What is a Network?
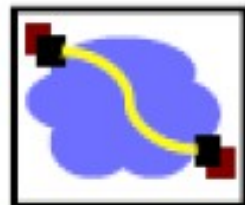
- An infrastructure that allows (distributed) "users" to communicate with each other
  - People, devices, …
  - By means of voice, video, text, …
  - We focus on electrical/optical/RF/.. (not trucks)
- It is assumed that the infrastructure is shared by many users
  - Point to point links are not very interesting
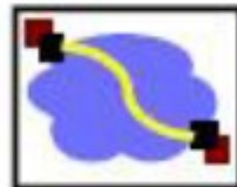 - Value increases with the number of users!

Basic Building Block: Nodes, Links

- Simplest example: 2 nodes
  - Sender changes voltage, frequency, …
  - Or maybe it is optical or wireless?

Node   Link   Node

- But receiver must "understand" sender – protocols
  - More on this later

- Okay… what about more nodes?

- How about a million?
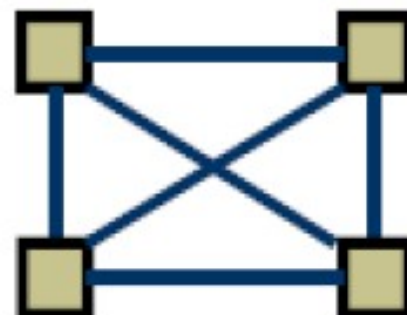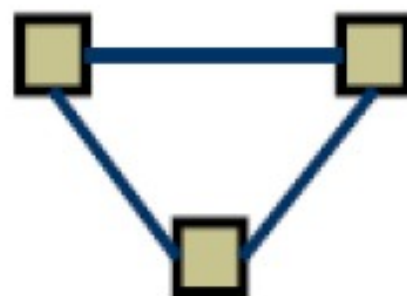
# Networks Have Been Around for a Long time!

- Courier: physical transport of the message
  - Messenger pigeons, pony express, FedEx
- Telegraph: message is transmitted across a network using signals – much faster!
  - Drums, beacons, mirrors, smoke, flags,
  - Light, electricity

Electric Telegraph Networks

- ## Electric telegraph networks exploded
  - ### Message switching & Store-and-Forward operation
  - ### Key elements: Addressing, Routing, Forwarding
- ## Optical telegraph networks disappeared
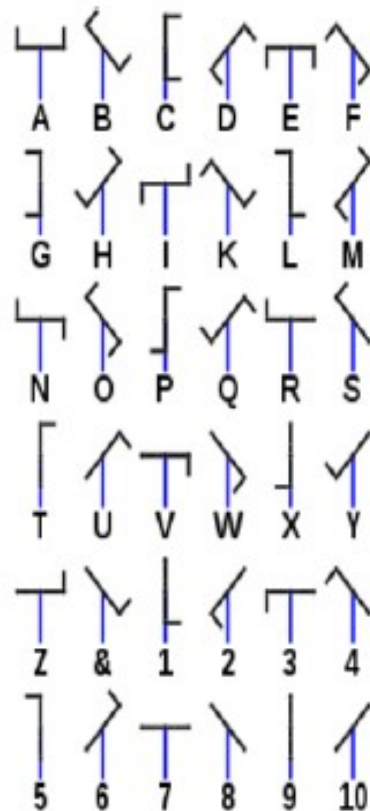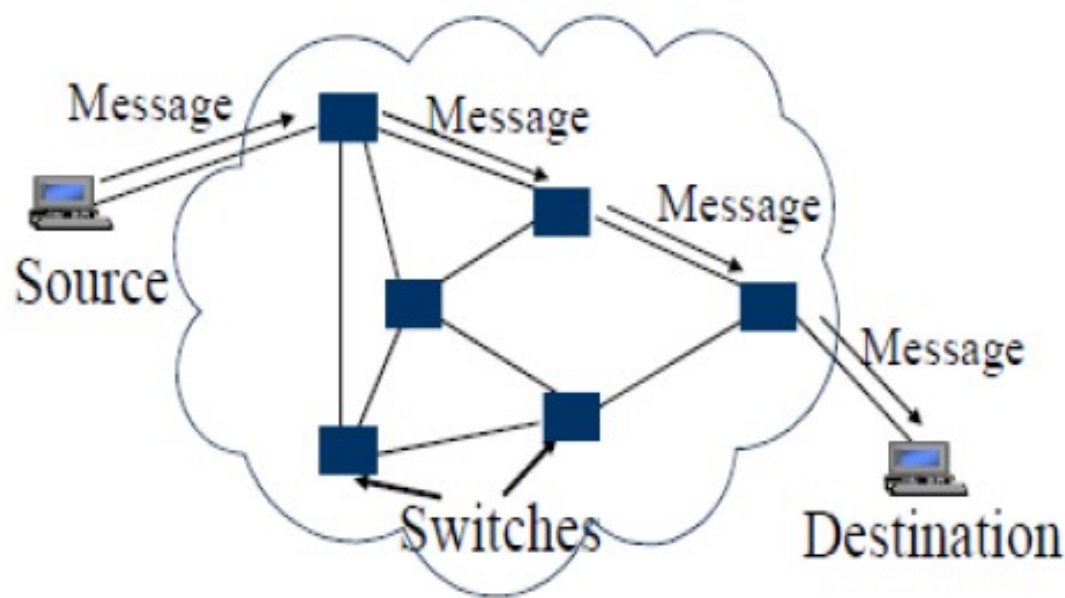
And Some More Examples ...

