# CDA 4506

## Design and Implementation of Data Communication Networks

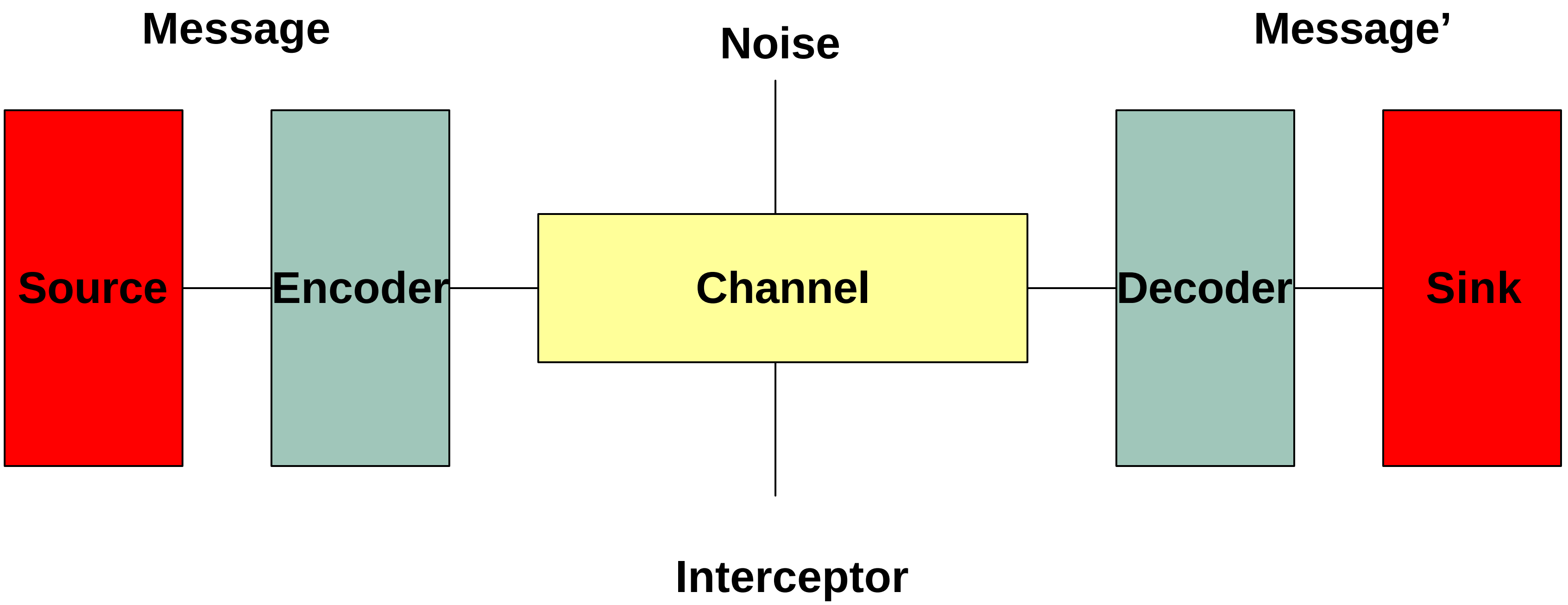
Lecture Notes 1

Dr. R. Lent

# Getting Organized

- Laboratories (You must be in one!)
- No food or drink allowed in the lab !
- Text: Kurose & Ross ($2^{nd}$ edition)
- Schedule and class notes: WebCT and http://www.cs.ucf.edu/course/cda4506 (check regularly!)
- Average use of time per week:
  - 3.5 hours of lecture/labs
  - 3 hours reading
  - 2.5 hours working on homework
- Grade = F(10 labs, 2 tests, 2-3 HW, 1 final exam)
- First Assignment: Read Chapter 1

# Basic Communication Model




Shannon, 1949

# Chapter 1: Introduction

## Our goal:

- get context, overview, “feel” of networking
- more depth, detail *later* in course
- approach:
  - descriptive
  - use Internet as example
  - Understand basic concepts

## Overview:

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

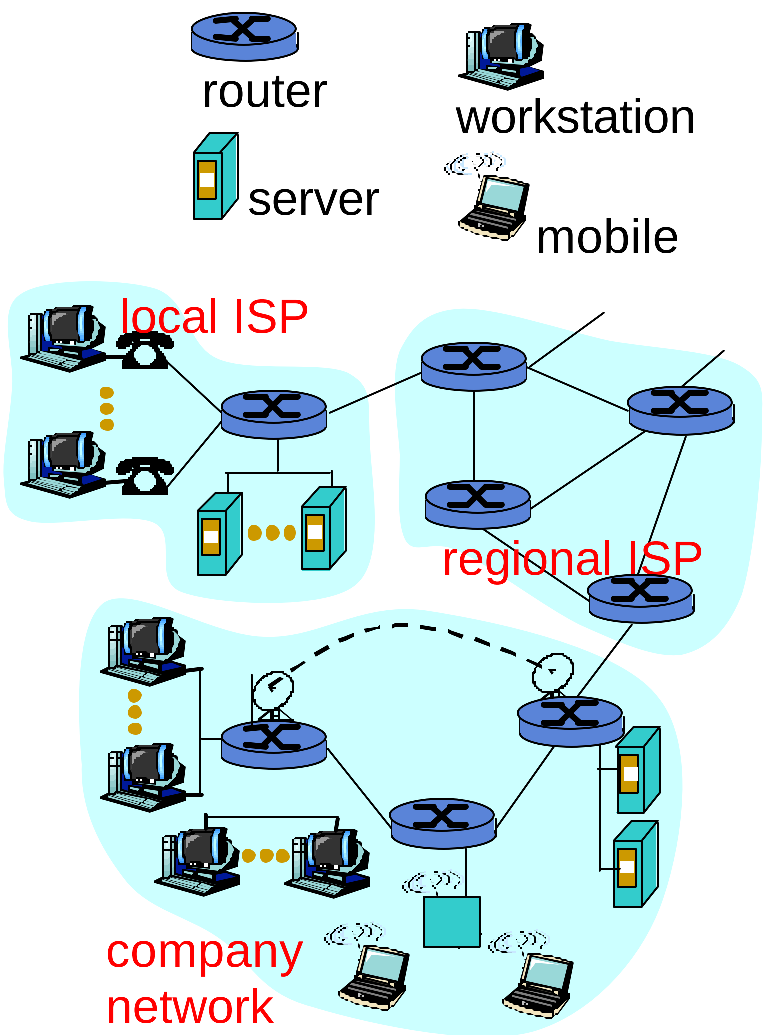
# Internet: “nuts and bolts” view

millions of connected computing devices (or *nodes*):

1. *hosts, end-systems*
   - PCs workstations, servers
   - PDAs phones, toasters

   running *network apps*
2. *communication links*
   - fiber, copper, radio, satellite
   - transmission rate
   - bandwidth
3. *routers:* forward packets (chunks of data)

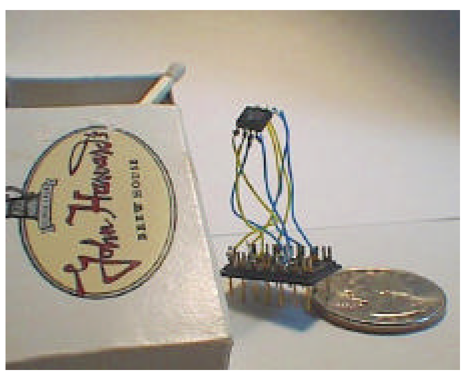
and protocols (FTP, TCP, HTTP, etc)



# “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

# 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

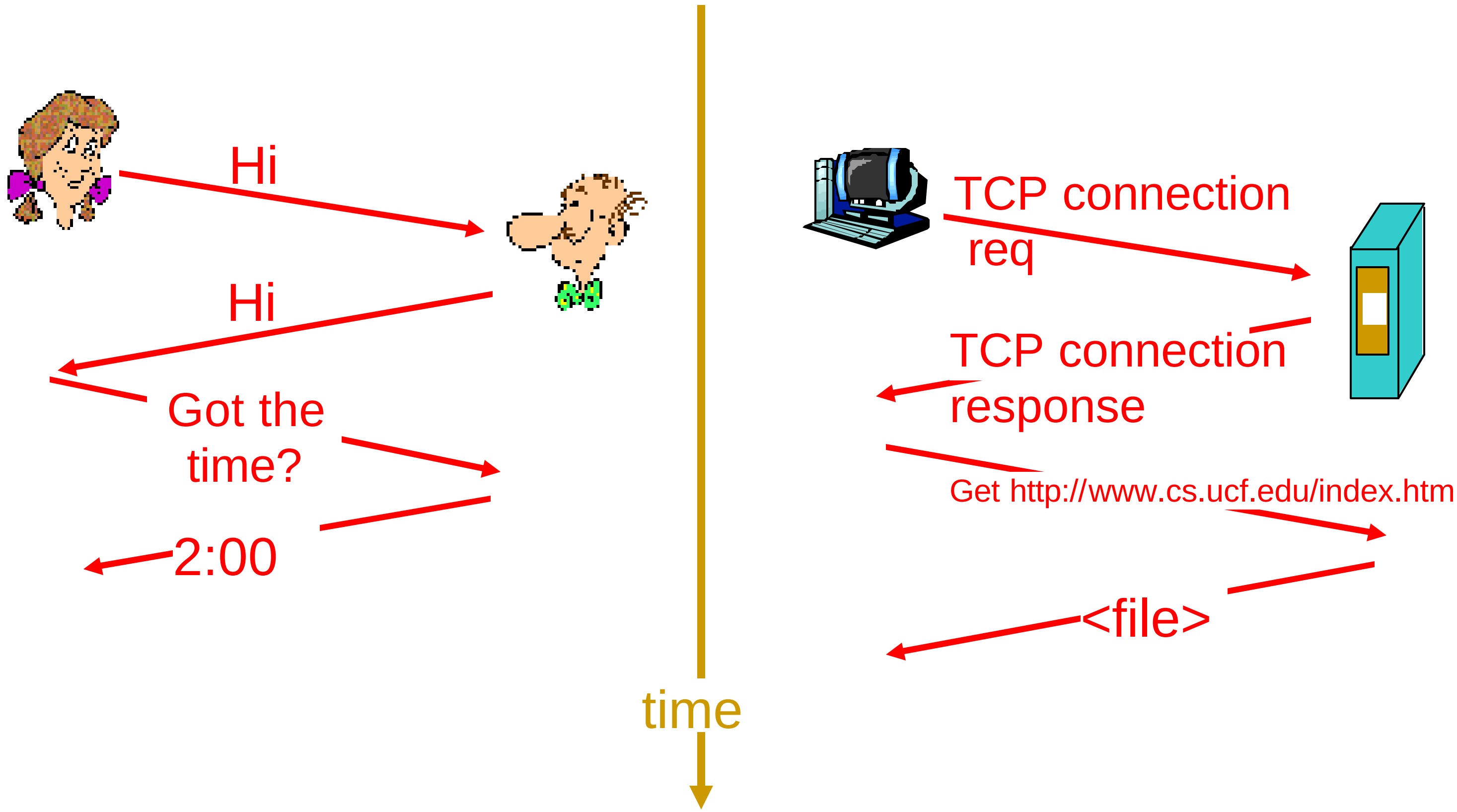
network protocols:

- machines rather than humans
- all communication activity in Internet governed 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?

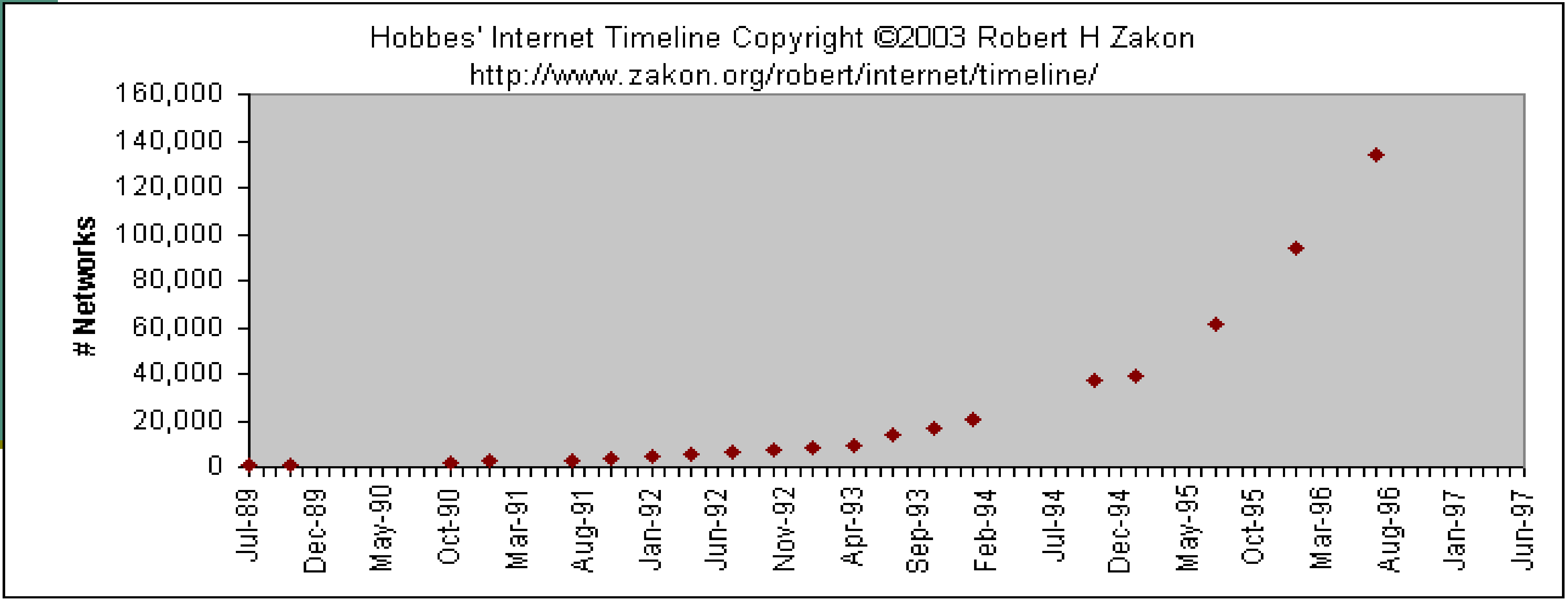
a human protocol and a computer network protocol:



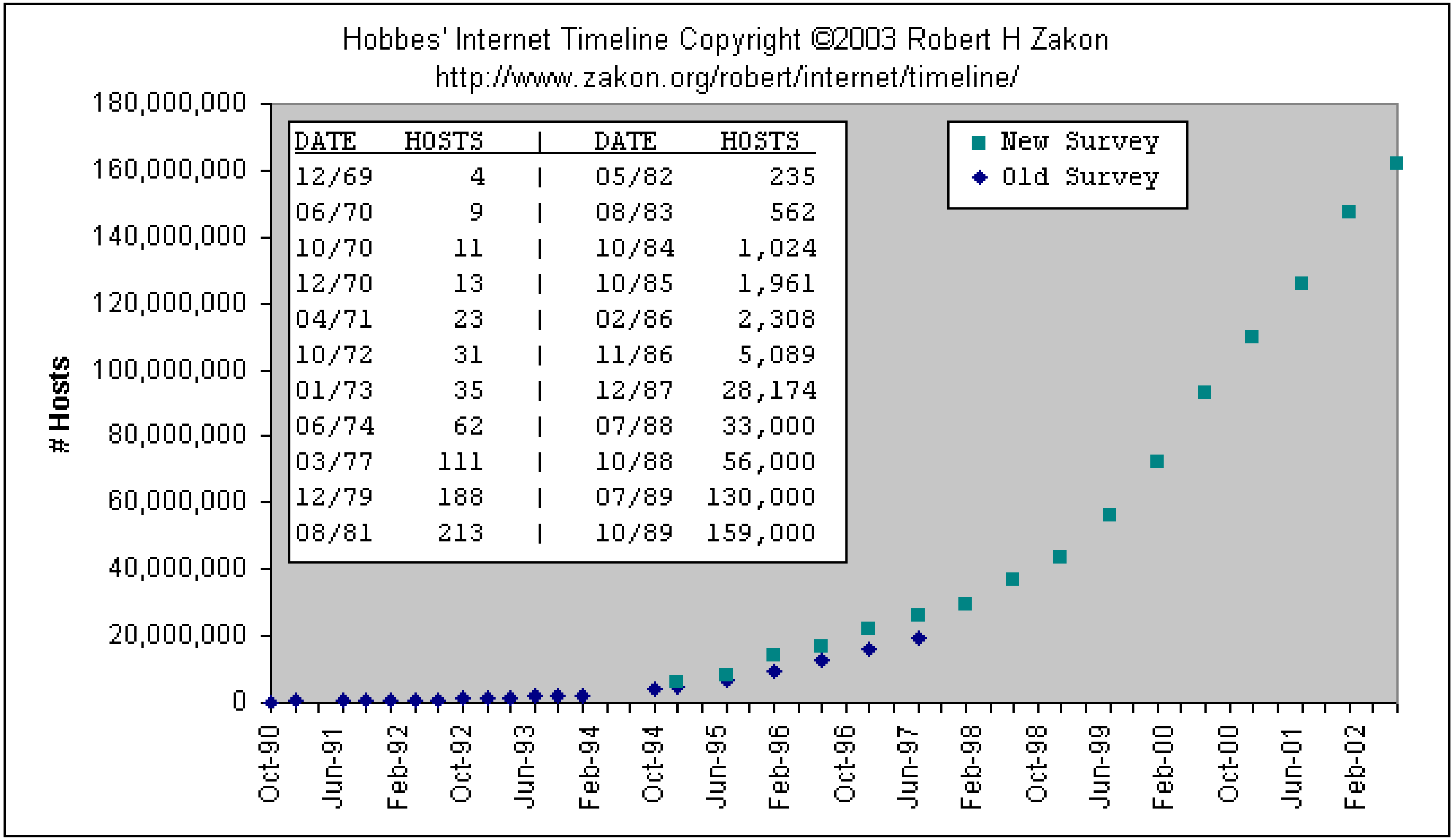
# Internet: A Service View

- communication *infrastructure* enables distributed applications:
  - Web, email, games, e-commerce, database., voting, file (MP3) sharing
- communication services provided to apps:
  - connectionless
  - connection-oriented

# How Large is the Internet?

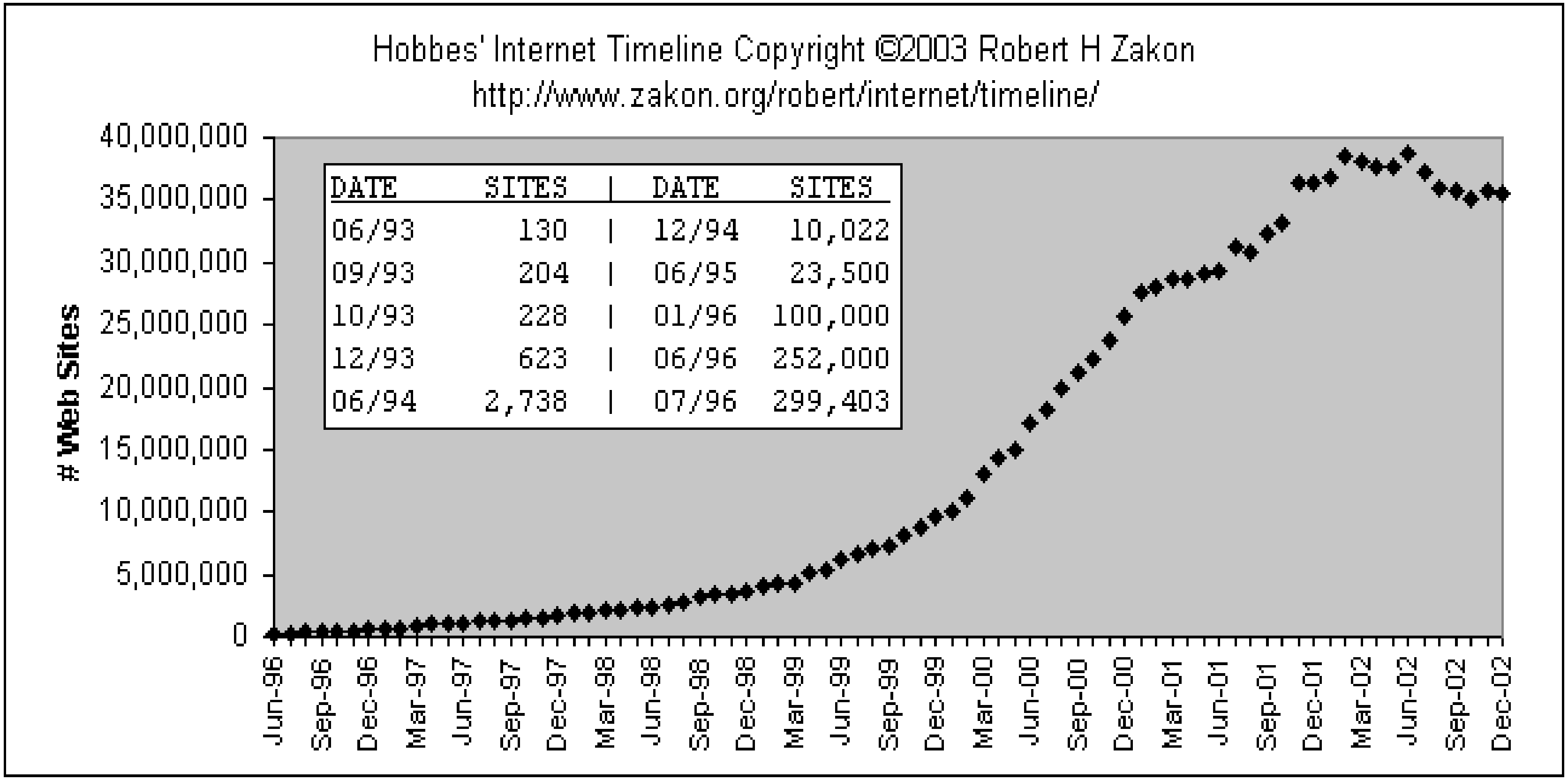


# Growth of Hosts in the Internet



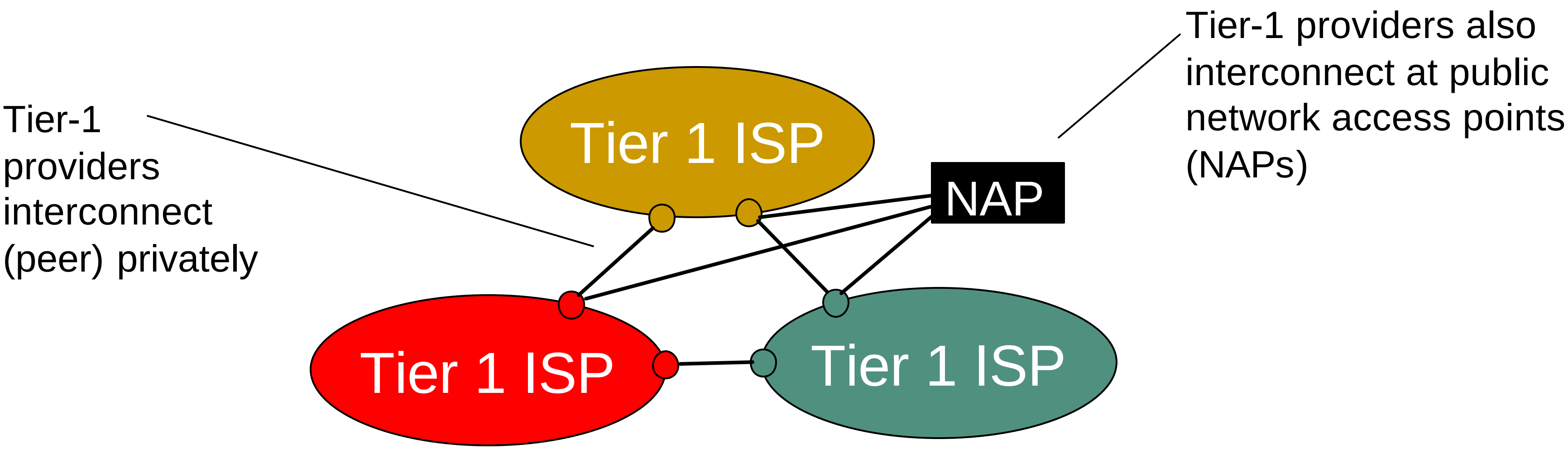

| DATE | HOSTS | DATE | HOSTS |
|---|---|---|---|
| 12/69 | 4 | 05/82 | 235 |
| 06/70 | 9 | 08/83 | 562 |
| 10/70 | 11 | 10/84 | 1,024 |
| 12/70 | 13 | 10/85 | 1,961 |
| 04/71 | 23 | 02/86 | 2,308 |
| 10/72 | 31 | 11/86 | 5,089 |
| 01/73 | 35 | 12/87 | 28,174 |
| 06/74 | 62 | 07/88 | 33,000 |
| 03/77 | 111 | 10/88 | 56,000 |
| 12/79 | 188 | 07/89 | 130,000 |
| 08/81 | 213 | 10/89 | 159,000 |

# Growth of WWW servers



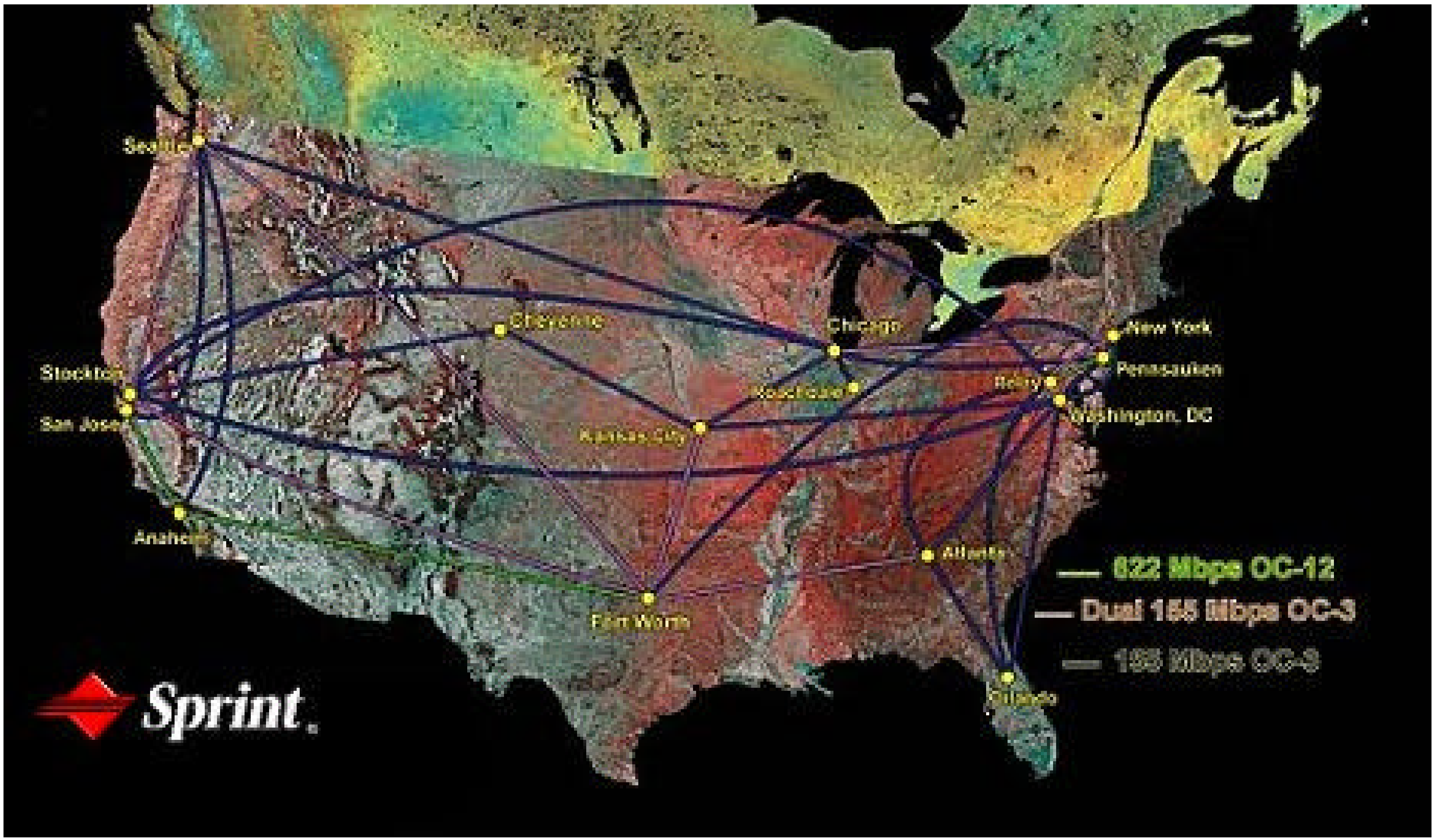
# Internet Structure: Network of Networks

- LAN, WAN, MAN
- Roughly hierarchical
- At center: “tier-1” ISPs (e.g., UUNet, BBN/Genuity, Sprint, AT&T), national/international coverage
  - treat each other as equals



