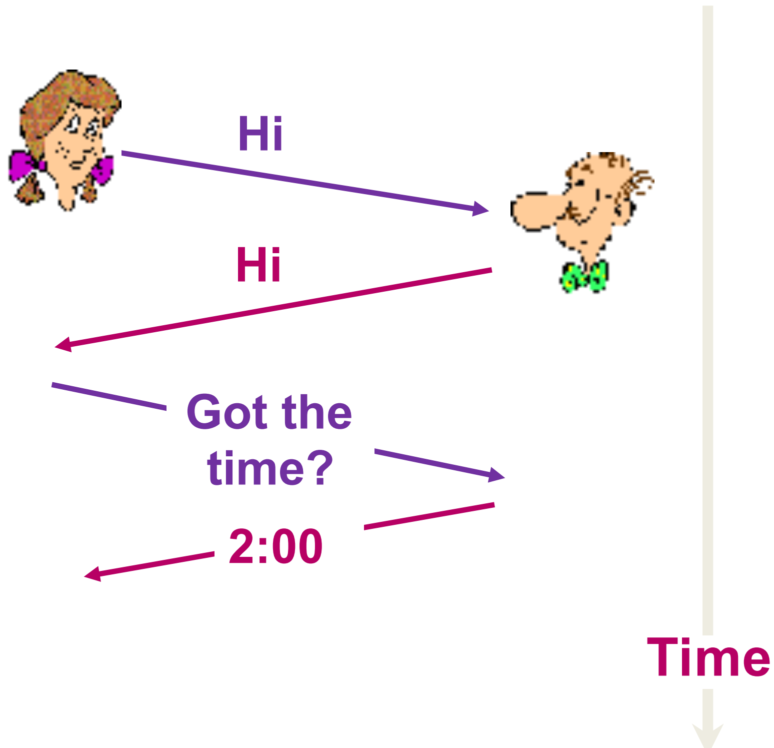
… specific messages sent

… actions taken when messages received, or other events

Many protocols used on Internet: TCP, IP, TLS, HTTP, …
- RFCs specify implementation standards for non-proprietary protocols

