Chapter 1: Introduction

Our goal:

- get "feel" and terminology
- more depth, detail *later* in course
- approach:
  - use Internet as example

Overview:

- what's the Internet
- what's a protocol?
- network edge
- access net, physical media
- network core
- Internet/ISP structure
- performance: loss, delay
- protocol layers, service models
- network modeling

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network access and physical media

1.4 Network core

1.5 Internet structure and ISPs

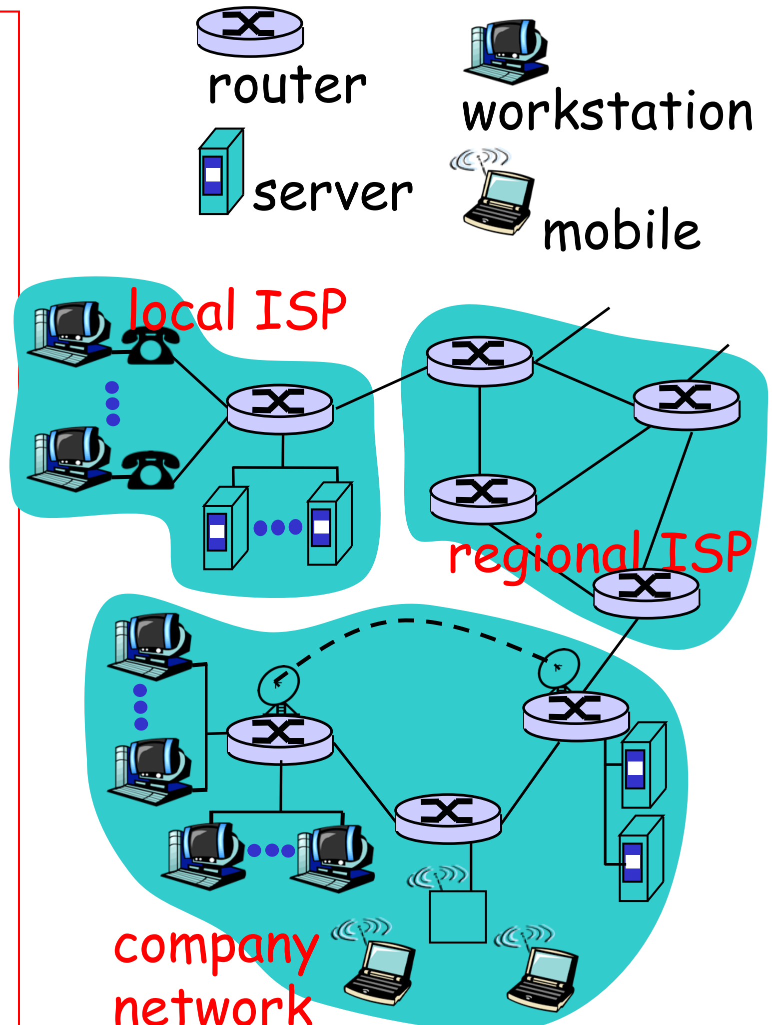
1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

What's the Internet: "nuts and bolts" view

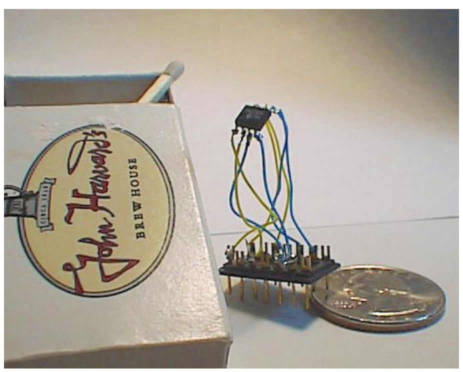
- millions of connected computing devices:
  - *hosts = end systems*
- running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - Different transmission rates
- *routers:* forward packets (chunks of data)



“Cool” internet appliances

IP picture frame
http://www.ceiva.com/

Web-enabled toaster +
weather forecaster

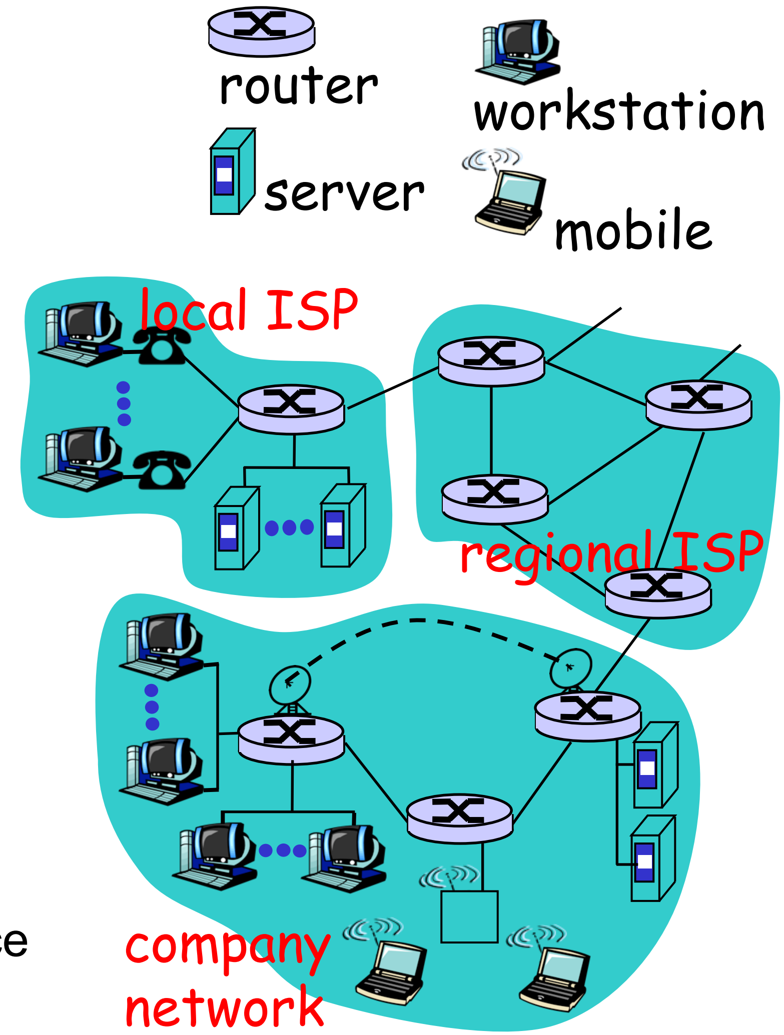


World’s smallest web server
http://www-ccs.cs.umass.edu/~shri/iPic.html

Internet phones

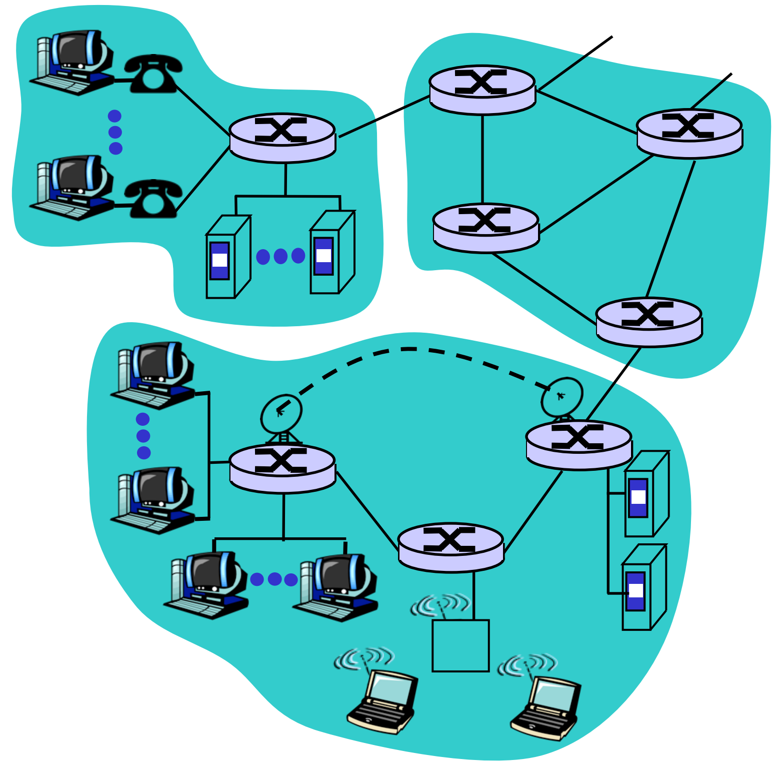
What's the Internet: "nuts and bolts" view

- *protocols* coordinate communication
  - Who gets to transmit?
  - What path to take?
  - What message format?
  - e.g., TCP, IP, HTTP, FTP, PPP
- *Internet:* "network of networks"
  - loosely hierarchical
  - public Internet Vs private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



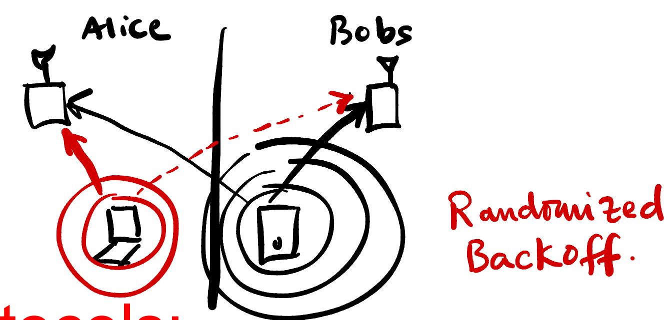
What's the Internet: a service view

- communication *infrastructure* enables distributed applications:
  - Web, email, games, e-commerce, file sharing
- communication services provided to apps:
  - Connectionless unreliable
  - connection-oriented reliable



Can you give an analogy of this in real life services

What's a protocol?



human protocols:

- "what's the time?"
- "I have a question"
- introductions

… specific msgs sent

… specific actions taken when msgs received, or other events

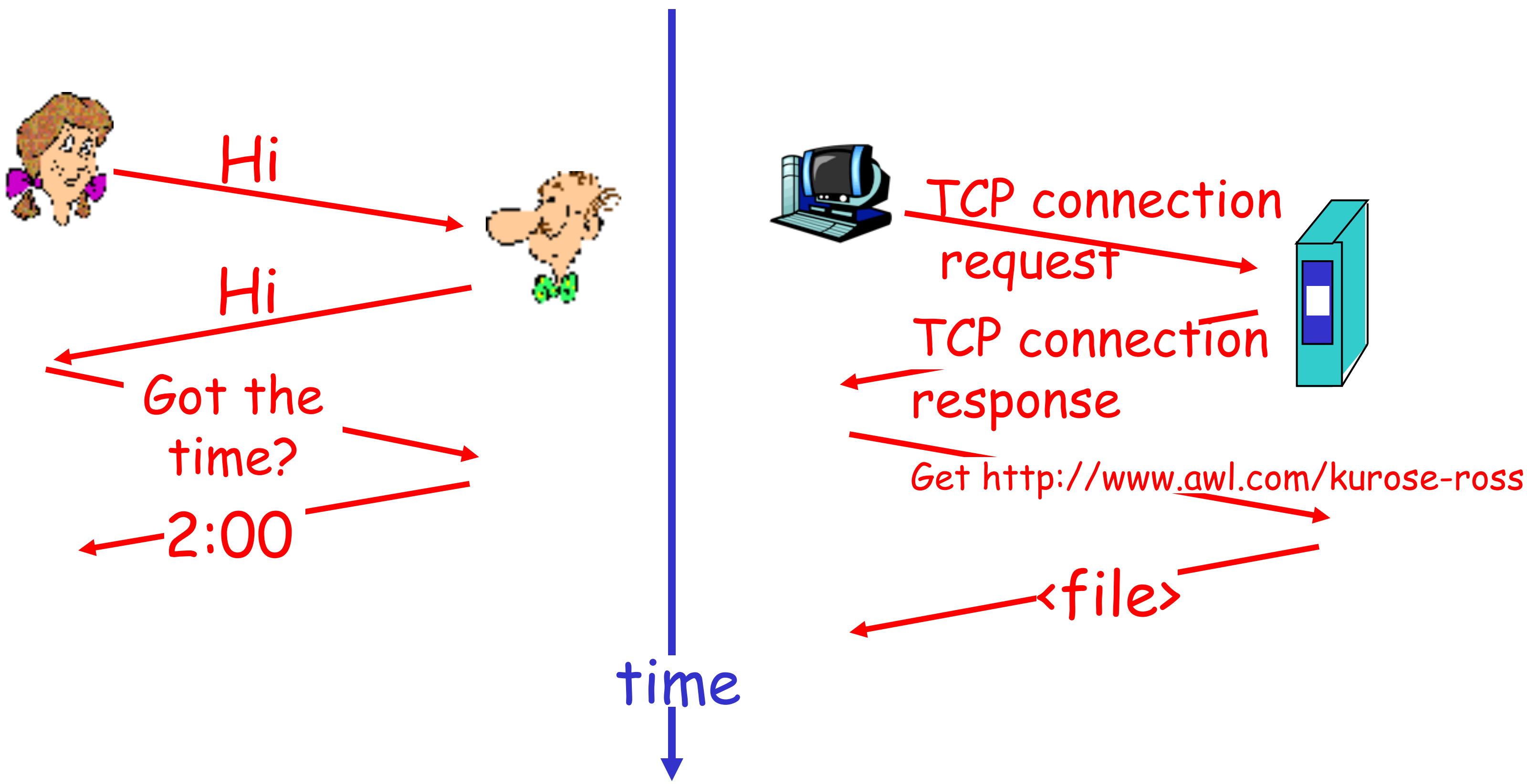
network protocols:

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

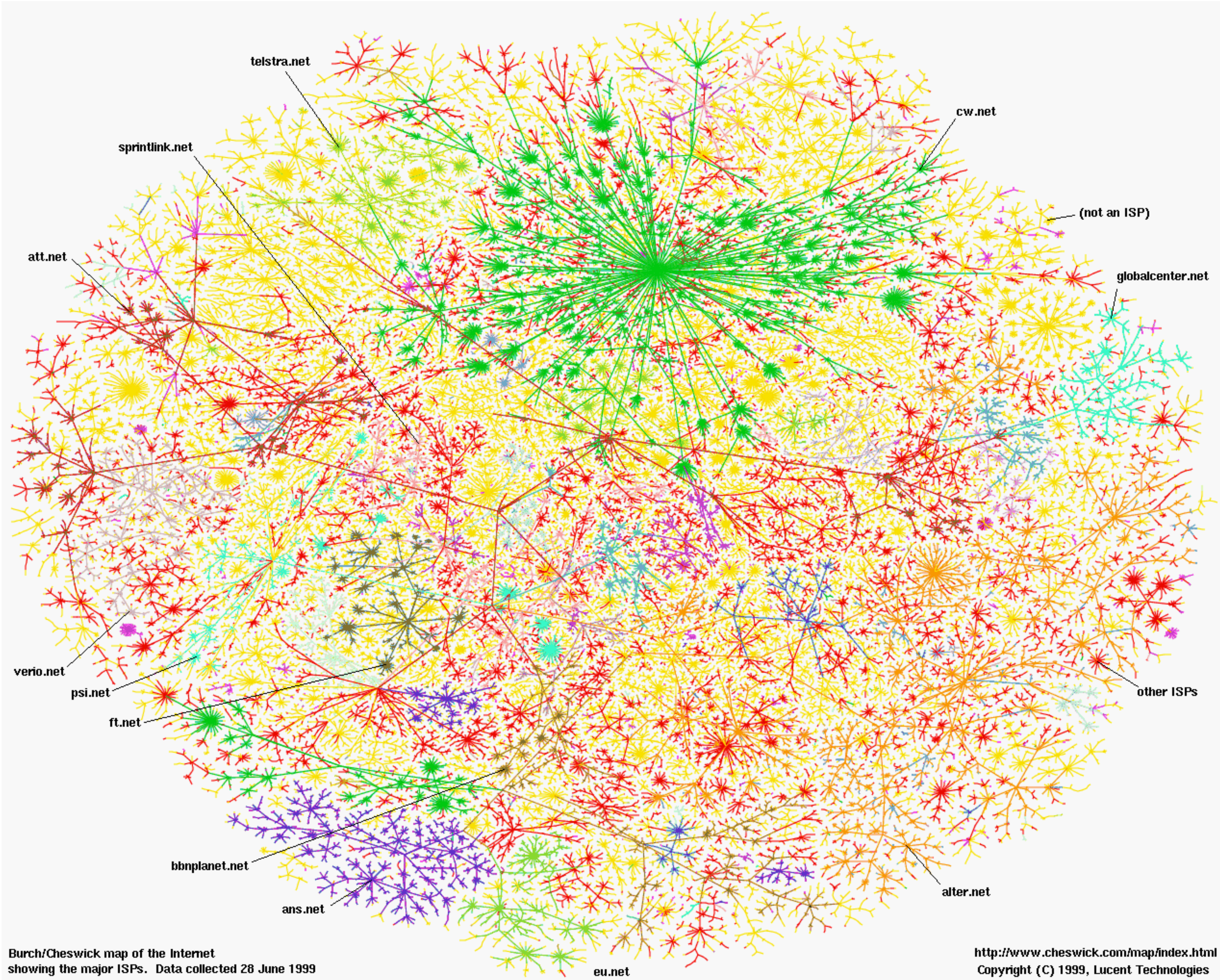
protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

What's a protocol?

a human protocol and a computer network protocol:



Q: This one trivial. Can u think of a more complex case?


telstra.net
sprintlink.net
cw.net
(not an ISP)
att.net
globalcenter.net
verio.net
psi.net
ft.net
other ISPs
bbnplanet.net
ans.net
alter.net
eu.net
Burch/Cheswick map of the Internet
showing the major ISPs. Data collected 28 June 1999
http://www.cheswick.com/map/index.html
Copyright (C) 1999, Lucent Technologies

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network access and physical media