- Television network
  - Over the air
  - Cable TV
  - Satellite

- Radio broadcast

- Many private networks
  - E.g., for first responders, military, ..

- They are designed for a single application!

- How about the Internet?

Networks Juggle Many Goals

- Support rich set of applications
- Efficiency – resource use, cost
- The "ilities":
  - Evolvability
  - Managability
  - Security (securability, if you must)
  - Scalability
  - Ease of:
    - Deployment, managability
    - Creating useful applications

Chapter 1: introduction

overview:

❖ what's the Internet?

❖ what's a protocol?

❖ network edge; hosts, access net, physical media

❖ network core: packet/circuit switching, Internet structure

❖ performance: loss, delay, throughput

❖ security

❖ protocol layers, service models

❖ history

Chapter 1: roadmap

What's the Internet

PC

server

wireless
laptop

smartphone

wireless
links

——— wired
links

router
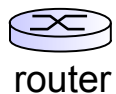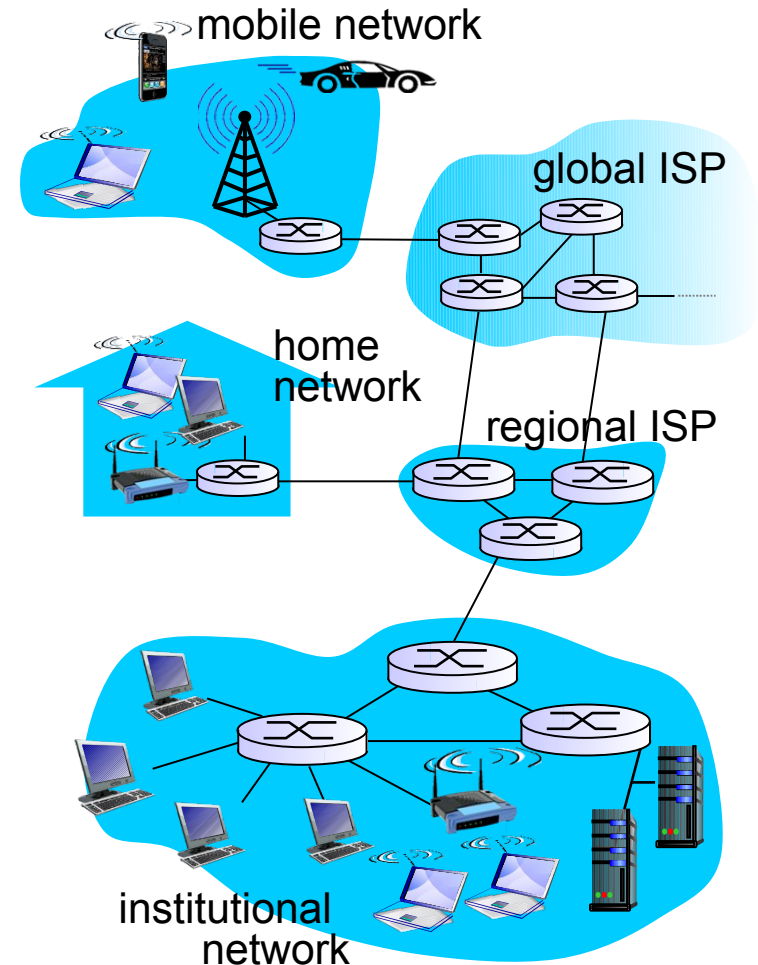
❖ millions of connected
computing devices:
  ▪ *hosts* = *end systems*
  ▪ running *network apps*