# Protocol example

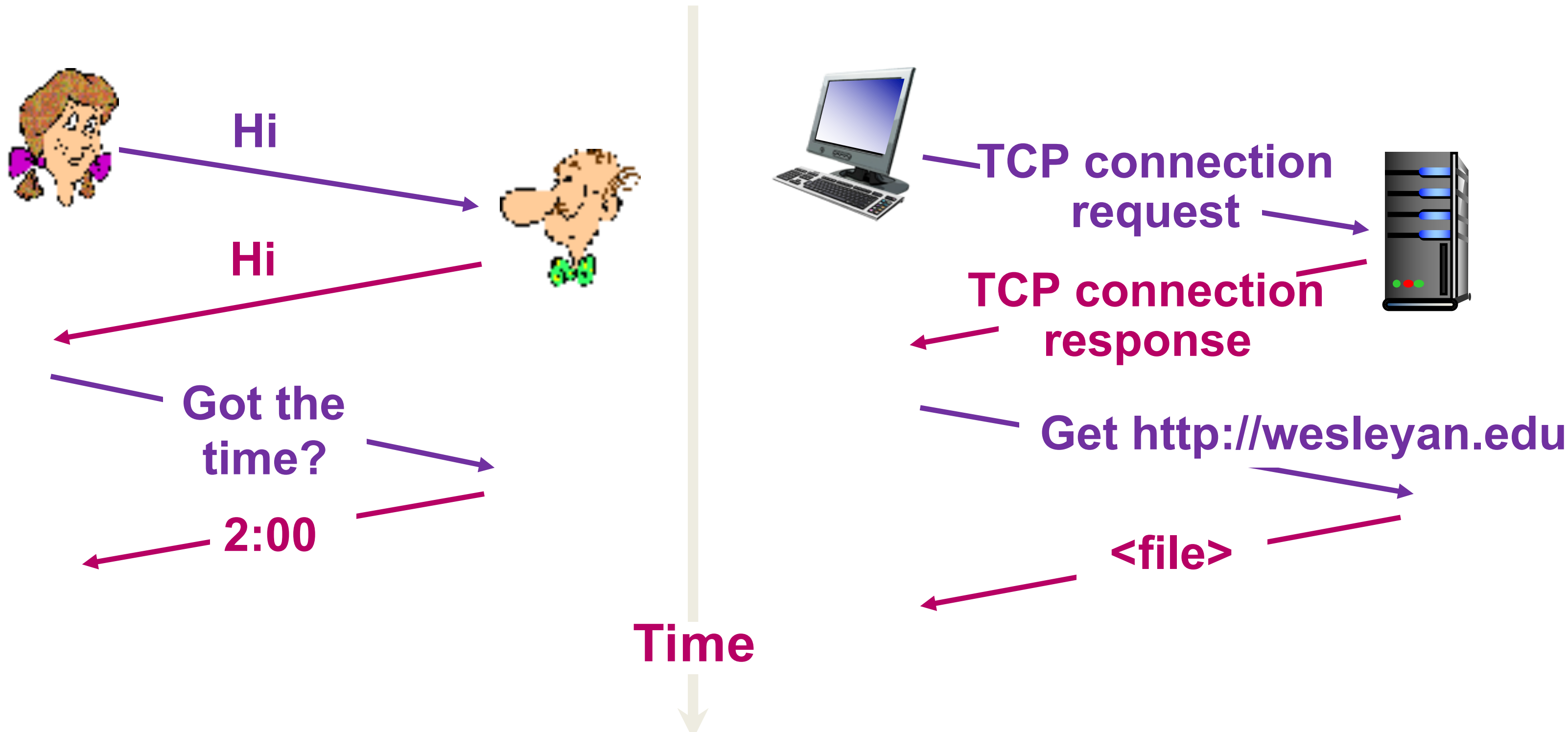
A human protocol and a computer network protocol:




Q: other human protocols?