1.4 Network core

1.5 Internet structure and ISPs

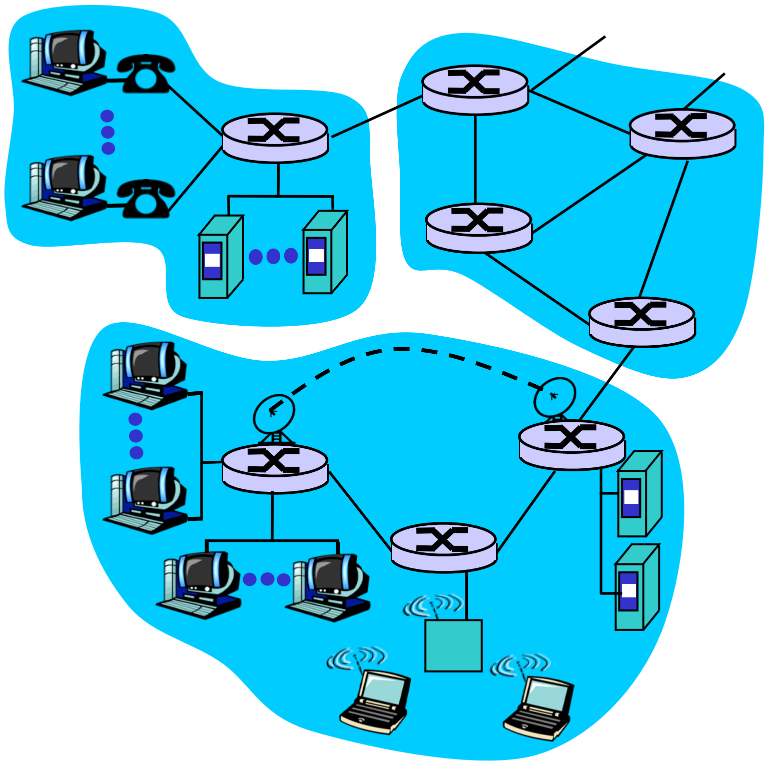
1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

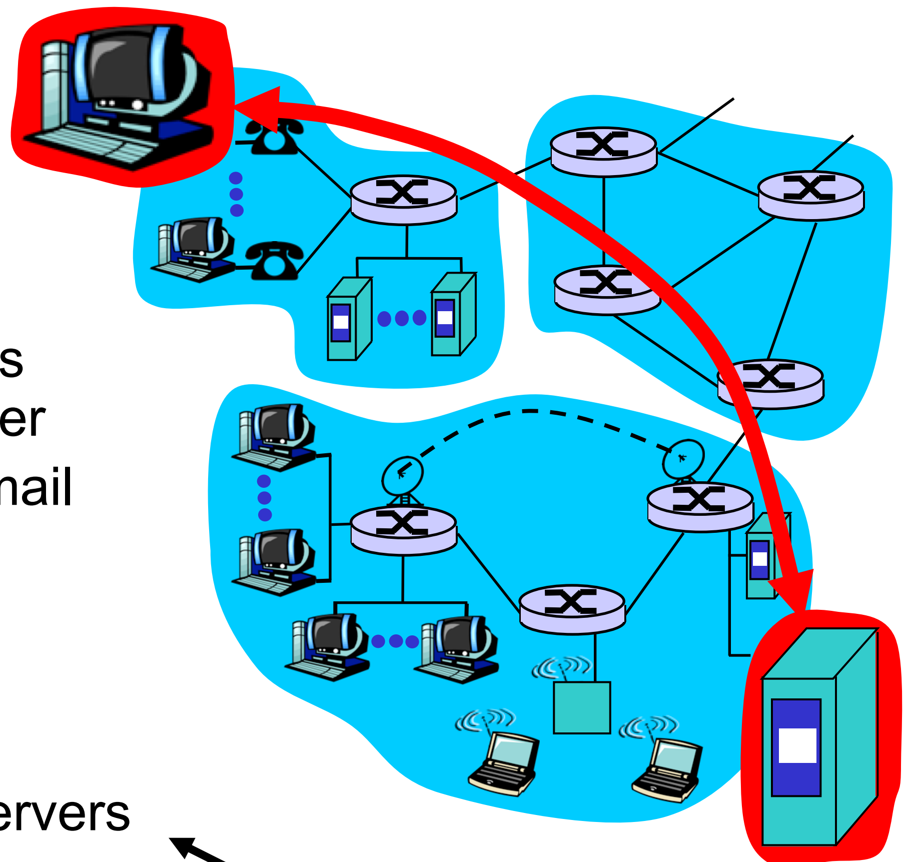
A closer look at network structure:

- network edge: applications and hosts
- network core:
  - routers
  - network of networks
- access networks, physical media: communication links



The network edge:

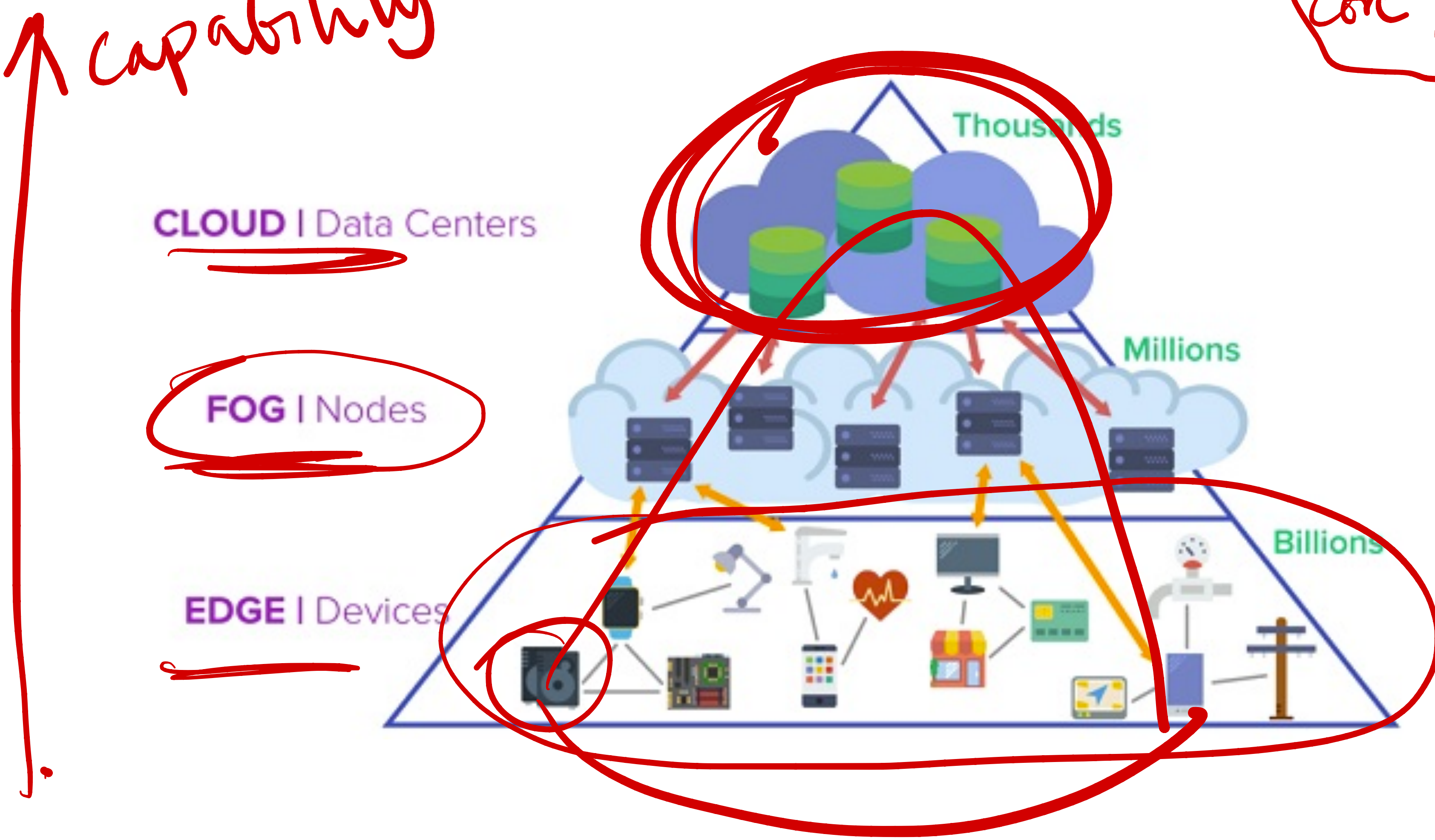
- end systems (hosts):
  - run application programs
  - e.g. Web, email
- client/server model
  - client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- peer-peer model:
  - minimal use of dedicated servers
  - e.g. Skype, BitTorrent, KaZaA

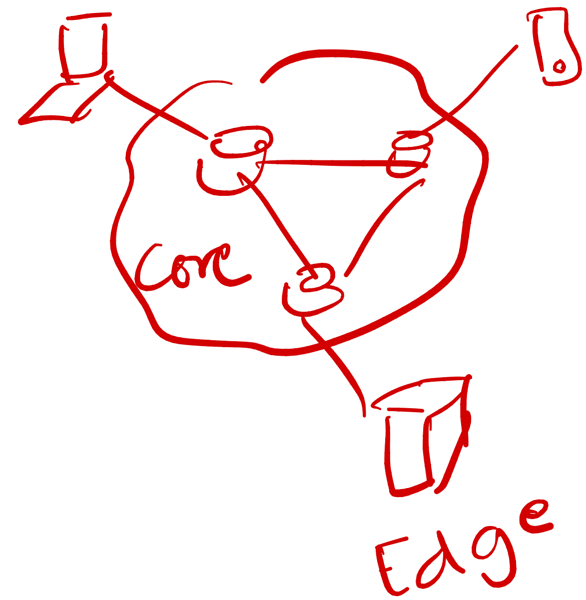


Any idea how?

Edge – Core

capability





Network edge: connection-oriented service

Goal: data transfer between end systems

- ***Connection:*** prepare for data transfer ahead of time
  - Request / Respond
  - *set up "state"* in two communicating hosts
- TCP - Transmission Control Protocol
  - Internet's connection-oriented service

TCP service [RFC 793]

- *reliable, in-order* byte-stream data transfer
  - loss: acknowledgements and retransmissions
- *flow control:*
  - sender won't overwhelm receiver
- *congestion control:*
  - senders "slow down sending rate" when network congested

Network edge: connectionless service

Goal: data transfer between end systems
- same as before!

- UDP - User Datagram Protocol [RFC 768]:
  - connectionless
  - unreliable data transfer
  - no flow control
  - no congestion control

App's using TCP:

- HTTP (Web), FTP (file transfer), Telnet (remote login), SMTP (email)

App's using UDP:

- streaming media, teleconferencing, DNS, Internet telephony

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network access and physical media

1.4 Network core

1.5 Internet structure and ISPs

1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

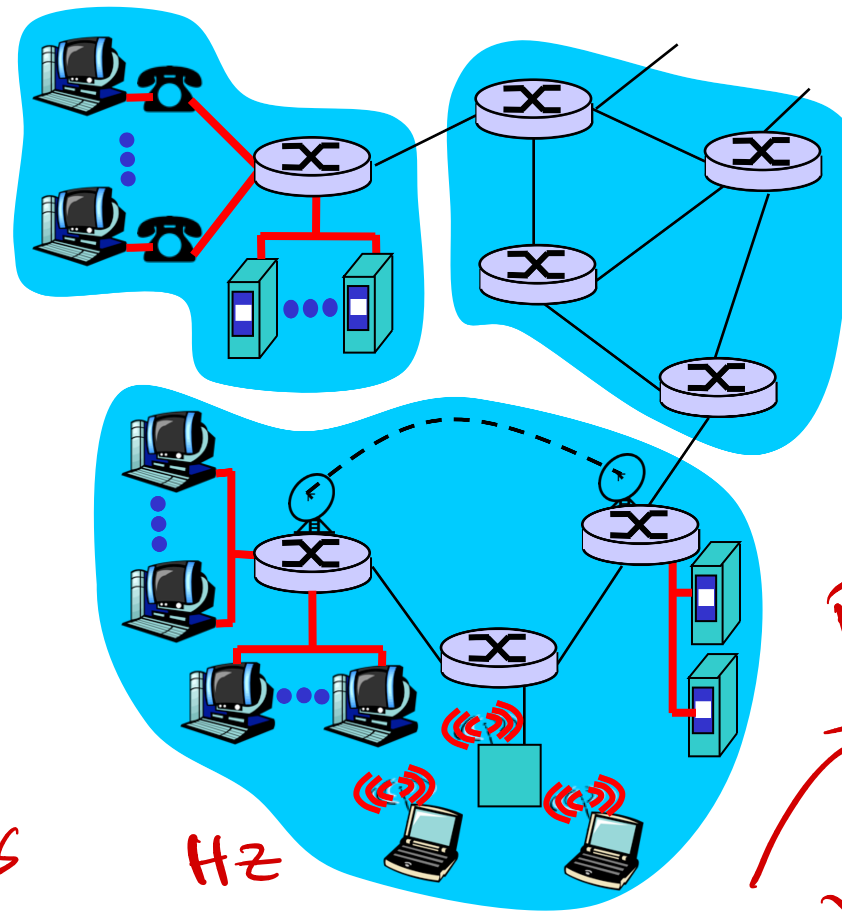
Access networks and physical media

Q: How to connect end systems to edge router?

- residential access nets
- institutional access networks (school, company)
- mobile access networks

Keep in mind:

- bandwidth (bits per second) of access network?
- shared or dedicated?



bits/s   Hz

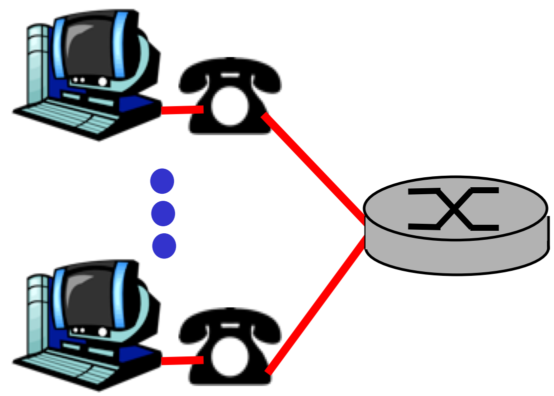
$$\text{Rate} = B \log(1 + SNR)$$

P_R / N_0

1 MHz

Residential access: point to point access

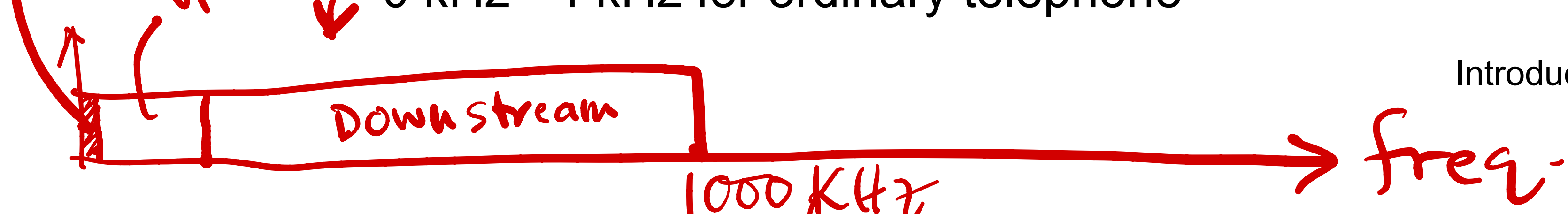
- Dialup via modem
  - up to 56Kbps direct access to router (often less)
  - Can't surf and phone at same time: can't be "always on"



- ADSL: asymmetric digital subscriber line
  - up to 1 Mbps upstream (today typically < 256 kbps)
  - up to 8 Mbps downstream (today typically < 1 Mbps)
  - FDM: 50 kHz - 1 MHz for downstream

    4 kHz - 50 kHz for upstream

    0 kHz - 4 kHz for ordinary telephone



Residential access: cable modems

- HFC: hybrid fiber coax
  - asymmetric: up to 30Mbps downstream, 2 Mbps upstream
- network of cable and fiber attaches home to ISP router
  - homes share access to router
- deployment: available via cable TV companies

