# Lecture 2: Internet Structure

COMP 332, Spring 2024
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. Announcements
   - Homework 1 posted, due Wednesday, Feb 7 by 11:59p
   - Help sessions: Sun/Mon/Tues, 7-9p in Exley 638. Also virtually

2. Building a network
   - Protocols
   - Layering
   - Key services

3. Internet organization
   - Edge
     - How you connect to Internet
   - Core
     - How your packets get to their destination
     - Circuit-switching vs. packet-switching:

## Building a Network

# PROTOCOLS

vumanfredi@wesleyan.edu

# Many, many things happening in a network

Networks are complex,
with many pieces

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

Questions:

How can we possibly organize and manage a network?

Some approaches: standards, protocols, and layering

# Standards

## If computers comply to same standards

- can interoperate even if computers are of different types or connected to different types of networks

## Standards organizations

- 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
- US: IETF, EIA, IEEE, ANSI, NIST, …
  - IETF RFCs define Internet standards for non-proprietary protocols
  - IEEE 802 define standards for links, e.g. Ethernet, WiFi

# Standards

https://www.ietf.org/rfc/rfc2616

ietf.org/rfc/rfc2616.txt

```
Network Working Group                                        R. Fielding
Request for Comments: 2616                                     UC Irvine
Obsoletes: 2068                                               J. Gettys
Category: Standards Track                                    Compaq/W3C
                                                               J. Mogul
                                                                 Compaq
                                                             H. Frystyk
                                                                W3C/MIT
                                                            L. Masinter
                                                                  Xerox
                                                               P. Leach
                                                              Microsoft
                                                         T. Berners-Lee
                                                                W3C/MIT
                                                              June 1999


               Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements.  Please refer to the current edition of the "Internet
```

# How do devices decide 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

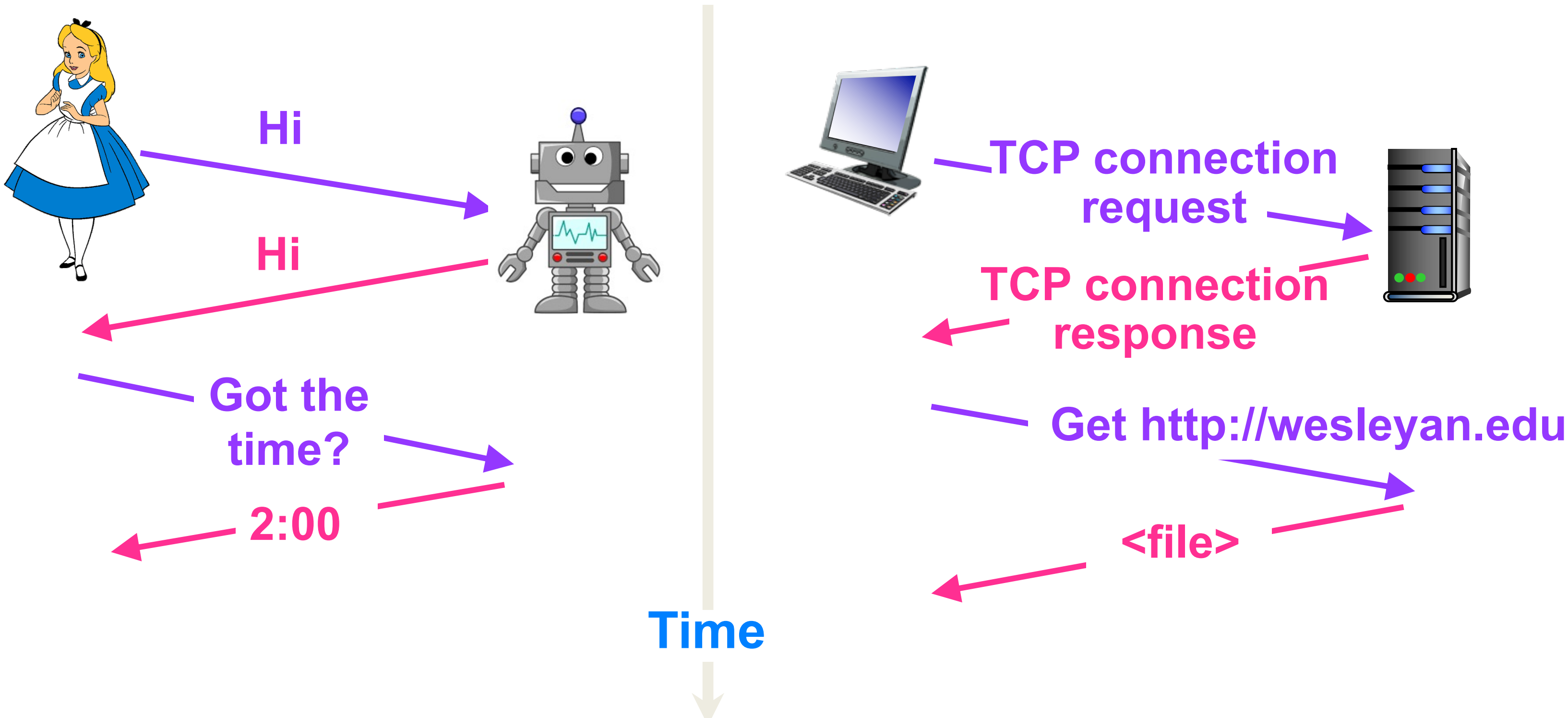
Network protocols
- machines rather than humans
- all communication activity in Internet governed by protocols

… specific messages sent

… actions taken when messages received, or other events

# Protocol example

A human protocol and a computer network protocol:




Q: other human protocols?

Many protocols used on Internet: TCP, IP, TLS, HTTP, …

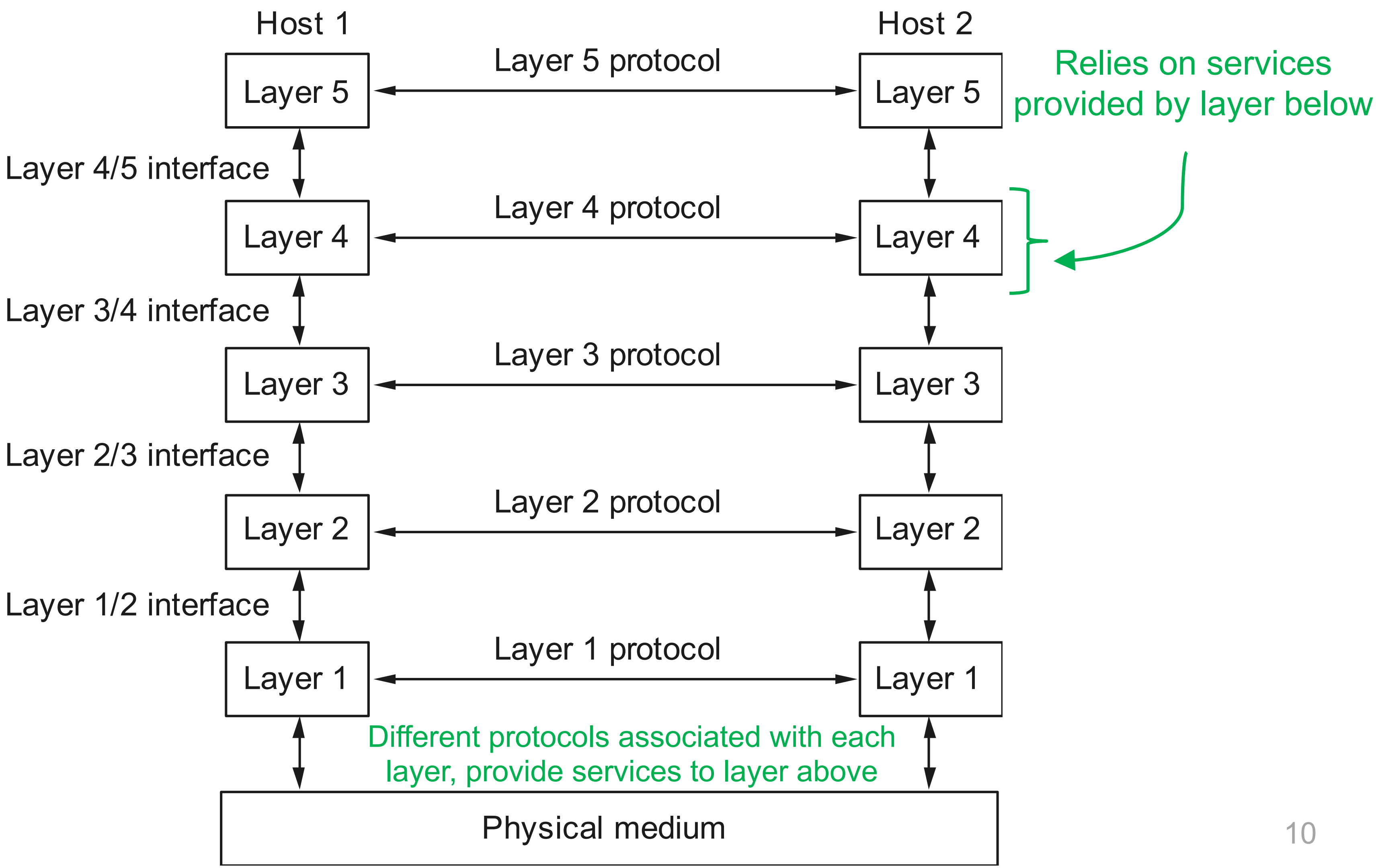
## Building a Network

# LAYERING

vumanfredi@wesleyan.edu

# Layered network architecture

Each layer of stack has certain protocols associated with it.
Different protocols provide different services



# 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
  - as long as layer still provides same 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

