

Lecture 2: Internet Structure

COMP 411, Fall 2022

Victoria Manfredi

W E S L E Y A N
U N I V E R S I T Y



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

- please do reading!
- homework 1 posted, due Wed. Sept. 16 by 5p

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**

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**

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

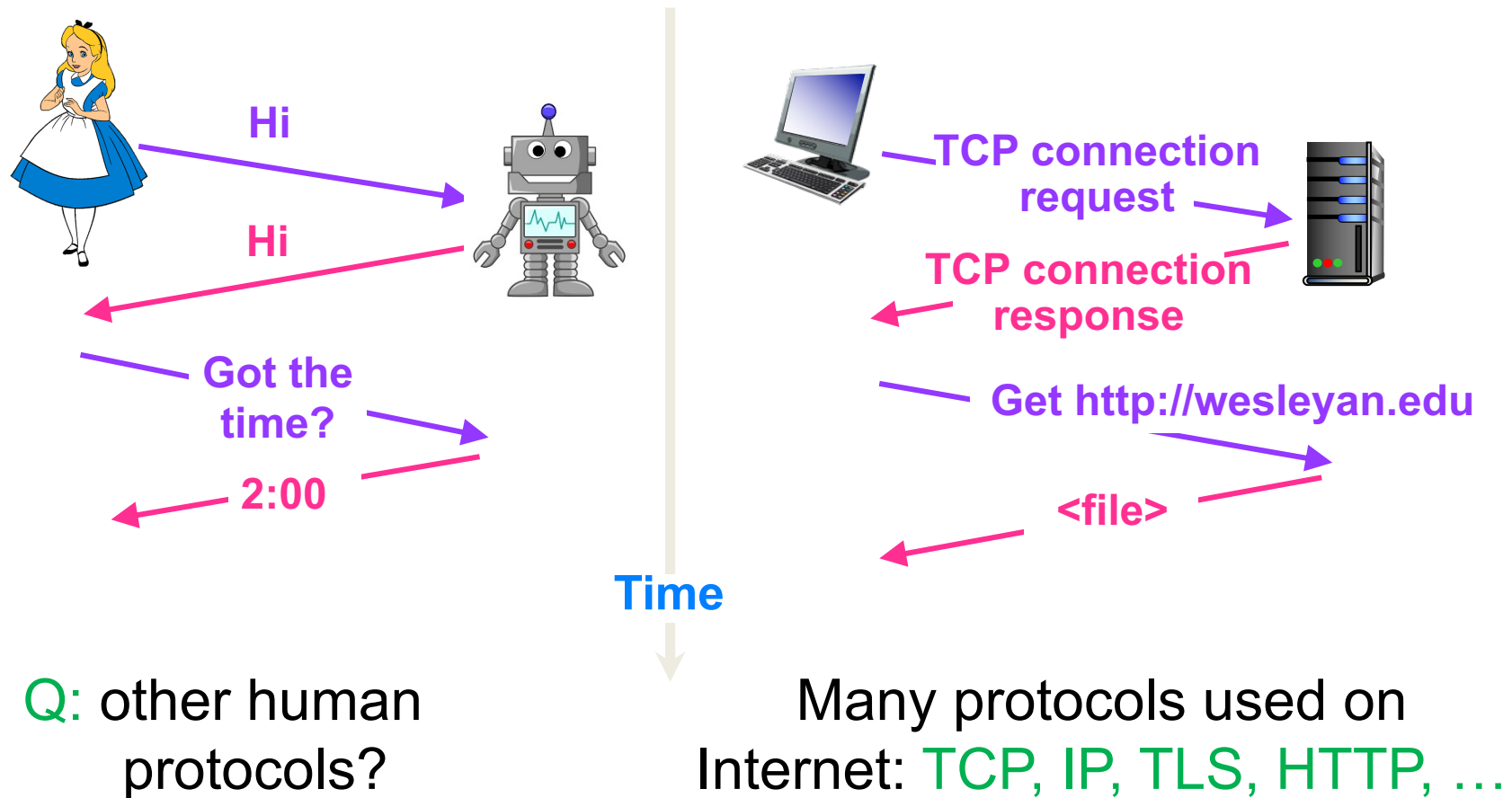
- machines rather than humans
- all communication activity in Internet governed by protocols