Residential access: cable modems

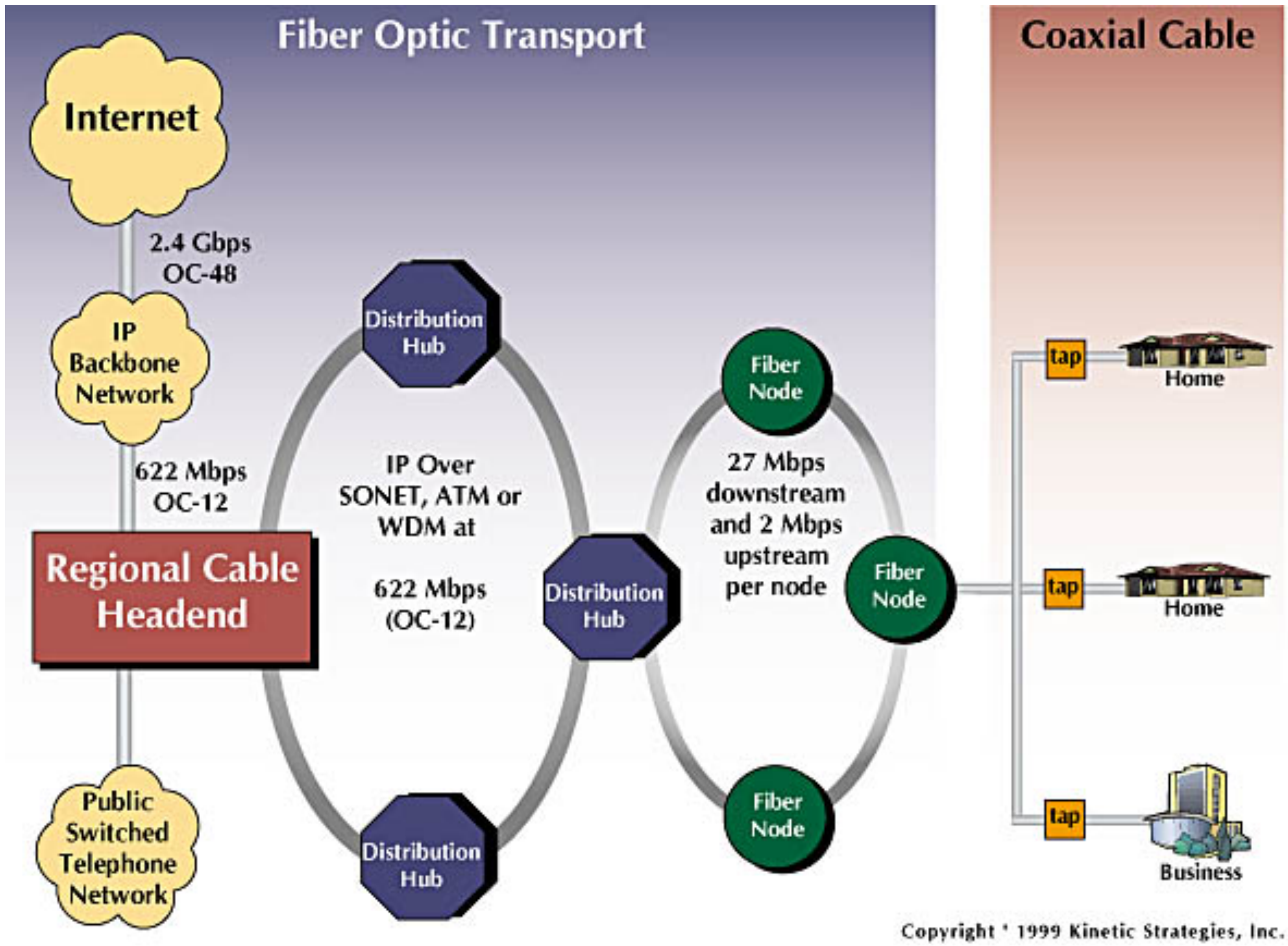
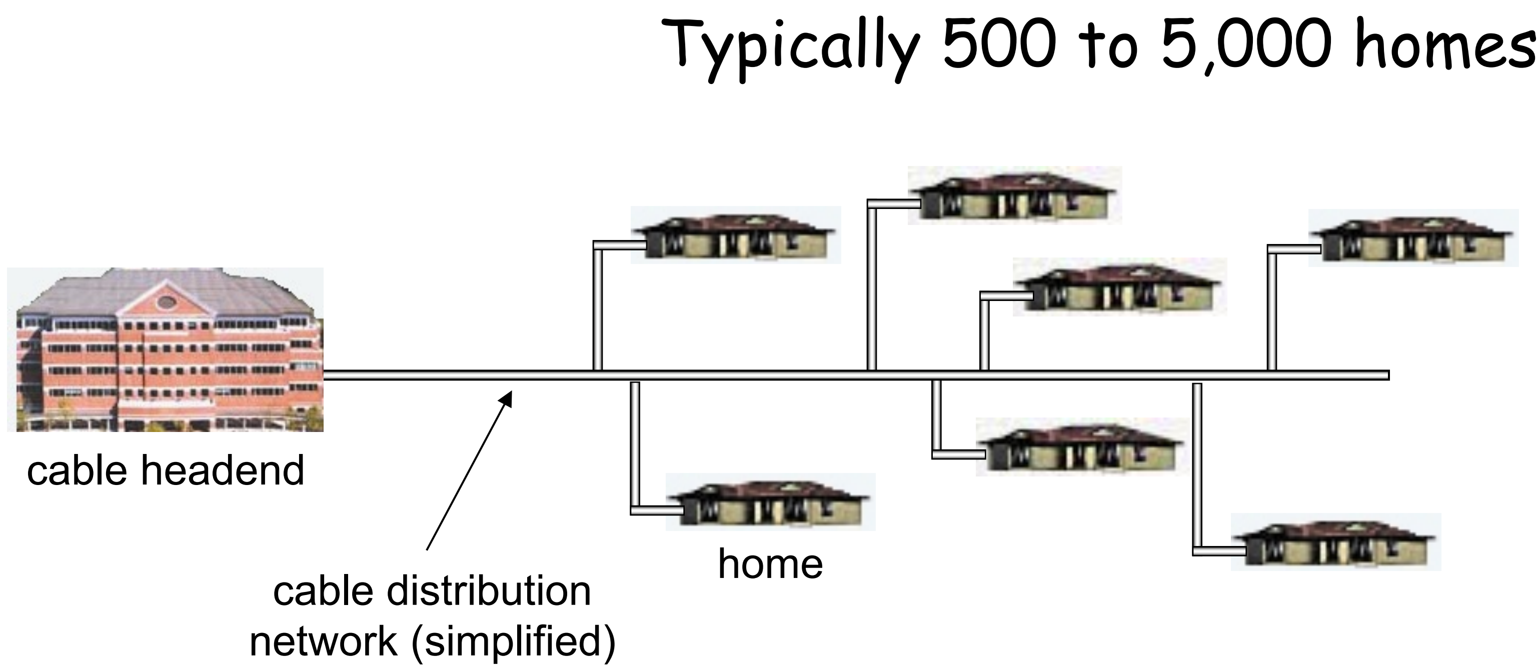
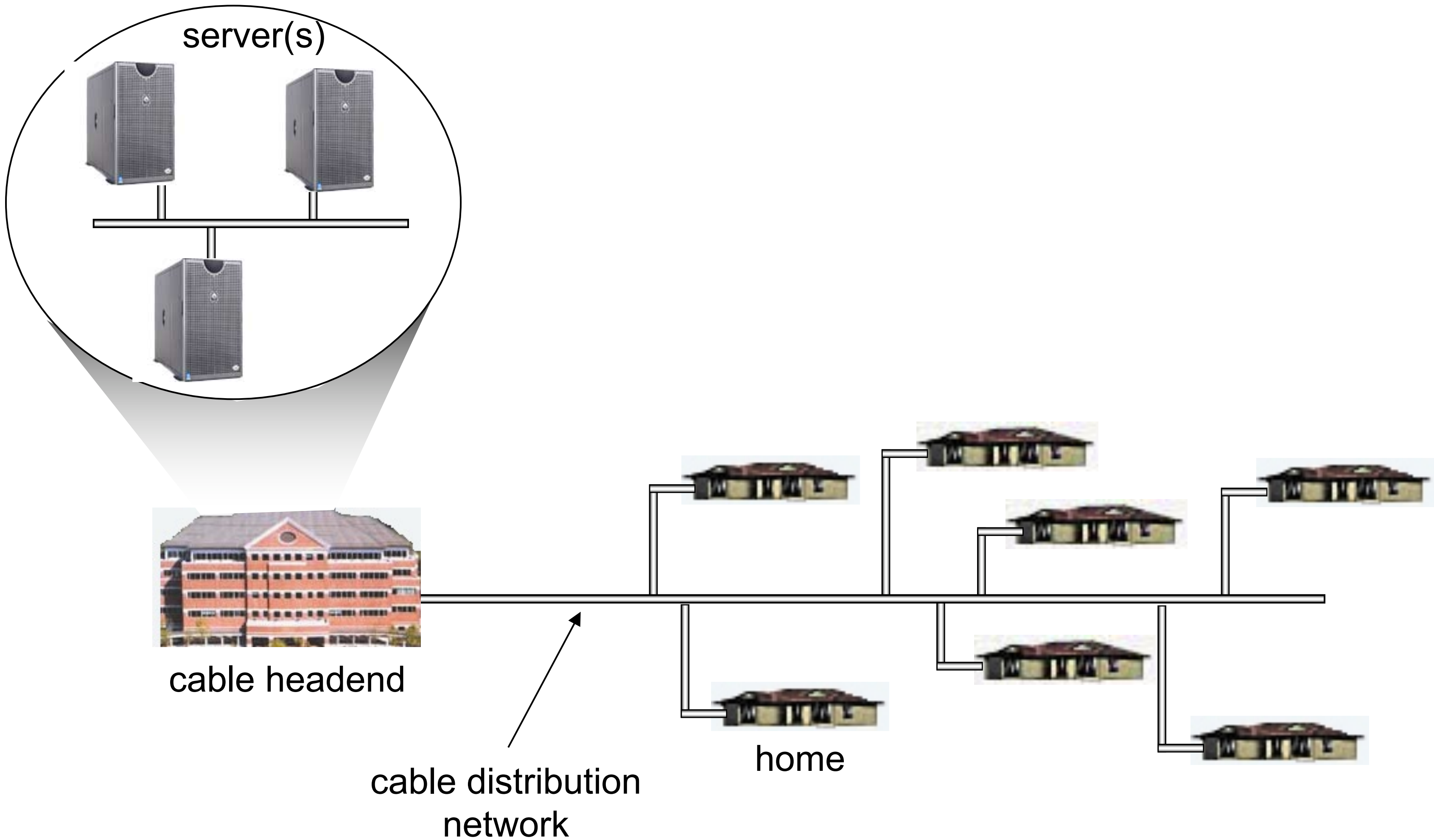


Diagram: http://www.cabledatacomnews.com/cmic/diagram.html

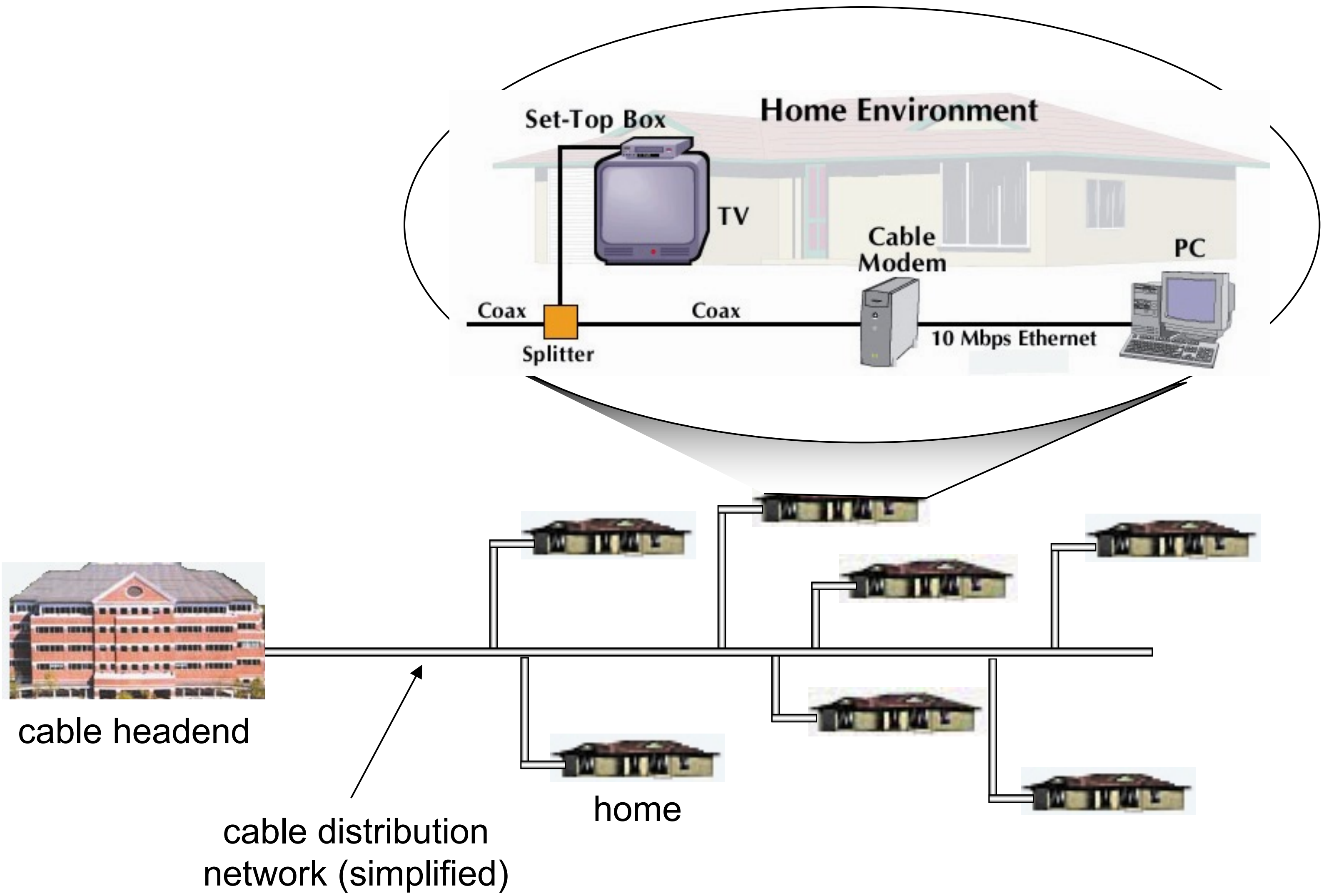
Cable Network Architecture: Overview



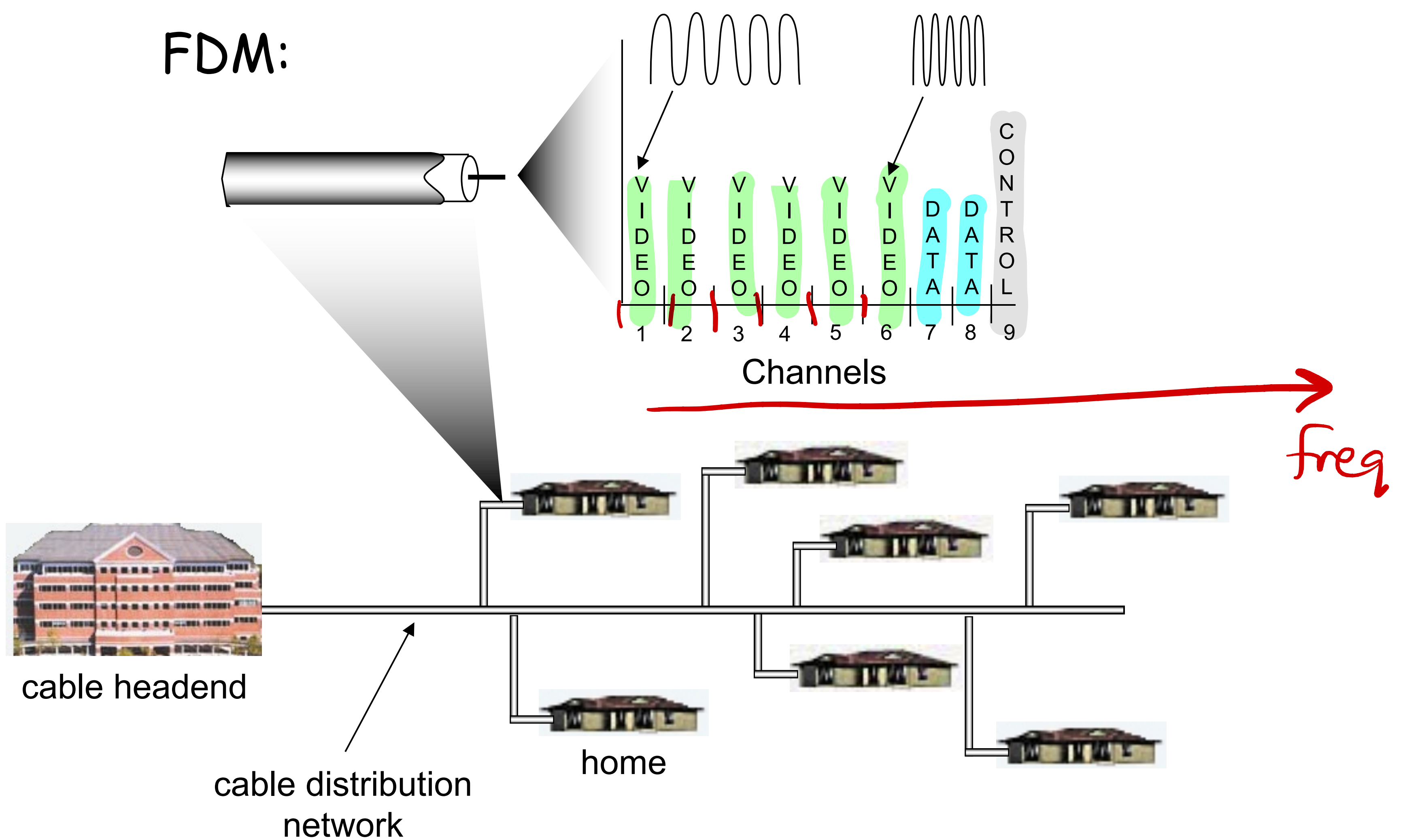
Cable Network Architecture: Overview



Cable Network Architecture: Overview



Cable Network Architecture: Overview



DSL vs Cable Modem

- DSL is point to point

  Thus data rate does not reduce when neighbor uses his/her DSL

- But, DSL uses twisted-pair, and transmission technology cannot support more than ~10Mbps

- Cable Modems share the pipe to the cable headend.

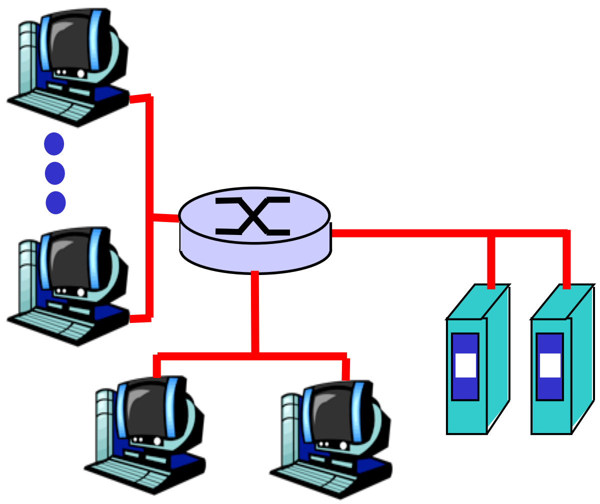
  Thus, your data rate can reduce when neighbors are surfing concurrently

- However, fibre optic lines have significantly higher data rate (fat pipe)

  Even if other users, data rate may still be higher

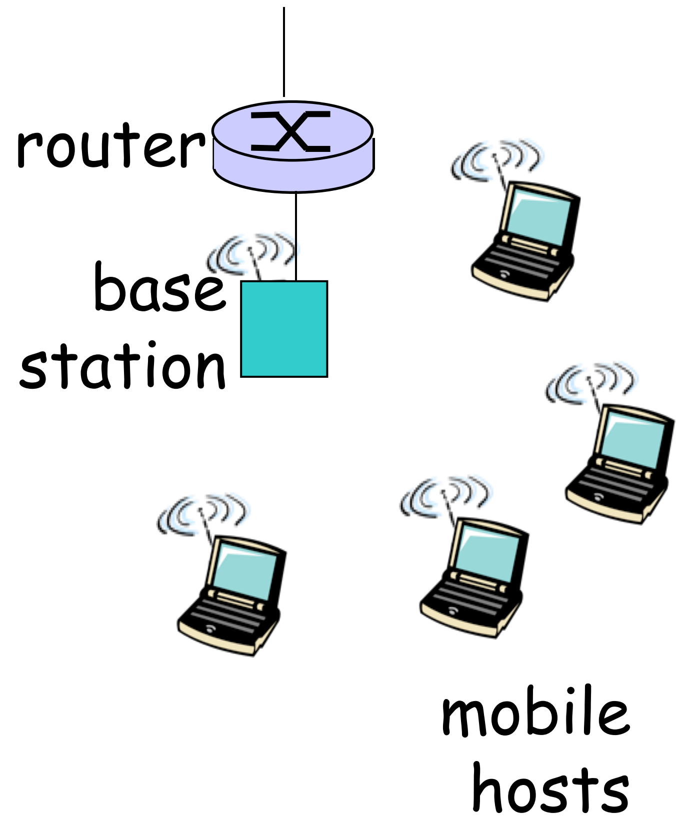
Company access: local area networks

- company/univ local area network (LAN) connects end system to edge router
- Ethernet:
  - shared or dedicated link connects end system and router
  - 10 Mbs, 100Mbps, Gigabit Ethernet
- LANs: chapter 5