# Tier-1 ISP: e.g., Sprint

Sprint US backbone network

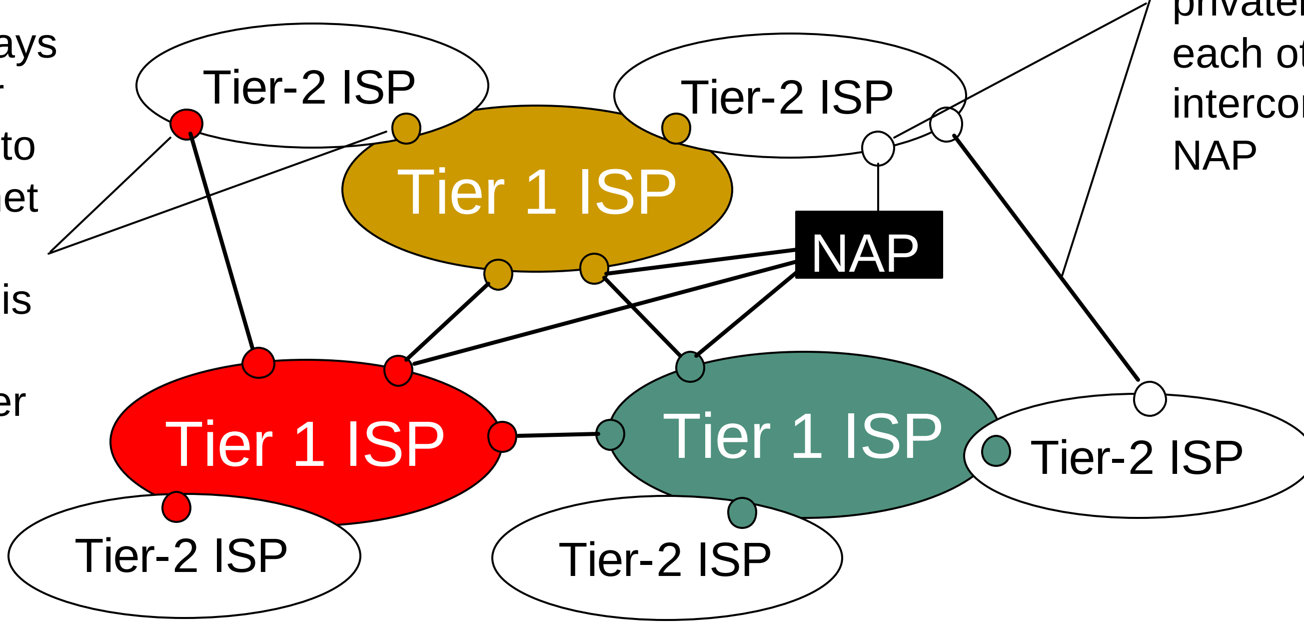


# Internet structure: network of networks

- "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

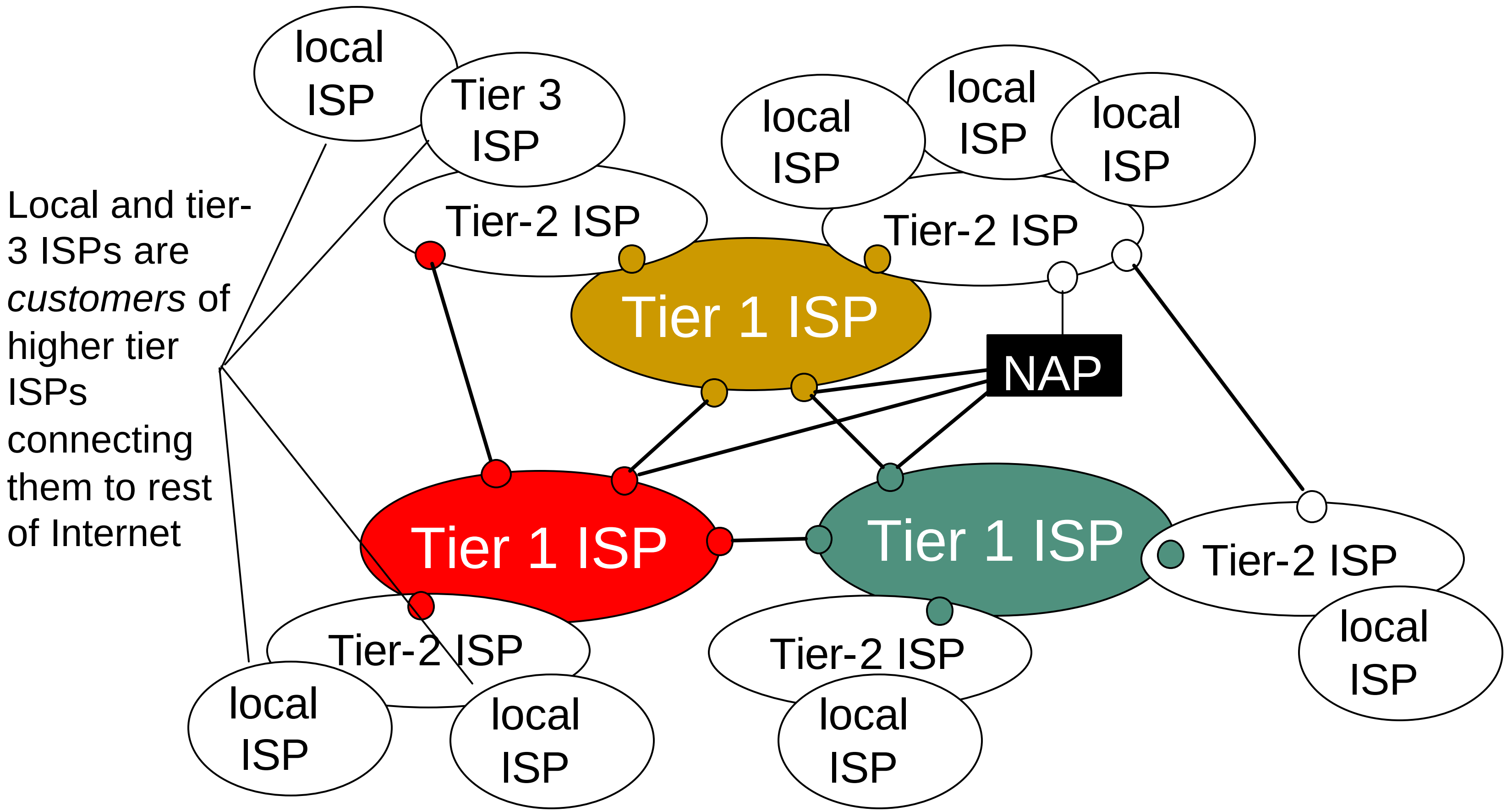
- tier-2 ISP is *customer* of tier-1 provider




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

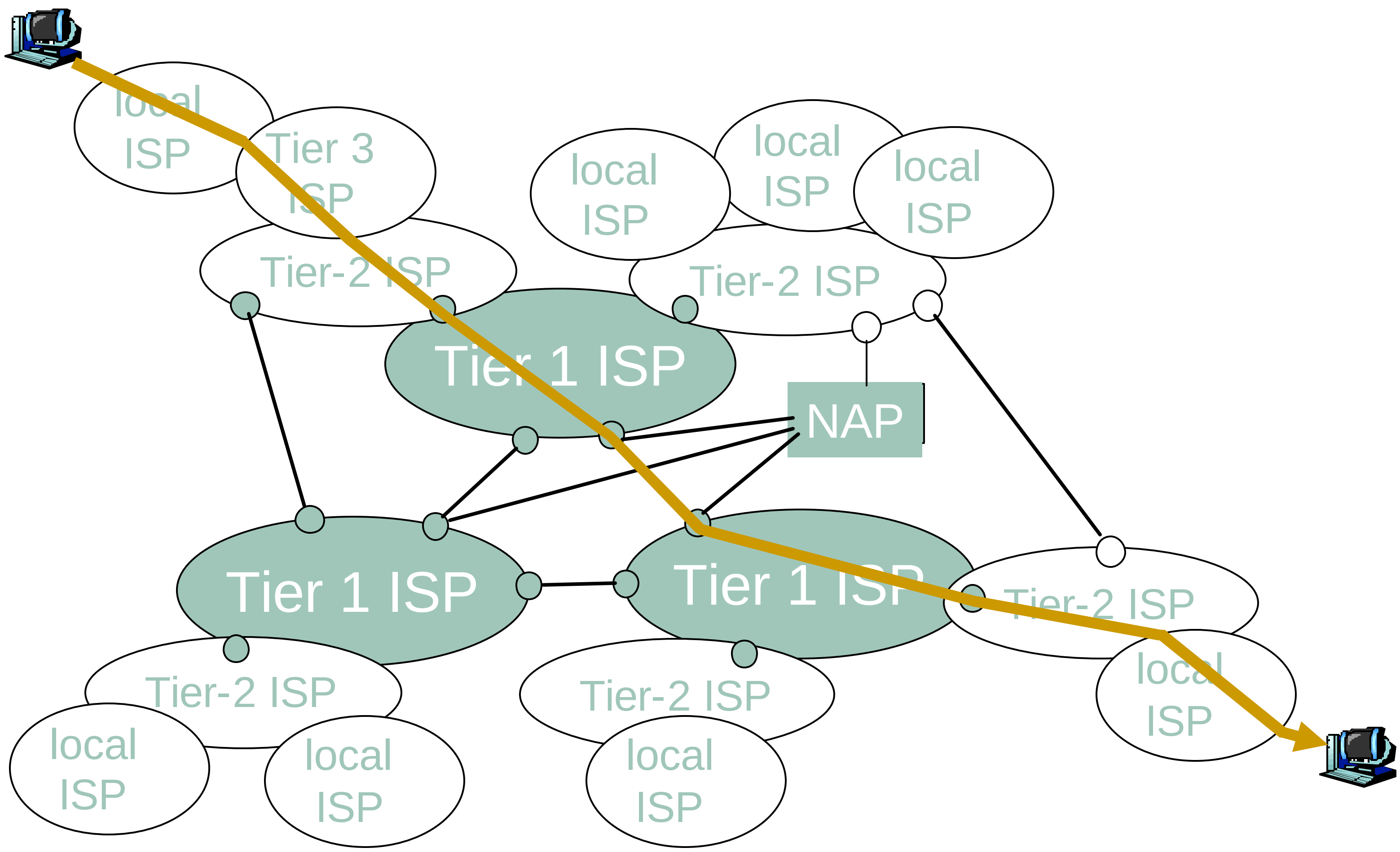
# Internet structure: network of networks

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



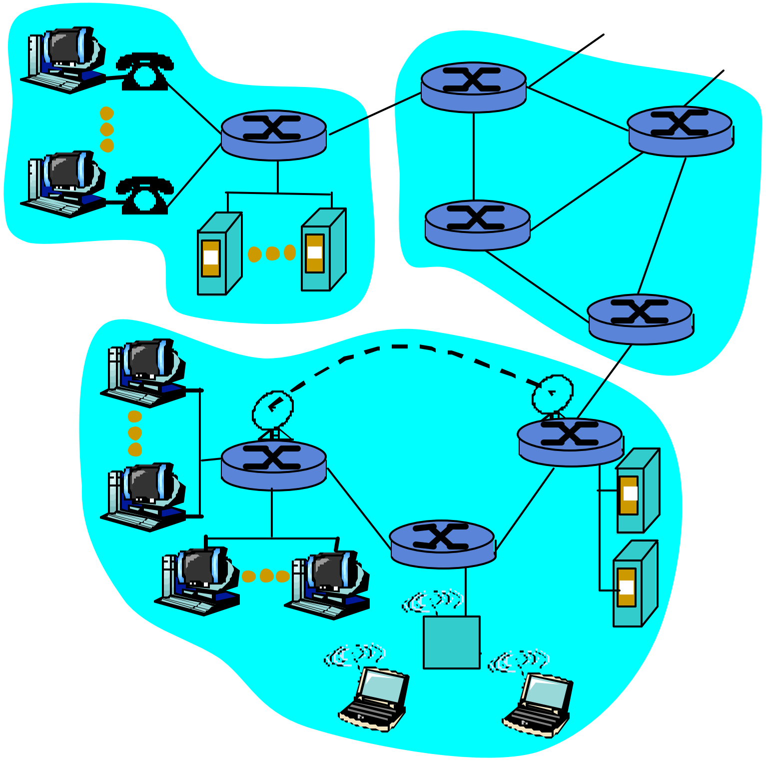
# Internet structure: network of networks

- a packet passes through many networks!
- Check Internet stats: www.caida.org



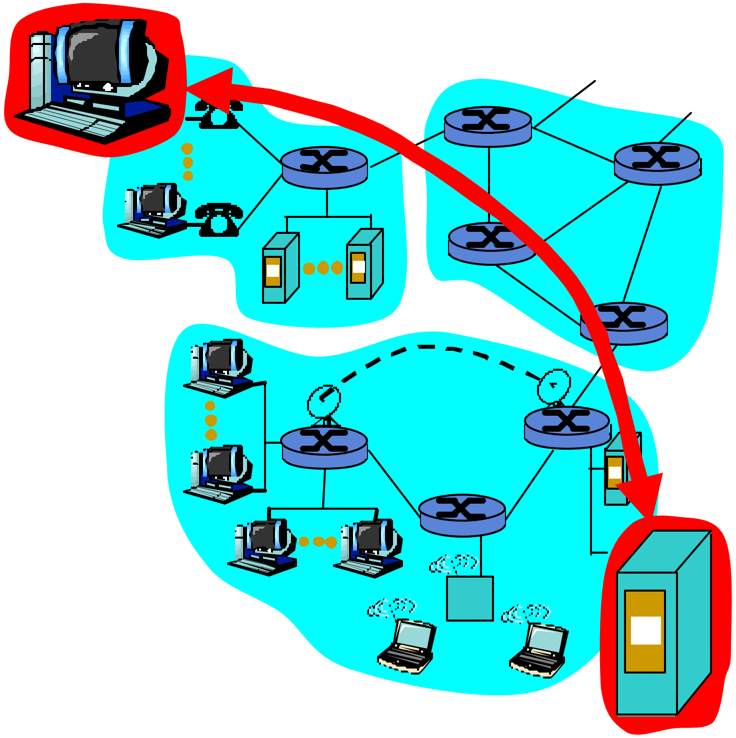
# A closer look at network structure:

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



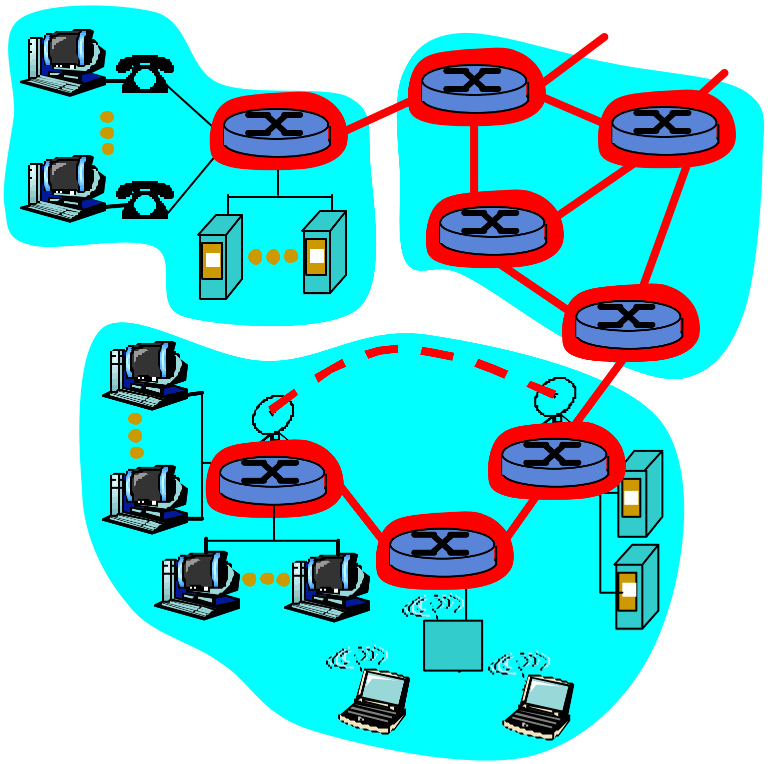
# The Network Edge:

- end systems (hosts): run application programs
- Models:
  - client/server model
    - client host requests, receives service from always-on server
  - peer-peer model:
    - minimal (or no) use of dedicated servers
- Service:
  - Connection oriented (TCP)
  - Connectionless (UDP)



# 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

Network resources (e.g., bandwidth) divided into “pieces”:

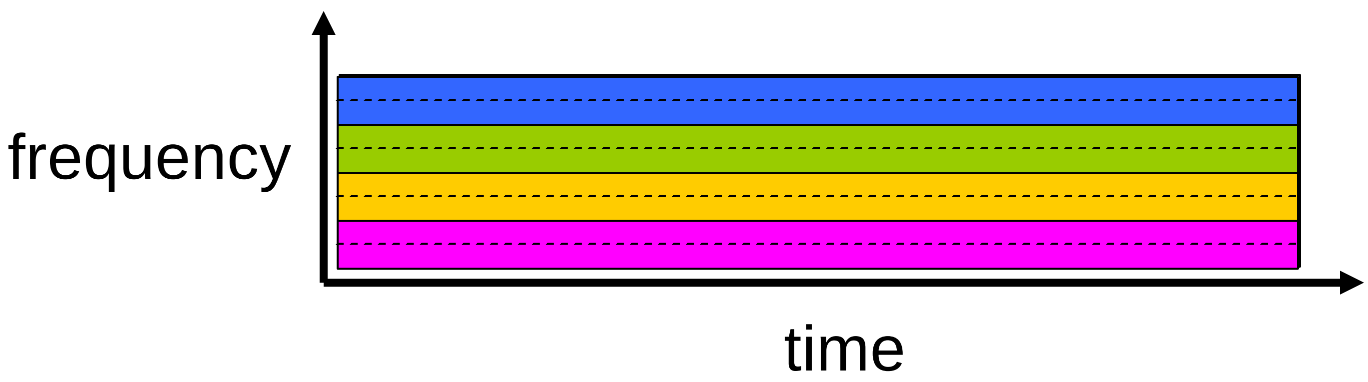
- pieces allocated to calls
- resource piece *idle* if not used by owning call *(no sharing)*
- dividing link bandwidth into “pieces”
  - frequency division
  - time division
  - Code division

# Circuit Switching: TDMA and TDMA

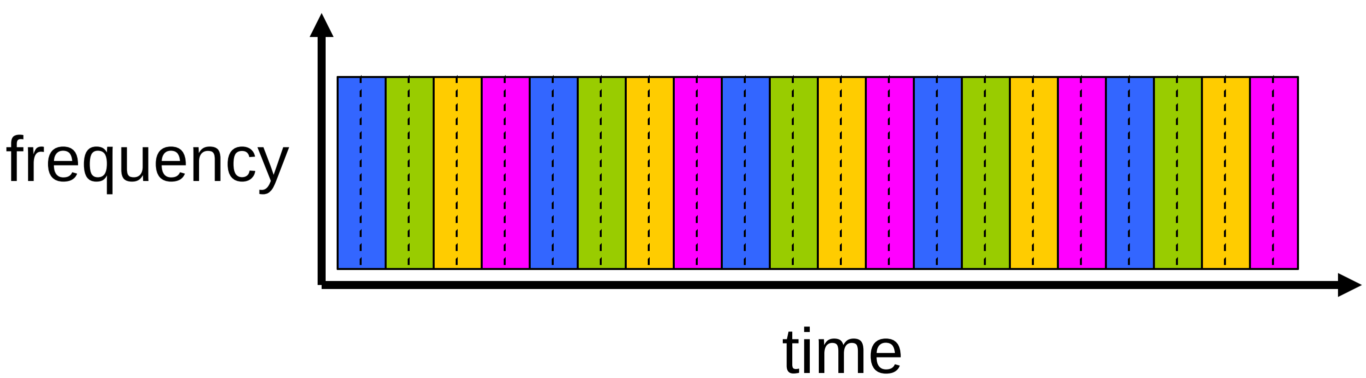
Example:

4 users

FDMA




TDMA



# 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*

resource contention:

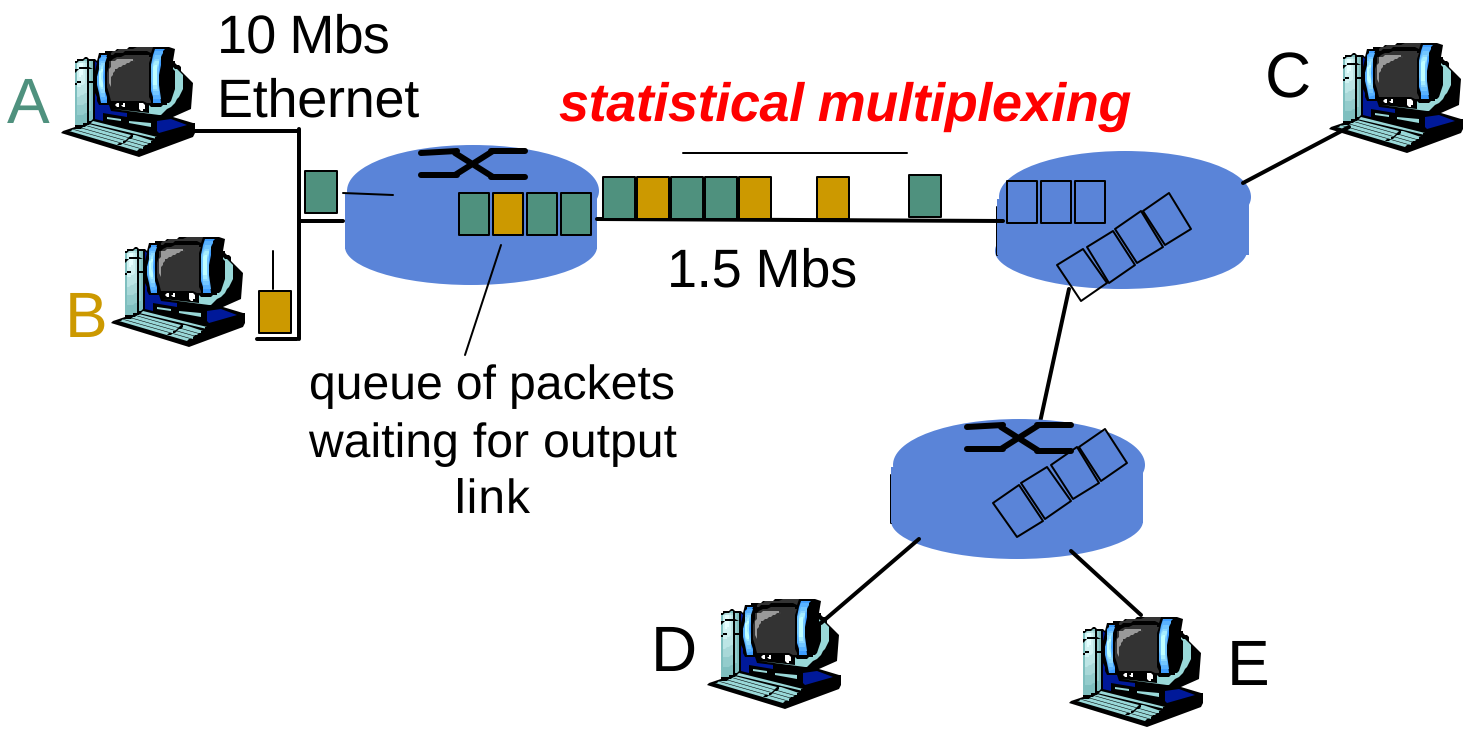
- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
  - transmit over link
  - wait turn at next link

Bandwidth division into “pieces”

Dedicated allocation

Resource reservation

# Packet Switching: Statistical Multiplexing



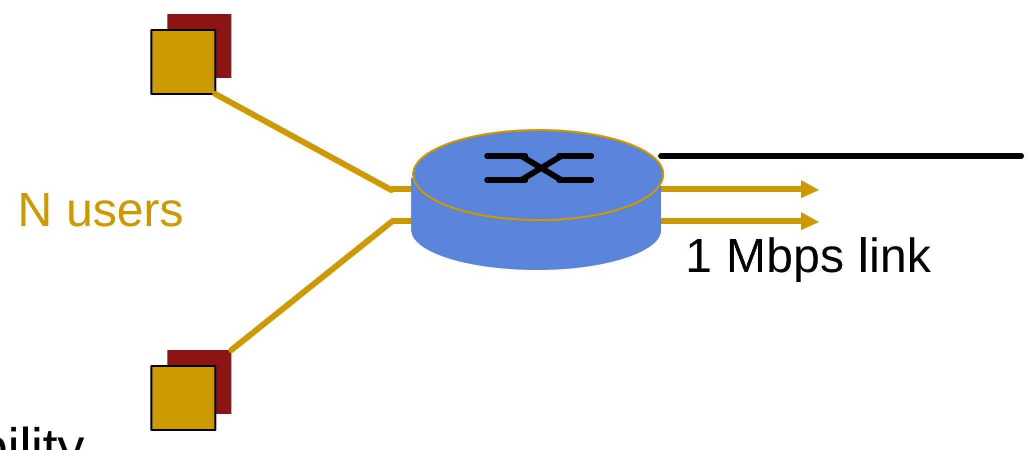

Sequence of A & B packets does not have fixed pattern ➔ ***statistical multiplexing***.

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

# Packet switching versus circuit switching

Packet switching allows more users to use network!

- 1 Mbit link
- each user:
  - 100 kbps when “active”
  - active 10% of time
- circuit-switching:
  - 10 users
- packet switching:
  - with 35 users, probability > 10 active less than .0004

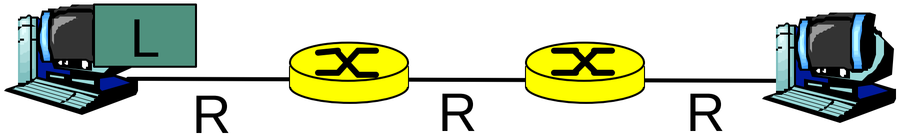


# Packet switching versus circuit switching

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 reliable data transfer, congestion control
- Q: How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps

# Packet-switching: store-and-forward

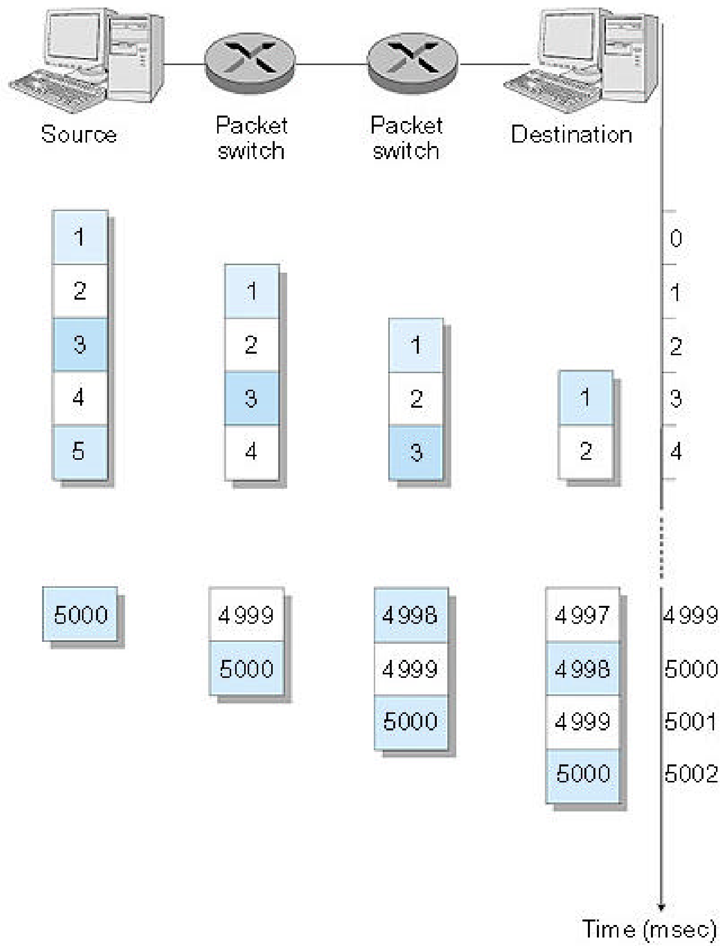



- Takes L/R seconds to transmit (push out) packet of L bits on to link or R bps
- Entire packet must arrive at router before it can be transmitted on next link: *store and forward*
- delay = 3L/R

Example:

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

# Packet Switching: Message Segmenting



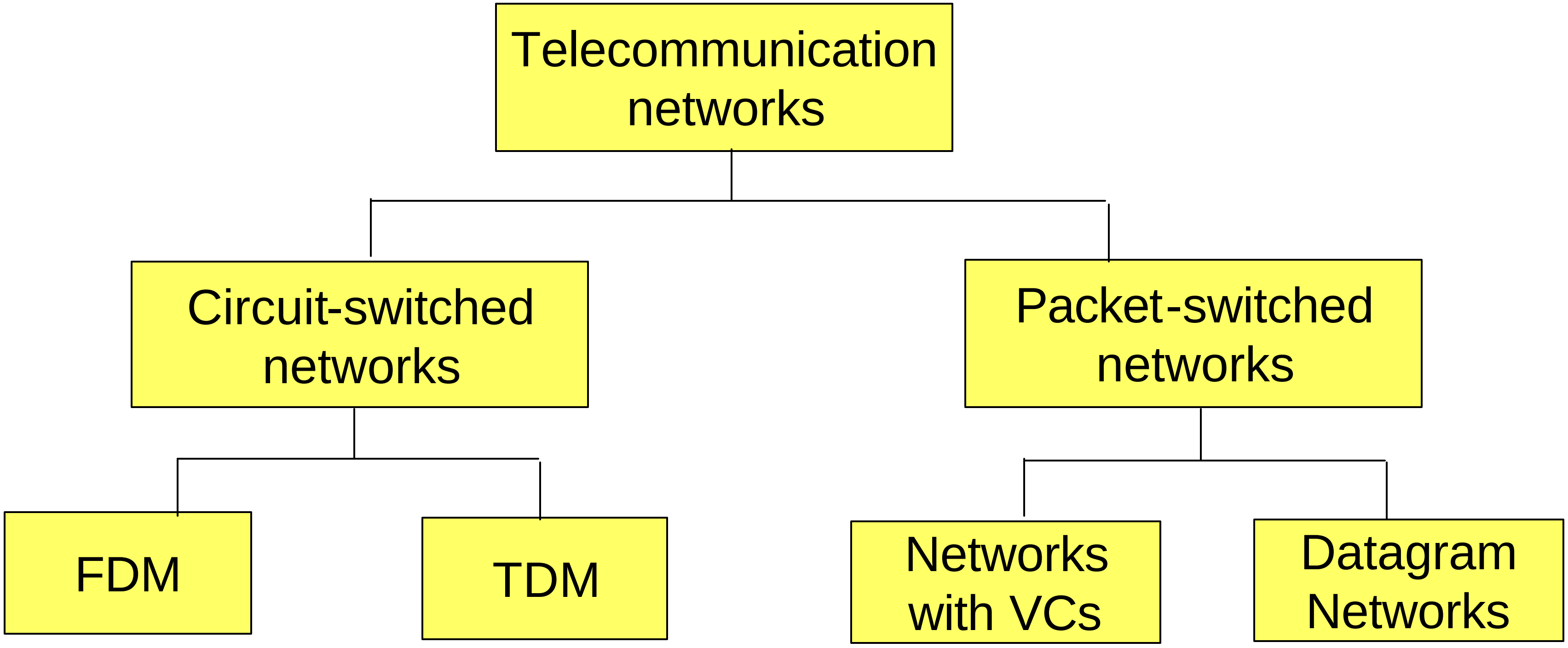

Now break up the message into 5000 packets

- Each packet 1,500 bits
- 1 msec to transmit packet on one link
- *pipelining:* each link works in parallel
- Delay reduced from 15 sec to 5.002 sec

# Packet-switched networks: forwarding

- *Goal:* move packets through routers from source to destination
  - we'll study several path selection (i.e. routing) algorithms (chapter 4)
- datagram network:
  - *destination address* in packet determines next hop
  - routes may change during session
  - analogy: driving, asking directions
- virtual circuit network:
  - each 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.