... specific messages sent

... actions taken when messages received, or other events

Protocol example

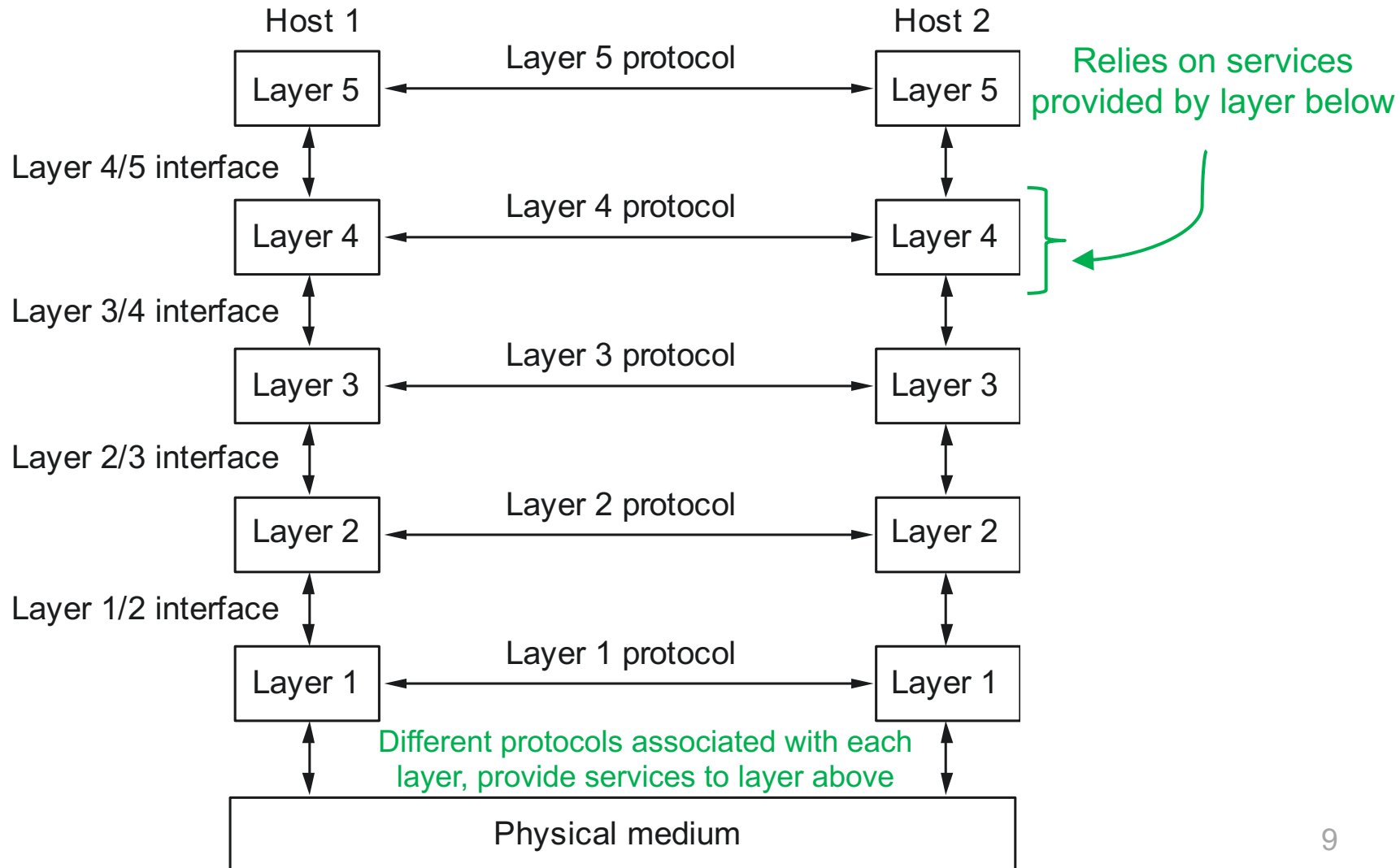
A human protocol and a computer network protocol:



Building a Network **LAYERING**

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**

7-Layer ISO/OSI Model

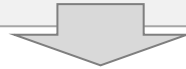
application: user interface

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?

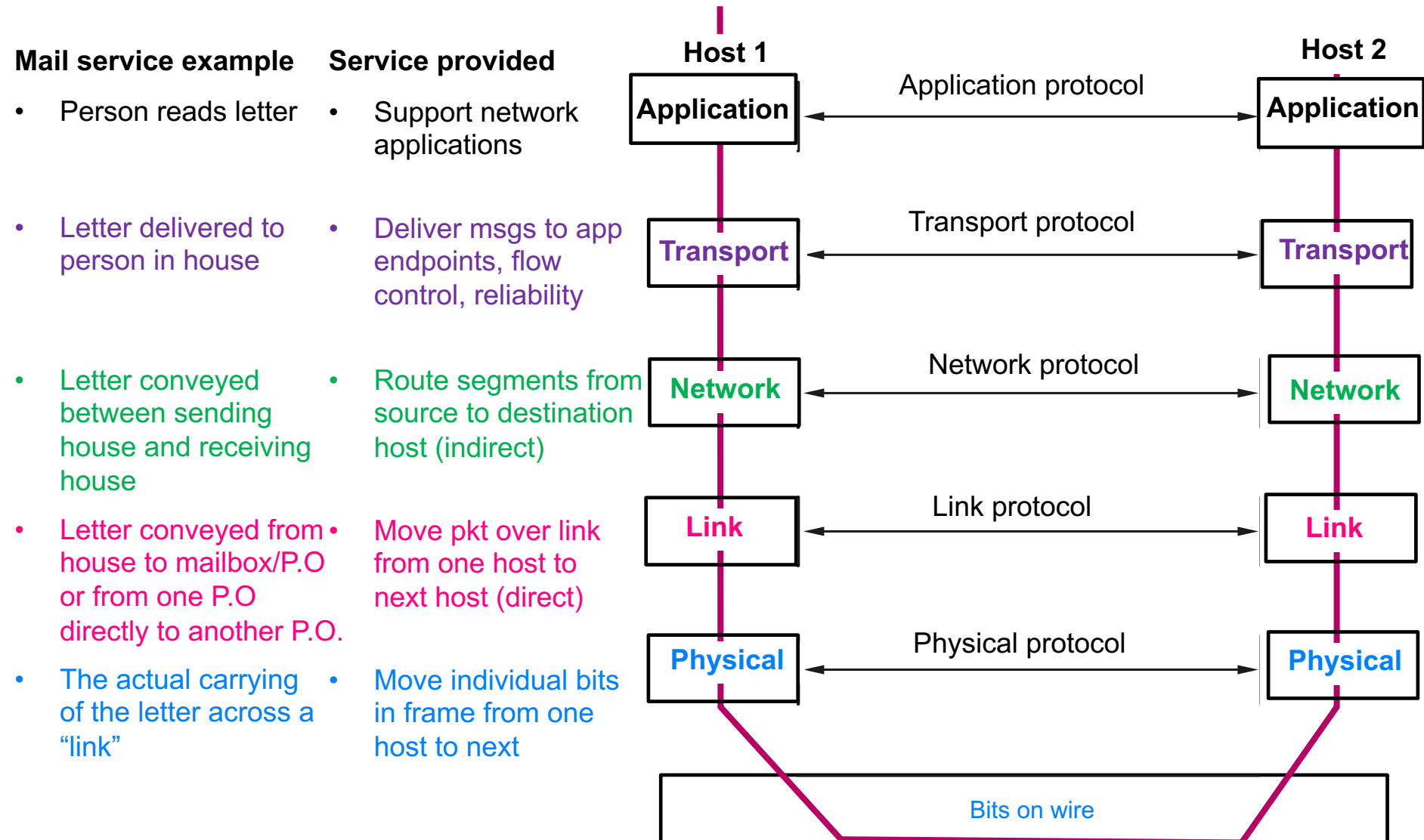
transport: multiplexing/demultiplexing, fragmentation/reassembly, end-to-end flow, congestion and error control

network: addressing and routing

data link: link flow and error control

physical: physical and electrical interfaces (normally 100% hardware)