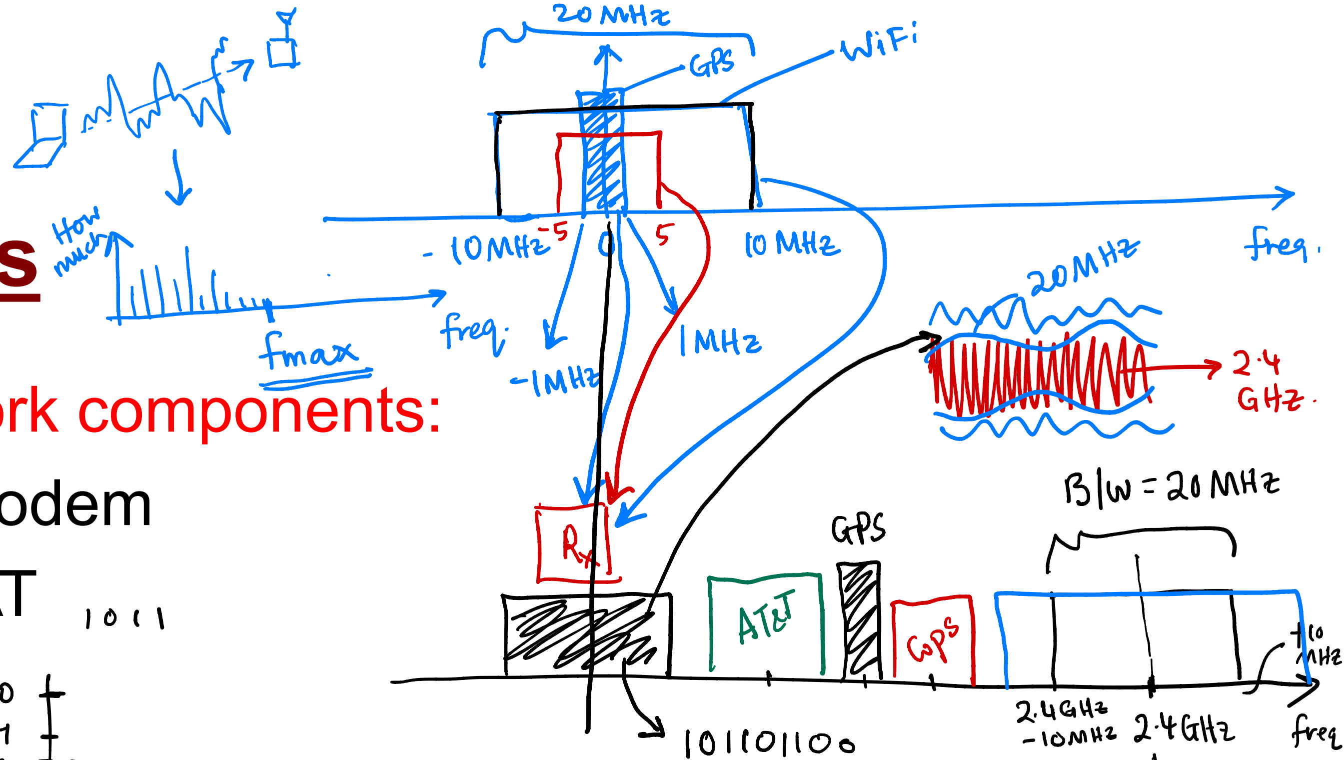


Wireless access networks

- shared *wireless* access network connects end system to router
  - via base station aka "access point"
- wireless LANs:
  - 802.11b/g (WiFi): 11 or 54 Mbps
- wider-area wireless access
  - provided by telco operator
  - 3G ~ 384 kbps
    - Will it happen??
  - GPRS in Europe/US

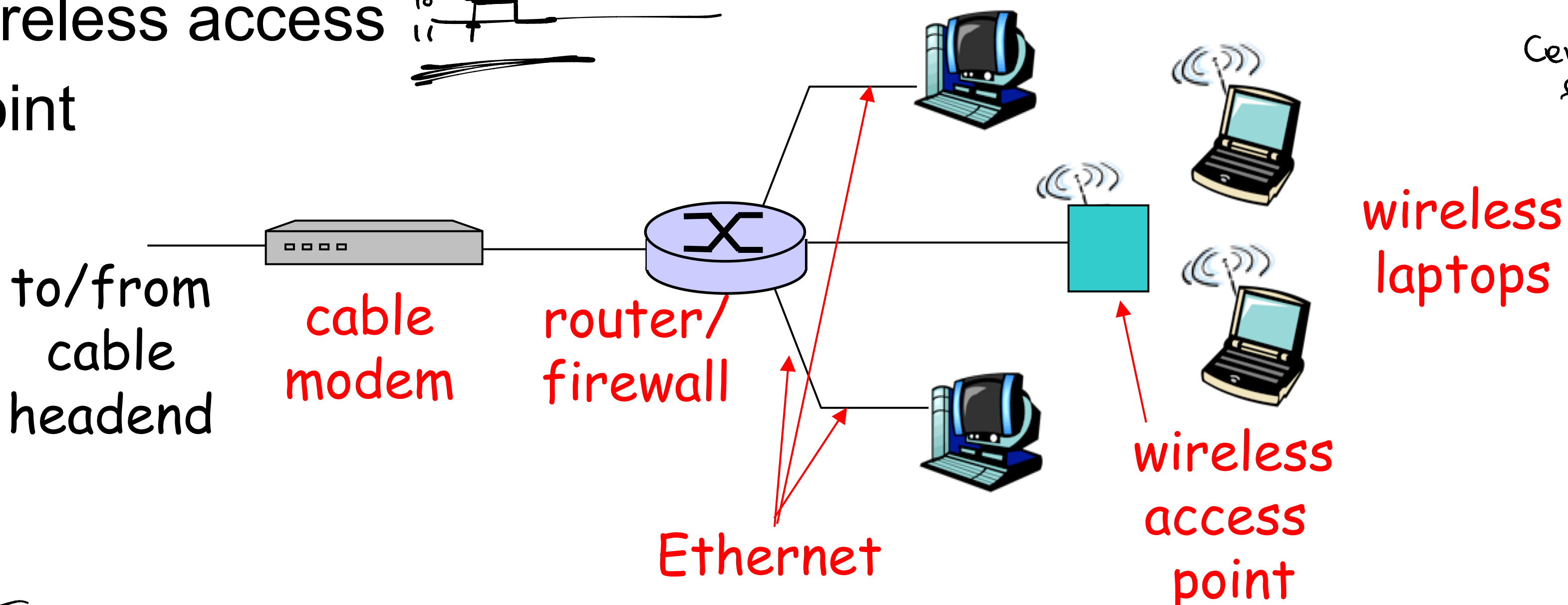


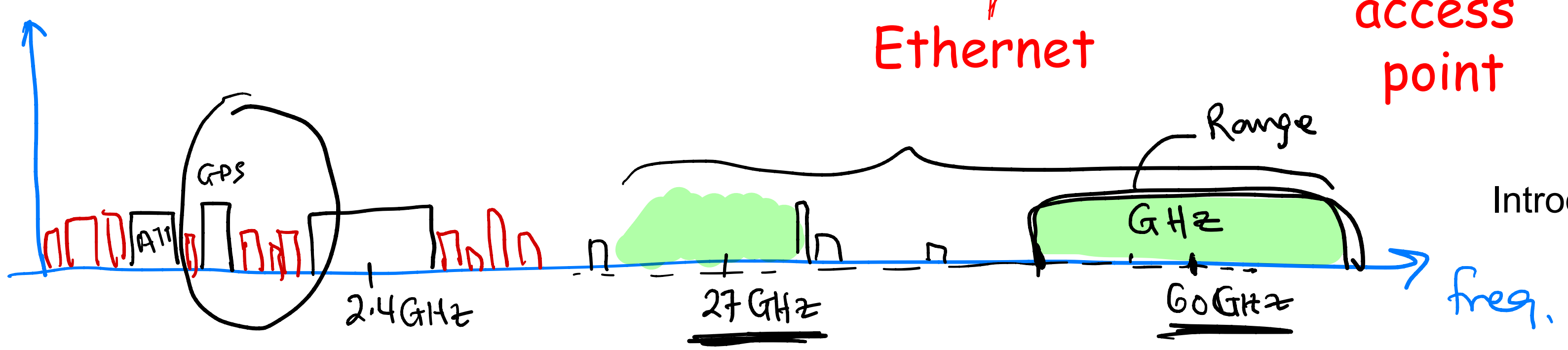
Home networks

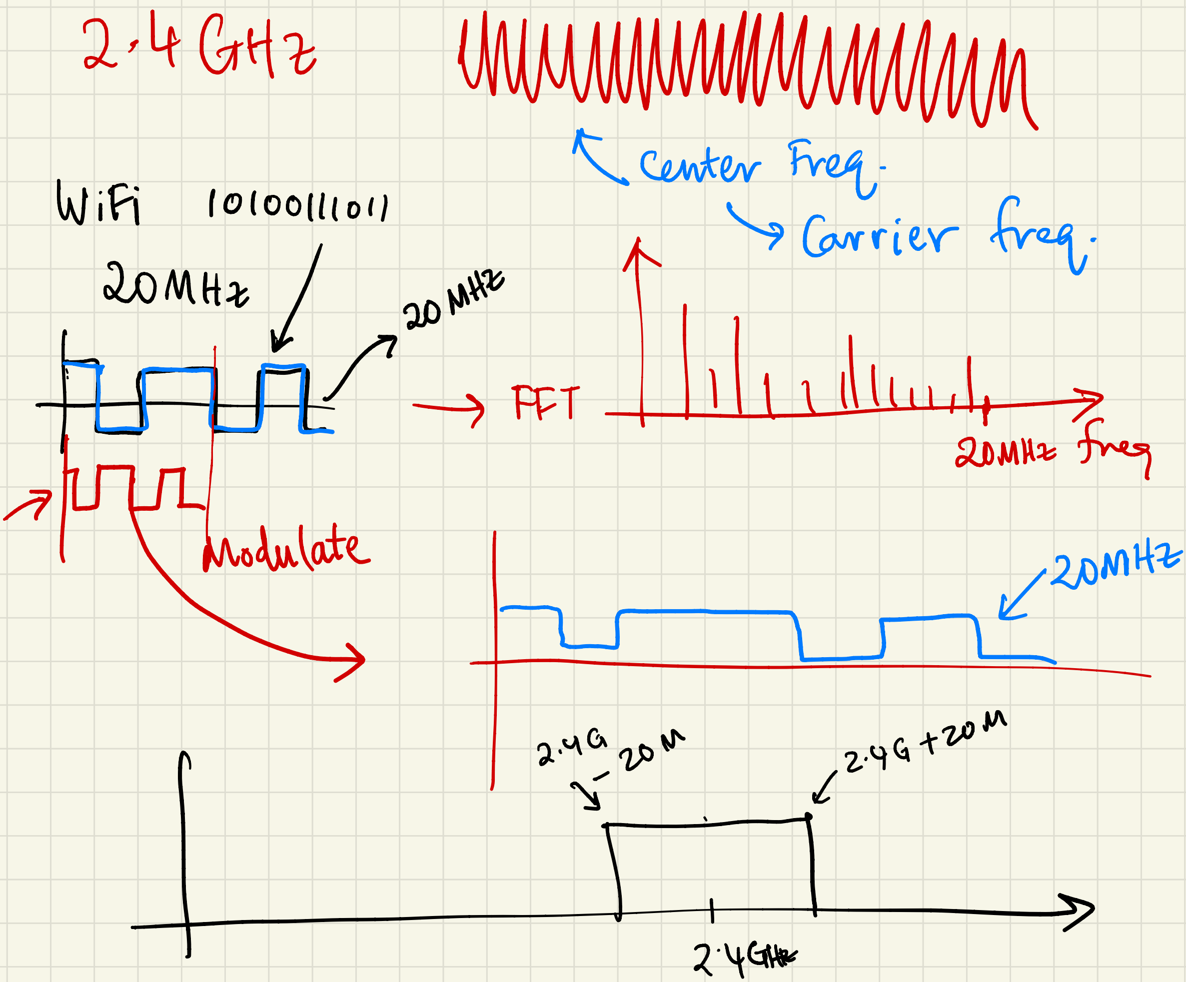


Typical home network components:

- ADSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point







2.4 GHz
Center Freq.
Carrier freq.
WiFi 1010011011
20MHz
20 MHz
FFT
20MHz freq
Modulate
20MHz
2.4G - 20M
2.4G + 20M
2.4GHz

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network access and physical media

1.4 Network core

1.5 Internet structure and ISPs

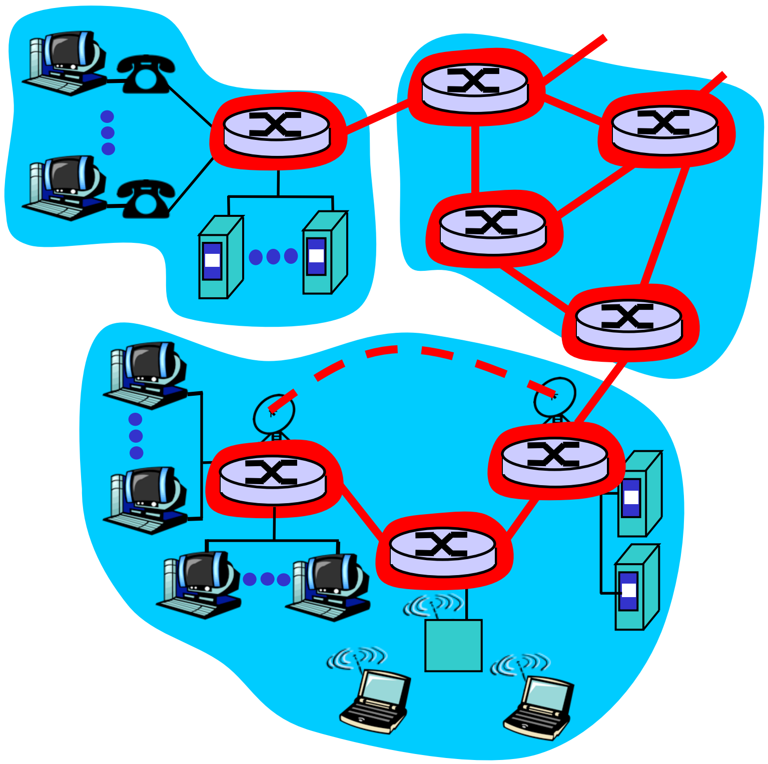
1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

The Network Core

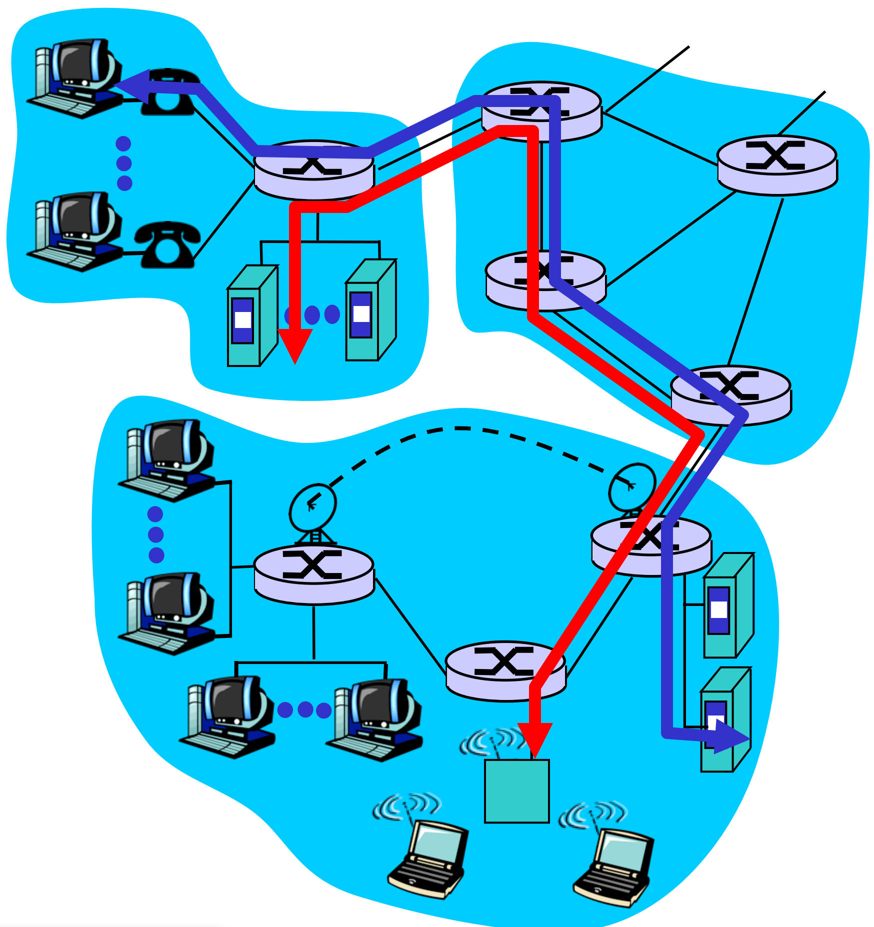
- mesh of interconnected routers
- *the* fundamental question: how is data transferred through net?
  - circuit switching: dedicated circuit per call: telephone net
  - packet-switching: data sent thru net in discrete "chunks"



Network Core: Circuit Switching

End-end resources reserved for "call"

- link bandwidth, switch capacity
- dedicated resources: no sharing
- circuit-like (guaranteed) performance
- call setup required



Analogy: When president travels, a CS path set up.