# Protocol example

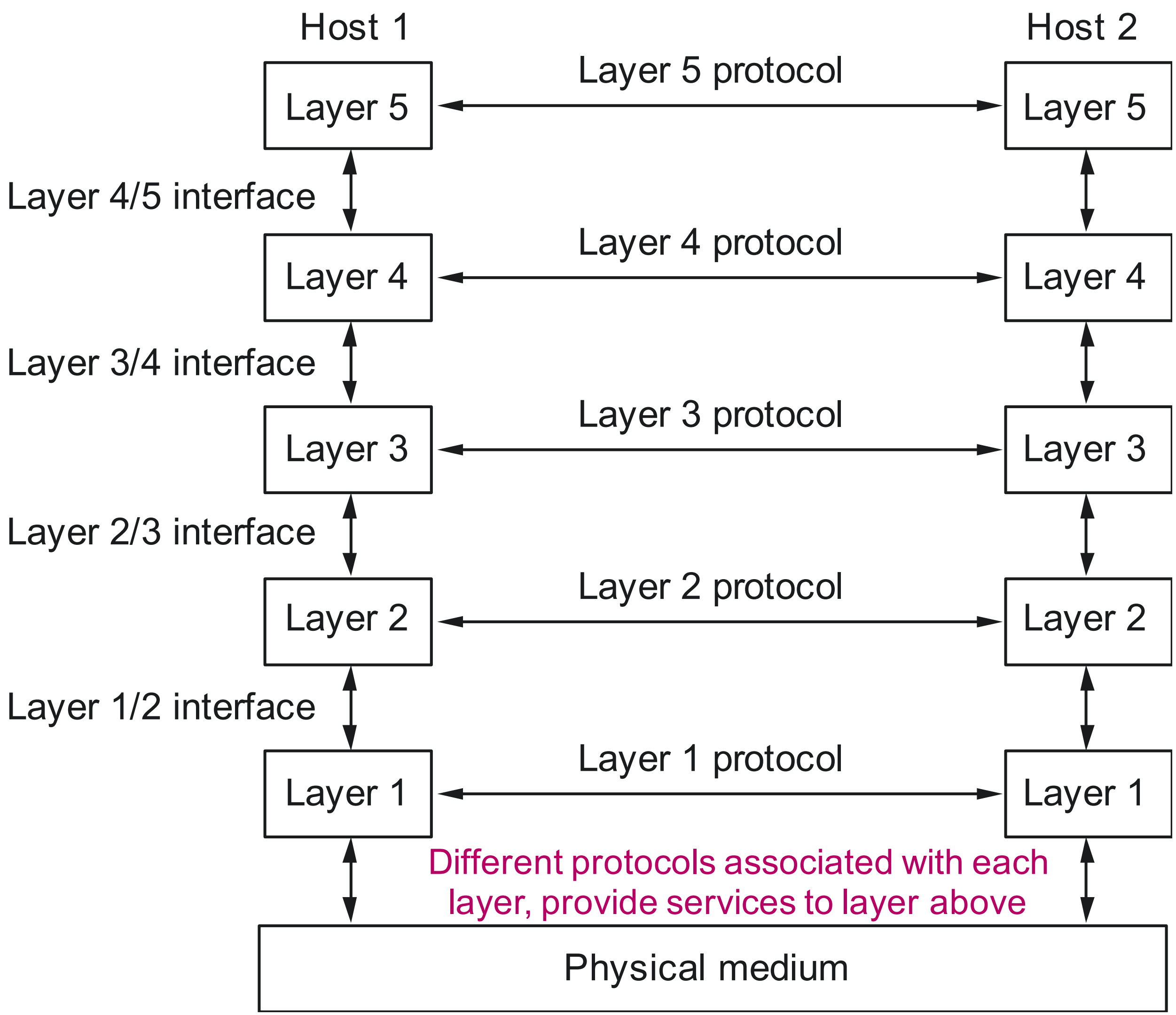
A human protocol and a computer network protocol:




Q: other human protocols?