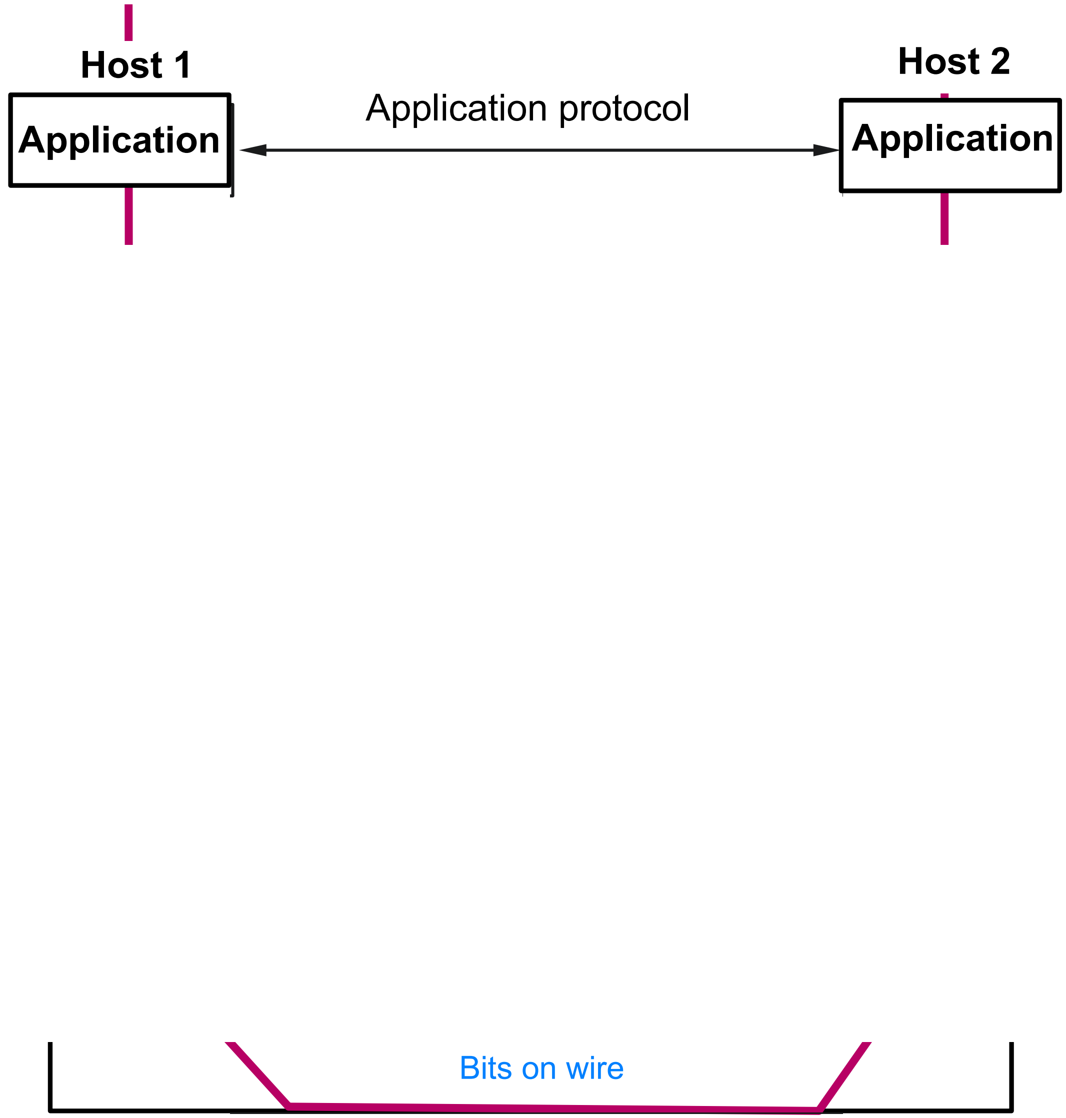
# 5-layer Internet protocol stack

**Mail service example**

- Person reads letter

**Service provided**

- Support network applications



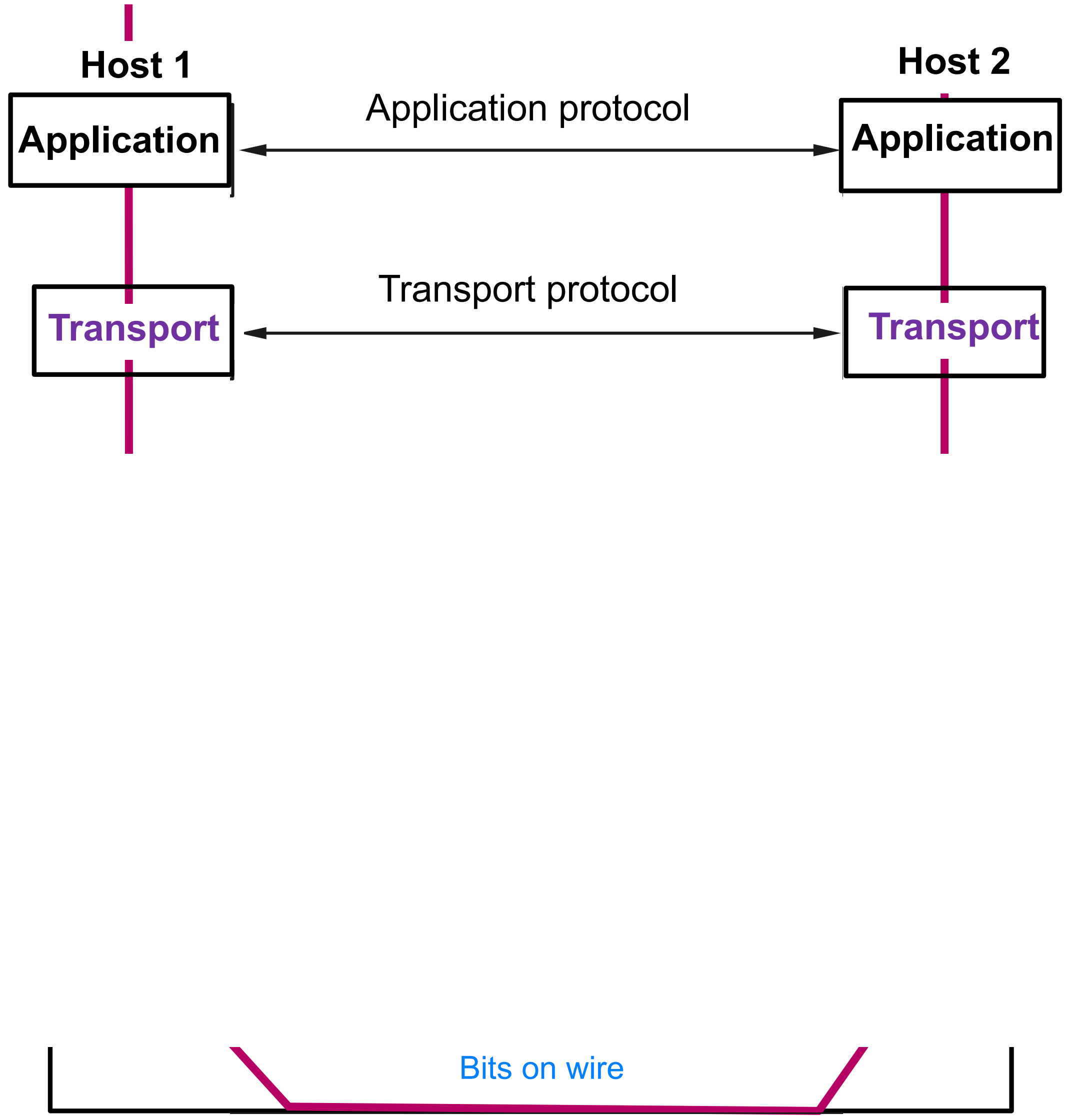
# 5-layer Internet protocol stack

**Mail service example**

- Person reads letter
- Letter delivered to person in house

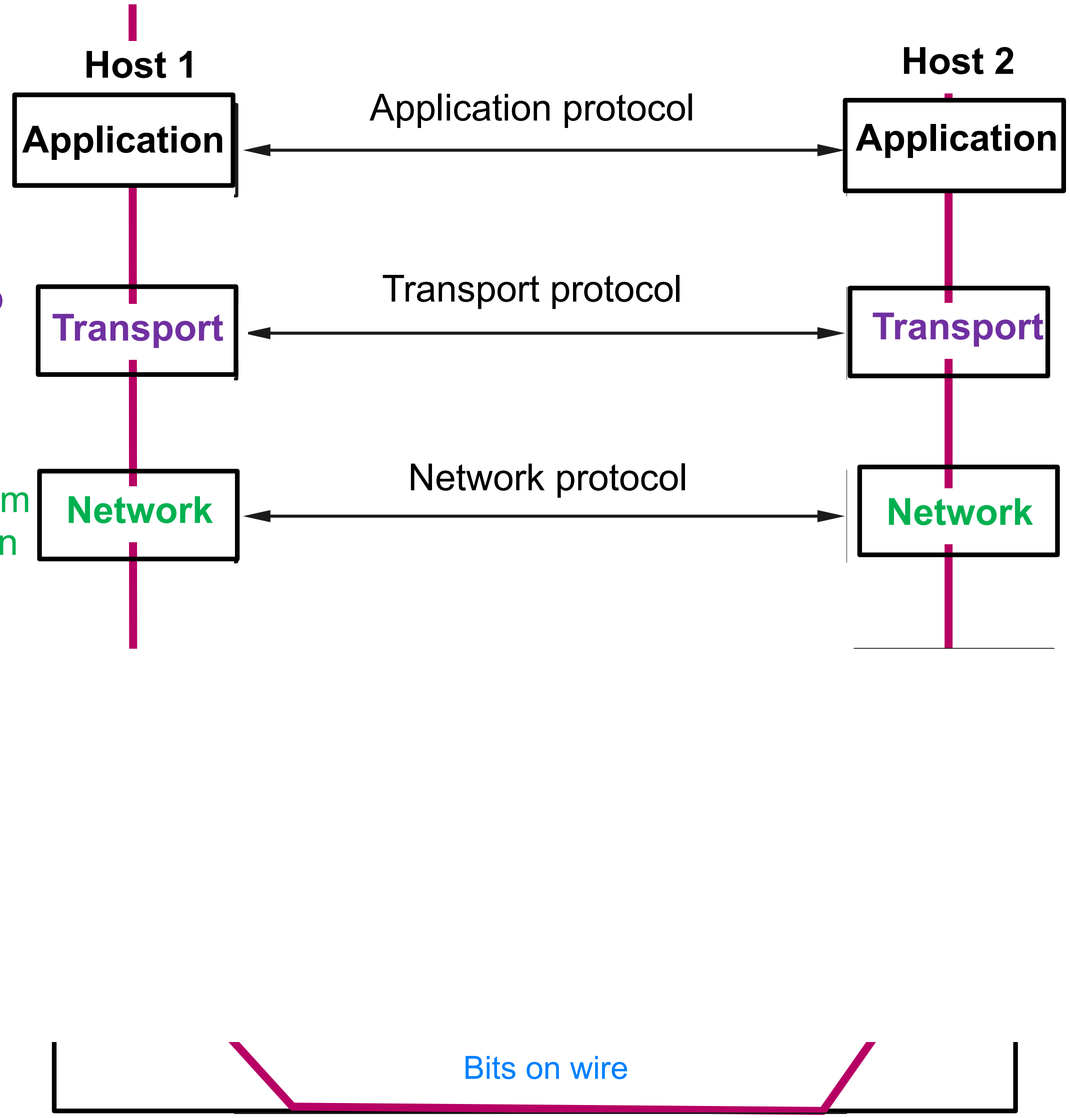
**Service provided**

- Support network applications
- Deliver msgs to app endpoints, flow control, reliability



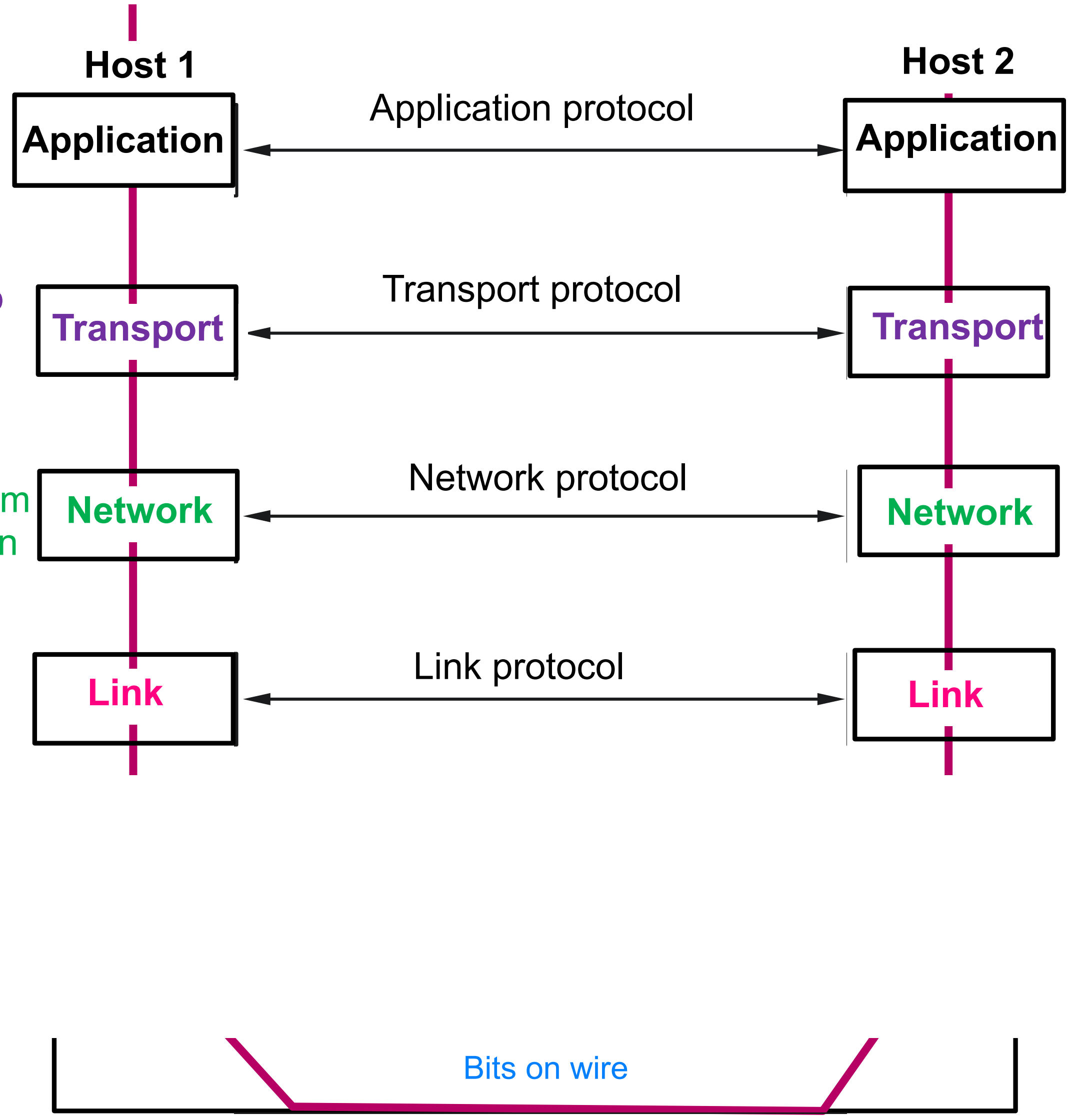
# 5-layer Internet protocol stack

| Mail service example | Service provided |
| --- | --- |
| Person reads letter | Support network applications |
| Letter delivered to person in house | Deliver msgs to app endpoints, flow control, reliability |