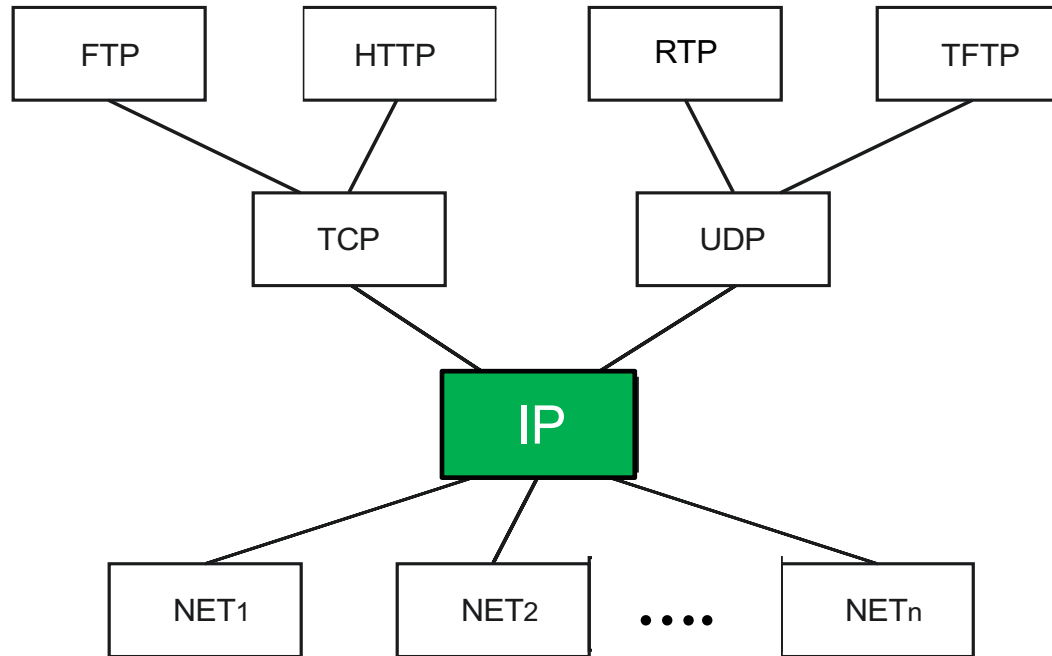
5-layer Internet protocol stack



5-layer Internet protocol stack

Layer	Service provided to upper layer	Protocols	Unit of information
5 Application	<ul style="list-style-type: none"> Support network applications 	FTP, DNS, SMTP, HTTP	Message 1 message may be split into multiple segments
4 Transport	<ul style="list-style-type: none"> Deliver messages to app endpoints Flow control Reliability 	TCP (reliable) UDP (best-effort)	Segment (TCP) Datagram (UDP) 1 segment may be split into multiple packets
3 Network	<ul style="list-style-type: none"> Route segments from source to destination host 	IP (best-effort) Routing protocols	Packet (TCP) Datagram (UDP)
2 Link	<ul style="list-style-type: none"> Move packet over link from one host to next host 	Ethernet, 802.11	Frame MTU is 1500 bytes
1 Physical	<ul style="list-style-type: none"> Move individual bits in frame from one host to next "bits on wire" 	Ethernet phy 802.11 phy Bluetooth phy 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

Layers

Physical

Link

Network

Transport

Application

87	8.578356	JuniperN_1e:18:01	Broadcast	ARP	64
88	8.622793	129.133.182.236	216.58.219.229	TCP	54
89	8.639661	216.58.219.229	129.133.182.236	TCP	66
90	9.602437	JuniperN_1e:18:01	Broadcast	ARP	64
91	9.848778	129.133.182.236	198.105.244.104	TCP	78

▶ Frame 77: 166 bytes on wire (1328 bits), 166 bytes captured (1328 bits) on inter
▶ Ethernet II, Src: JuniperN_1e:18:01 (3c:8a:b0:1e:18:01), Dst: Apple_c5:b4:9a (78
▶ Internet Protocol Version 4, Src: 129.133.6.11, Dst: 129.133.178.53
▶ User Datagram Protocol, Src Port: 53 (53), Dst Port: 44065 (44065)
▶ Domain Name System (response)

```
0000  78 31 c1 c5 b4 9a 3c 8a b0 1e 18 01 08 00 45 00  x1....<. ....E.
0010  00 98 20 98 00 00 3e 11 a0 72 81 85 06 0b 81 85  .. ...>. .r.....
0020  b2 35 00 35 ac 21 00 84 ee d2 24 fc 81 80 00 01  .5.5.!... ..$. ....
0030  00 03 00 00 00 00 03 69 6e 74 03 6e 79 74 03 63  .....i nt.nyt.c
0040  6f 6d 00 00 01 00 01 c0 0c 00 05 00 01 00 00 01  om.....
0050  ad 00 22 08 77 69 6c 64 63 61 72 64 07 6e 79 74  ..".wild card.nyt
0060  69 6d 65 73 03 63 6f 6d 07 65 64 67 65 6b 65 79  imes.com .edgekey
```

wireshark_pcapng_en0_20160824155218_HN8Ru3 Packets: 48516 · Displayed:

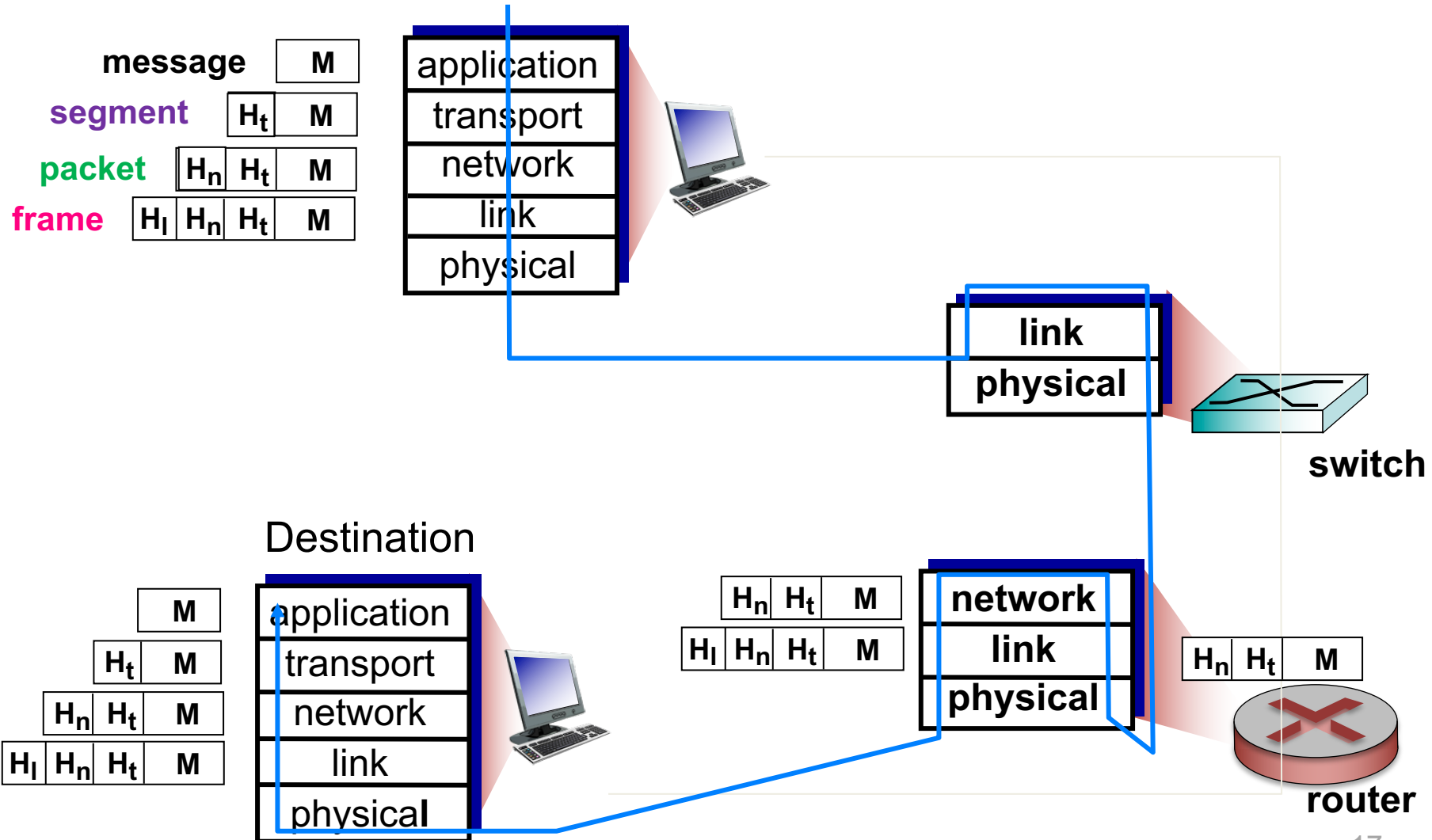
We'll talk in depth about Wireshark and how to use next week

Building a Network

KEY SERVICES

Encapsulation/Decapsulation

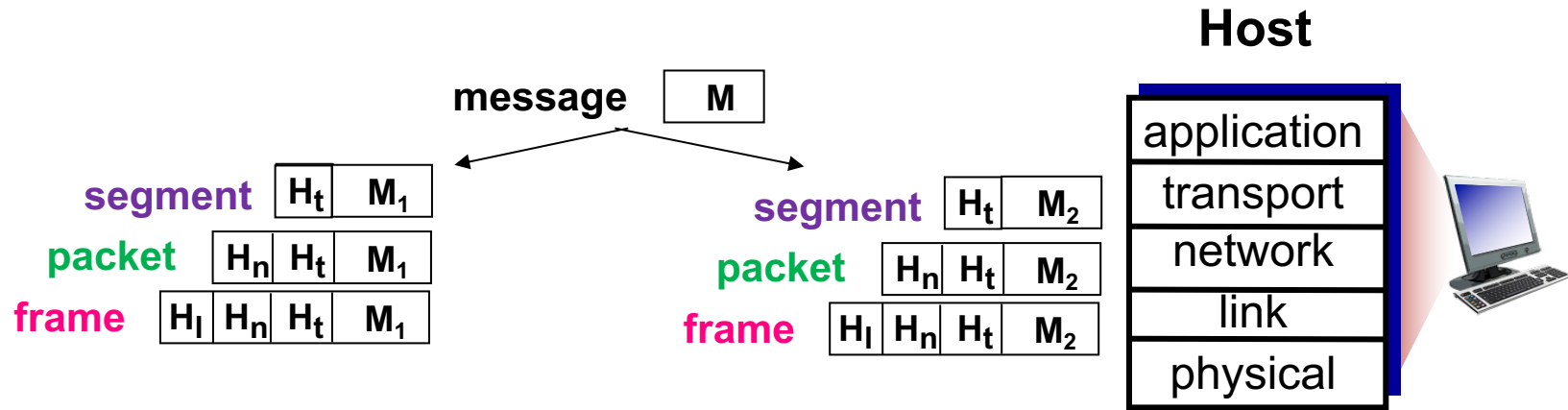
Headers must be added/removed from data unit at each layer



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

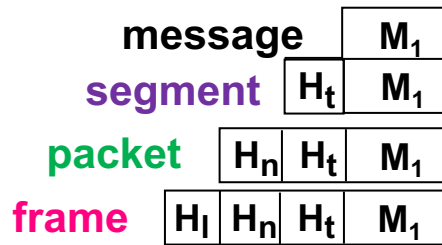


Need additional book-keeping to keep track of which **segments** belong to which **message**