❖*communication links*
  ▪ fiber, copper, radio,
    satellite
  ▪ transmission rate:
    bandwidth

❖*Packet switches:* forward
packets (chunks of data)
  ▪ *routers* and *switches*

mobile network

global ISP

home
network

regional ISP

institutional
network

"Fun" internet appliances

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

Web-enabled toaster +
weather forecaster

Tweet-a-watt:
monitor energy use

Internet
refrigerator

Slingbox: watch,
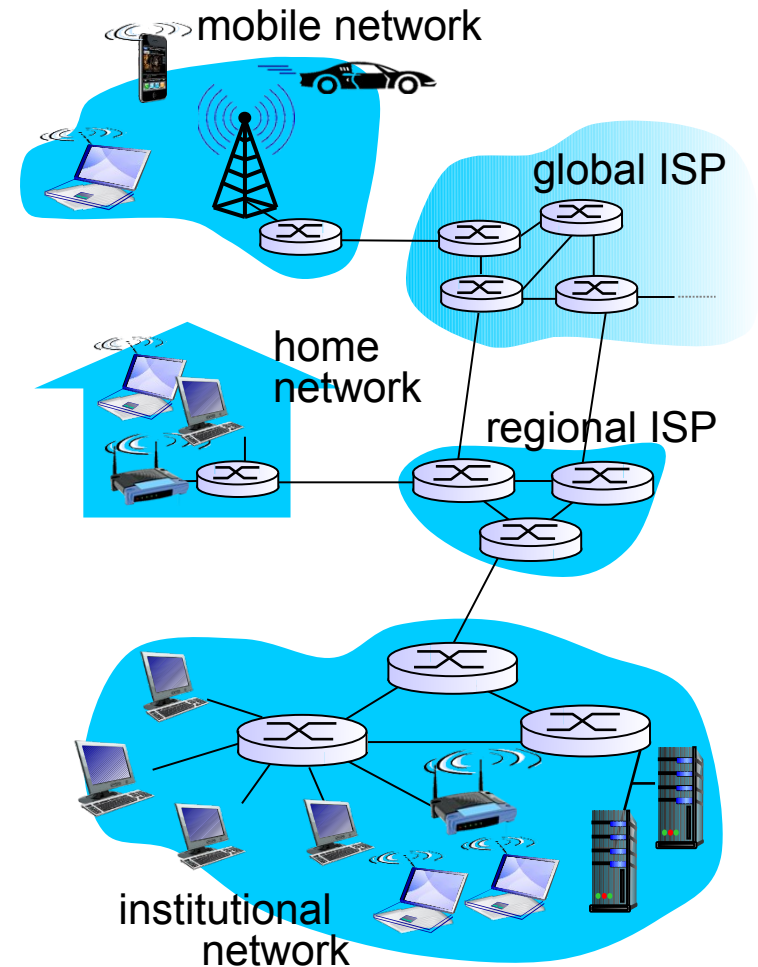control cable TV remotely