| Letter conveyed between sending house and receiving house | Route segments from source to destination host (indirect) |



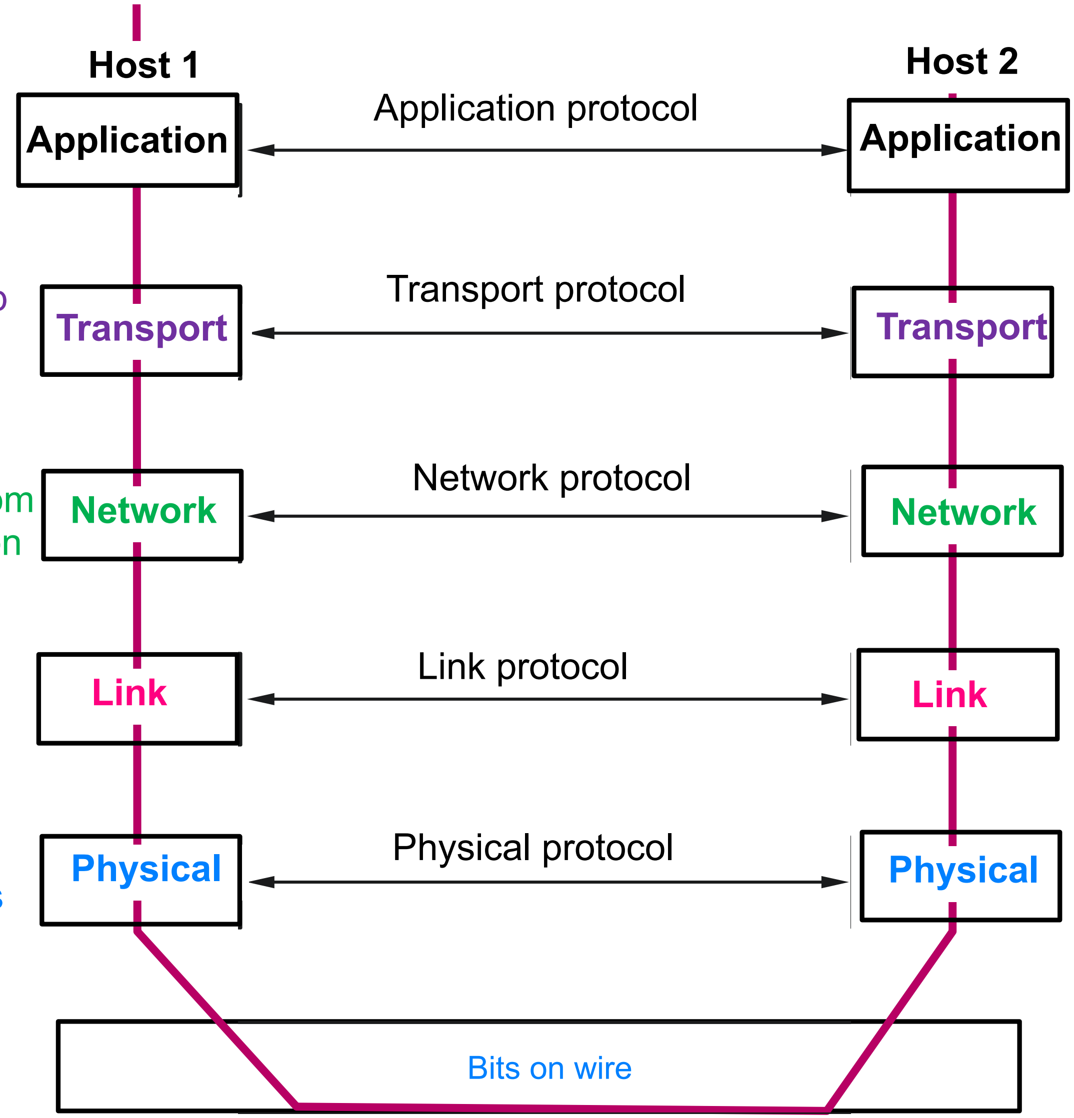
# 5-layer Internet protocol stack

| Mail service example | Service provided |
|---|---|
| • Person reads letter | • Support network applications |
| • Letter delivered to person in house | • Deliver msgs to app endpoints, flow control, reliability |
| • Letter conveyed between sending house and receiving house | • Route segments from source to destination host (indirect) |
| • Letter conveyed from house to mailbox/P.O or from one P.O directly to another P.O. | • Move pkt over link from one host to next host (direct) |