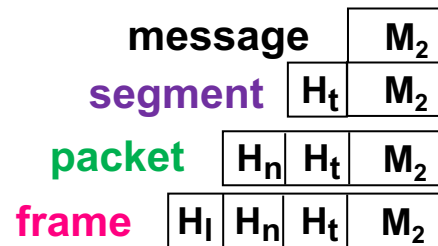
Multiplexing/Demultiplexing

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

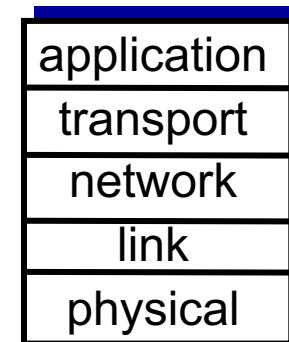
Process 1: Web



Process 2: Email



Host



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

Internet Organization

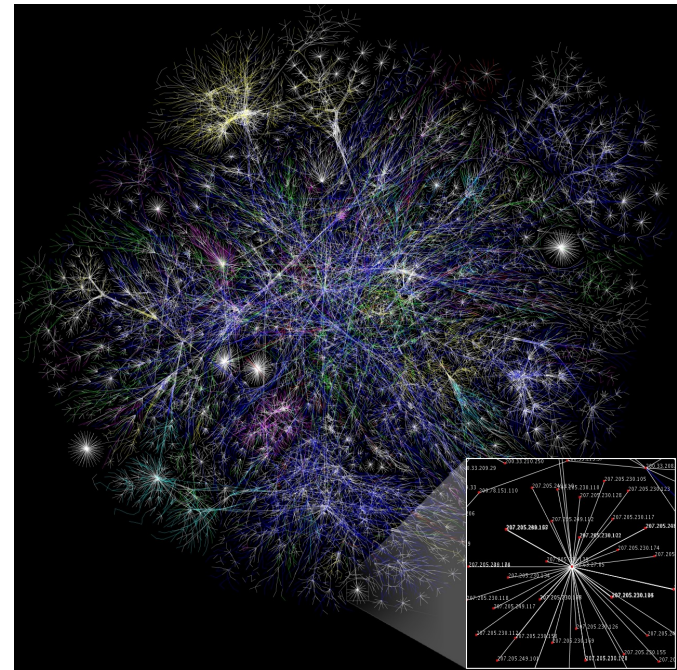
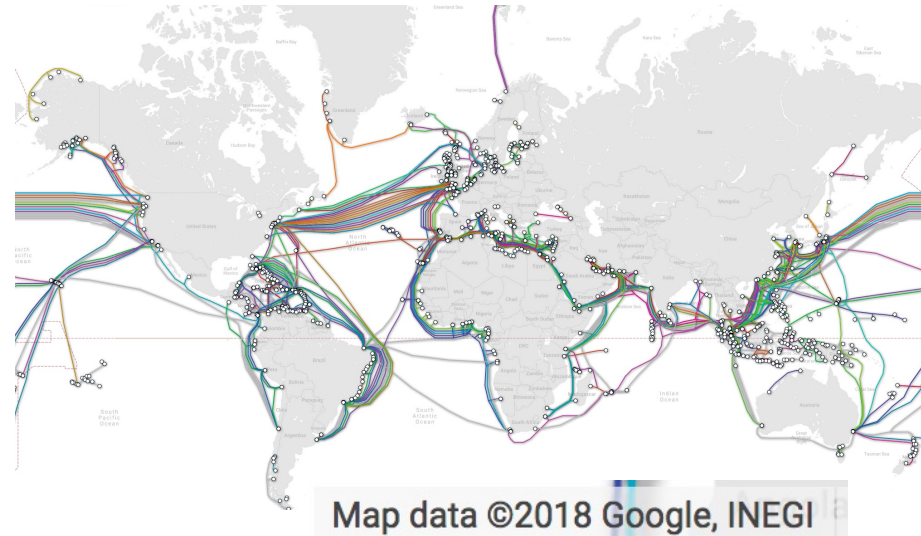
OVERVIEW

The story of the Internet

Not so long ago, a new kind of **communication network** was needed ...

... one that could withstand a **nuclear war**

... and in 1969 the **Internet** was born



By The Opte Project [CC BY 2.5
(<http://creativecommons.org/licenses/by/2.5>)],
via Wikimedia Commons

The Internet has been constantly evolving

...over the past 45+ years

Early Internet design decisions

- have far-reaching consequences today
- impact security, privacy, scalability, quality of service

To understand impact today

- we need to first understand Internet structure

How is the Internet organized?