# Protocol "Layers"

Networks are complex!

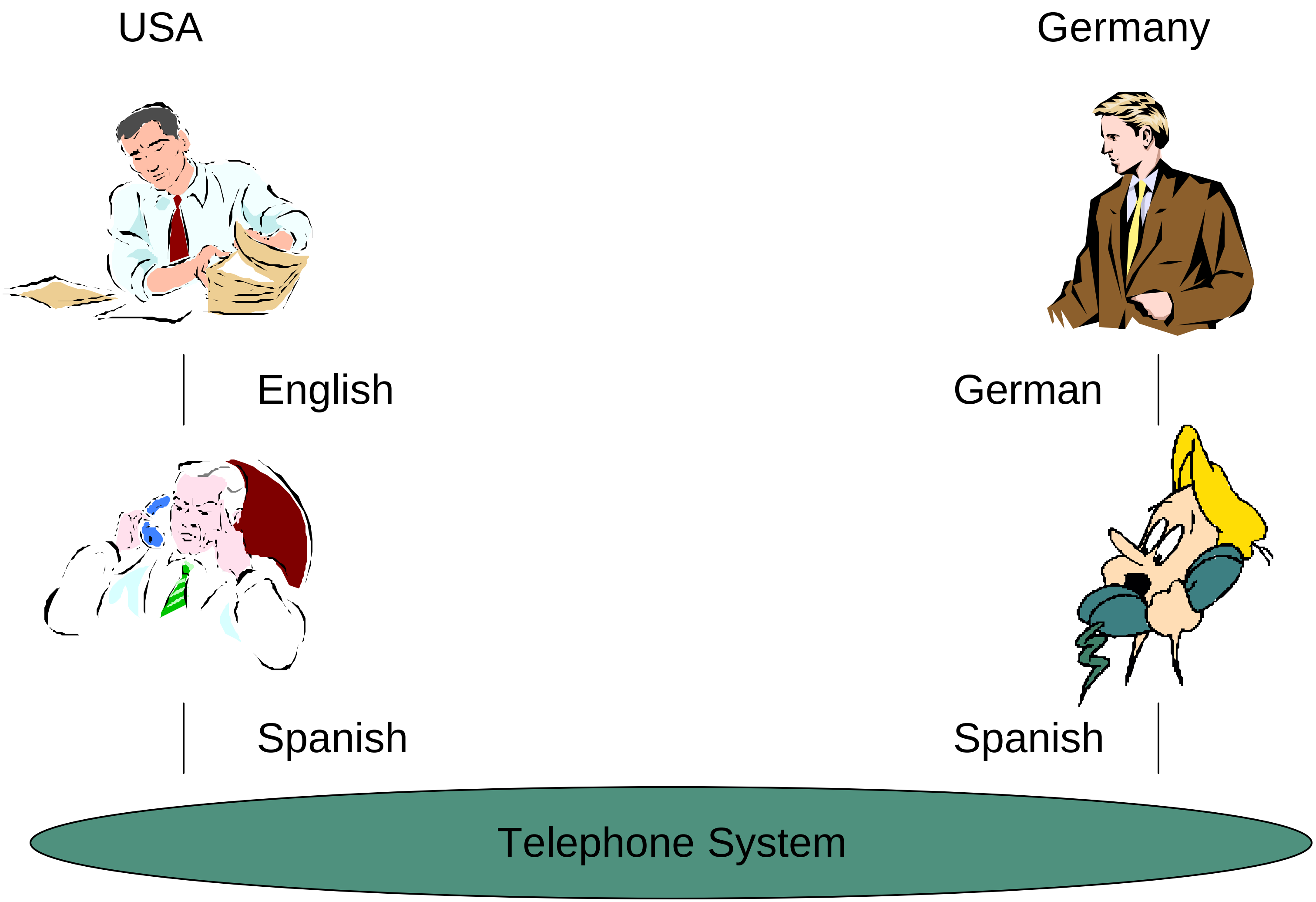
- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

Question:

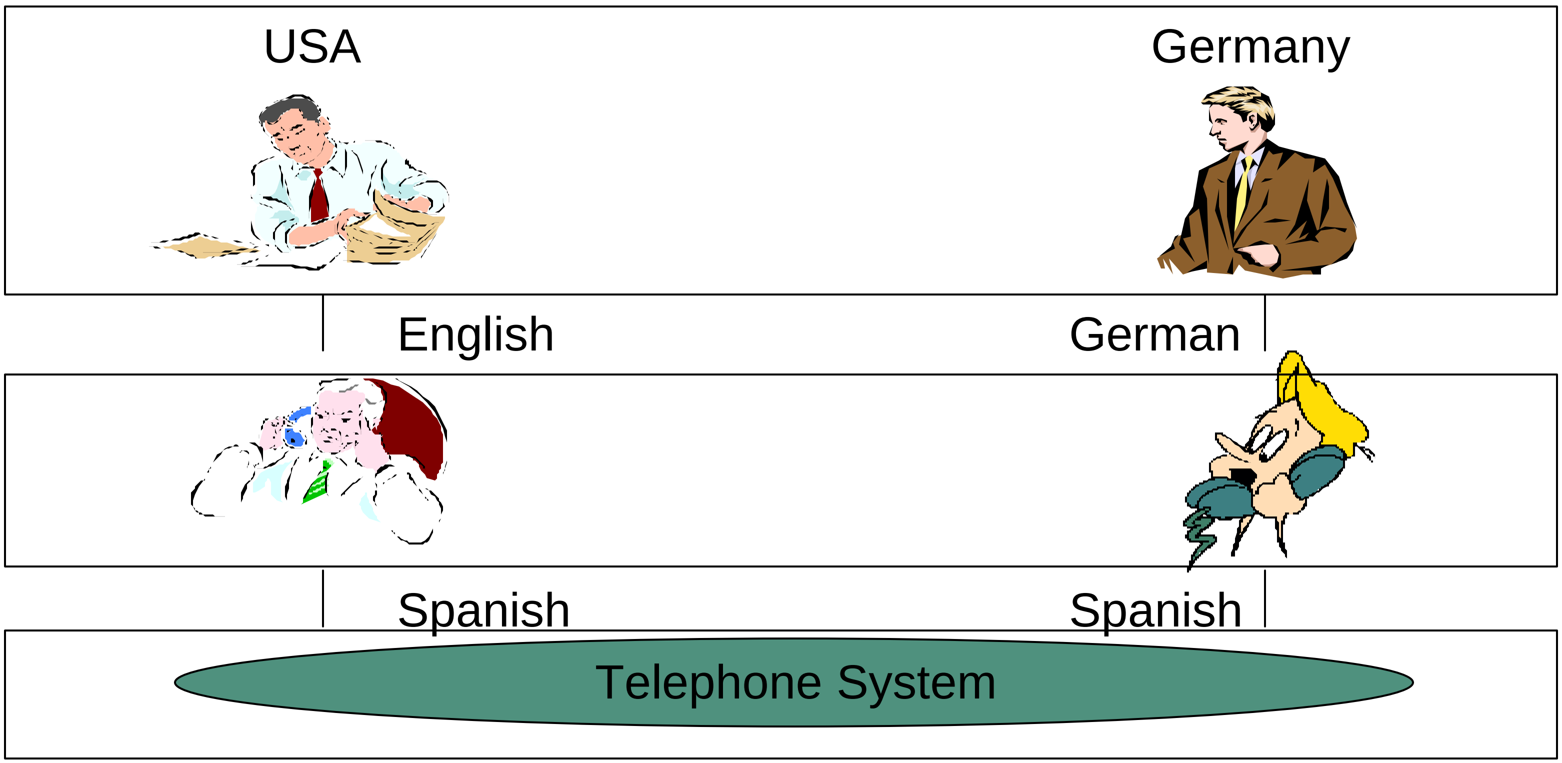
Is there any hope of *organizing* structure of network?

Or at least our discussion of networks?

# Layering Example



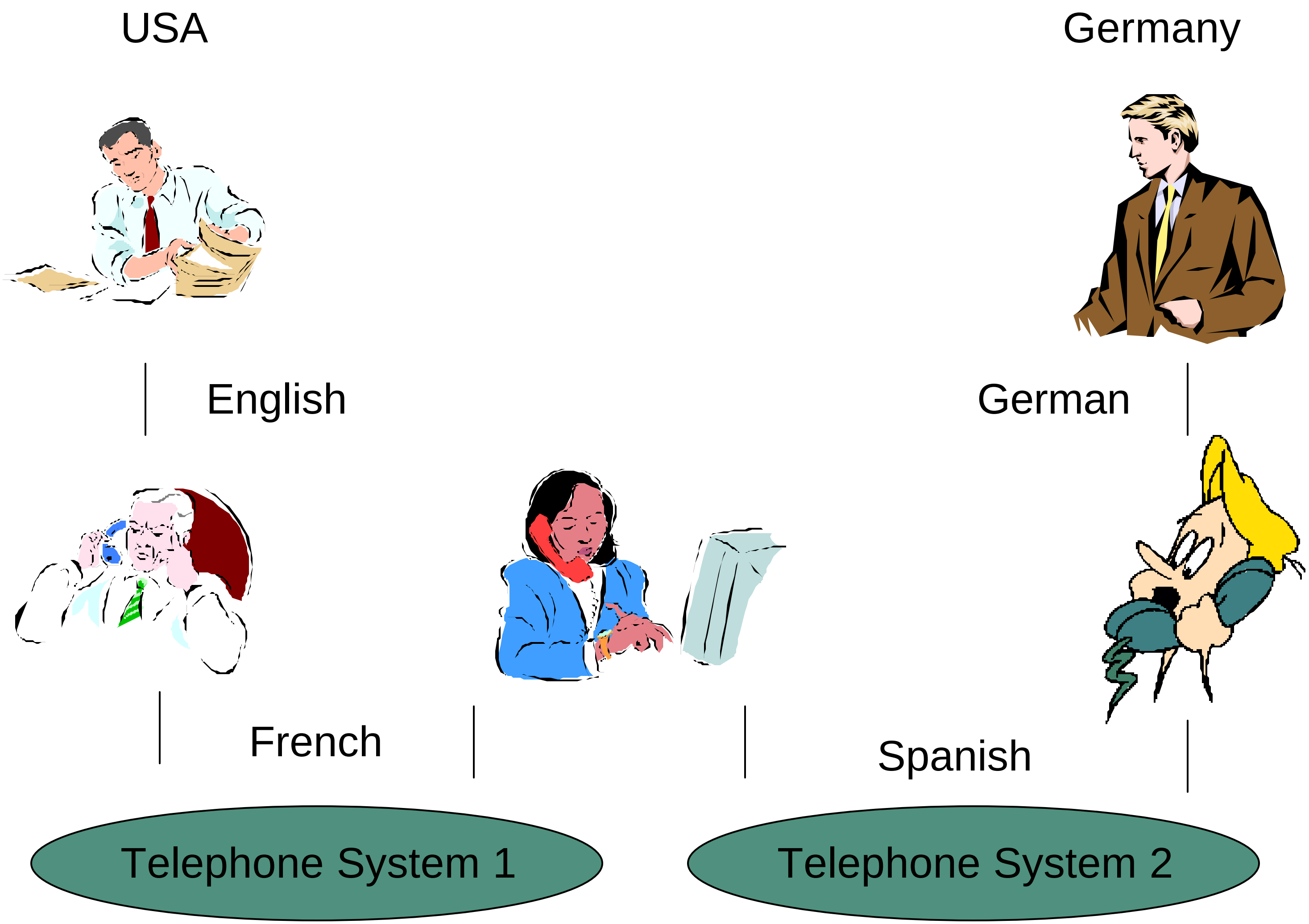
# A different view




Layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

# Another view



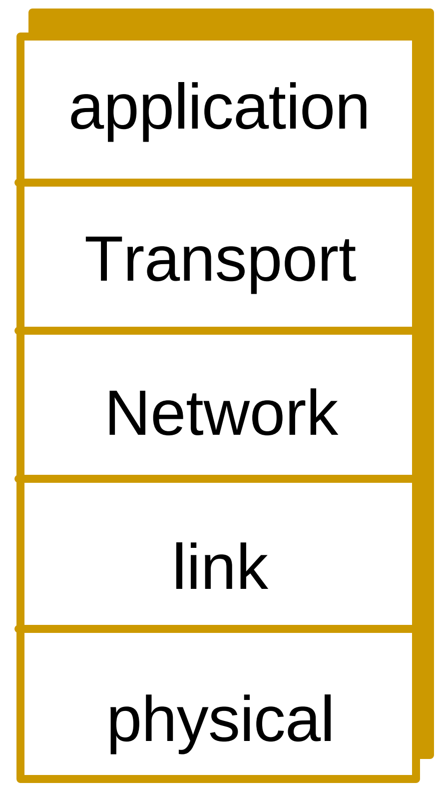
# Why layering?

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 gate procedure doesn't affect rest of system
- layering considered harmful?