# 5-layer Internet protocol stack

| **Mail service example** | **Service provided** |
| --- | --- |
| • Person reads letter | • Support network applications |
| • Letter delivered to person in house | • Deliver msgs to app endpoints, flow control, reliability |
| • Letter conveyed between sending house and receiving house | • Route segments from source to destination host (indirect) |
| • Letter conveyed from house to mailbox/P.O or from one P.O directly to another P.O. | • Move pkt over link from one host to next host (direct) |
| • The actual carrying of the letter across a “link” | • Move individual bits in frame from one host to next |



# Protocols and units of information

| | 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** | | | |
| 3 | **Network** | | | |
| 2 | **Link** | | | |
| 1 | **Physical** | | | |

# Protocols and units of information

| | 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** | | | |
| 2 | **Link** | | | |
| 1 | **Physical** | | | |

# Protocols and units of information

| | 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** | | | |
| 1 | **Physical** | | | |

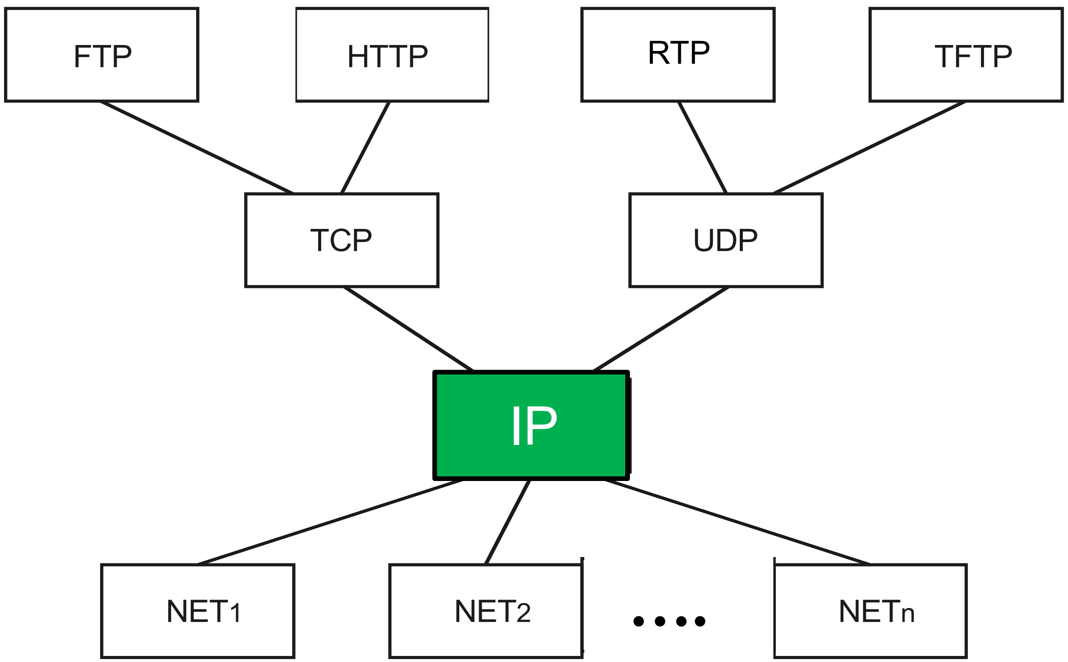
# Protocols and units of information

| | 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** | | | |

# Protocols and units of information

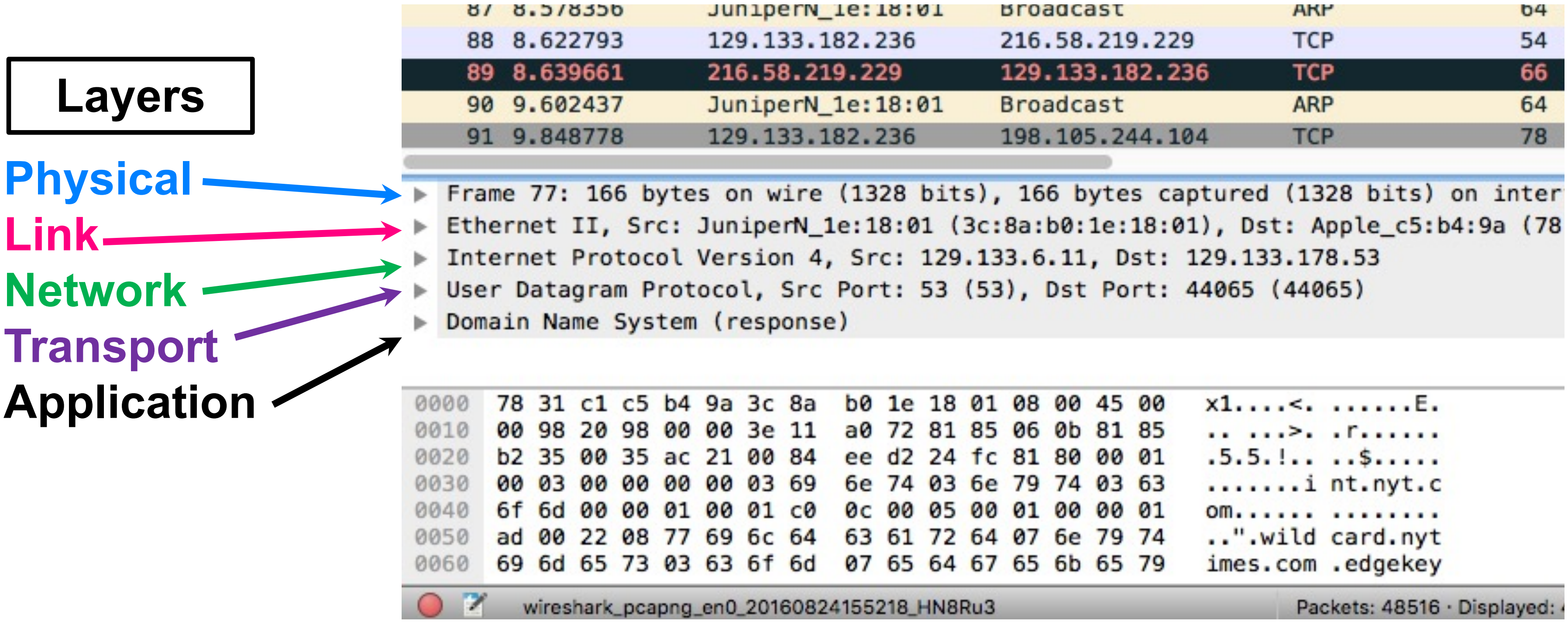
| | Layer | Service provided to upper layer | Protocols | Unit of information |
|---|---|---|---|---|
| 5 | Application | • Support network applications | FTP, DNS, SMTP, HTTP | **Message**<br>1 message may be split into multiple segments |
| 4 | **Transport** | • Deliver messages to app endpoints<br>• 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 Internet




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

# Looking at protocol stack in Wireshark




We'll talk in depth about Wireshark and how to use next week

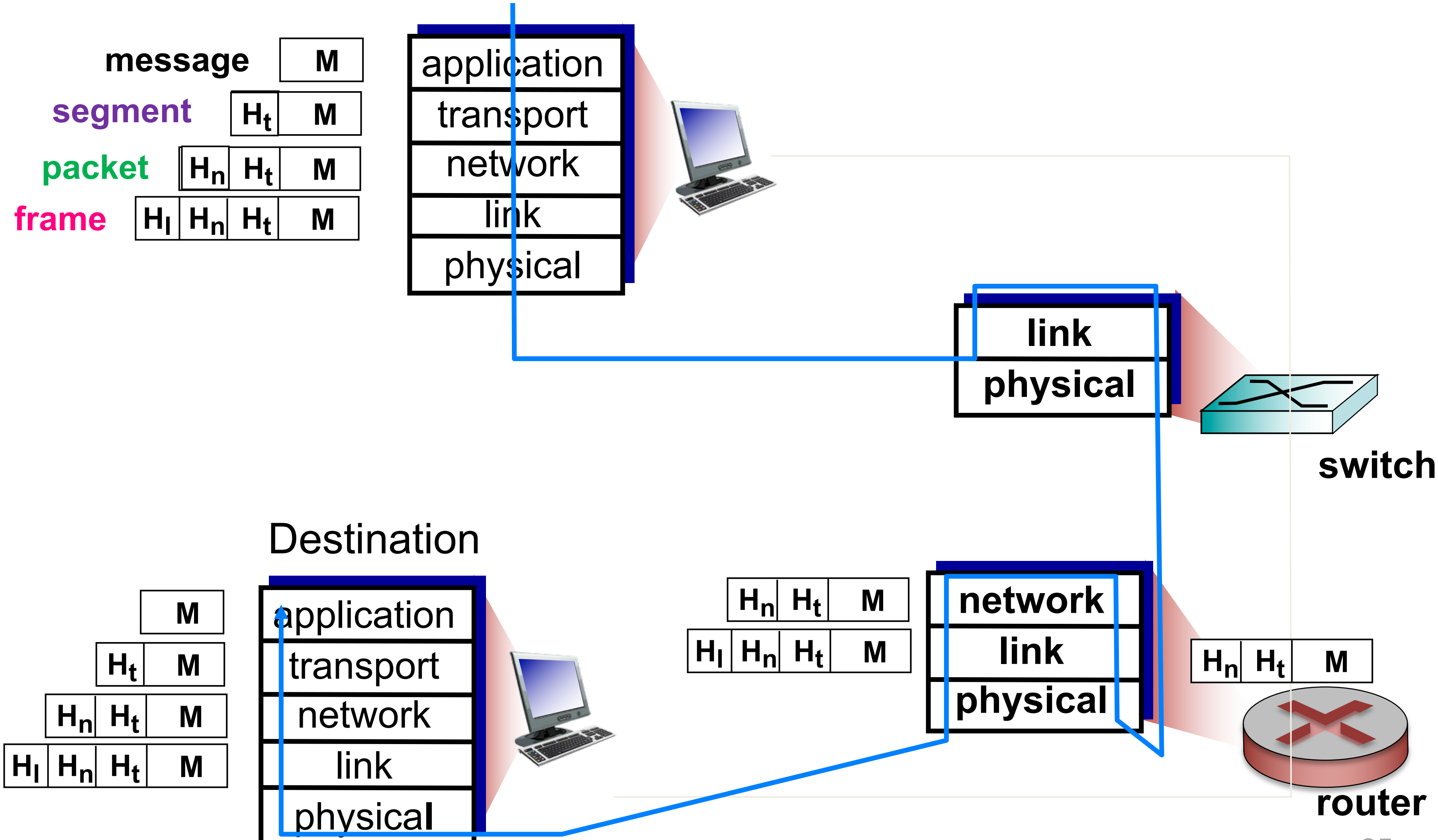
Please download and install Wireshark for class next week!