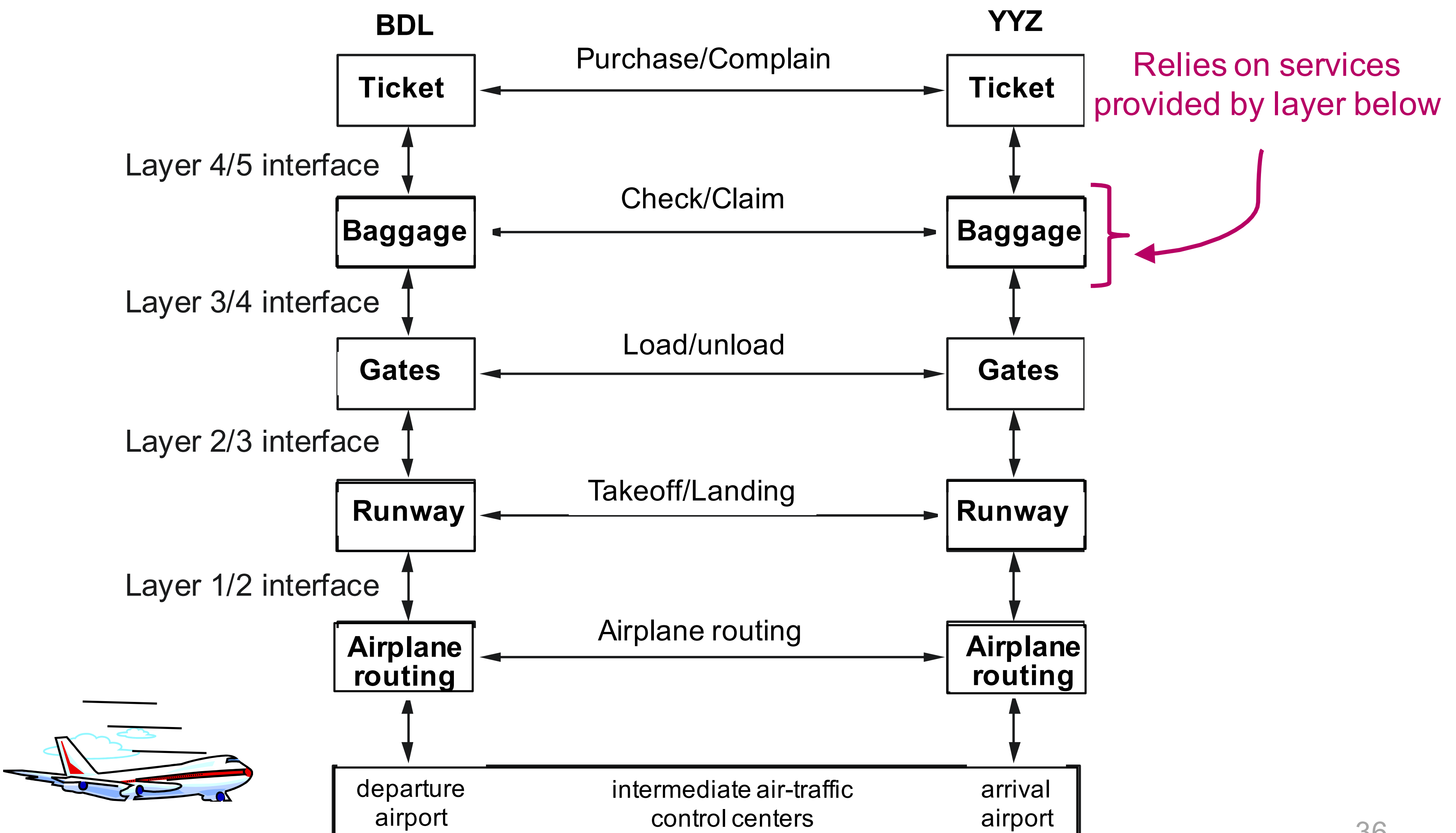
# Layered network architecture

Based on divide-and-conquer concept



# Layering of airline functionality

## Each layer implements a service via its own internal actions



# Why layering?

## Pros

- identifies and captures how parts of system relate
- information hiding
  - hide info in one part of system from another
  - higher layer shielded from how lower layer implemented
- modularity
  - easy to change implementation of service provided by layer
    - e.g., change in gate procedure doesn't affect rest of system
  - as long as layer still provides some services to higher layer, higher layers can stay unchanged

## Cons

- duplicate functionality
  - higher layer may duplicate functionality in lower layer
  - e.g., error checking; link by link, end to end
- one layer may need info from another layer
- no cross-layer optimization