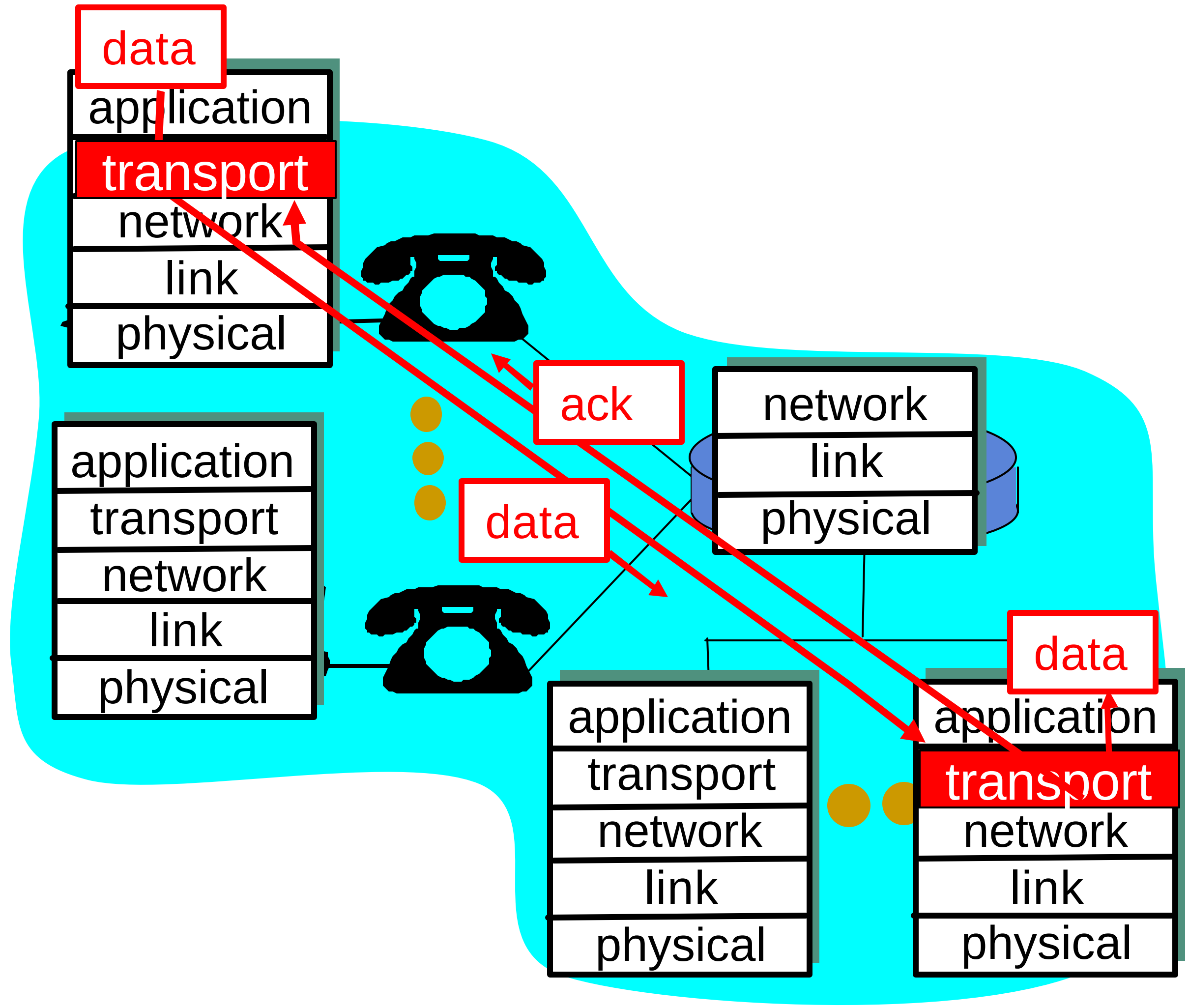
# Internet protocol stack

- application: supporting network applications
  - FTP, SMTP, STTP
- transport: host-host data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - PPP, Ethernet
- physical: bits “on the wire”

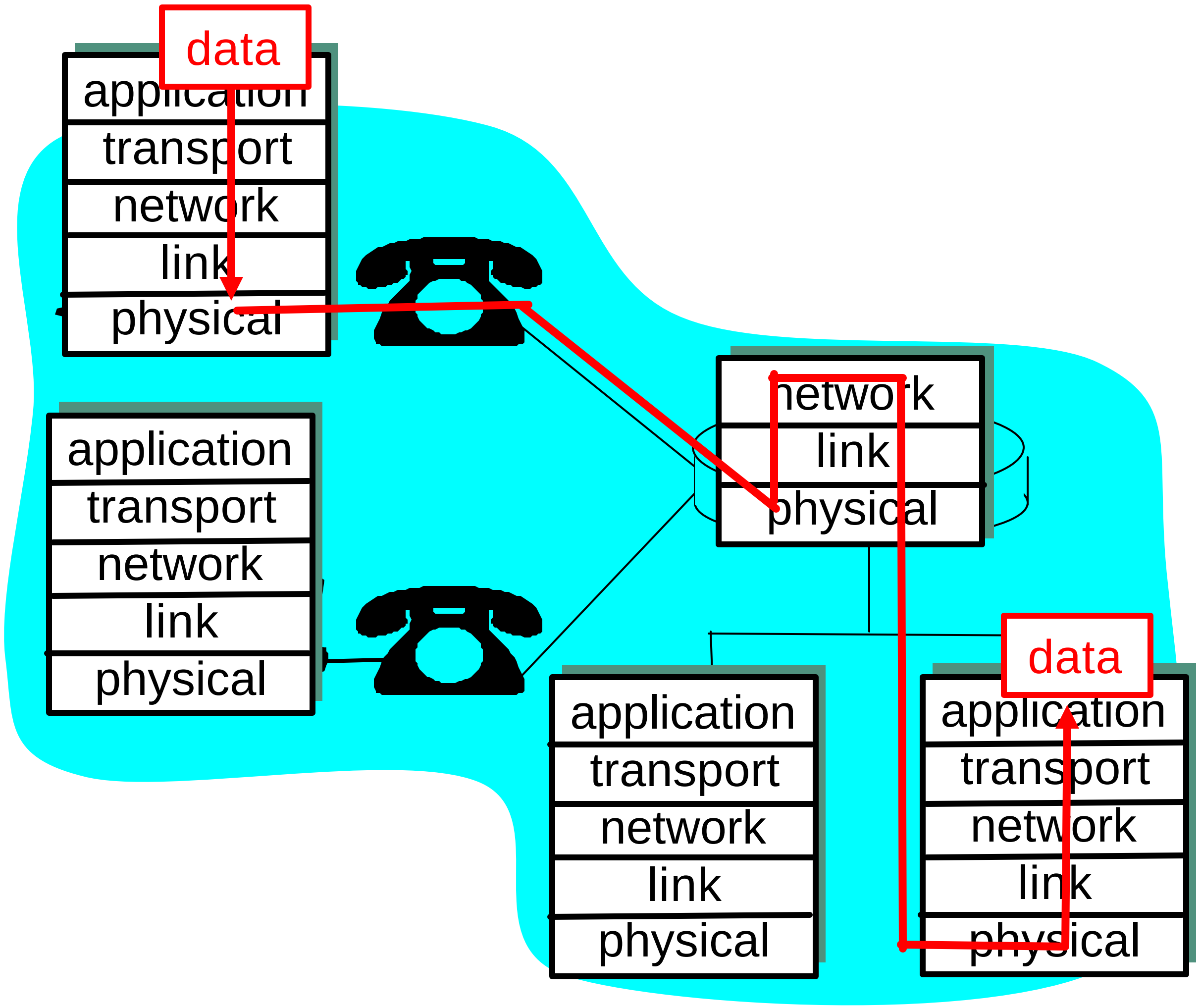


# Layering: *logical* communication

- take data from application
- add addressing, reliability check info to form "datagram"
- send datagram to peer
- wait for peer to ack receipt



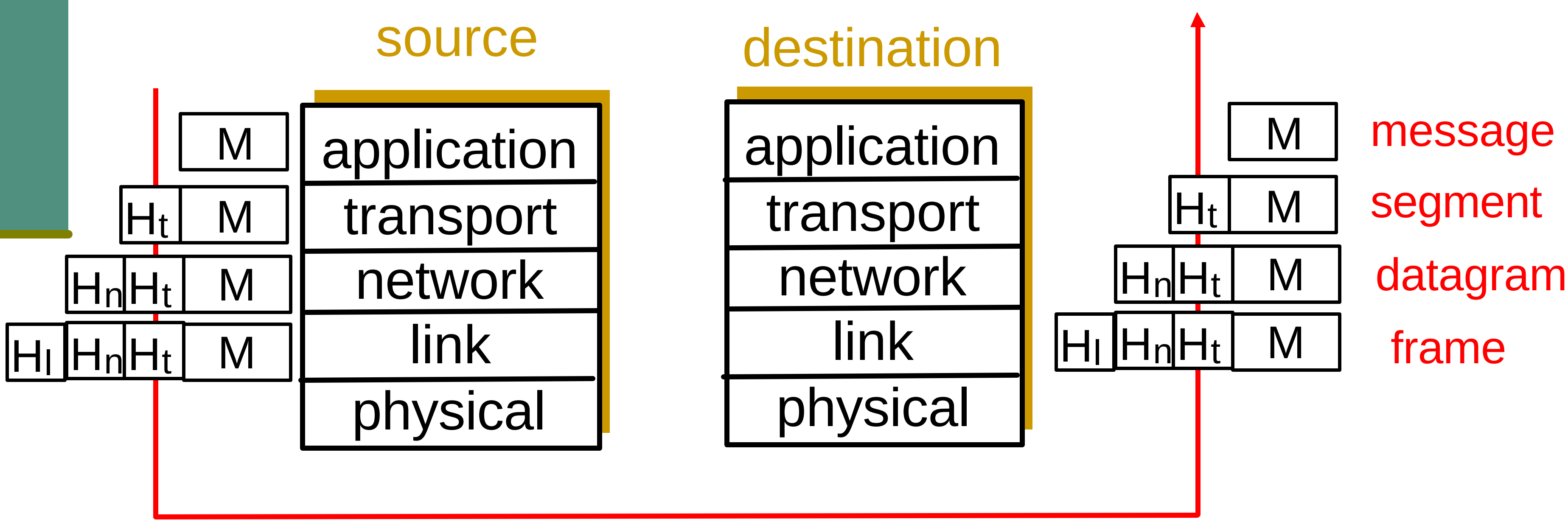
# Layering: physical communication



# Protocol layering and data

Each layer takes data from above

- adds header information to create new data unit
- passes new data unit to layer below



# Physical Layer

- 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

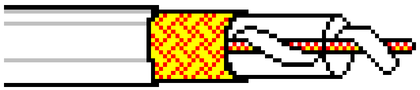
Twisted Pair (TP)

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

# Physical Media: coax, fiber

## Coaxial cable:

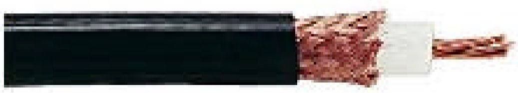
- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channel on cable
  - HFC

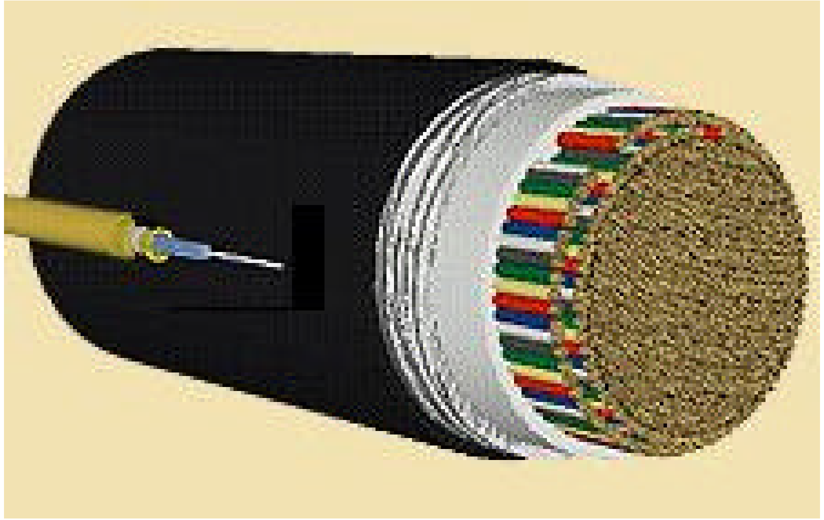


## Fiber optic cable:

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 5 Gps)
- low error rate: repeaters spaced far apart ; immune to electromagnetic noise

# Example





# 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., WaveLAN)
  - 2Mbps, 11Mbps
- wide-area (e.g., cellular)
  - e.g. 3G: hundreds of kbps
- satellite
  - up to 50Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus LEOS