Internet phones

What's the Internet

- *Internet:* "network of networks"
  - Interconnected ISPs
- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, 802.11
- *Internet standards*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force

mobile network

global ISP

home network

regional ISP

institutional network

What's the Internet: a service view

❖ *Infrastructure that provides services to applications:*
- Web, VoIP, email, games, e-commerce, social nets, …

❖ *provides programming interface to apps*
- hooks that allow sending and receiving app programs to "connect" to Internet
- provides service options, analogous to postal service

mobile network

global ISP

home network

regional ISP

institutional network
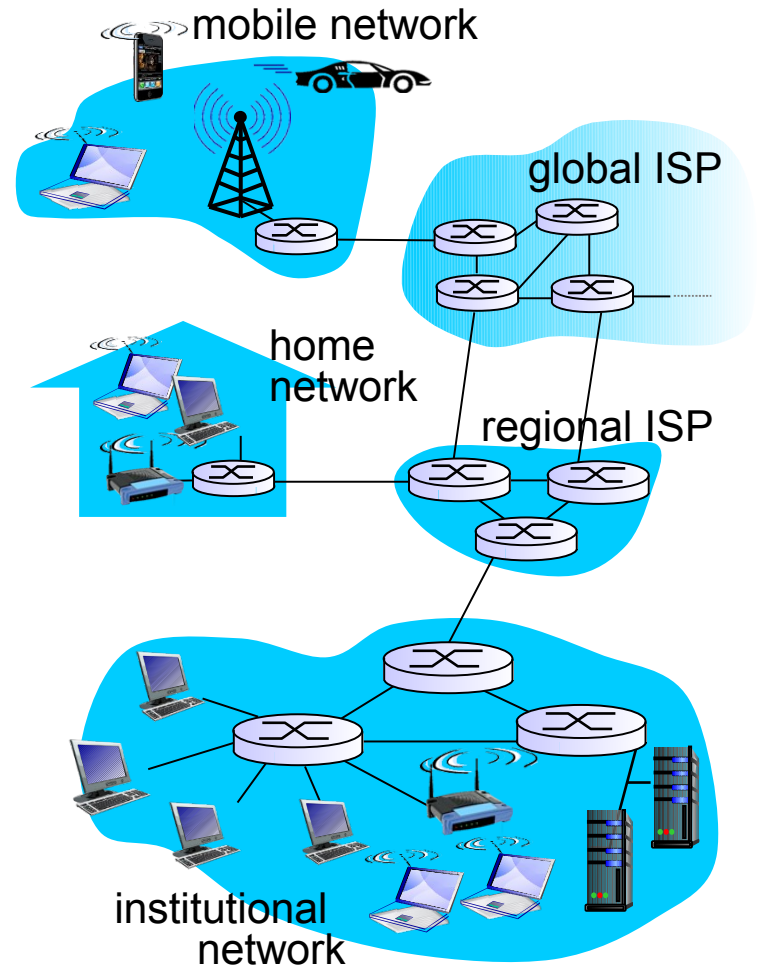
What's a protocol?

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:

time
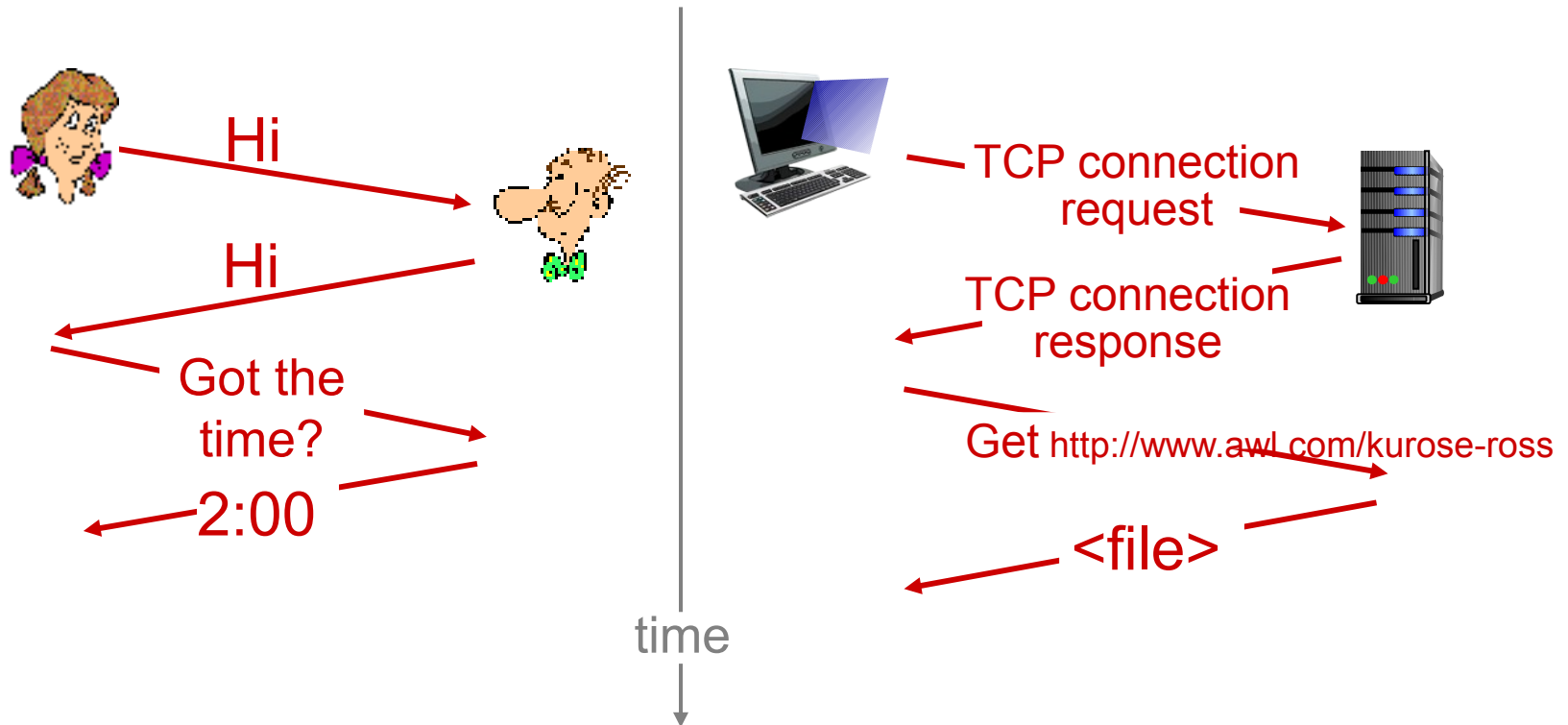
Q: other human protocols?

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security
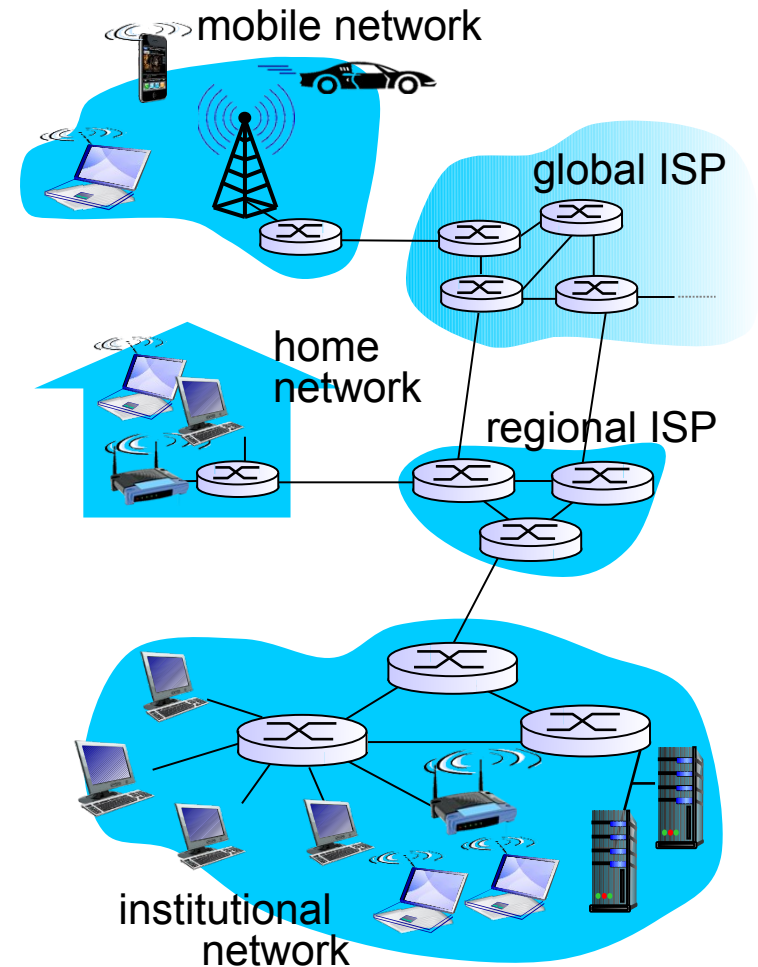
1.7 history