## Building a Network

# KEY SERVICES

# Encapsulation/Decapsulation

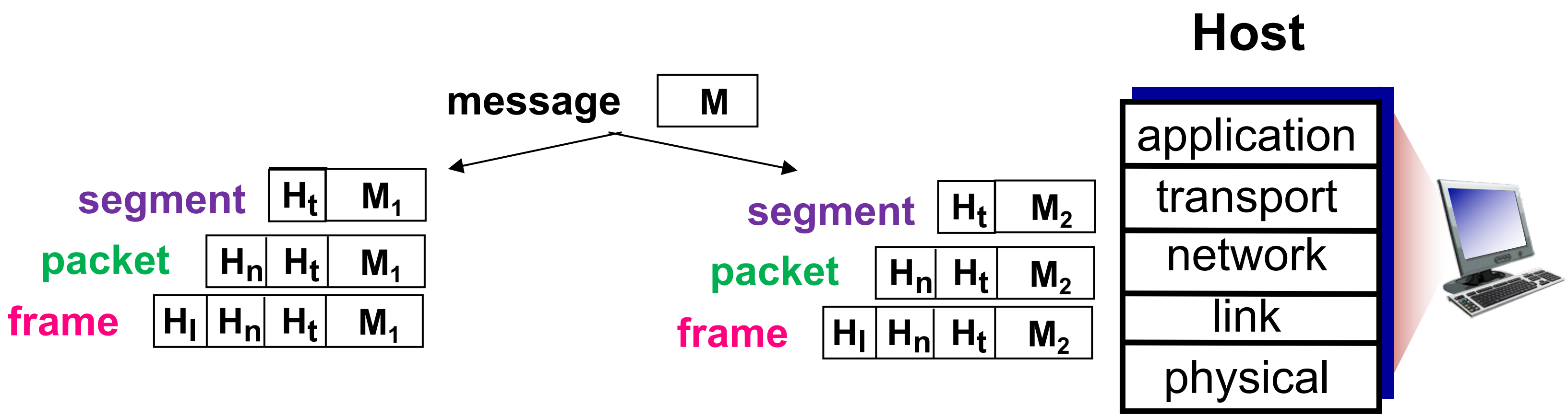
Headers must be added/removed from data unit at each layer



# Fragmentation/Assembly

If data unit too large for layer below, must fragment/assemble

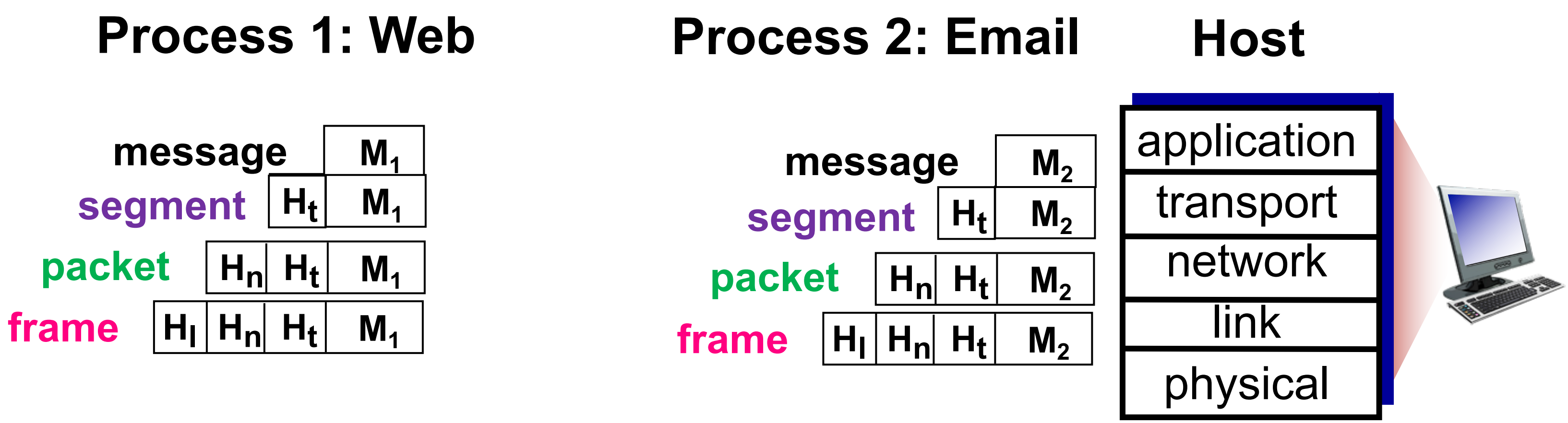
**Why fragment?** Max size of Ethernet frame is specified to be 1522 bytes




Need additional book-keeping to keep track of which **segments** belong to which **message**

# Multiplexing/Demultiplexing

Many processes sending network traffic simultaneously on host, many hosts sharing network




Need additional book-keeping to keep track of which **segments** belong to which **process** on host

# Internet Organization

# OVERVIEW

vumanfredi@wesleyan.edu

# How is the Internet organized?

## Billions of connected hosts

– run network applications


## Communication links

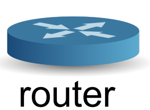
– carry info between apps on hosts

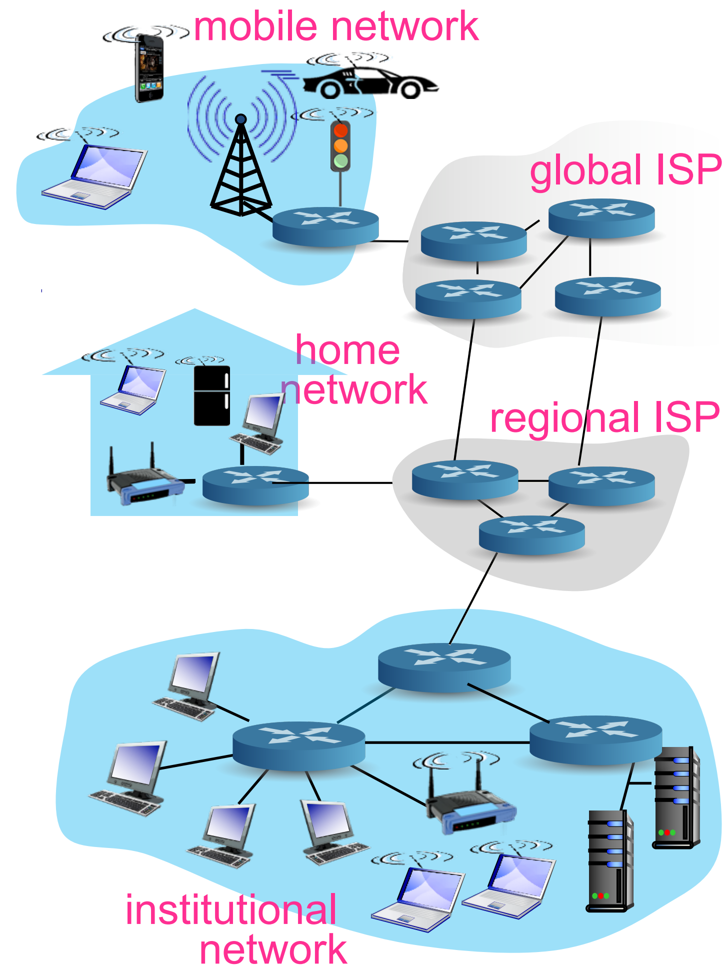
– fiber, copper, radio, satellite

– transmission rate: data per second


## Routers (like post offices)

– forward packets (like letters)