Network Core: Circuit Switching

network resources (e.g., bandwidth) divided into "pieces"

- pieces allocated to calls
- resource piece *idle* if not used by owning call *(no sharing)*

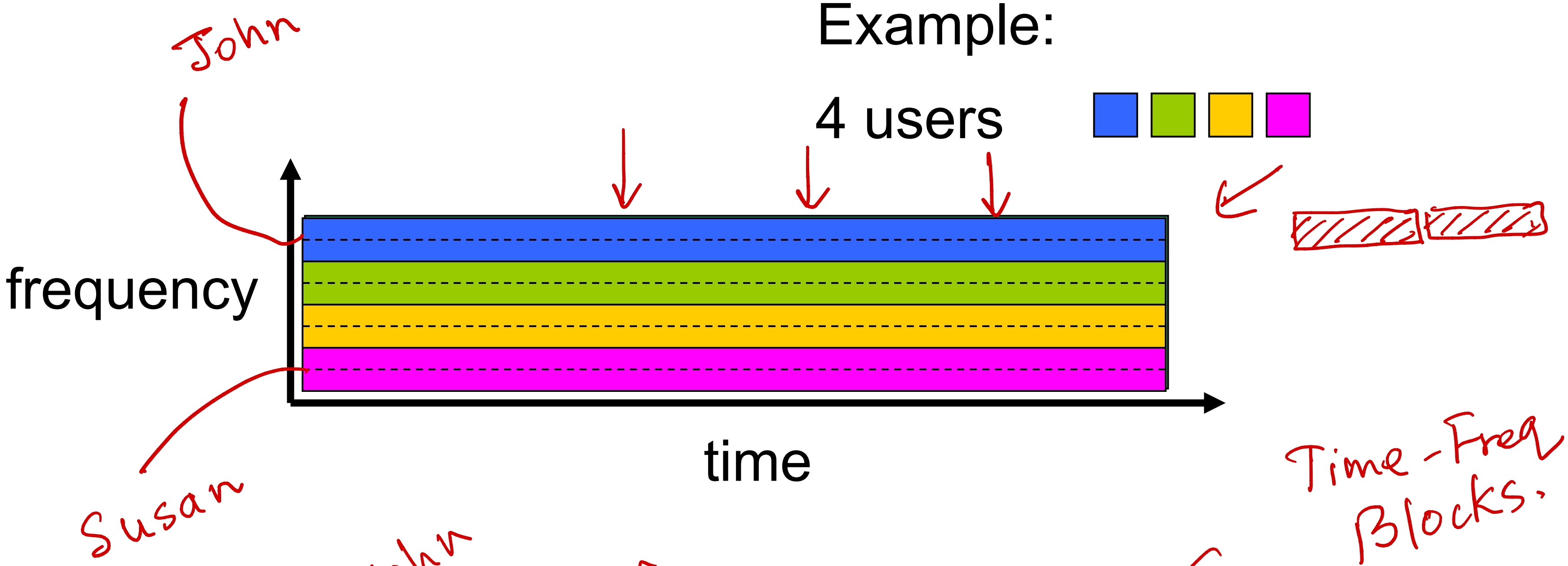
- dividing link bandwidth into "pieces"
  - frequency division
  - time division

Circuit Switching: FDM and TDM

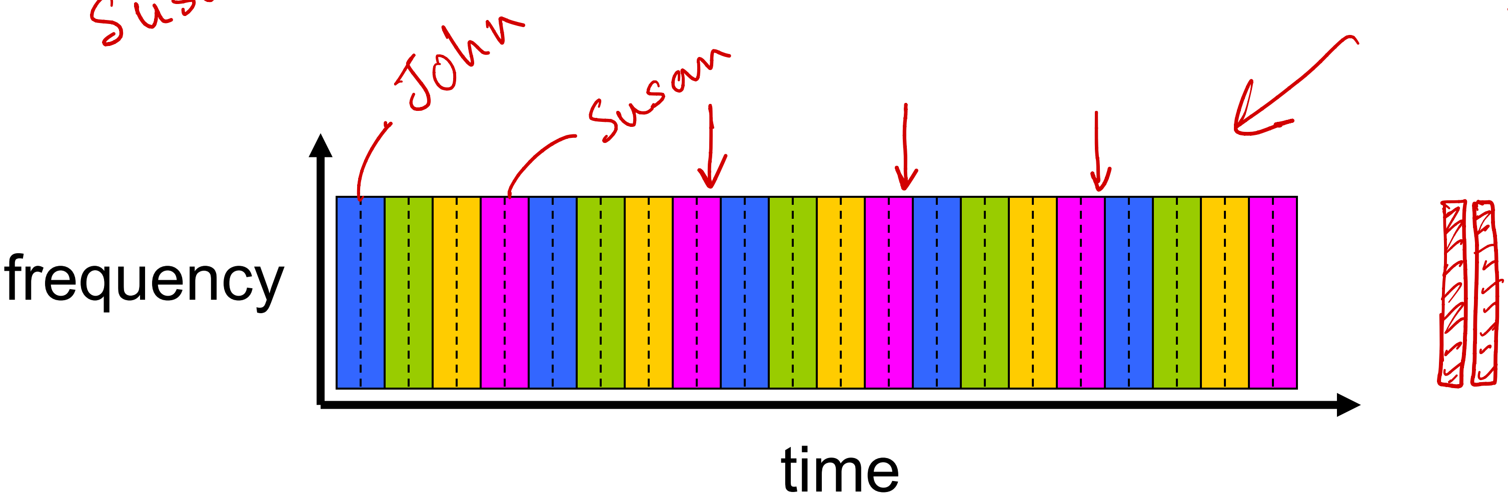
Example:

4 users

FDM



TDM

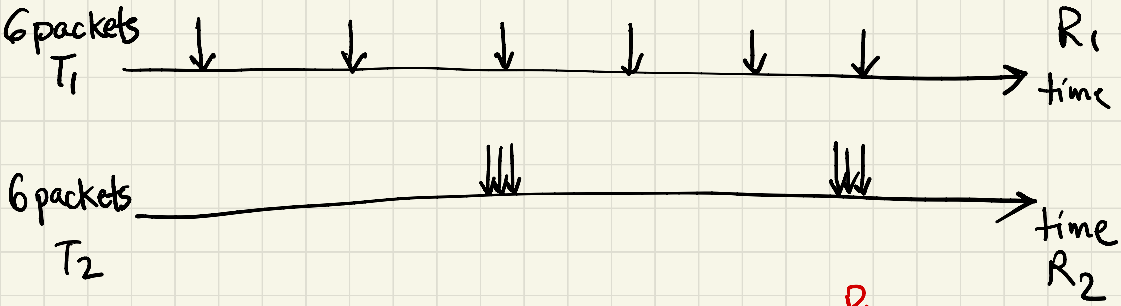


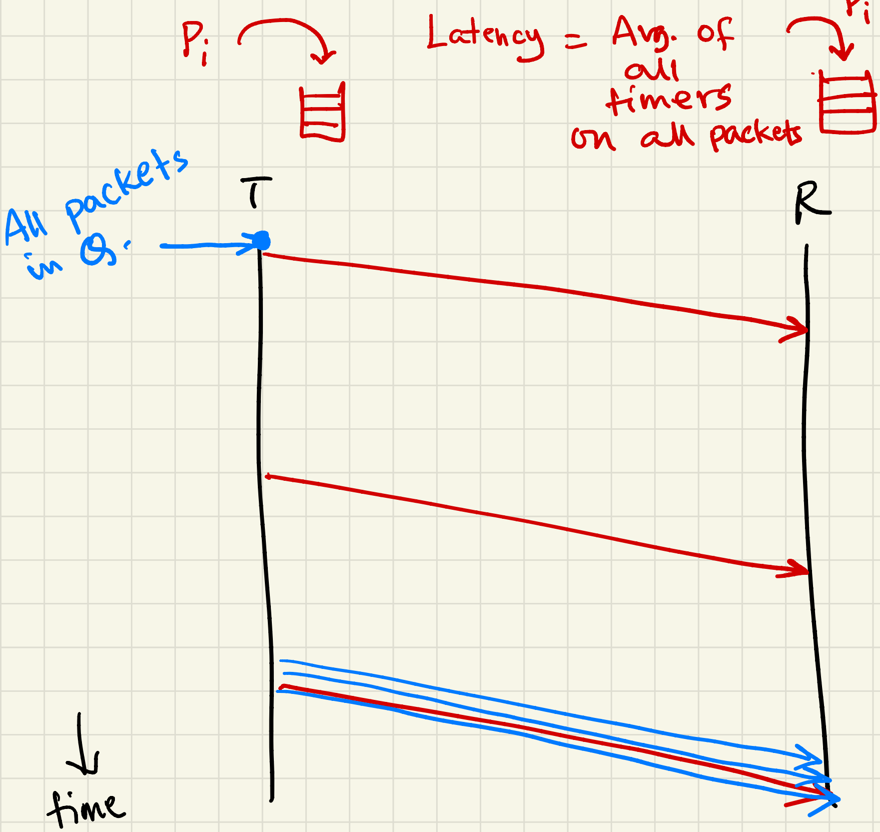
Throughput

packets / unit time delivered.

Latency.

time to deliver packets





FDM Vs TDM

- What are the tradeoffs?
  - (Dis)Advantage of dividing frequency ?
  - (Dis)Advantage of dividing time ?

Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit

Let's work it out!

Another numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses FDM with 24 channels (frequencies)
  - 500 msec to establish end-to-end circuit

Let's work it out!

Network Core: Packet Switching

each end-end data stream divided into *packets*

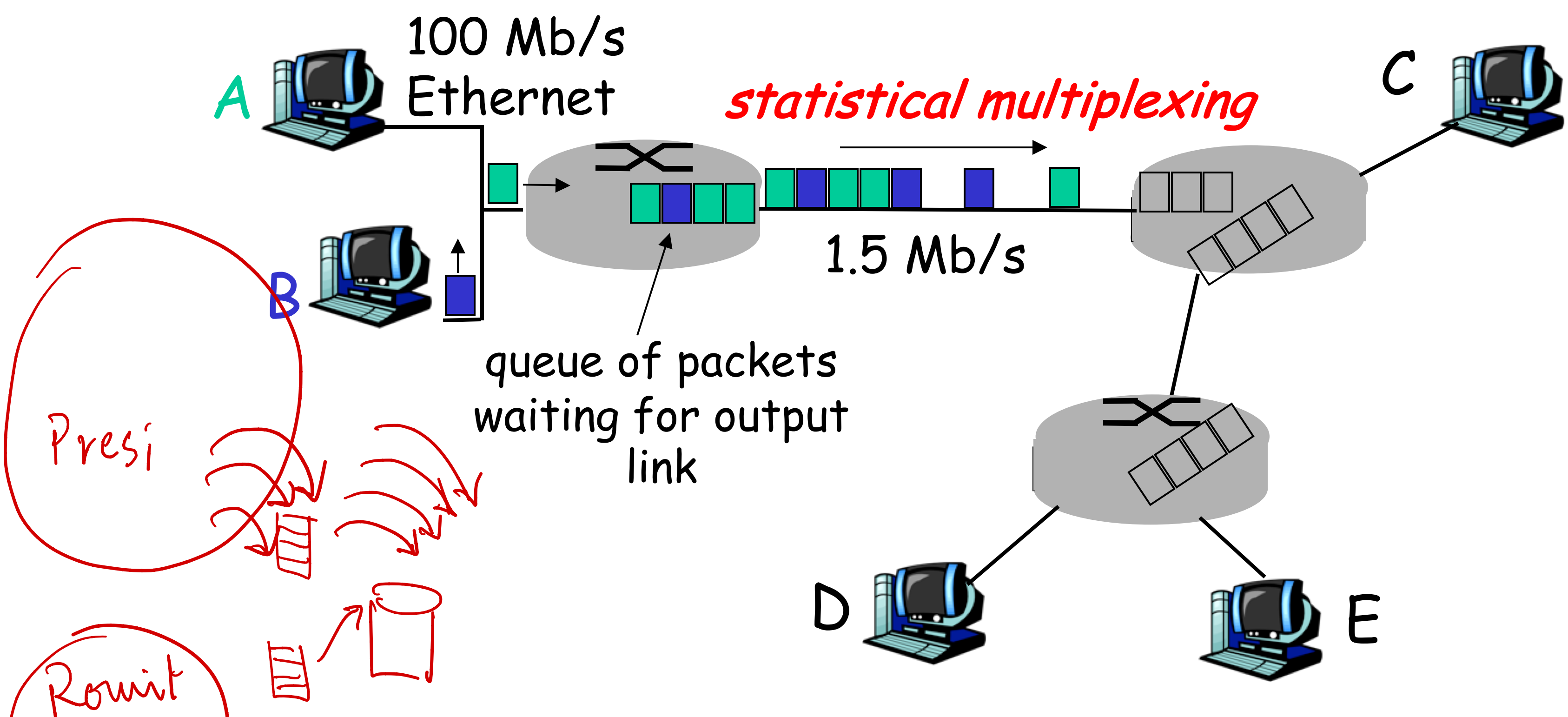
- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

Bandwidth division into "pieces"
Dedicated allocation
Resource reservation

resource contention:

- aggregate resource demand can exceed amount available
  - Packets queue up
- store and forward: packets move one hop at a time
  - Node receives complete packet before forwarding

Packet Switching: Statistical Multiplexing



Sequence of A & B packets does not have fixed pattern, shared on demand ➡ ***statistical multiplexing***.

TDM: each host gets same slot in revolving TDM frame.

Compare

Thoughts on tradeoffs between packet switching and circuit switching?

Which one would you take?

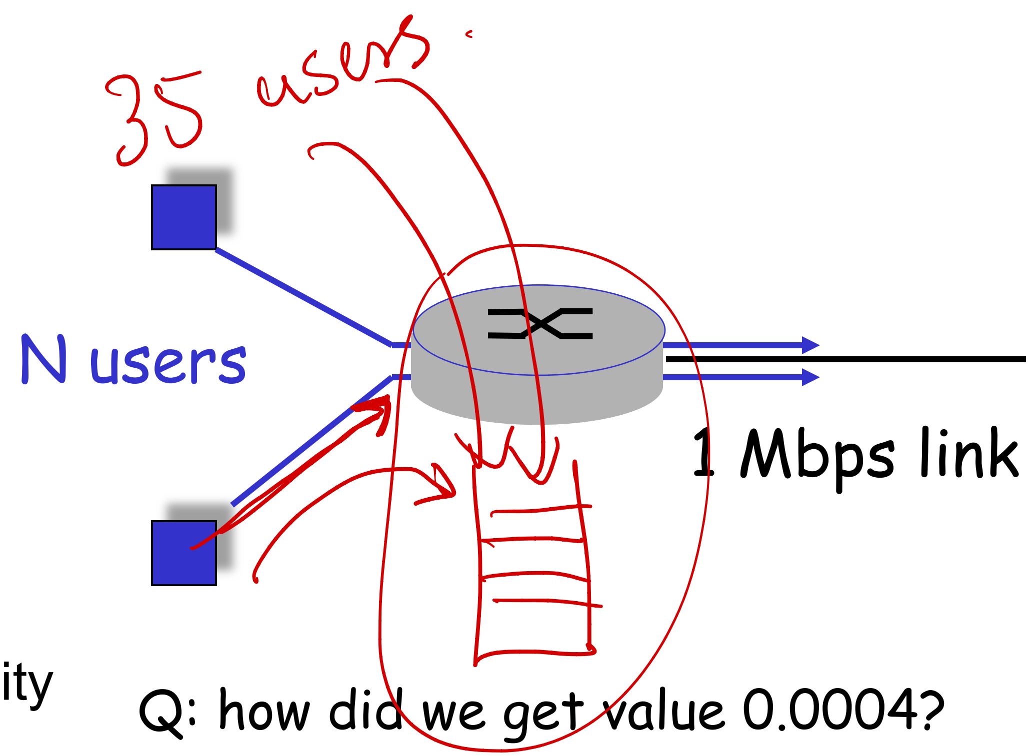
Under what circumstances?

Why?

Packet switching versus circuit switching

Packet switching allows more users to use network!

- 1 Mb/s link
- each user:
  - 100 kb/s when "active"
  - active 10% of time
- circuit-switching:
  - 10 users
- packet switching:
  - with 35 users, probability > 10 active less than .0004



Q: how did we get value 0.0004?

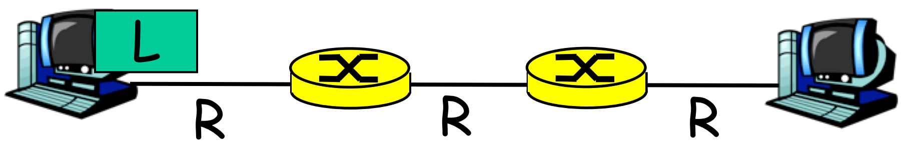
Packet switching versus circuit switching

Is packet switching a "slam dunk winner?"

- Great for bursty data
  - resource sharing
  - simpler, no call setup
- Excessive congestion: packet delay and loss
  - protocols needed for reliability, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still unsolved (chapter 7)

Why?

Packet-switching: store-and-forward



- Takes L/R seconds to transmit (push out) packet of L bits on to link or R bps
- Entire packet must arrive at router before it can be transmitted on next link: *store and forward*
- delay = 3L/R (assuming zero propagation delay)

Example:

- L = 7.5 Mbits
- R = 1.5 Mbps
- delay = 15 sec

more on delay shortly …

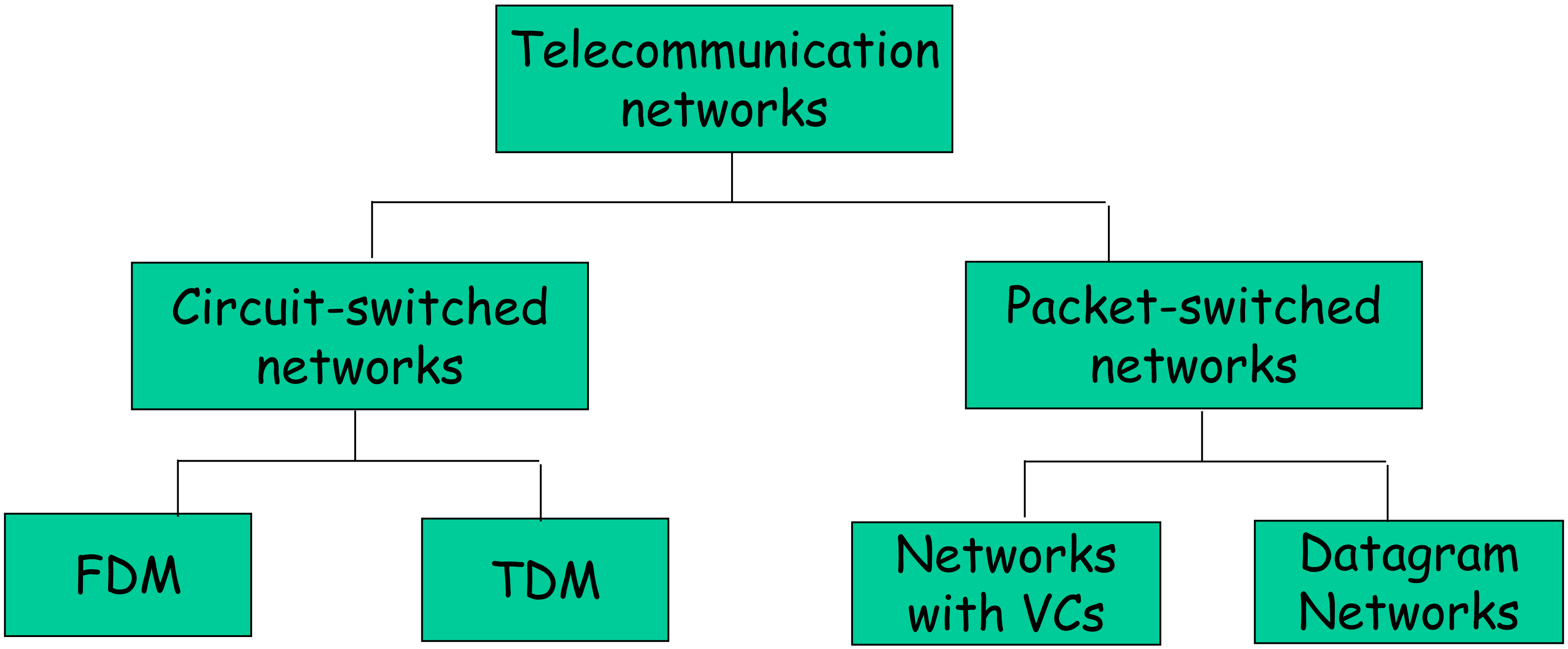
Packet-switched networks: forwarding

- *Goal:* move packets through routers from source to destination
  - we'll study several path selection (routing) algorithms (chap 4)

- datagram network:
  - *destination address* in packet determines next hop
  - routes may change during session
  - analogy: driving, asking directions

- virtual circuit network:
  - packet carries tag (virtual circuit ID), tag determines next hop
  - fixed path determined at *call setup time*, remains fixed thru call
  - *routers maintain per-call state*

Network Taxonomy



- Datagram network is *not* either connection-oriented or connectionless.
- Internet provides both connection-oriented (TCP) and connectionless services (UDP) to apps.