A closer look at network structure:

❖ *network edge:*
- hosts: clients and servers
- servers often in data centers

❖*access networks, physical media:* wired, wireless communication links

❖*network core:*
- interconnected routers
- network of networks

mobile network

global ISP

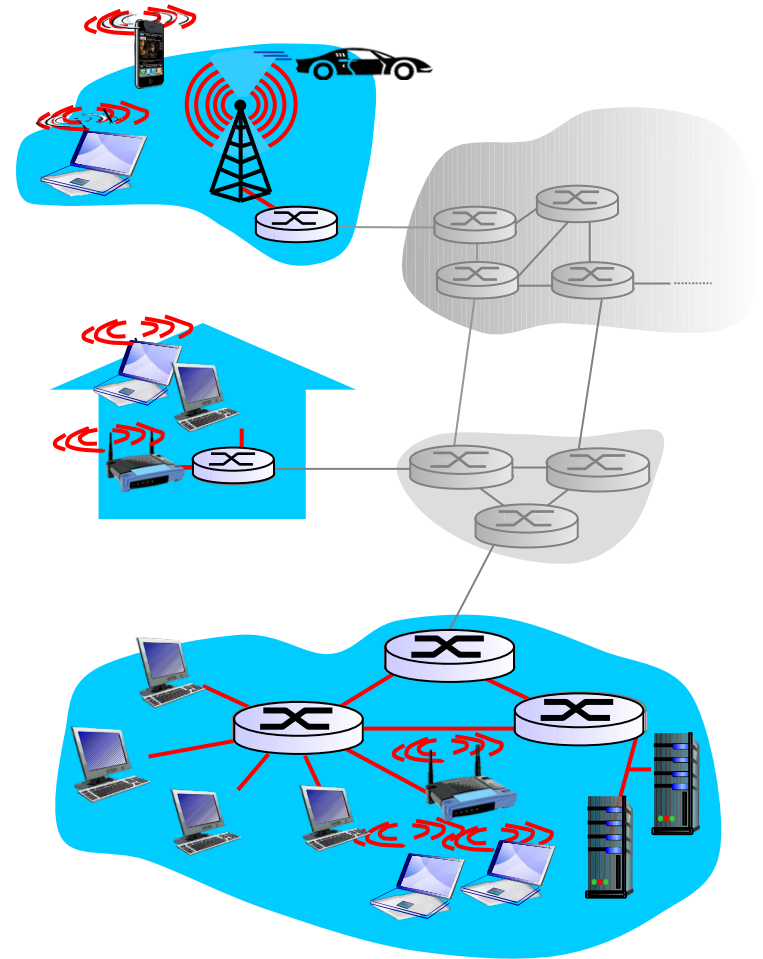home network

regional ISP

institutional network

Access networks and physical media

Q: How to connect end systems to edge router?