ISP: Internet Service Provider

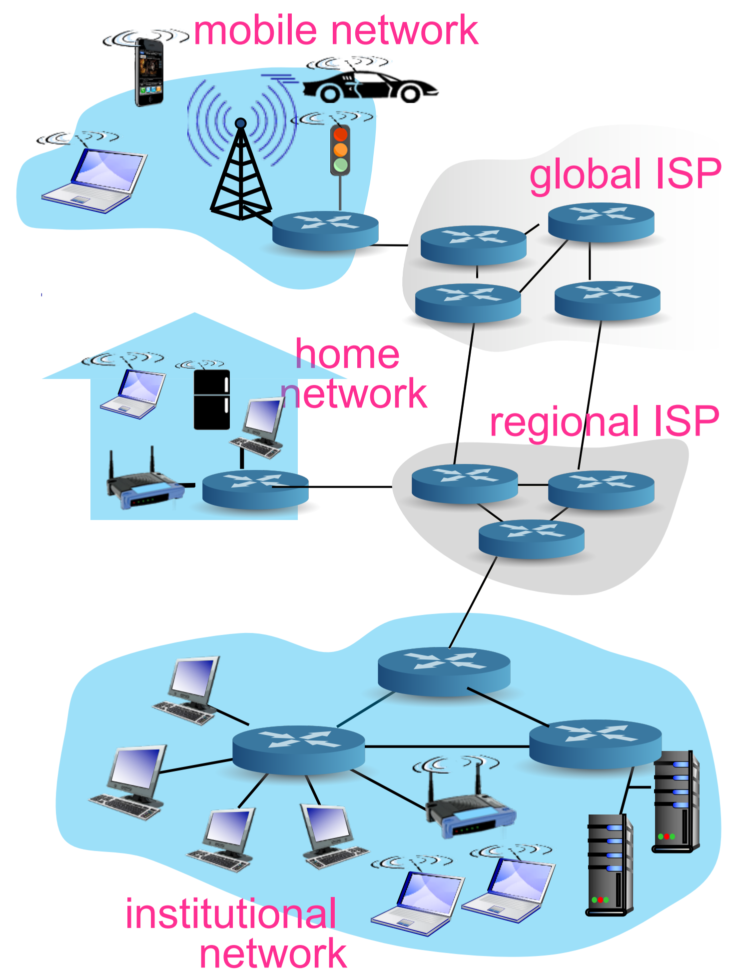
# Digging deeper

## Network edge

- hosts: clients and servers
- servers often in data centers

## Network core

- interconnected routers
- network of networks




**Internet is network of networks: i.e., internetwork**

Every device must implement IP (Internet Protocol) and have IP address

ISP: Internet Service Provider

# Internet provides services

## Services to applications

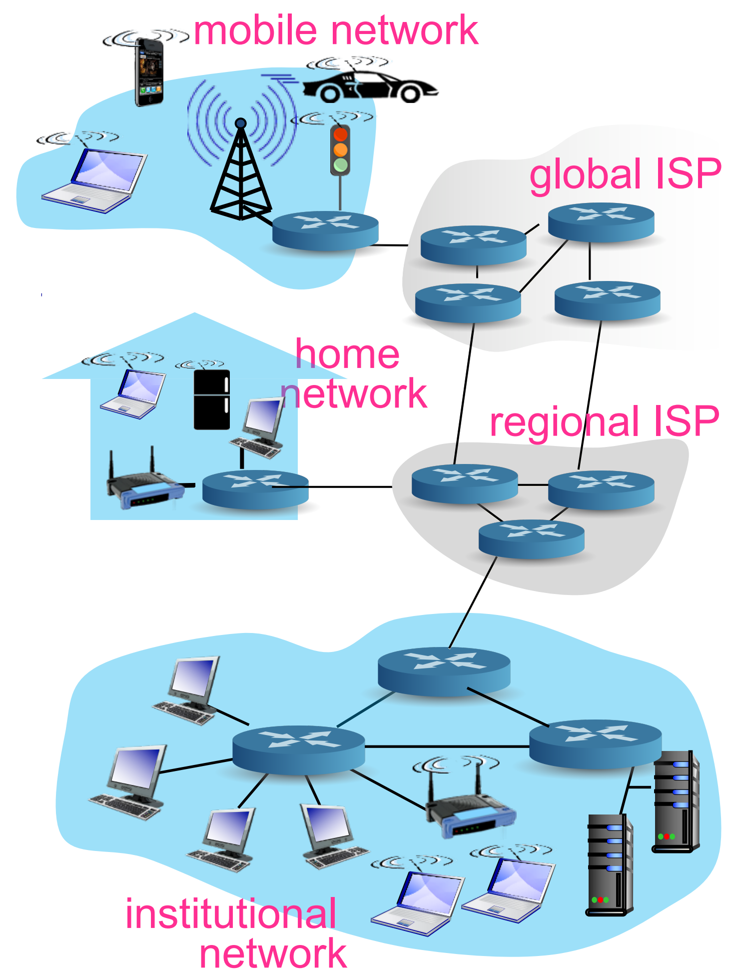
- e.g., web, VoIP, email, games, ecommerce, social networks

## Programming interface to apps

- **hooks**
  - for sending and receiving app programs to connect to Internet
- **service options**
  - analogous to postal service

## Protocols

- control message sending, receiving




ISP: Internet Service Provider

# Where to place functionality in Internet?

Option 1

- inside network (switches/routers)

Option 2

- at edges (hosts)

Illustrates end-end principle

- some functionality can only be correctly implemented at end-hosts
  - e.g., file transfer
    - should each link check or end hosts check for loss/errors?
    - what if a link on path fails?

# Internet Organization

# EDGE

# How do you connect to Internet?

## Hosts connect to edge router

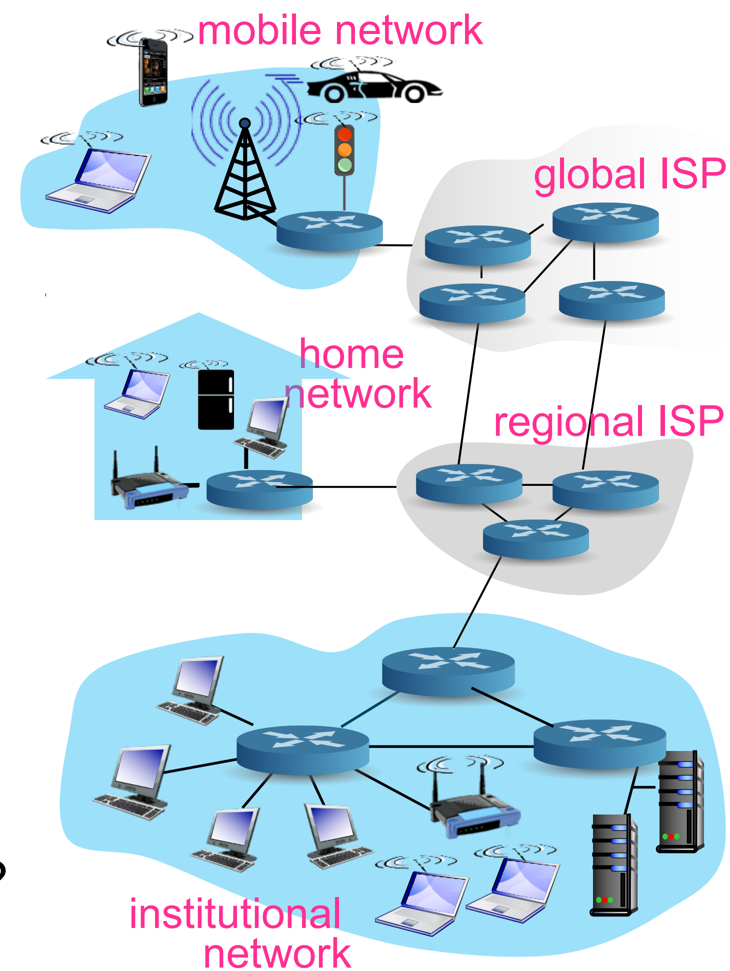
– access network/ISP

## Access networks

– **residential**
  • DSL (telephone), cable,
– **institutional**
  • school, company
– **mobile**

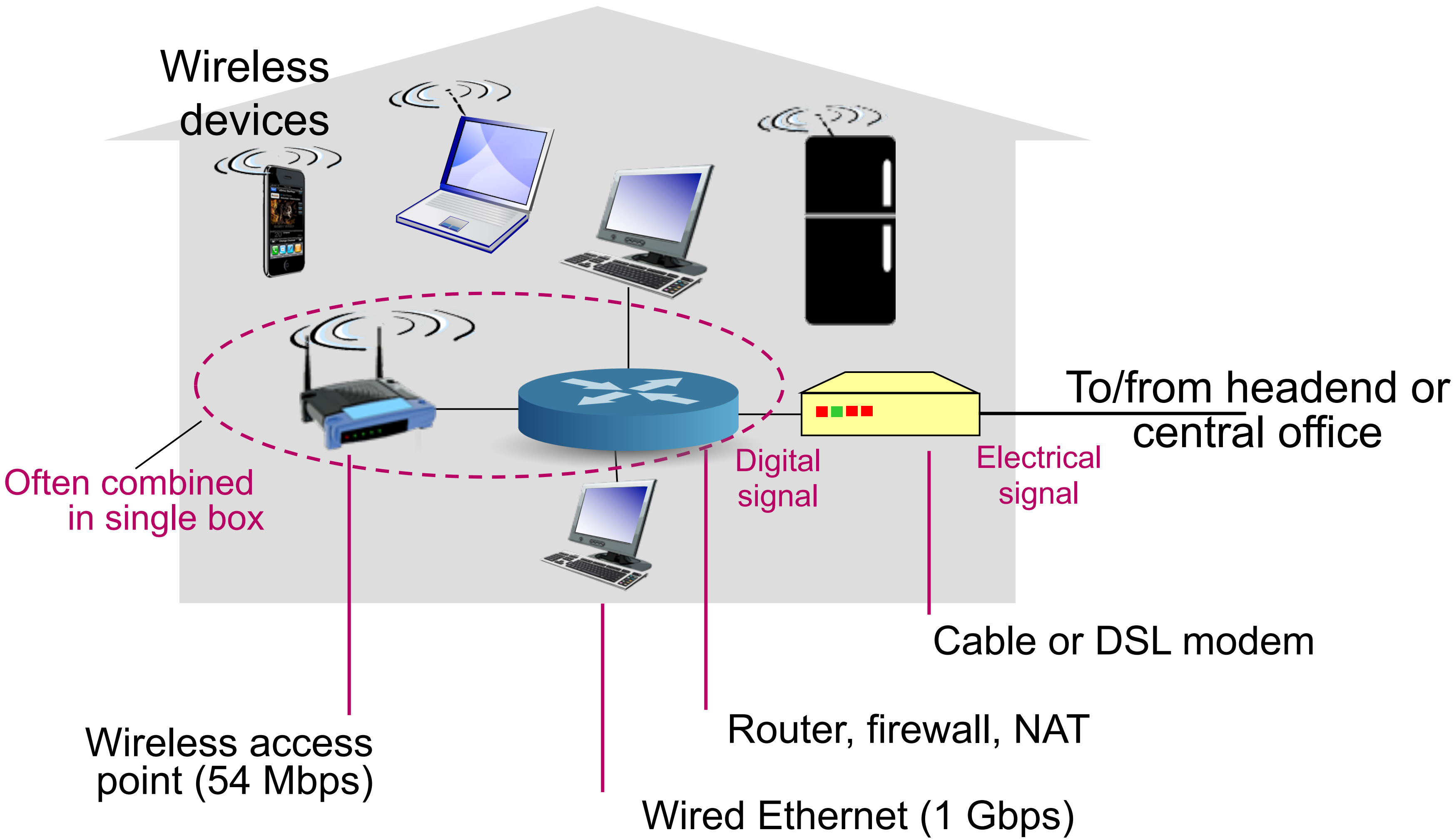
## Issues

– bandwidth (bps) of access network?
– shared or dedicated?