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network access and physical media

1.4 Network core

1.5 Internet structure and ISPs

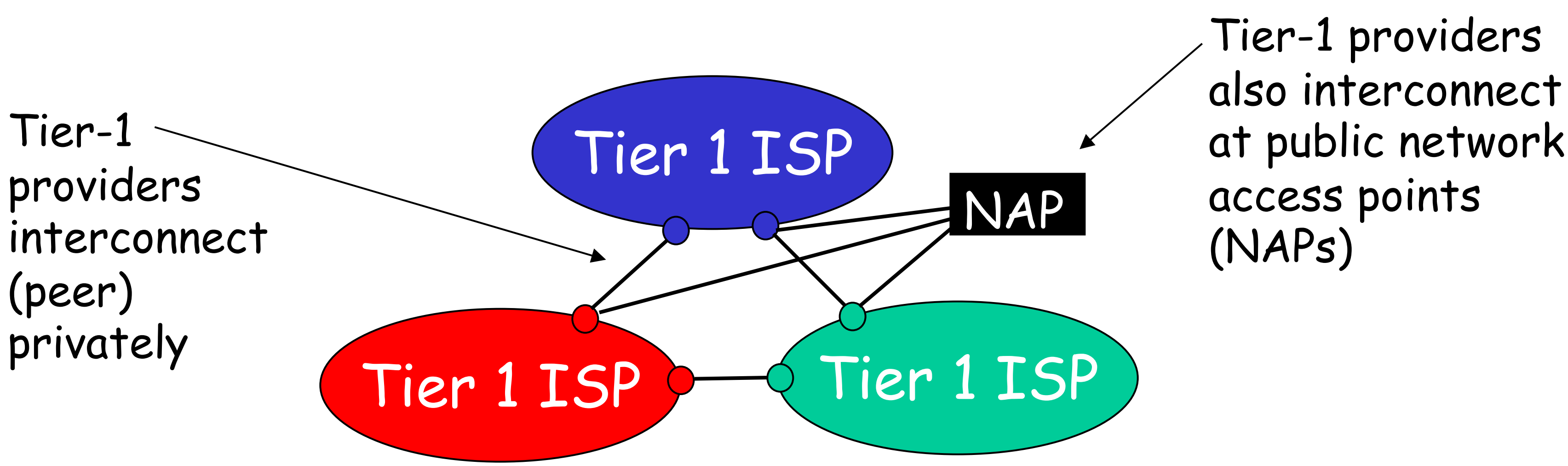
1.6 Delay & loss in packet-switched networks

1.7 Protocol layers, service models

1.8 History

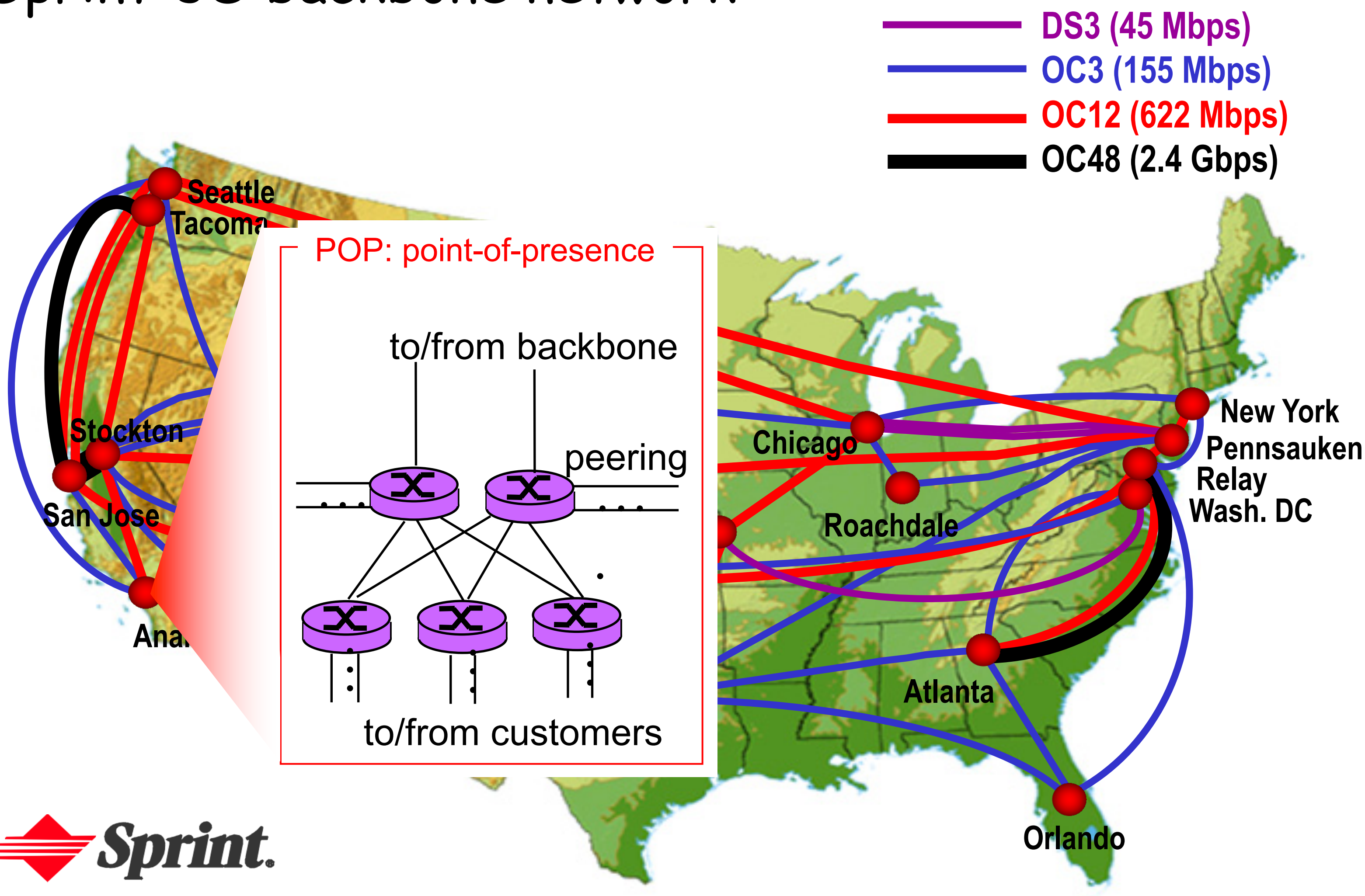
Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (e.g., MCI, Sprint, AT&T, Cable and Wireless), national/international coverage
  - treat each other as equals



Tier-1 ISP: e.g., Sprint

Sprint US backbone network

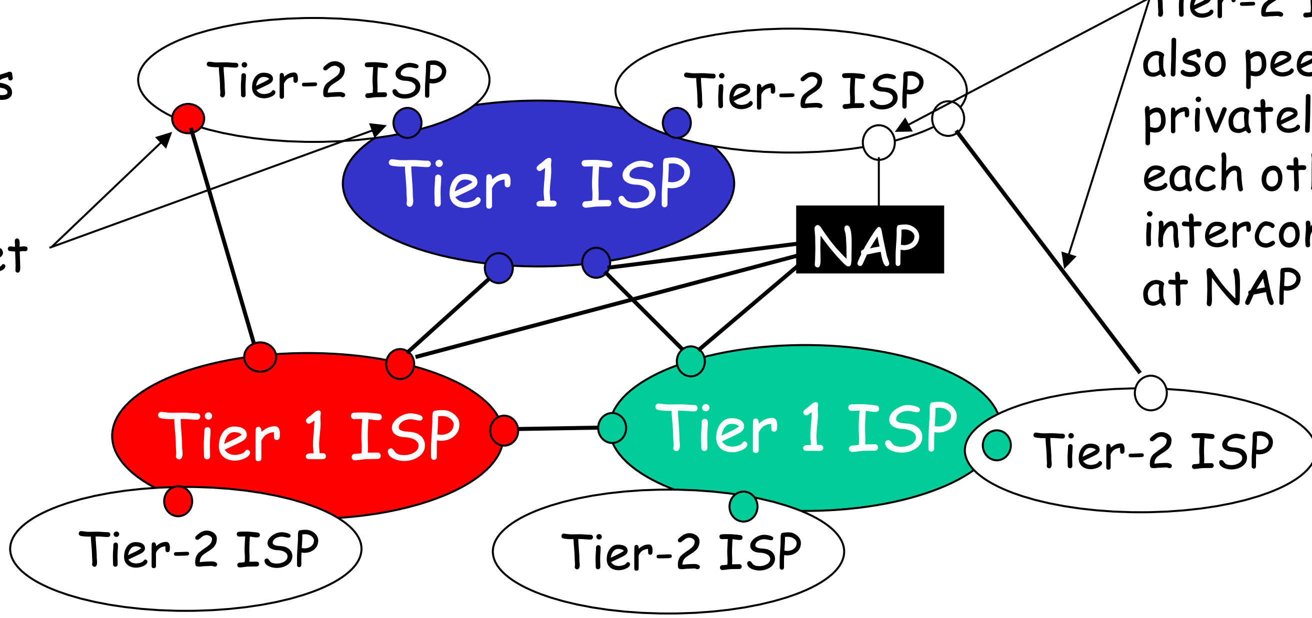


Internet structure: network of networks

- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet

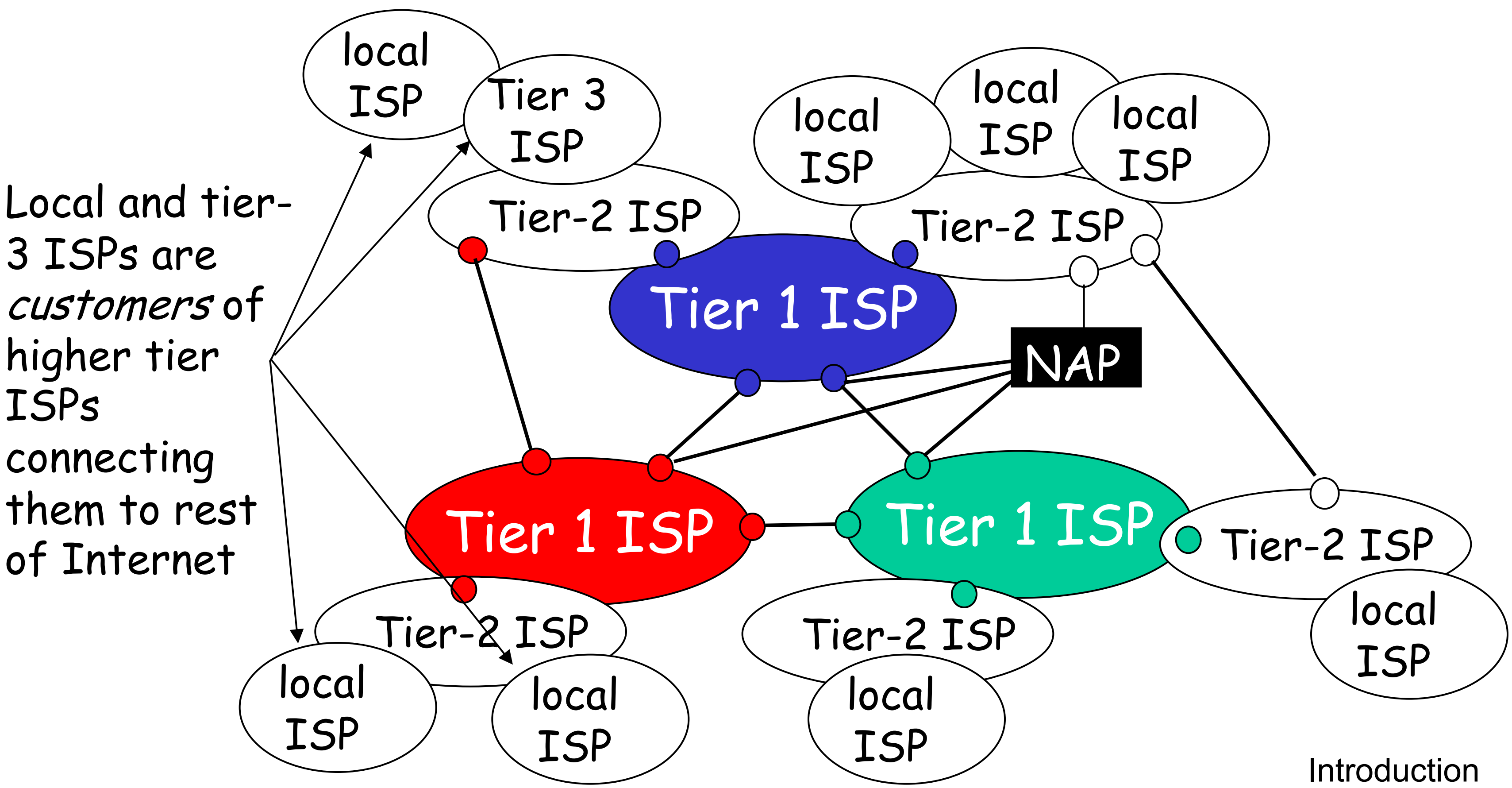
- tier-2 ISP is *customer* of tier-1 provider



Tier-2 ISPs also peer privately with each other, interconnect at NAP

Internet structure: network of networks

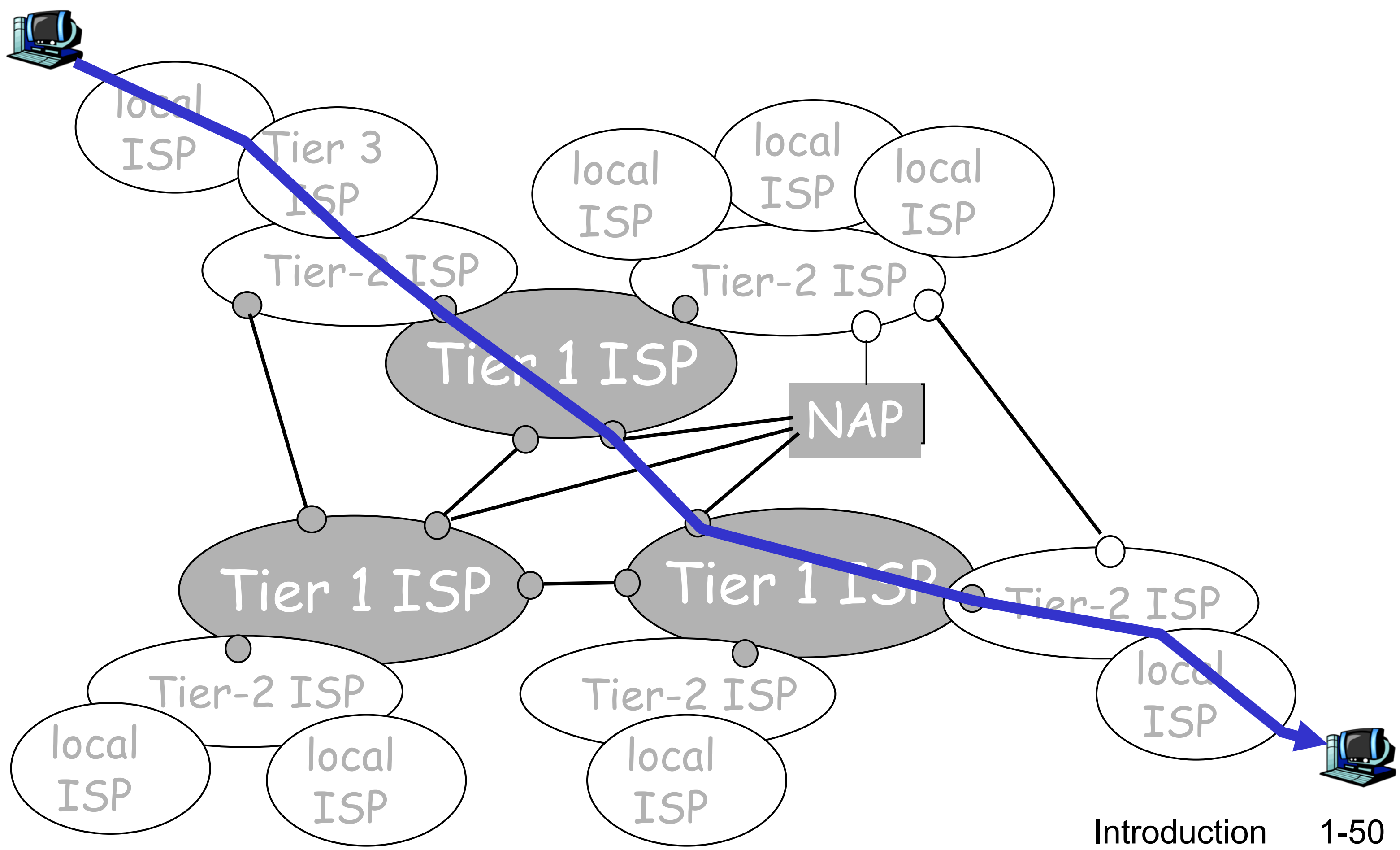
- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)



Internet structure: network of networks

- a packet passes through many networks!

  local (taxi) → T1 (bus) → T2 (domestic) → T3 (international)



Organizing the giant structure

Networks are complex!