# ISO/OSI Model

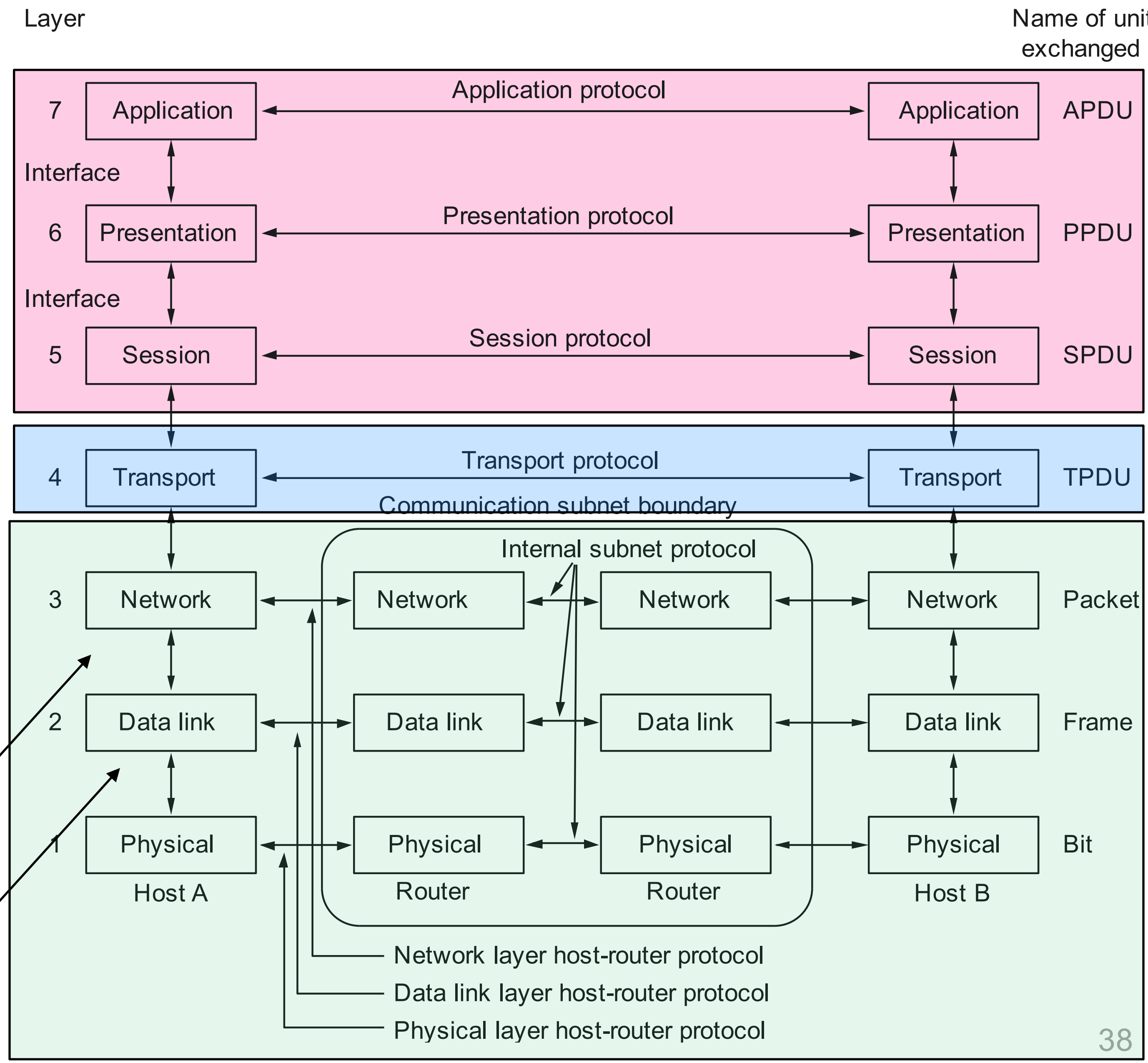
Application-oriented layers

Interface between 5-7 and 1-3

Network-dependent layers

Internetwork sublayer to connect networks

MAC sublayer to handle multi-access (shared) links



# ISO/OSI Model

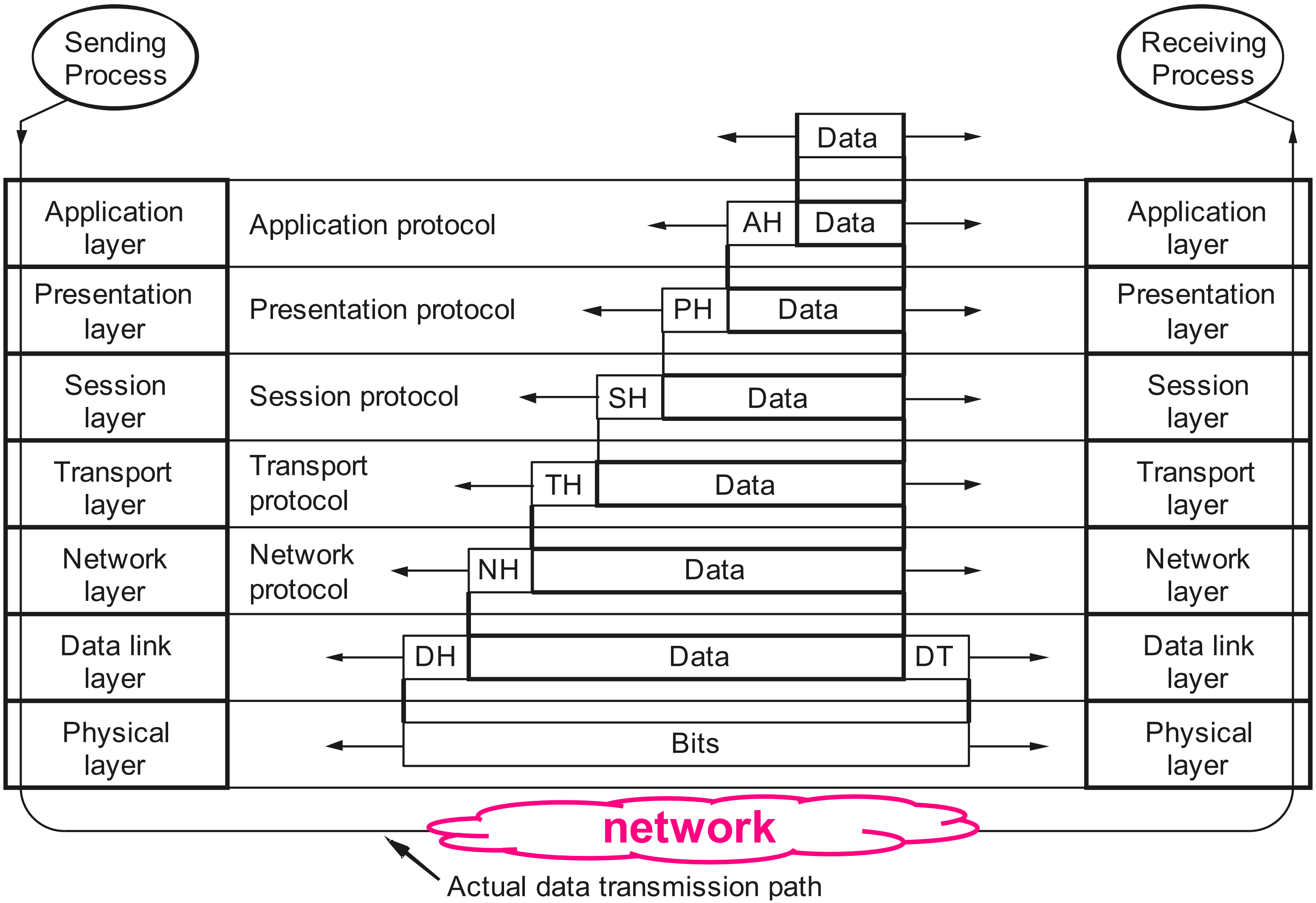
Seven layers with following typical functions

- **application**: user interface
- **presentation**:
  - allow applications to interpret meaning of data
  - e.g., encryption, compression, machine-specific conventions
- **session**: synchronization, check-pointing, recovery of data exchange

Internet protocol stack is "missing" these layers. These services, if needed, must be implemented in application. Needed?