ISP: Internet Service Provider

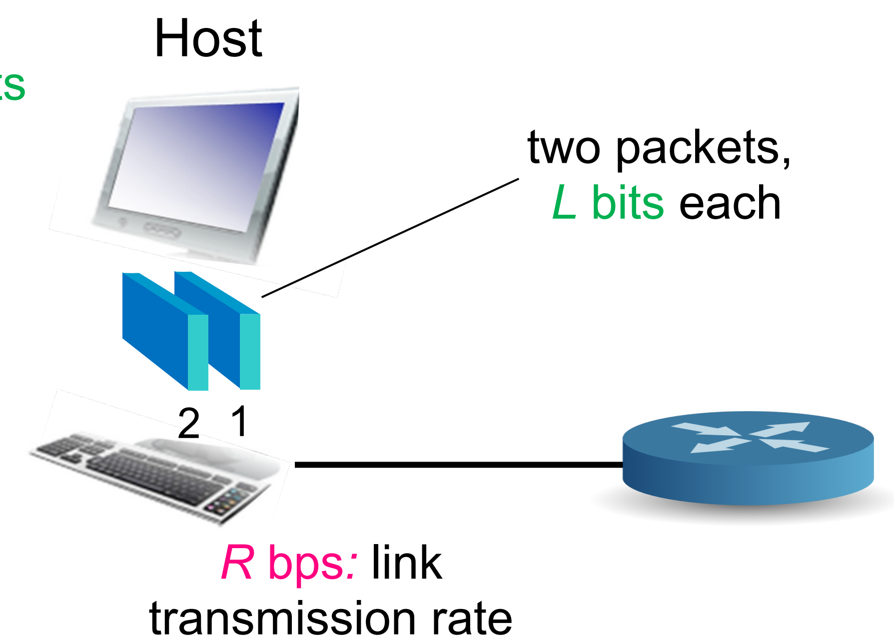
# Home access network




Delay of getting bits into Internet?

# Host sends packets into access network

1. Given application message
   - breaks into packets
     - smaller chunks of length *L* bits
2. Transmit packets into access network
   - at transmission rate *R*
     - aka link capacity
     - aka link bandwidth




| Transmission delay | = | Time to transmit *L*-bit packet into link | = | $\frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$ |
|---|---|---|---|---|

# Quantifying channel performance

## Delay

- time to send packet from host A to host B
  - Total Delay = **Processing** + **Transmission** + **Queu**e + **Propagation**
  - **Transmission** Delay = Packet length / Bit Rate
  - **Propagation** Delay = Distance / SpeedOfLight

- 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

Q: How to compute delay accrued as packet travels through Internet?

# Internet Organization

# CORE

vumanfredi@wesleyan.edu

# How to move data through Internet core?

Internet core

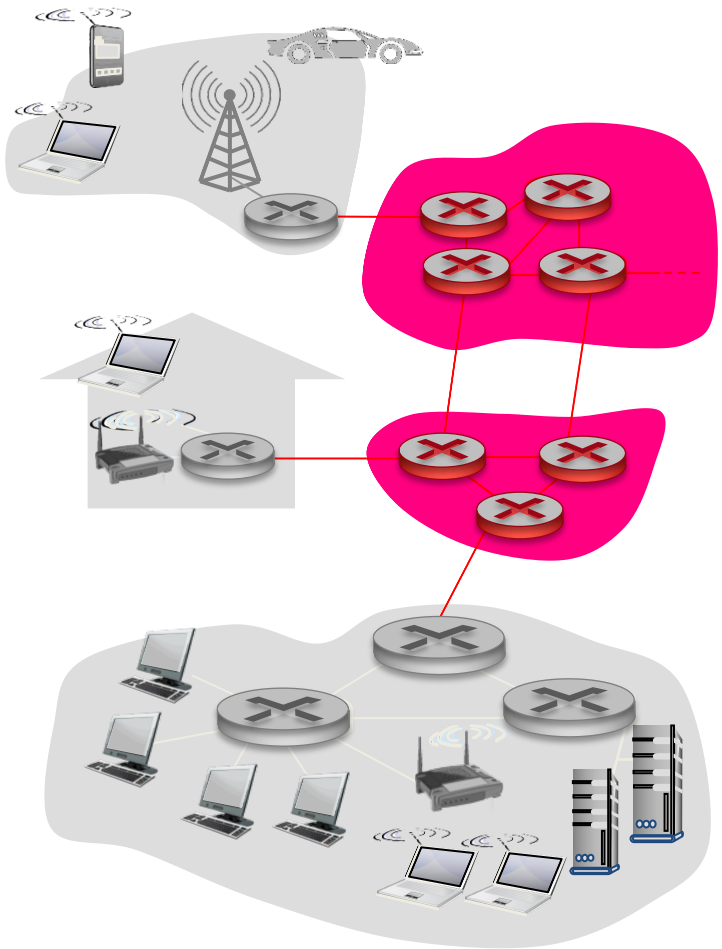
- mesh of interconnected routers

Option 1: Packet-switching

- on-demand resource allocation
- best effort service
- good bandwidth use

Option 2: Circuit-switching

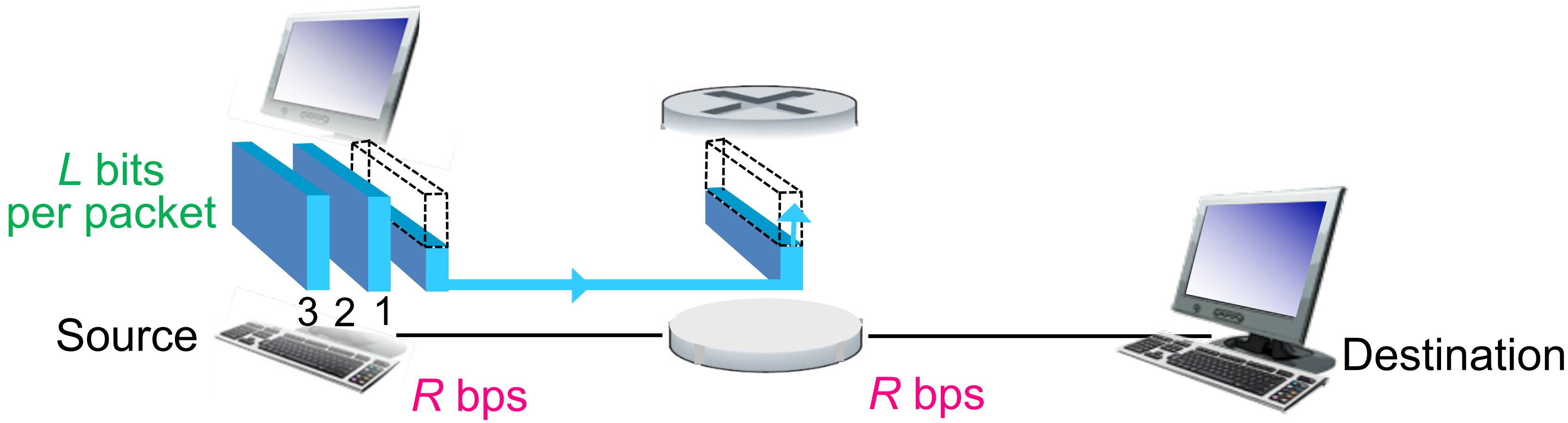
- reserved resources
- guaranteed service
- may waste bandwidth



Q: human analogies of reserved versus on-demand allocation

# Packet switching

1. Hosts break app-layer messages into packets




2. Time to transmit (push out) *L*-bit packet into *R* bps link: *L* / *R* *seconds*

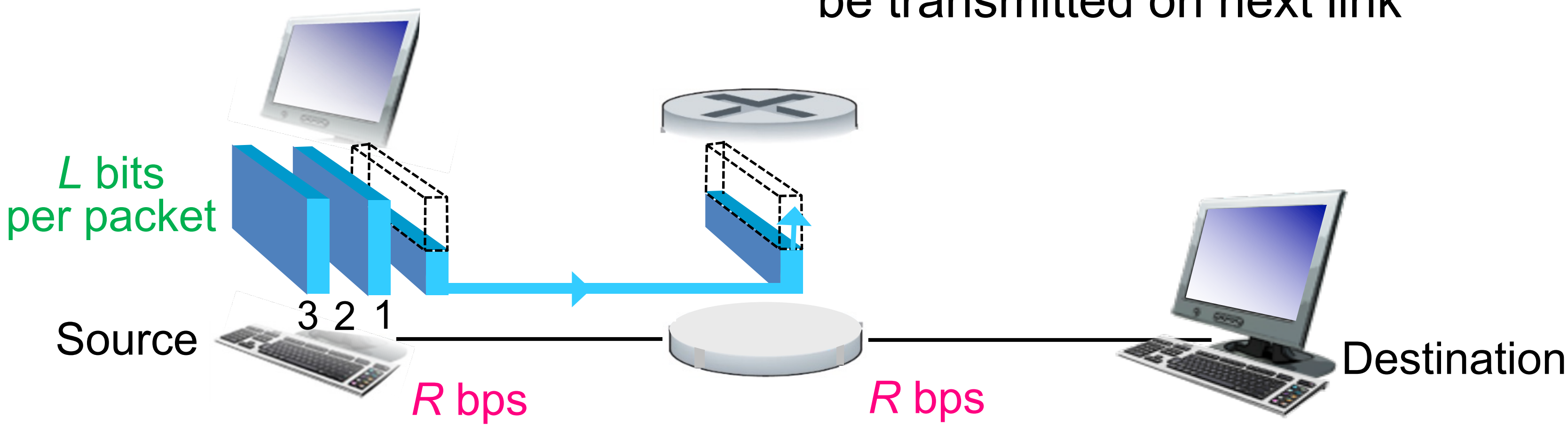
Example

- L = 7.5 Mbits
- R = 1.5 Mbps
- 1-hop transmission delay = 5s

# Packet switching

1. Hosts break app-layer messages into packets

3. Store-and-forward: entire packet must arrive at router before it can be transmitted on next link




2. Time to transmit (push out) *L*-bit packet into *R* bps link: *L* / *R seconds*

4. Time to transmit (push out) *L*-bit packet into *R* bps link: *L* / *R seconds*

**End-end transmission delay = 2 *L* / *R seconds***

(assuming zero propagation, queuing, processing delay… )