Billions of connected hosts

- run network applications



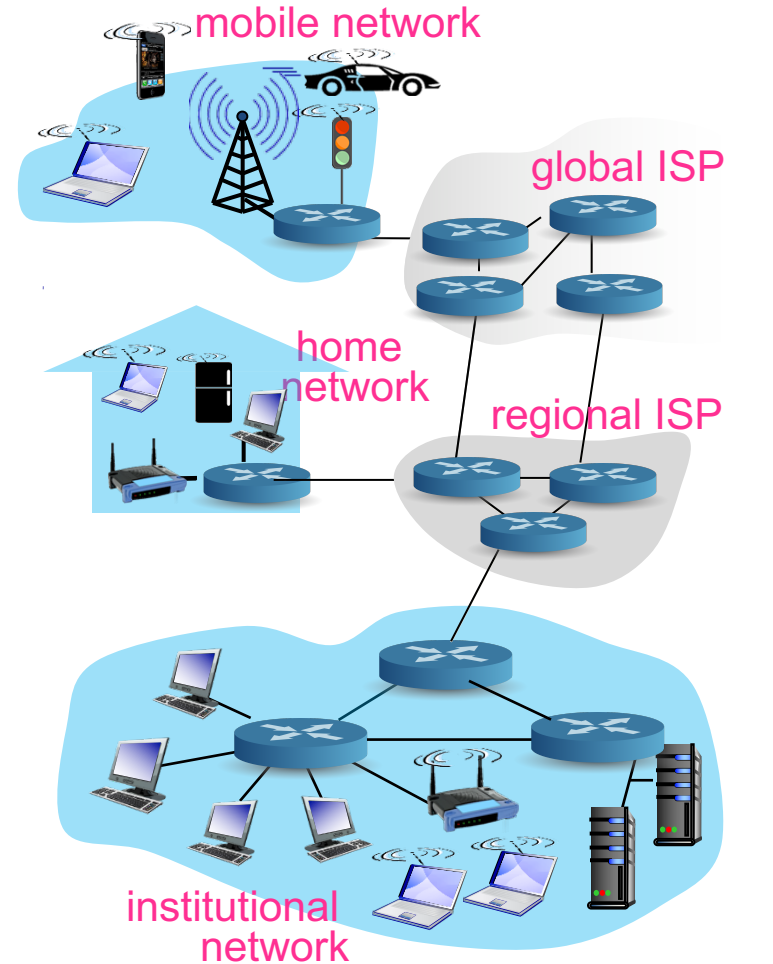
Communication links

- 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

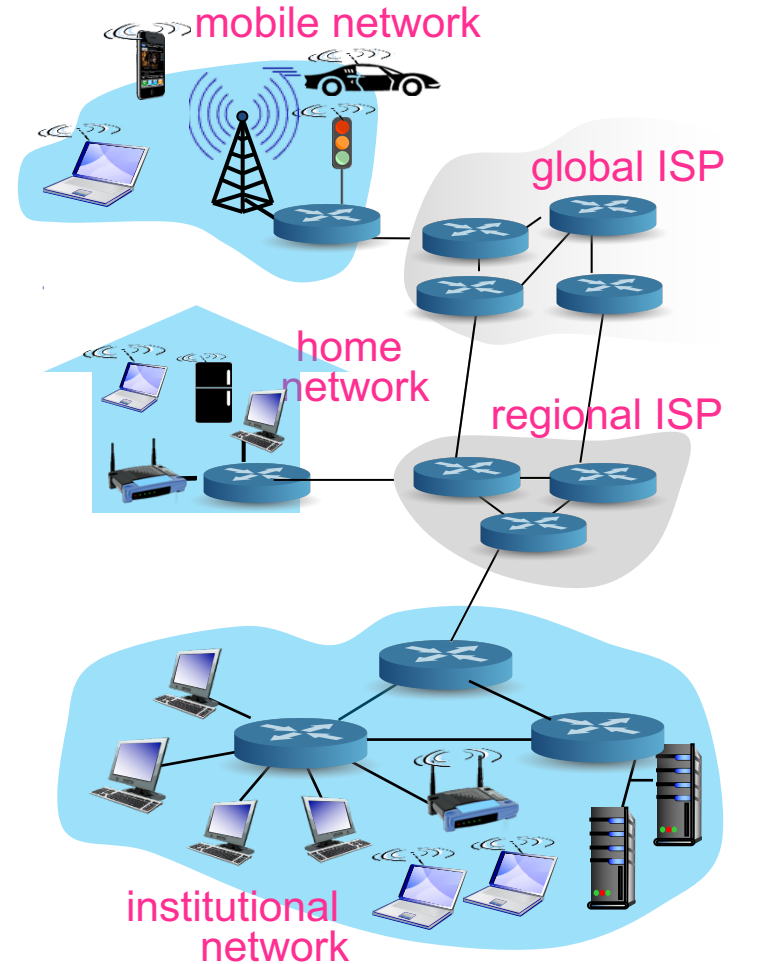
- **hosts**: clients and servers
- servers often in data centers

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

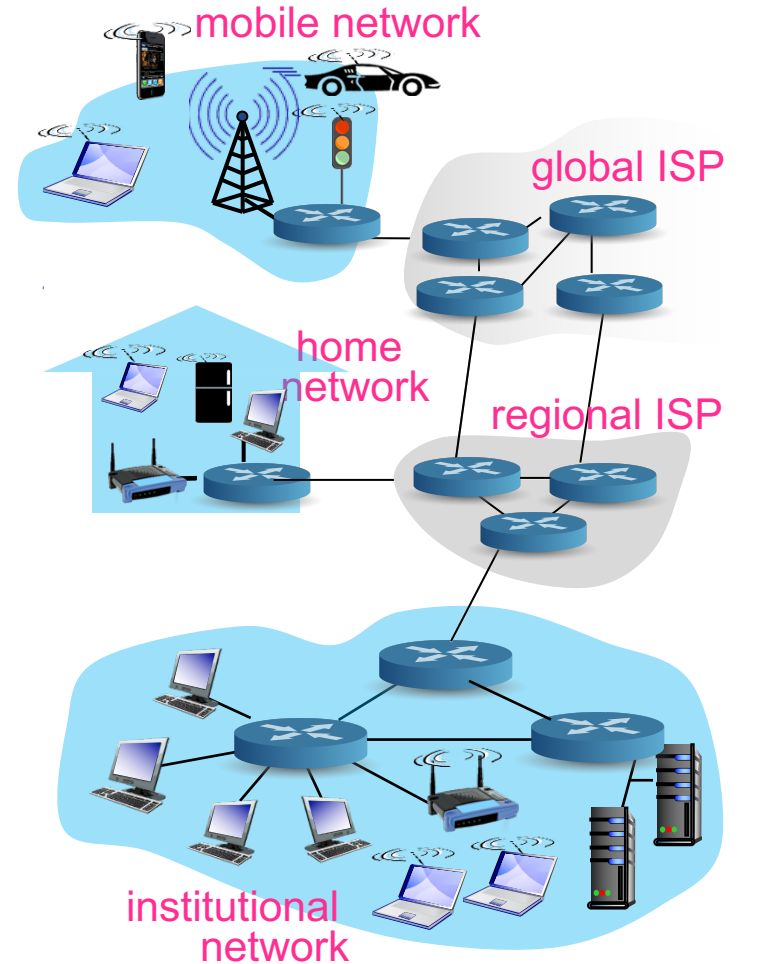
- e.g., web, VoIP, email, games, ecommerce, social networks