- **transport**: multiplexing/demultiplexing, fragmentation/reassembly, end-to-end flow, congestion and error control
- **network**: addressing and routing
- **data link**: link flow and error control
- **physical:** physical and electrical interfaces (normally 100% hardware)

# How is the ISO/OSI Model used?




vumanfredi@wesleyan.edu

# Internet organization

## Internet protocol stack

- each layer of stack has certain protocols associated with it
- different services are provided by different protocols

## Protocol

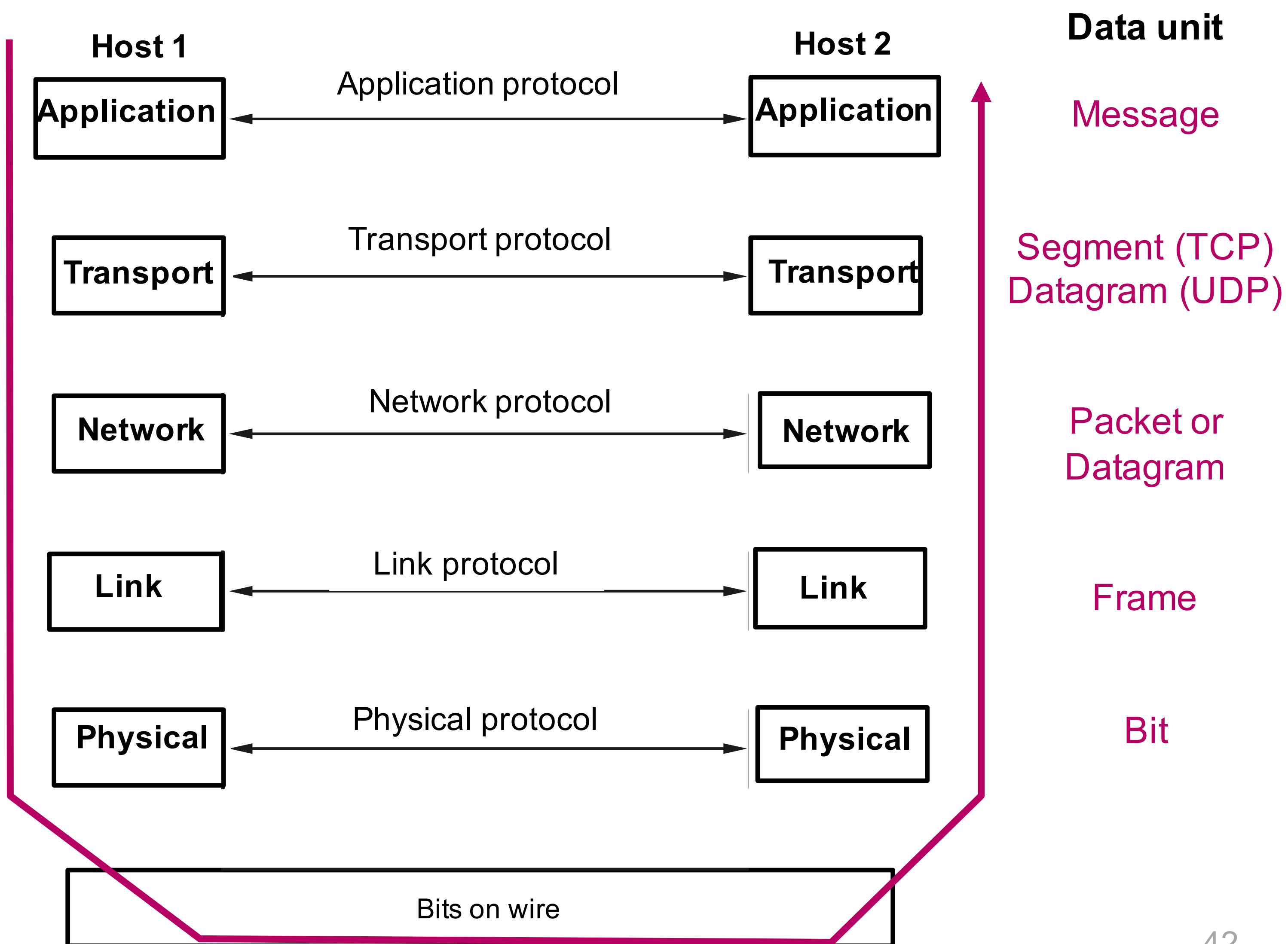
- dictates what behavior is acceptable
- defines format and order of messages exchanged between 2 or more communicating hosts/routers/entities
- defines actions taken on msg transmission/reception or other event