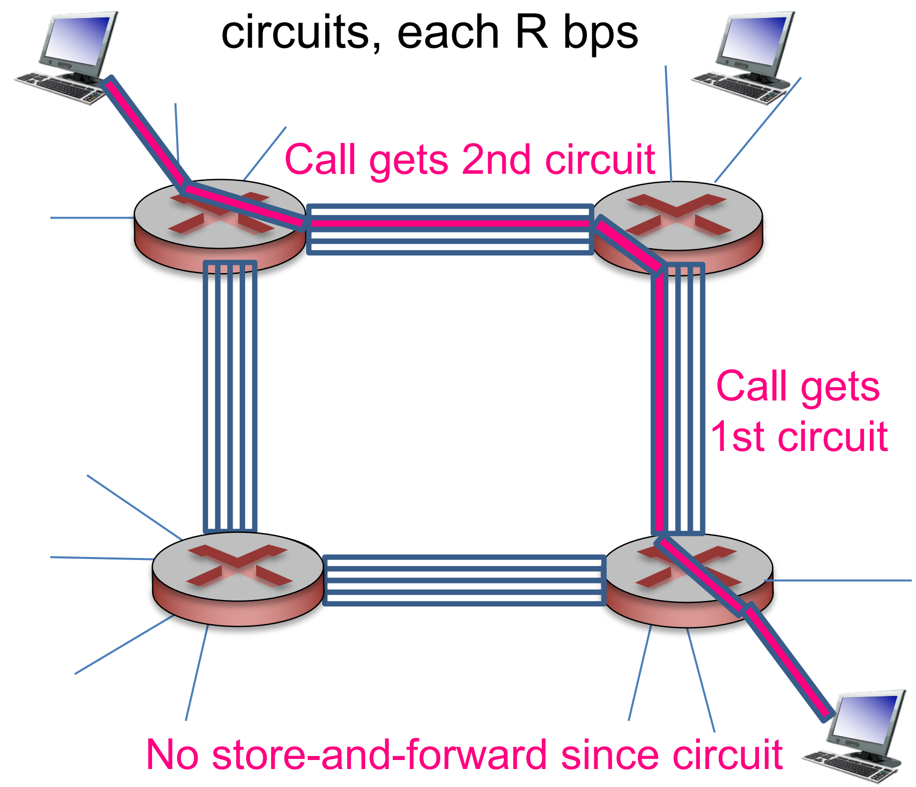
# Circuit switching

## End-end resources allocated

- reserved for "call" between source & dest
- commonly used in traditional telephone networks

## Dedicated resources

- no sharing
- circuit-like (guaranteed) performance
- circuit segment idle if not used by call (no sharing)

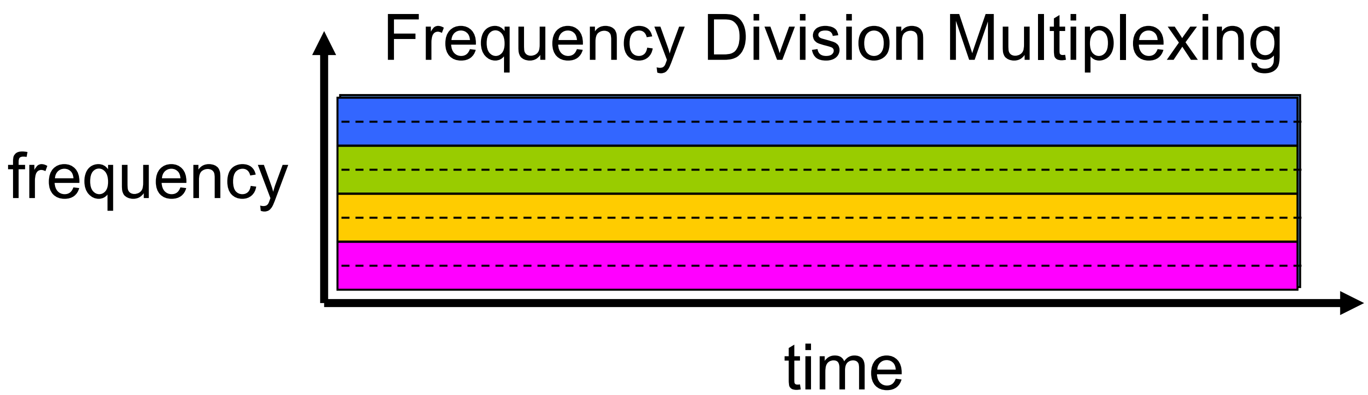


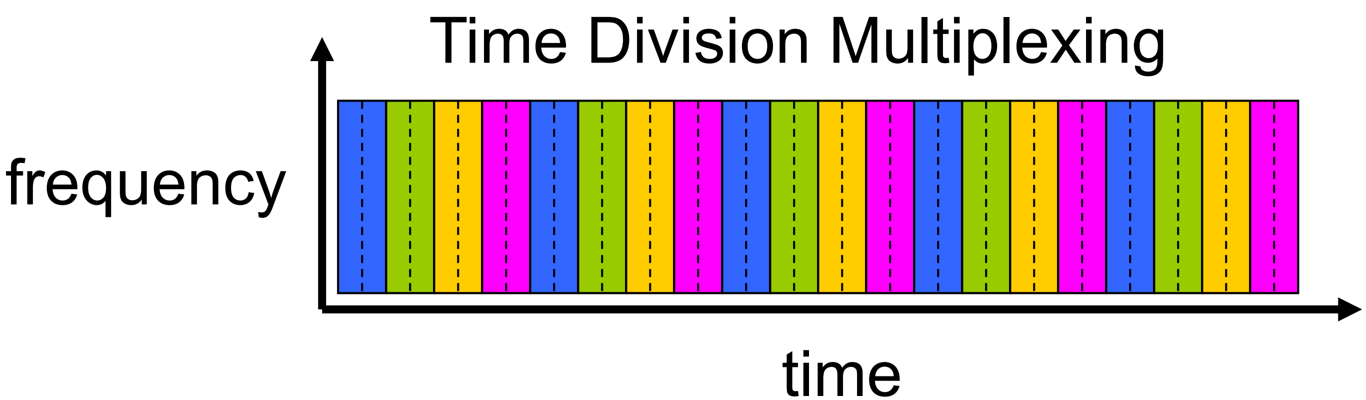

Q: what is end-end transmission delay? *L / R seconds*

Q: what happens if there is a lull in conversation?

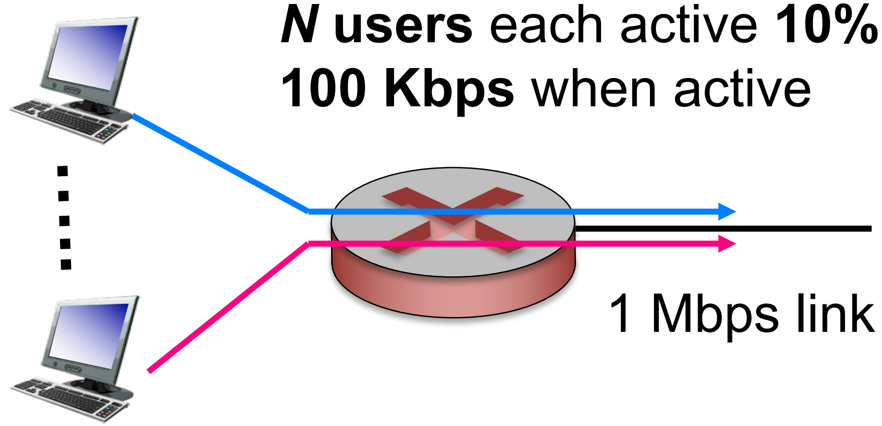
# Circuit switching

Multiplexing 4 users



# Packet switching versus circuit switching




***N* users** each active **10%** of time,
**100 Kbps** when active

Q: How many users can be supported?

Circuit switching

- 1 Mbps / 100 Kbps = 10
- N = 10 users

Packet switching

- N = 35 users
- prob > 10 users active simultaneously < .0004

Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

Packet switching allows more users to use network!

# Is packet switching always better?

Great for bursty data

- resource sharing
- simpler, no call setup

Excessive congestion possible

- packet delay and loss
- protocols needed for reliable data transfer, congestion control

Q: How to provide circuit-like behavior?

- bandwidth guarantees needed for audio/video apps
- still an unsolved problem (chapter 7)

# Binomial random variable (homework)

Suppose we do n independent experiments

- each experiment succeeds with probability p
- each experiment fails with probability 1-p

Independent experiment

- knowledge about one experiment occurring does not affect probability of other experiment occurring: e.g., coin toss

$$P(A \text{ and } B) = P(A) \times P(B)$$

$$P(A \text{ or } B) = P(A) + P(B)$$

$$P(X=4 \text{ and } X=5) = P(X=4) \times P(X=5)$$

$$P(X=4 \text{ or } X=5) = P(X=4) + P(X=5)$$

# Binomial random variable (homework)

Suppose we do n independent experiments

- each experiment succeeds with probability p
- each experiment fails with probability 1-p

X = Random Variable indicating # of successes that occur in n experiments

Probability of i successes

$$P(X = i) = \binom{n}{i} p^i (1-p)^{n-i}$$

Probability of n-i failures

n choose i
different ways to get i
successful experiments

vumanfredi@wesleyan.edu