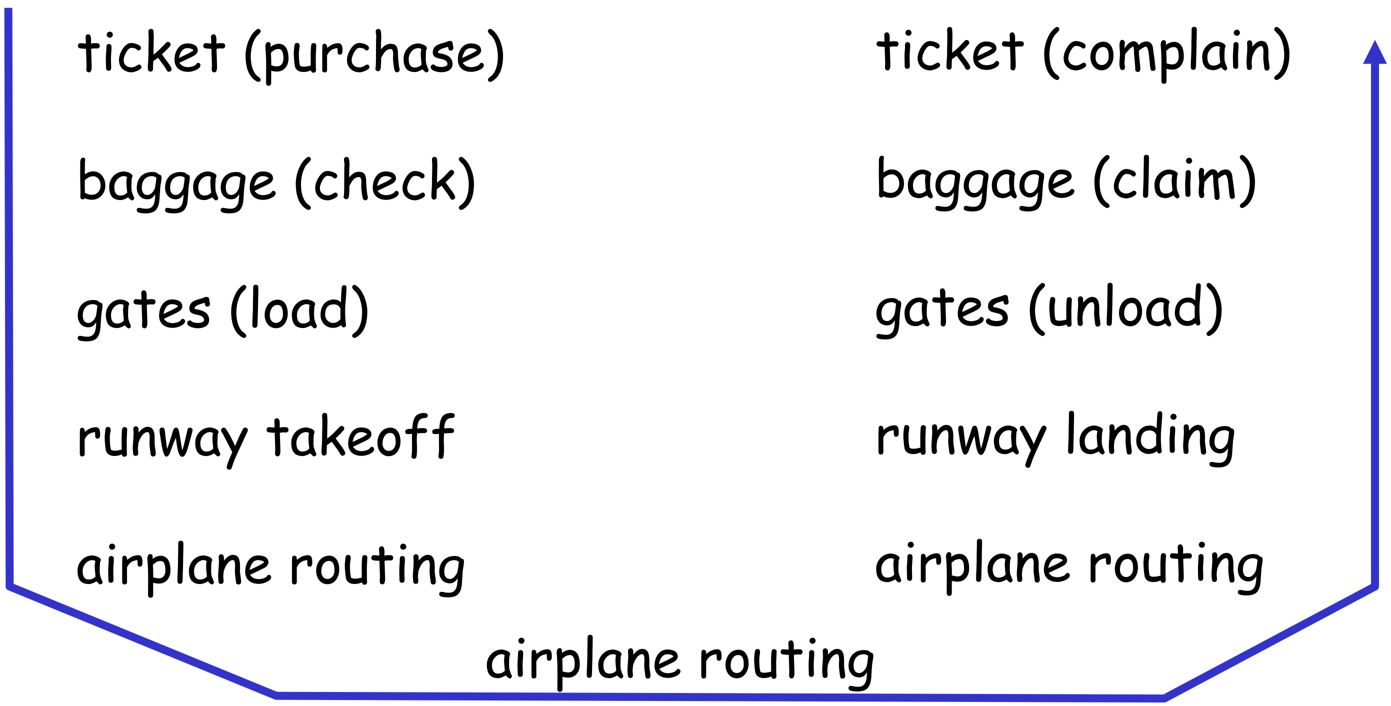
- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

Question:

Is there any hope of *organizing* structure of network?

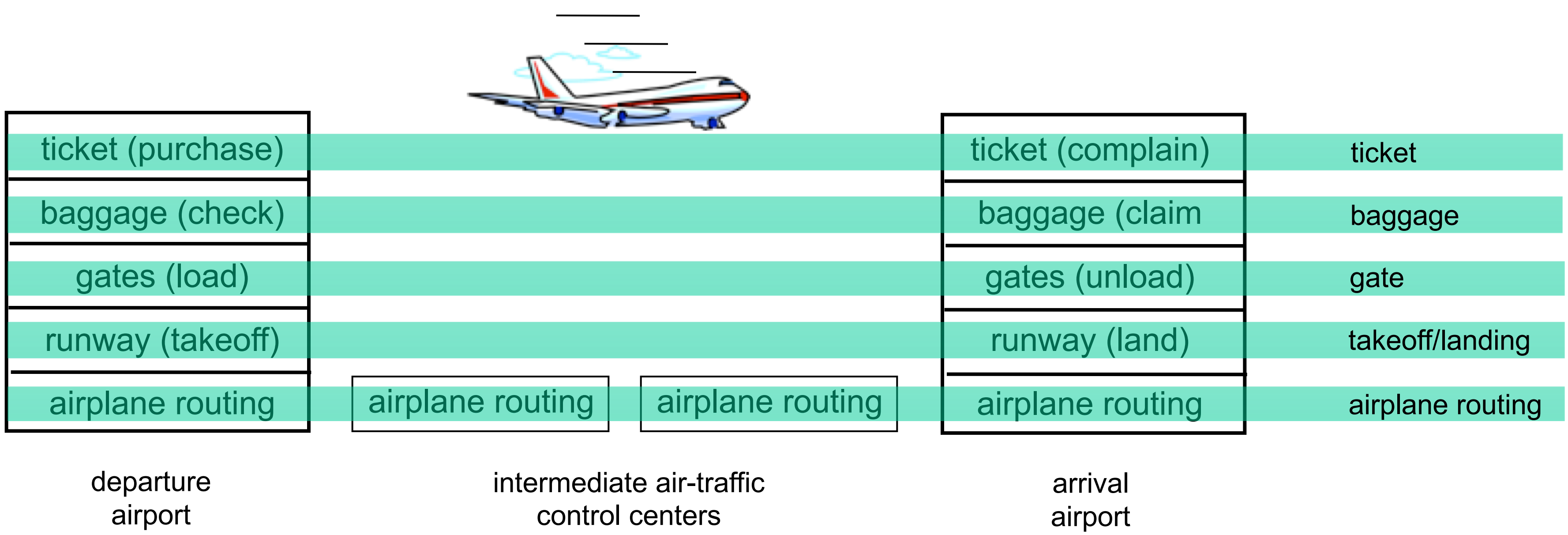
Or at least our discussion of networks?

Turn to analogies in air travel



- a series of steps

Layering of airline functionality



Layers: each layer implements a service

- layers communicate with peer layers
- rely on services provided by layer below

Why layering?

- Explicit structure allows identification, relationship of complex system's pieces

- Modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., runway delay (wheels up time) depends on clearance of destination runway … doesn't change the baggage tagging systems … or flight to gate assignment

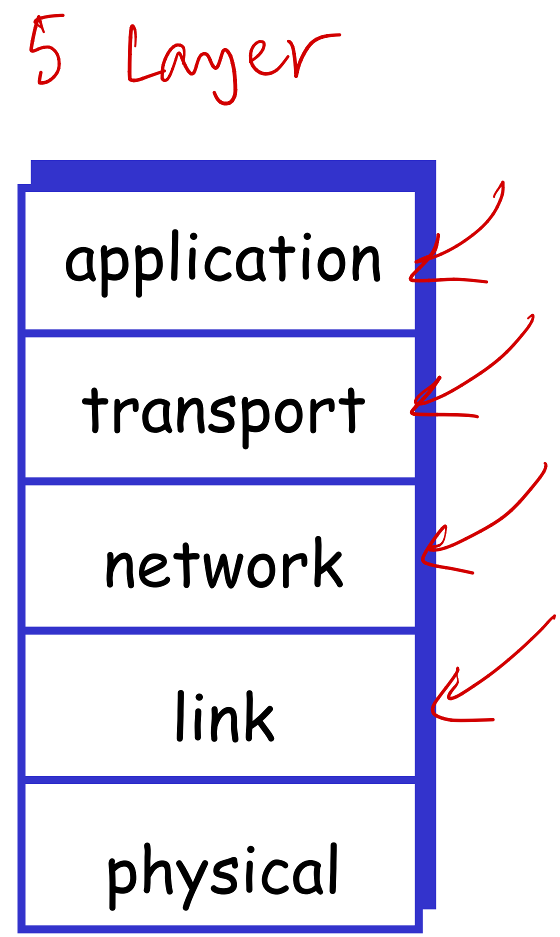
Protocol "Layers"

- Service of each layer encapsulated
- Universally agreed services called
  PROTOCOLS

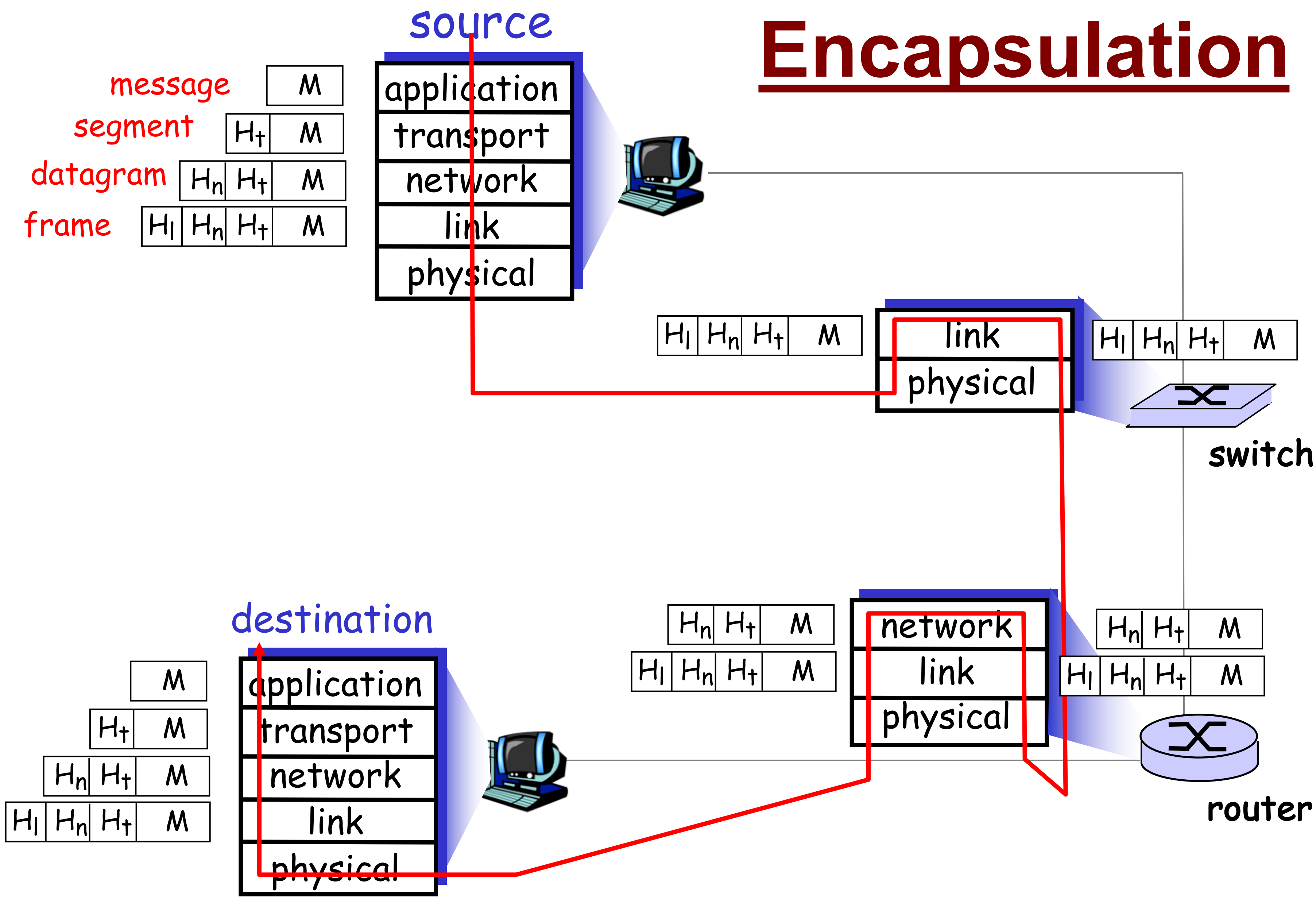
A large part of this course will focus on understanding protocols for networking systems

Internet protocol stack

- application: supporting network applications
  - FTP, SMTP, HTTP, DNS …
- transport: host-host data transfer
  - TCP, UDP …
- network: routing of datagrams from source to destination
  - IP, BGP, routing protocols …
- link: data transfer between neighboring network elements
  - PPP, Ethernet, WiFi, Bluetooth …
- physical: bits "on the wire"
  - OFDM, DSSS, CDMA, Coding …



Encapsulation



Success of Layering

- Protocol stack successful in Internet
- Internet uses wired physical layer links
  - Very reliable
  - Bit Error Rate (BER) = 10^{-8}
- What about wireless networks
  - Very unreliable due to channel fluctuations
  - Due to co-channel interference
  - Due to external noise
- Does horizontal layering still hold ?

minus 1

Assignment # -1

Watch "City in the Sky" documentary on Netflix

You will appreciate both airline systems and
The Internet much more than you do now …

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network access and physical media

1.4 Network core

1.5 Internet structure and ISPs

1.6 Delay & loss in packet-switched networks

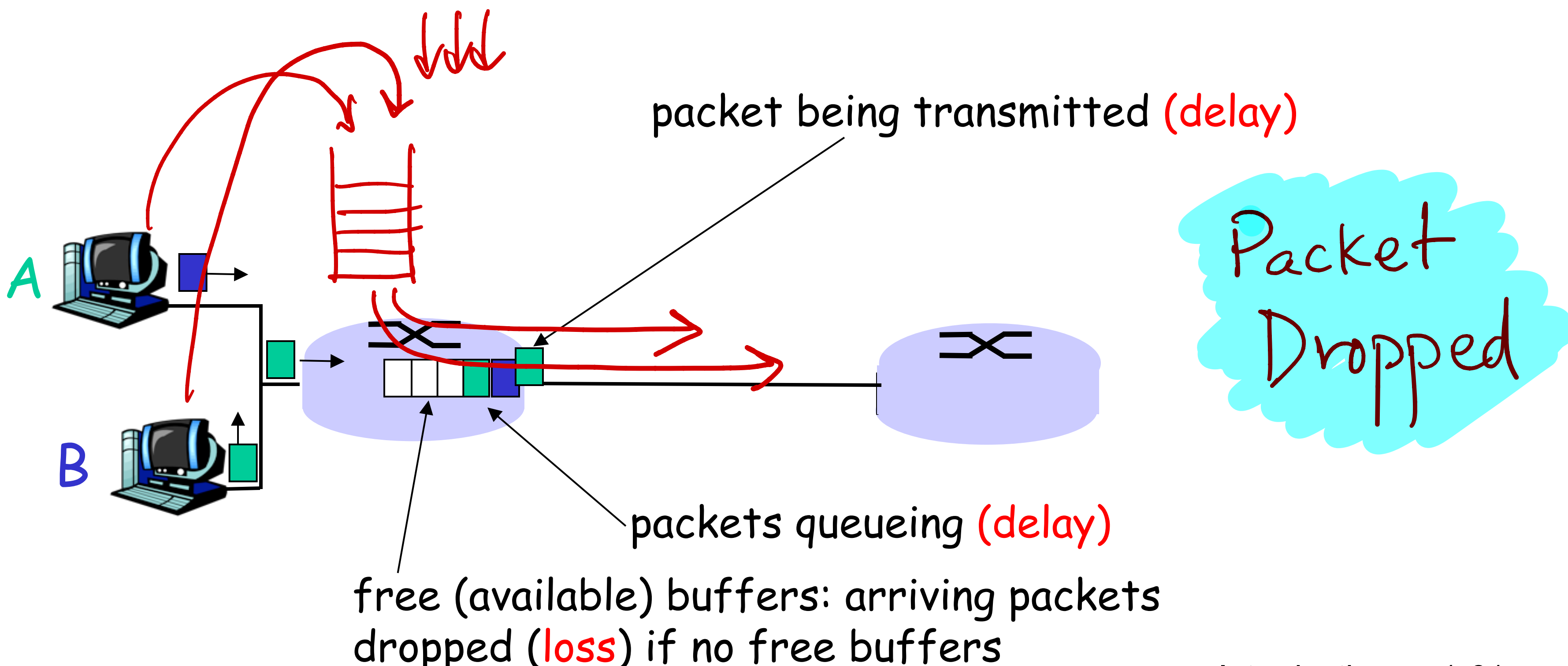
1.7 Protocol layers, service models

1.8 History

How do loss and delay occur?

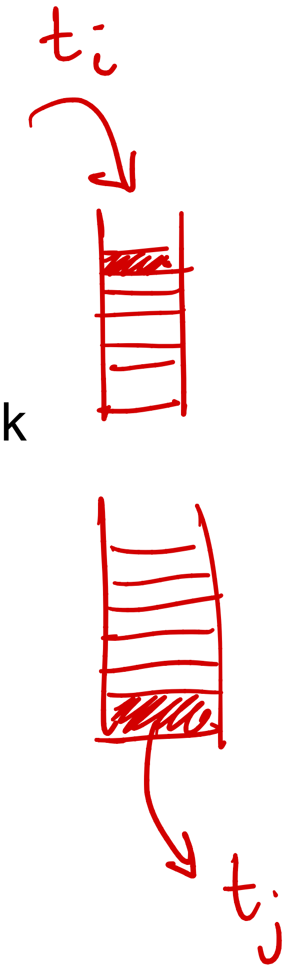
packets *queue* in router buffers

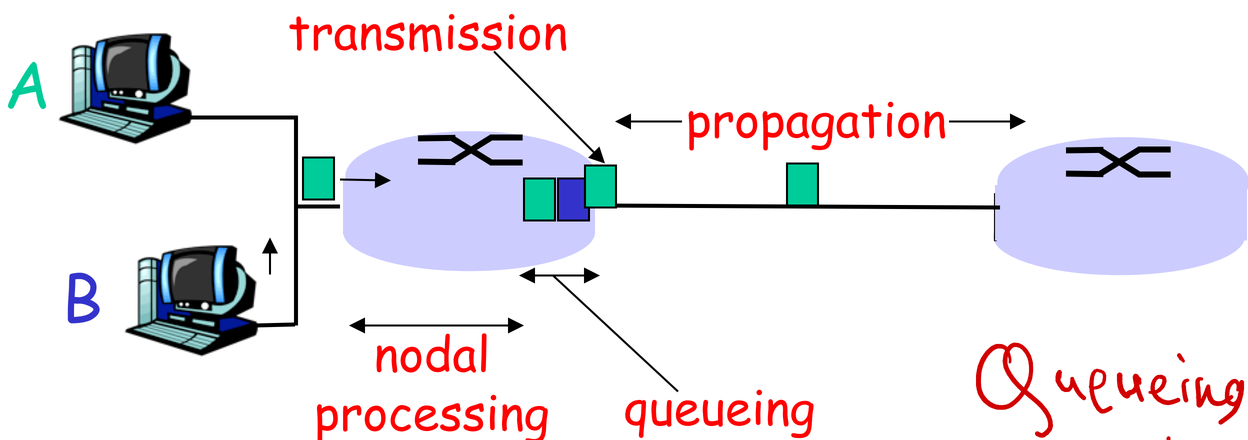
- packet arrival rate to link exceeds output link capacity
- packets queue, wait for turn



Four sources of packet delay

- 1. nodal processing:
  - check bit errors
  - determine output link
- 2. queueing
  - time waiting at output link for transmission
  - depends on congestion level of router