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

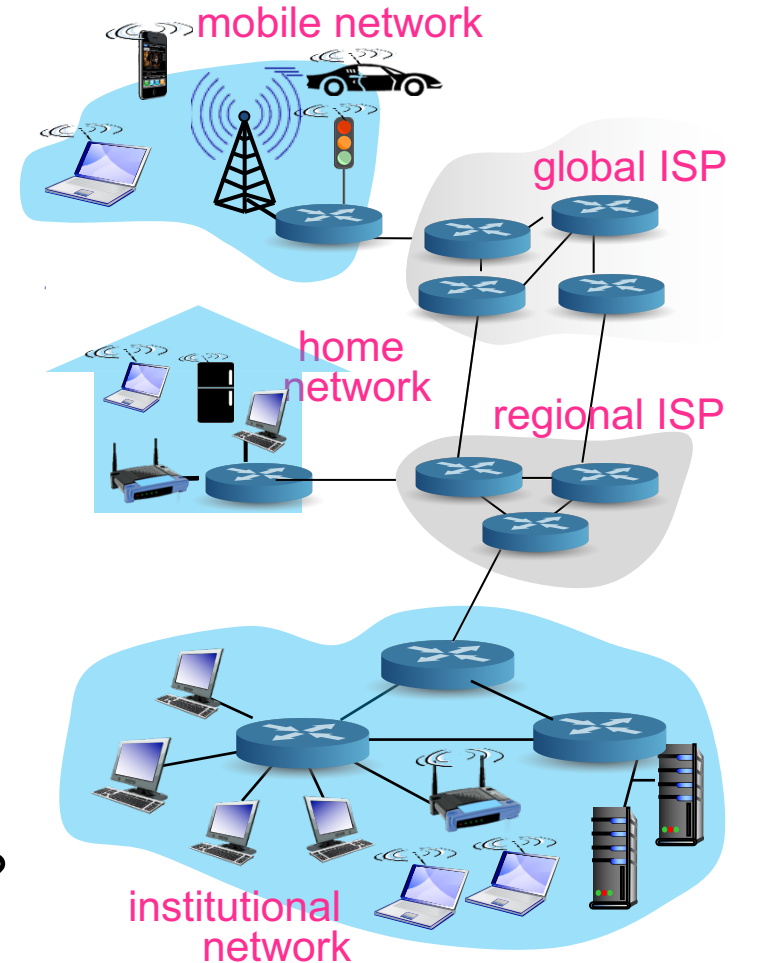
- access network/ISP

Access networks

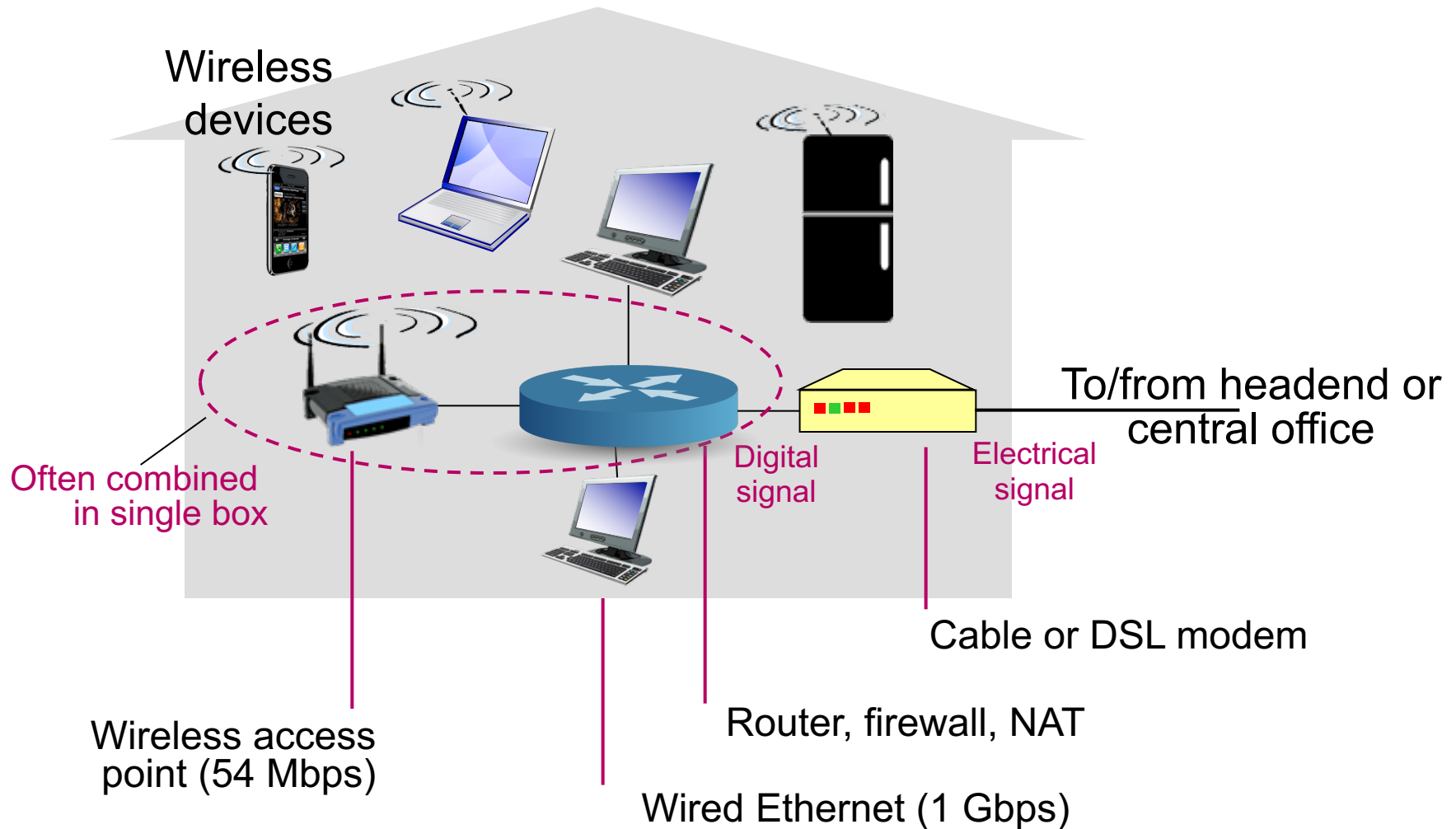
- **residential**
 - DSL (telephone), cable,
- **institutional**
 - school, company
- **mobile**