## Example human protocols

- asking for time
- answering or asking a question in class
- introductions

# Internet protocol stack

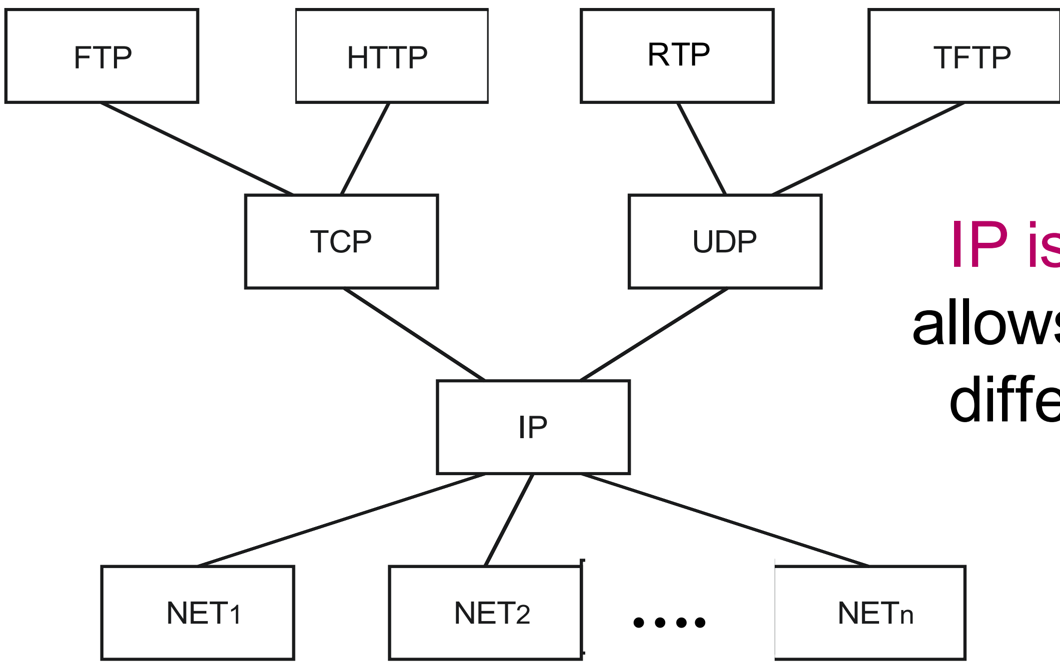
Each layer implements a service via its own internal actions



# Internet protocol stack summary

| | Layer | Service provided to upper layer | Protocols | Unit of information |
|---|---|---|---|---|
| 5 | Application | • Support network applications | FTP, DNS, SMTP, HTTP | Message<br>1 message may be split into multiple segments |
| 4 | Transport | • Deliver messages to app endpoints<br>• Flow control<br>• Reliability | TCP (reliable)<br>UDP (best-effort) | Segment (TCP)<br>Datagram (UDP)<br>1 segment may be split into multiple packets |
| 3 | Network | • Route segments from source to destination host | IP (best-effort)<br>Routing protocols | Packet (TCP)<br>Datagram (UDP) |
| 2 | Link | • Move packet over link from one host to next host | Ethernet, 802.11 | Frame<br>MTU is 1500 bytes |
| 1 | Physical | • Move individual bits in frame from one host to next<br>• "bits on wire" | Ethernet phy<br>802.11 phy<br>Bluetooth phy<br>DSL | Bit |

# Protocol Graph for the Internet




IP is narrow waist of Internet: allows interconnectivity of many different kinds of networks as long as they use IP