$t_j > t_i$

Queueing Delay $= t_j - t_i$

Delay in packet-switched networks

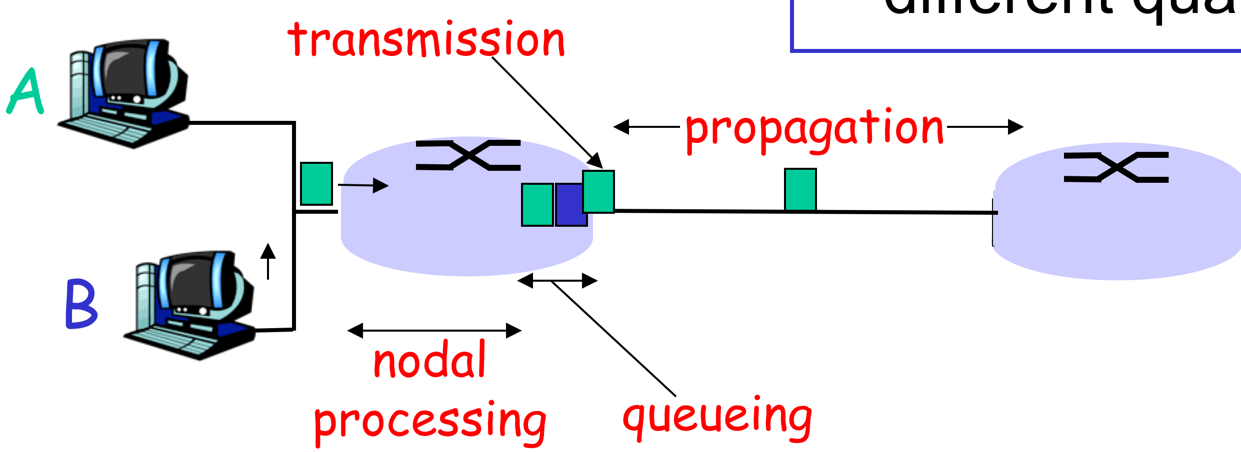
3. Transmission delay:

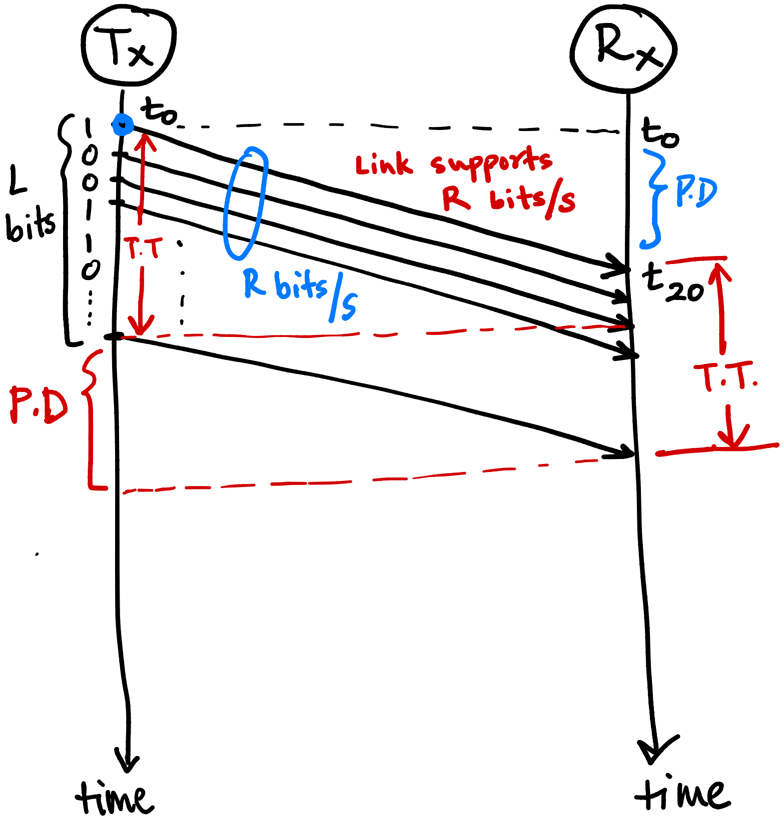
- R=link ~~bandwidth~~ Data Rate (bps)
- L=packet length (bits)
- time to send bits into link = L/R

4. Propagation delay:

- d = length of physical link
- s = propagation speed in medium (~2×10^8 m/sec)
- propagation delay = d/s

Note: s and R are *very* different quantities!





Prop. Delay

$$P.D. = (t_{20} - t_0)$$

$$= \frac{\text{Length of wire}}{\text{Speed of light}}$$

Transmit Time

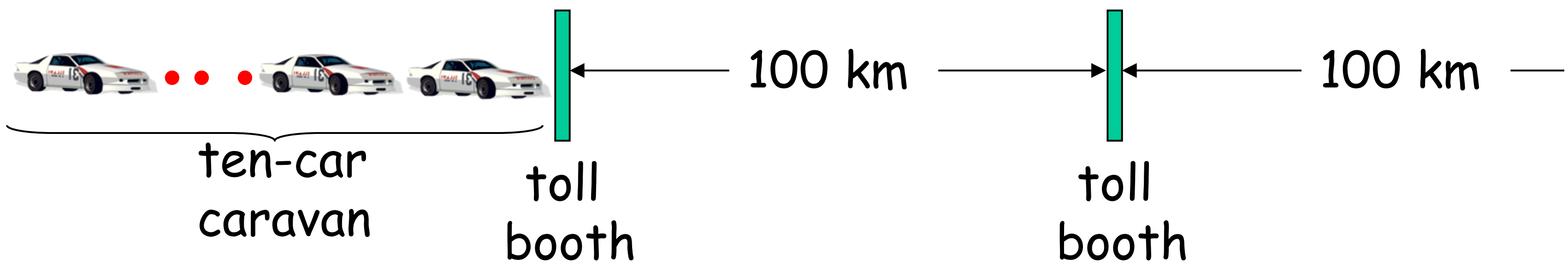
$$T.T = \frac{L \text{ bits}}{R \text{ bits/s}}$$

$$= L/R \text{ seconds}$$

Total time to completely transmit a pkt = P.D + T.T + P.D. ✗

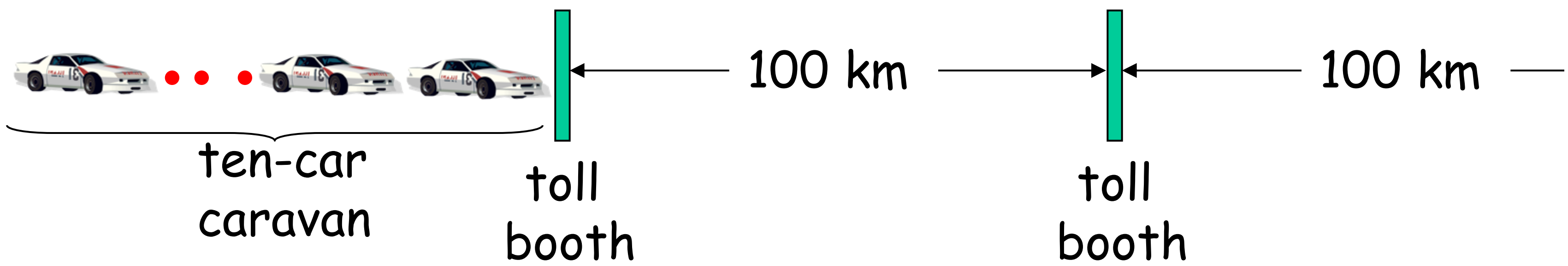
→ P.D + T.T. ✓

Caravan analogy

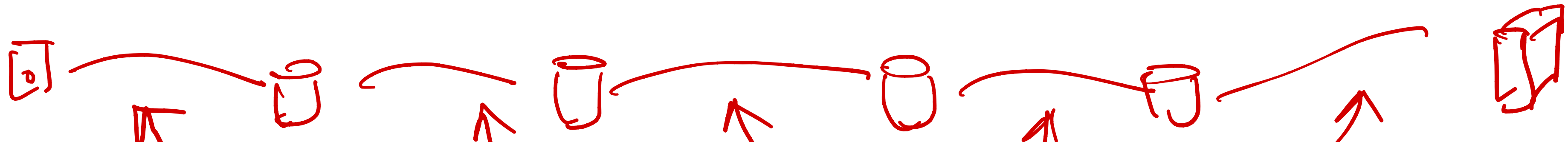


- Cars "propagate" at 100 km/hr
- Toll booth takes 12 sec to service a car (transmission time)
- car~bit; caravan ~ packet
- Q: How long until caravan is lined up before 2nd toll booth?
- Time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- Time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- A: 62 minutes

Caravan analogy (more)



- Cars now "propagate" at 1000 km/hr
- Toll booth now takes 1 min to service a car
- Q: Will cars arrive to 2nd booth before all cars serviced at 1st booth?
- Yes! After 7 min, 1st car at 2nd booth and 3 cars still at 1st booth.
- 1st bit of packet can arrive at 2nd router before packet is fully transmitted at 1st router!
  - See Ethernet applet at AWL Web site



Nodal delay

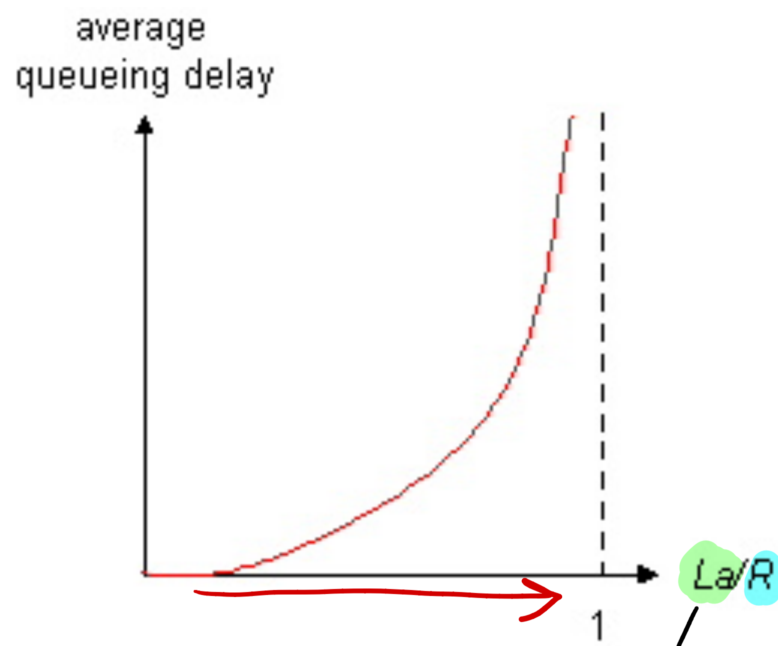
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

- d_{proc} = processing delay
  - typically a few microsecs or less
- d_{queue} = queuing delay
  - depends on congestion
- d_{trans} = transmission delay
  - = L/R, significant for low-speed links
- d_{prop} = propagation delay
  - a few microsecs to hundreds of msecs

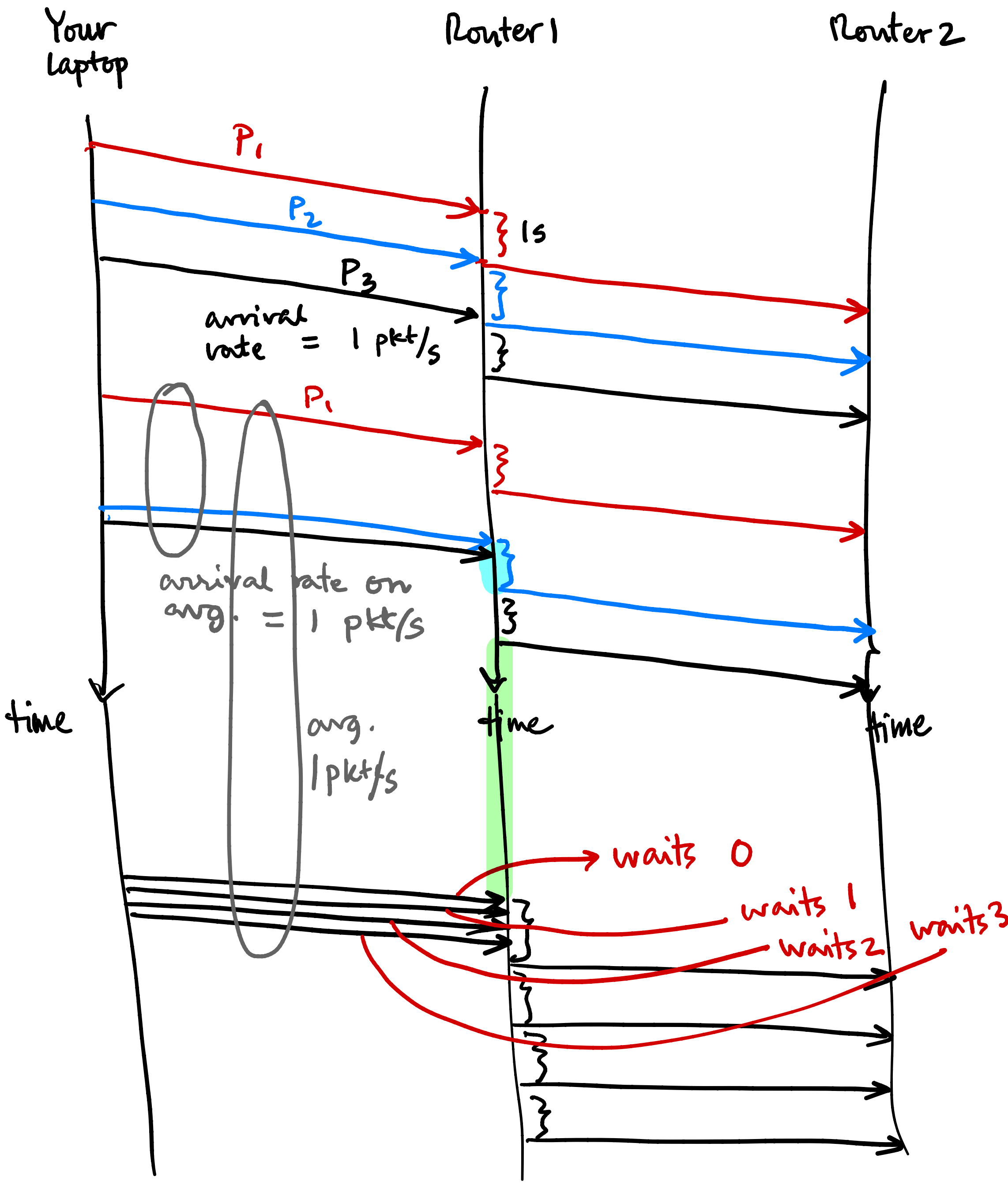
Queueing delay (revisited)

- R=link ~~bandwidth~~ (bps)
- L=packet length (bits)
- a=average packet arrival rate

traffic intensity = La/R



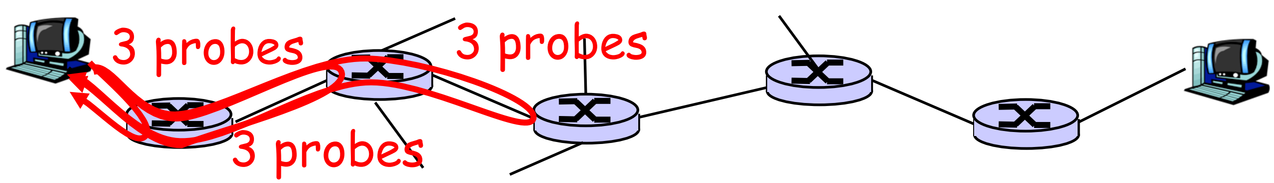
- La/R ~ 0: average queueing delay small
- La/R -> 1: delays become large
- La/R > 1: more "work" arriving than can be serviced, average delay infinite!


Your laptop
Router 1
Router 2
P_1
P_2
P_3
1s
arrival rate = 1 pkt/s
P_1
arrival rate on avg. = 1 pkt/s
avg. 1 pkt/s
time
time
time
waits 0
waits 1
waits 2
waits 3

traceroute google.com

"Real" Internet delays and routes

- What do "real" Internet delay & loss look like?
- **Traceroute** program: provides delay measurement from source to router along end-end Internet path towards destination. For all *i:*
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.



"Real" Internet delays and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

Three delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136 ms
```

trans-oceanic link

* means no response (probe lost, router not replying)

Packet loss

- queue (aka buffer) preceding link has finite capacity
- when packet arrives to full queue, packet is dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not retransmitted at all

Questions?

Introduction: Summary

Covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
- Internet/ISP structure
- performance: loss, delay
- layering and service models

You now have:

- context, overview, "feel" of networking
- more depth, detail *to follow!*

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

1.3 Network core

1.4 Network access and physical media

1.5 Internet structure and ISPs

1.6 Delay & loss in packet-switched networks

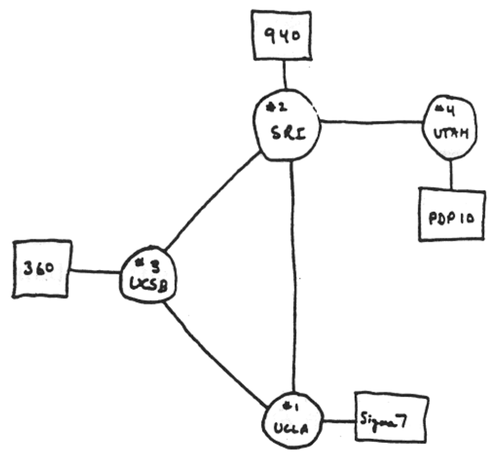
1.7 Protocol layers, service models

1.8 History

Internet History

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational
- 1972:
  - ARPAnet public demonstration
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



THE ARPA NETWORK

Internet History

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet History

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control

- new national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

Internet History

1990, 2000's: commercialization, the Web, new apps

- Early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

Late 1990's – 2000's:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

Physical Media

- Bit: propagates between transmitter/rcvr pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fiber, coax
- unguided media:
  - signals propagate freely, e.g., radio

Twisted Pair (TP)

- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 5: 100Mbps Ethernet

Physical Media: coax, fiber

Coaxial cable:

- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channels on cable
  - HFC

Fiber optic cable:

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10's-100's Gps)
- low error rate: repeaters spaced far apart ; immune to electromagnetic noise

Physical media: radio

- signal carried in electromagnetic spectrum
- no physical “wire”
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

Radio link types:

- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., Wifi)
  - 11Mbps, 54 Mbps
- wide-area (e.g., cellular)
  - e.g. 3G: hundreds of kbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude

Why layering?

Dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., change in baggage procedure doesn't affect rest of system (as long as all baggage sections know)
- layering considered harmful?