Issues

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



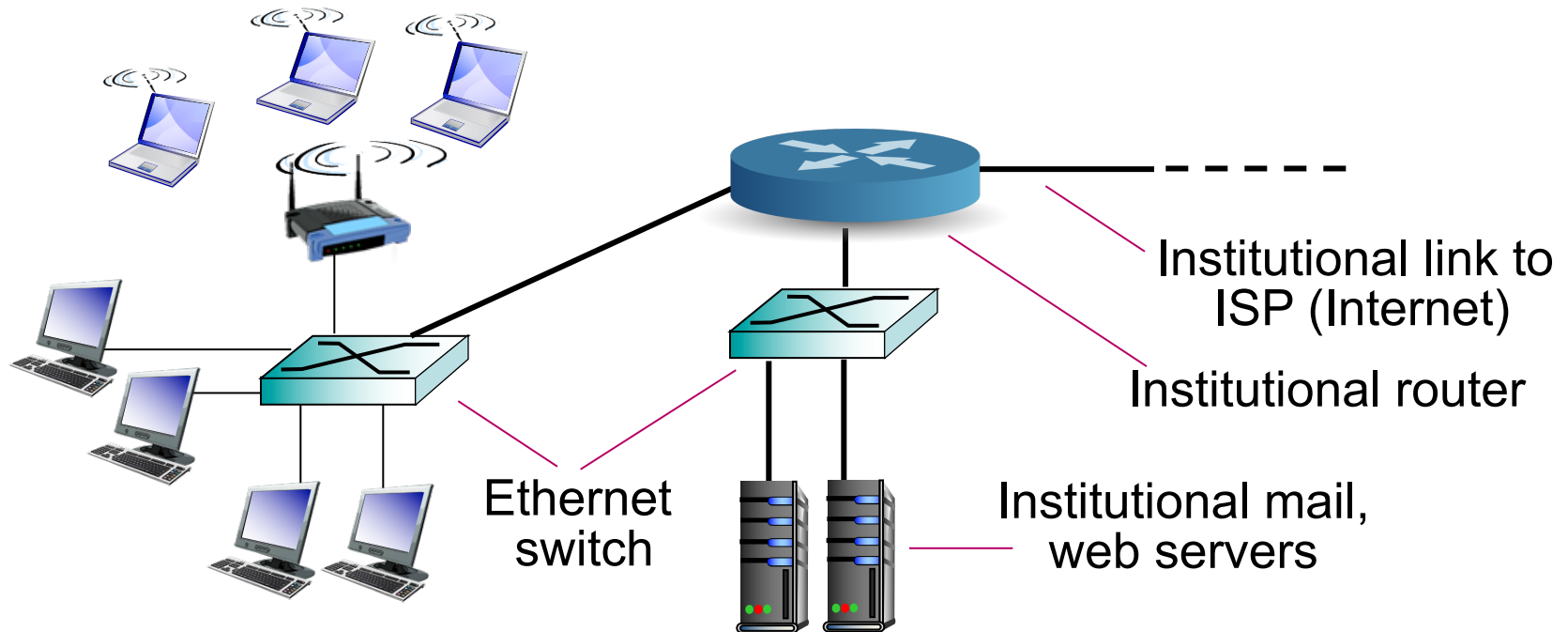
Home access network



Enterprise access network (Ethernet)

Typically used in companies, universities, etc.

- 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- today, end systems typically connect into Ethernet switch



Wireless access network

Shared wireless access network

- connects end system to router via base station (aka “access point”)

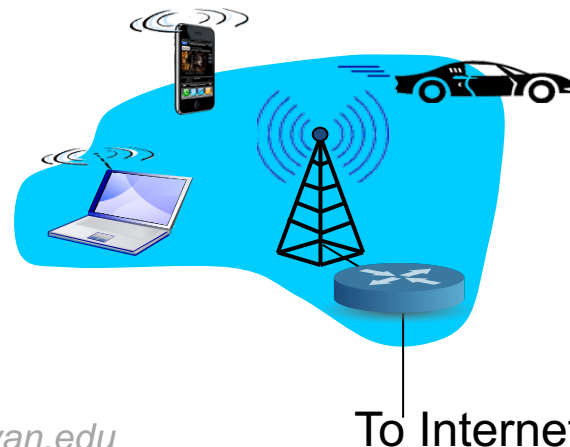
Wireless LANs

- within building (100 ft.)
- 802.11b/g/n (WiFi):
 - 11, 54, 450 Mbps



Wide-area wireless access

- provided by telco (cellular) operator, 10's km
- between 1 and 10 Mbps
- 3G, 4G: LTE



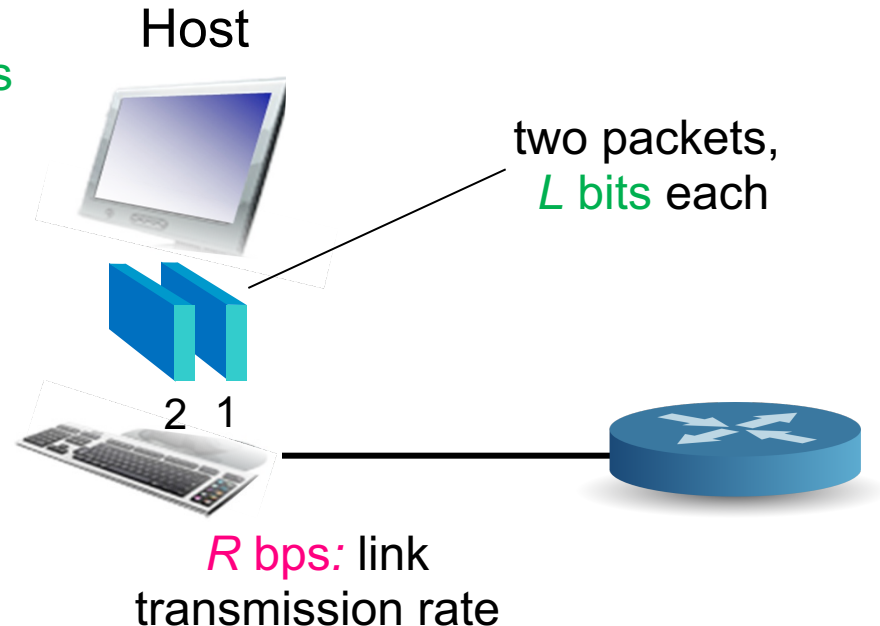
Host sends packets into access network

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



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