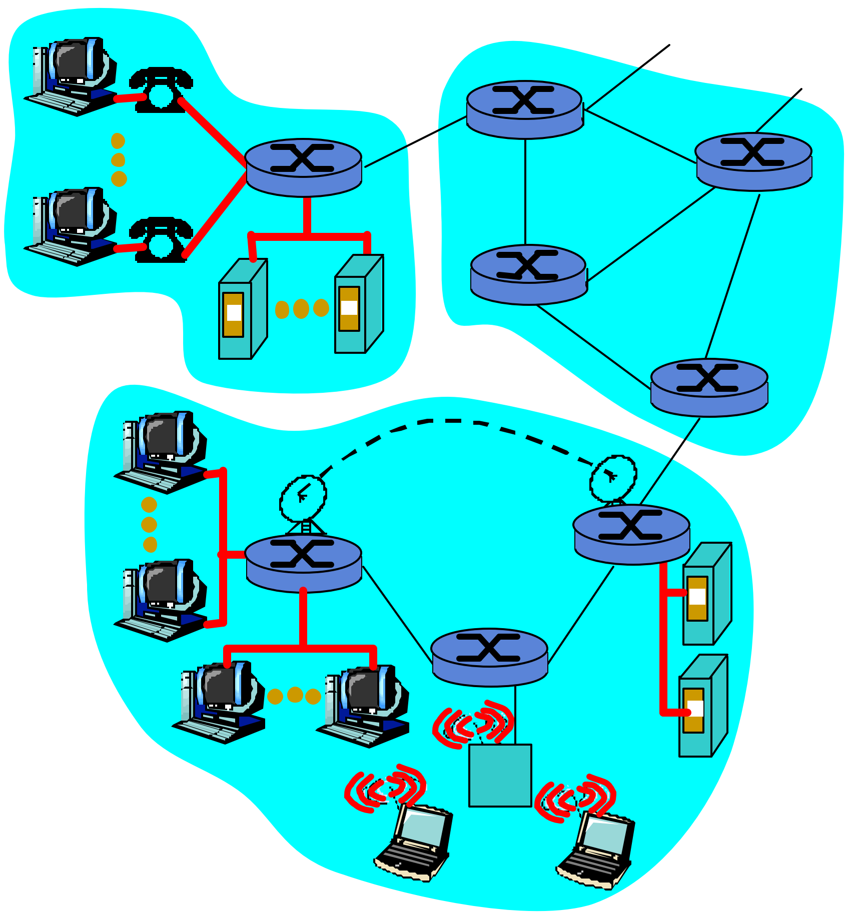
# Access networks and physical media

*Q: How to connection end systems to edge router?*

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

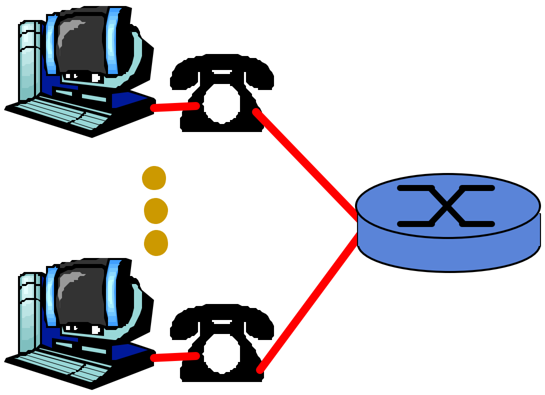
*Keep in mind:*

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



# 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

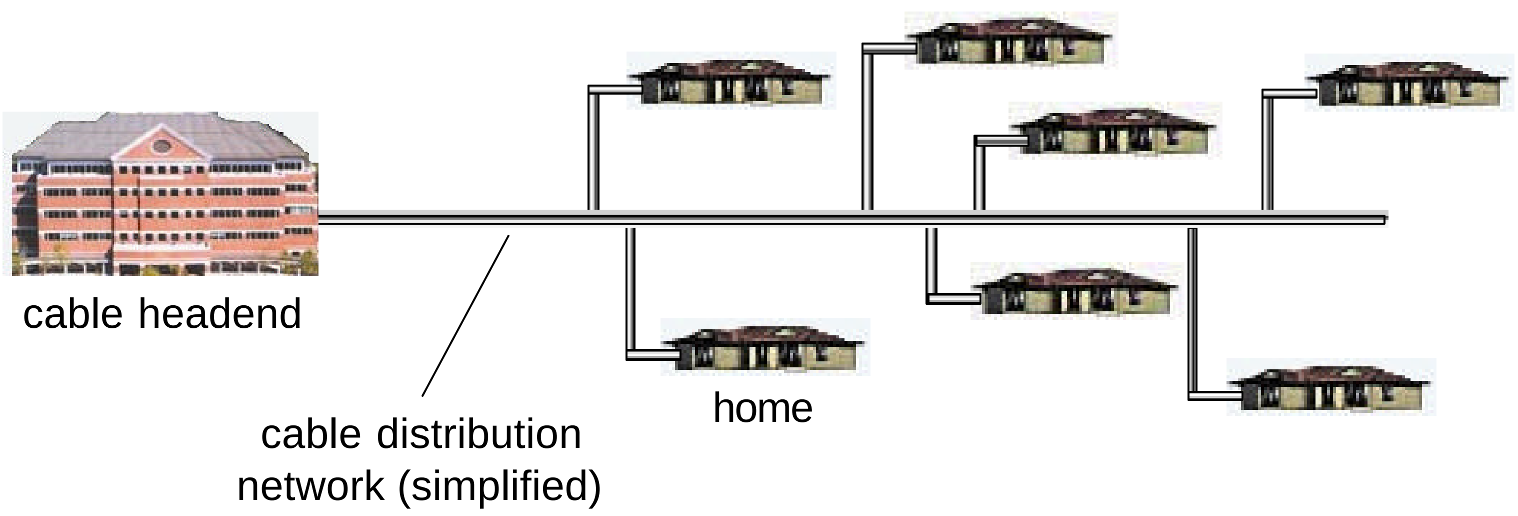
    0 kHz - 4 kHz for ordinary telephone

# Residential access: cable modems

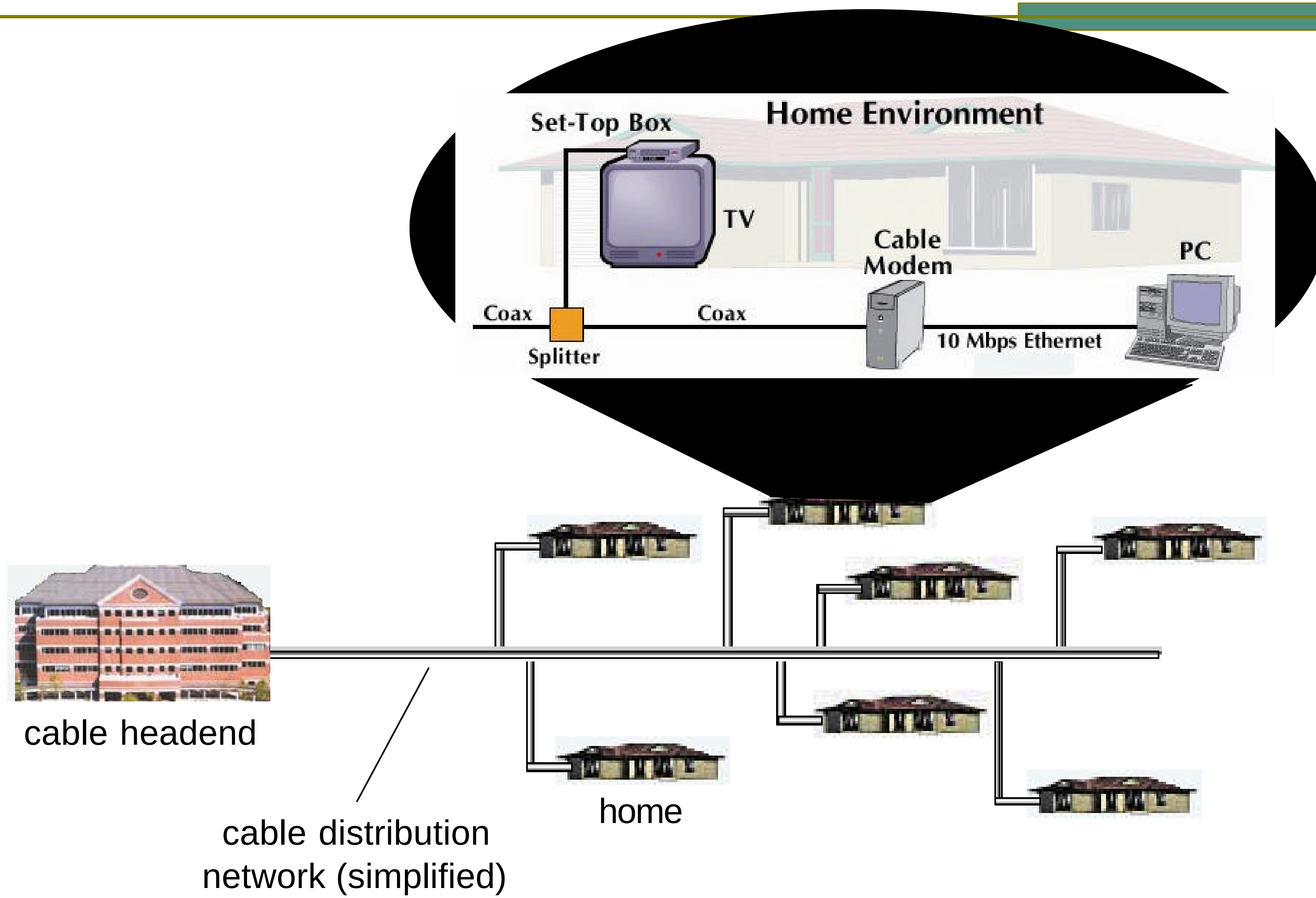
- HFC: hybrid fiber coax
  - asymmetric: up to 10Mbps upstream, 1 Mbps downstream
- network of cable and fiber attaches homes to ISP router
  - shared access to router among home
  - issues: congestion, dimensioning
- deployment: available via cable companies, e.g., Road Runner

# Cable Network Architecture: Overview

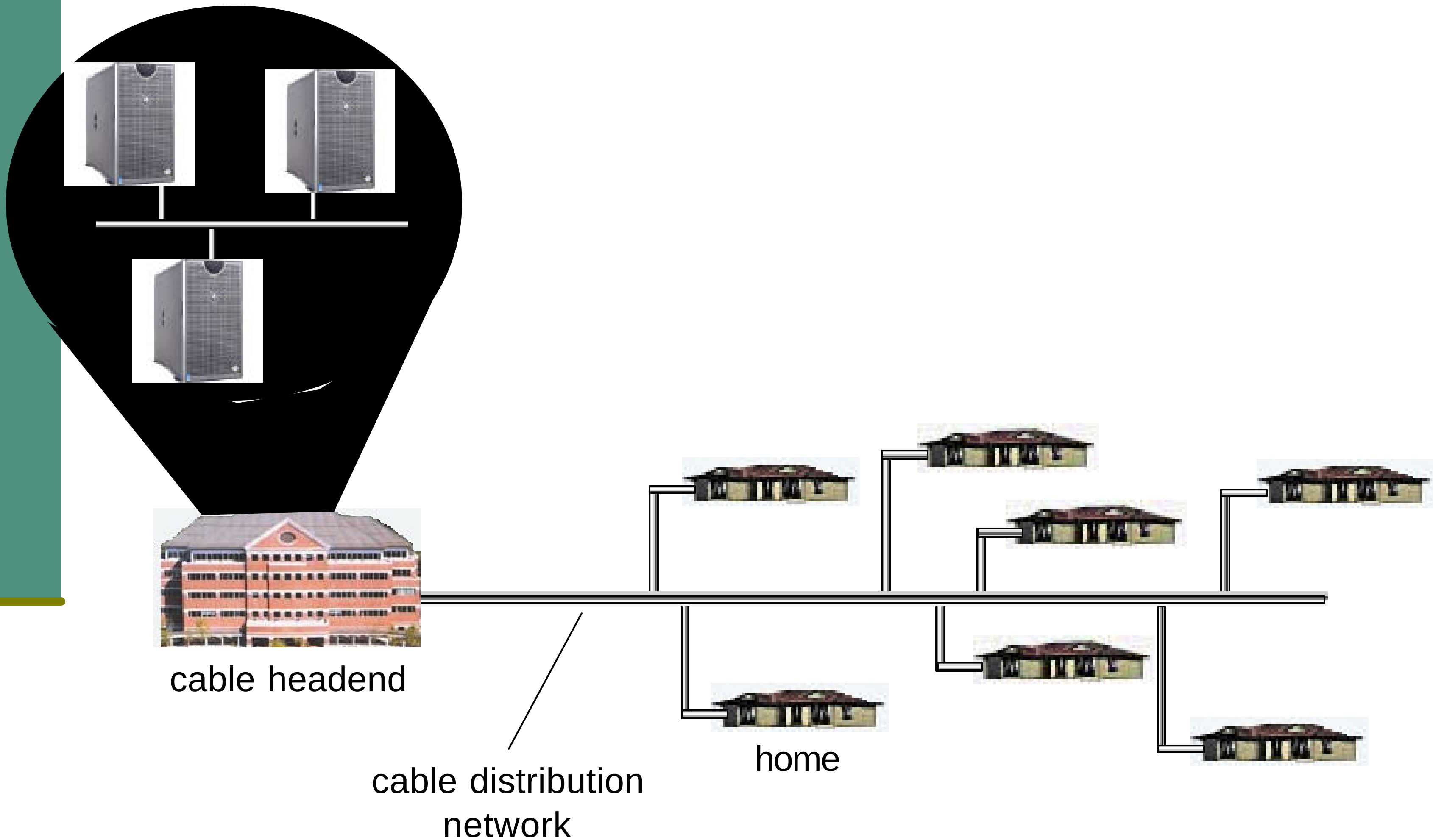
Typically 500 to 5,000 homes



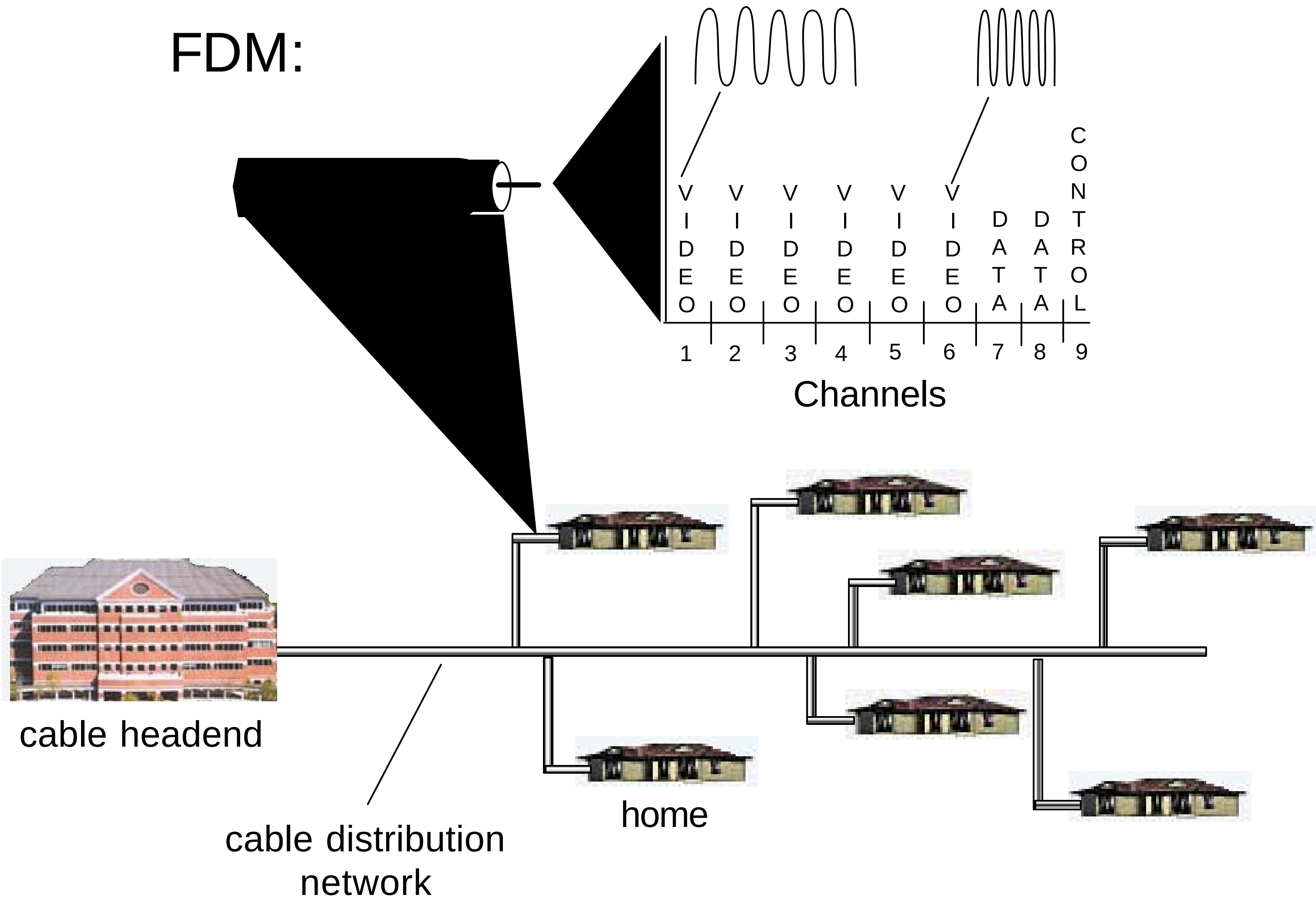
# Cable Network Architecture: Overview



# Cable Network Architecture: Overview

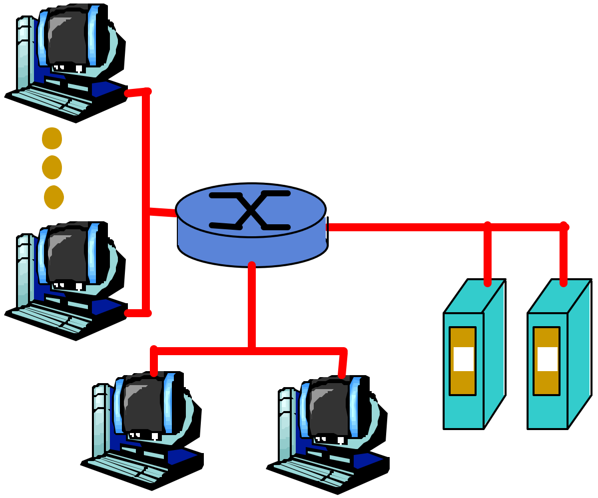


# Cable Network Architecture: Overview



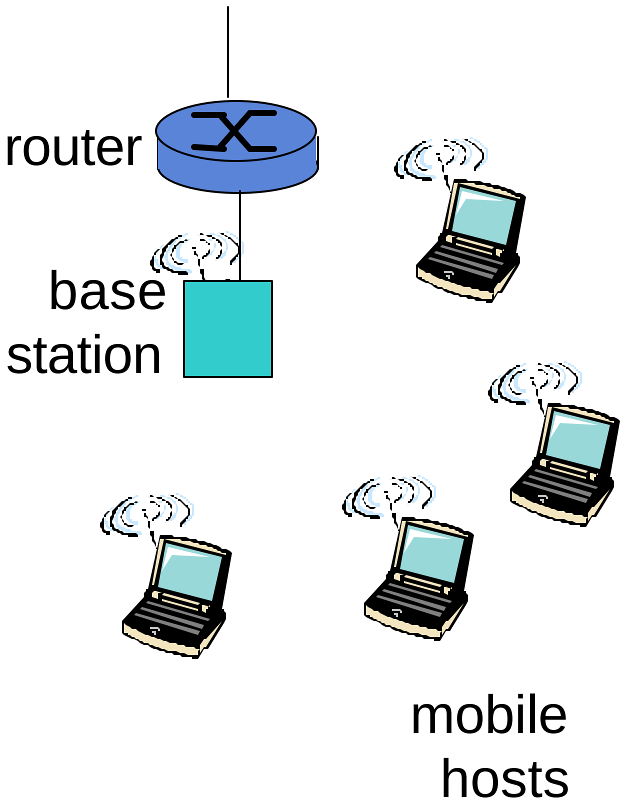
# Company access: local area networks

- 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
- deployment: institutions, home LANs happening now
- LANs: chapter 5



# Wireless access networks

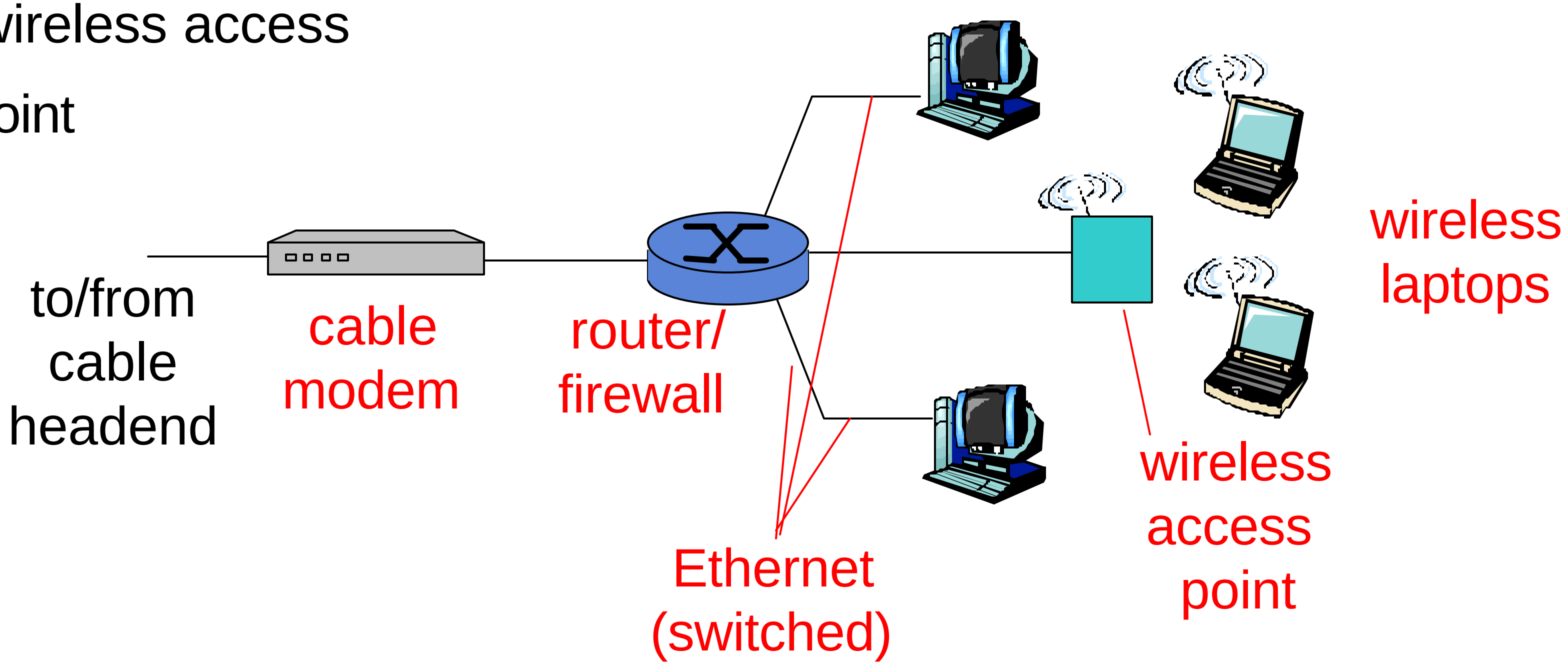
- shared *wireless* access network connects end system to router
  - via base station aka "access point"
- wireless LANs:
  - 802.11b (WiFi): 11 Mbps
- wider-area wireless access
  - provided by telco operator
  - 3G ~ 384 kbps
    - Will it happen??
  - WAP/GPRS in Europe



# Home networks

Typical home network components:

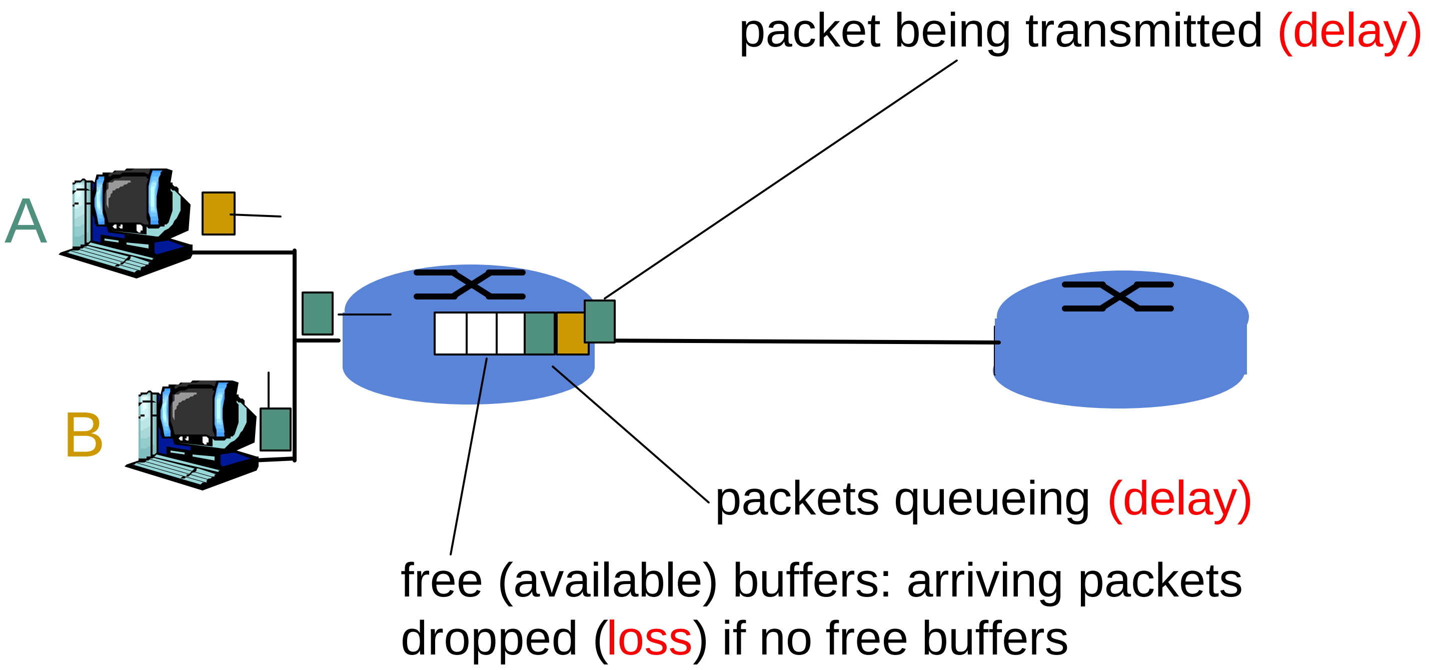
- ADSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point



# How do loss and delay occur?

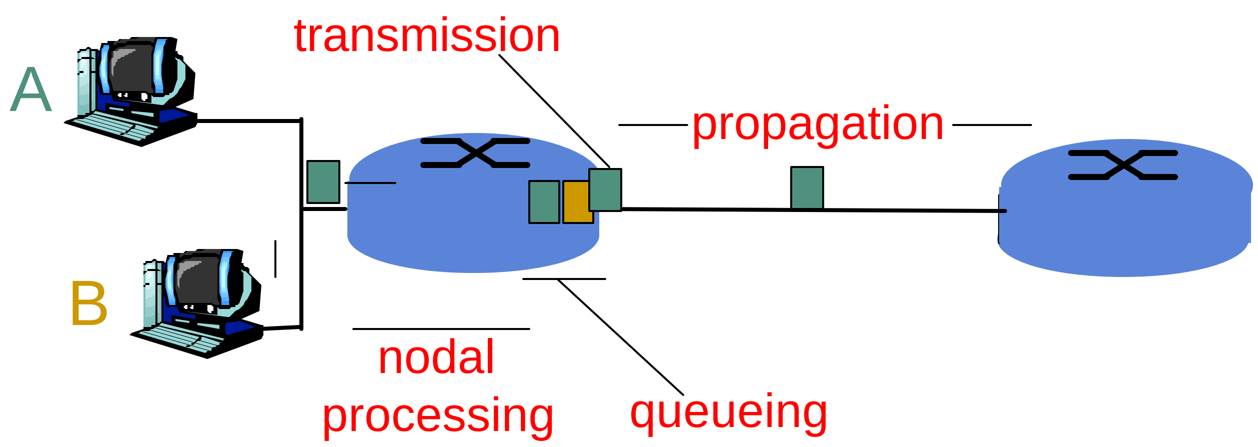
packets *queue* in router buffers

- 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



# Delay in packet-switched networks

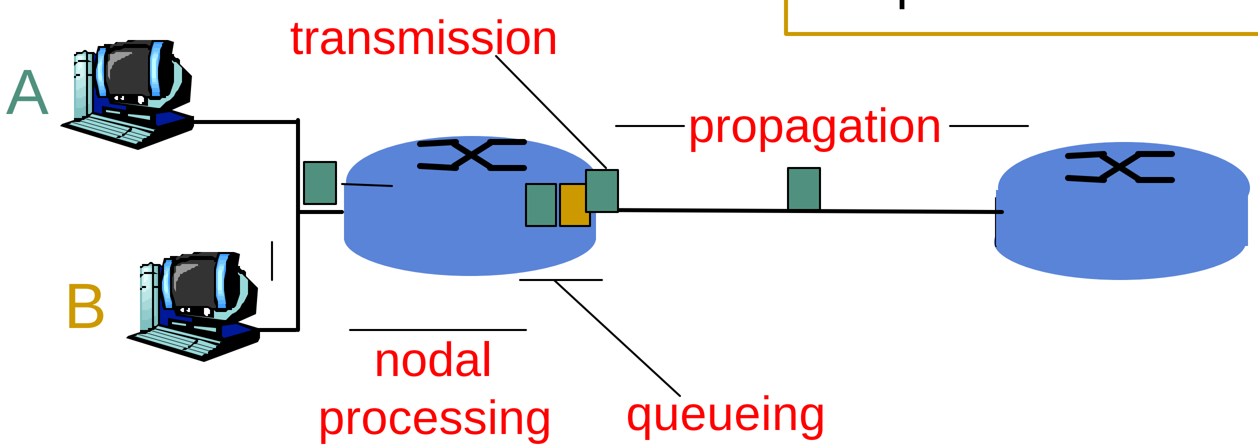
3. Transmission delay:

- R=link bandwidth (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 (~$2 \times 10^8$ m/sec)
- propagation delay = d/s

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



# Nodal delay

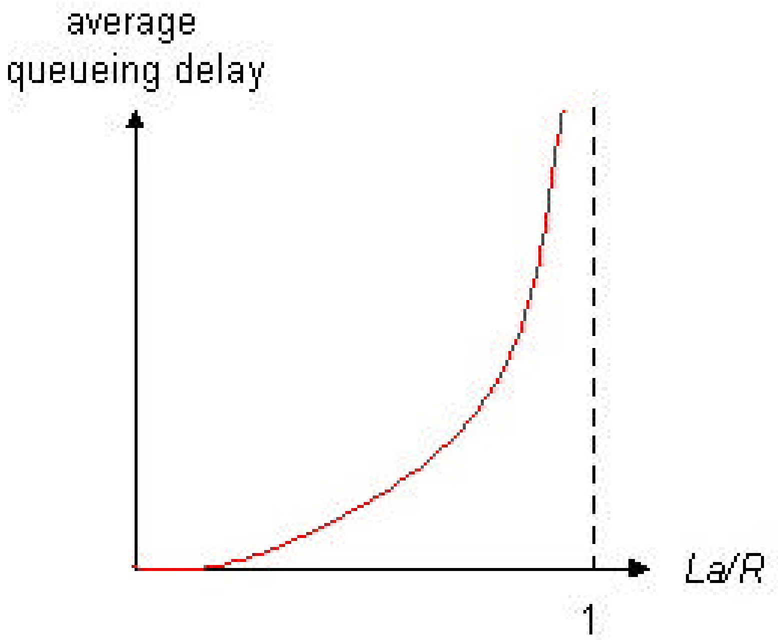
$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

- $d_{proc}$ = 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

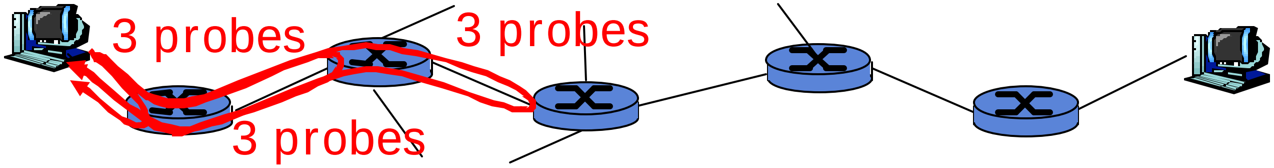
traffic intensity = La/R




- 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!

# “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 measements 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 reponse (probe lost, router not replying)

# Packet loss (Typical)

- queue (aka buffer) preceding link in buffer 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