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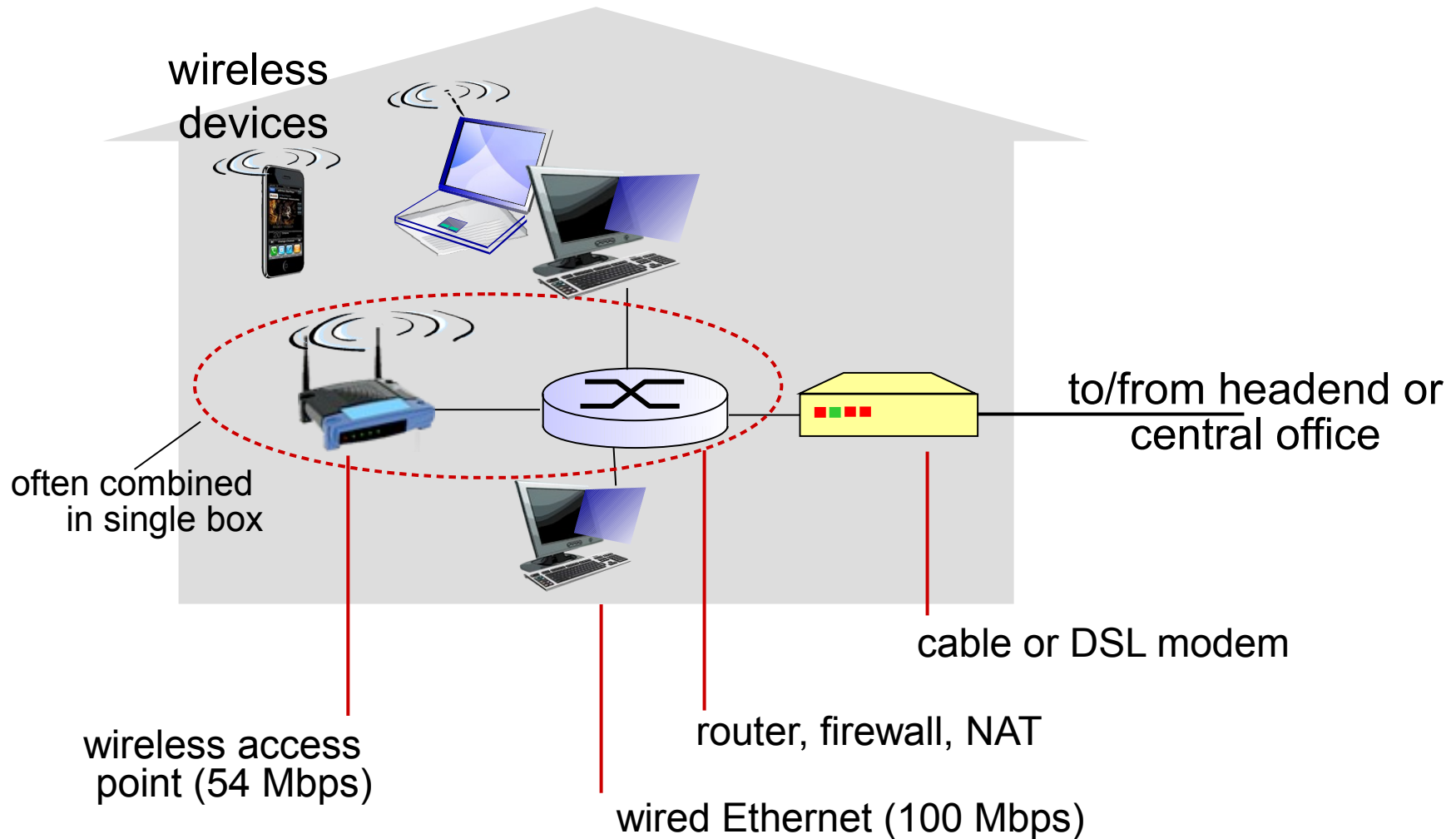- ❖ institutional access networks (school, company)
- ❖ mobile access networks

keep in mind:

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

Access net: home network

wireless devices

often combined in single box

to/from headend or central office

cable or DSL modem

wireless access point (54 Mbps)

router, firewall, NAT

wired Ethernet (100 Mbps)

Enterprise access networks (Ethernet)

institutional link to
ISP (Internet)

institutional router

Ethernet
switch

institutional mail,
web servers
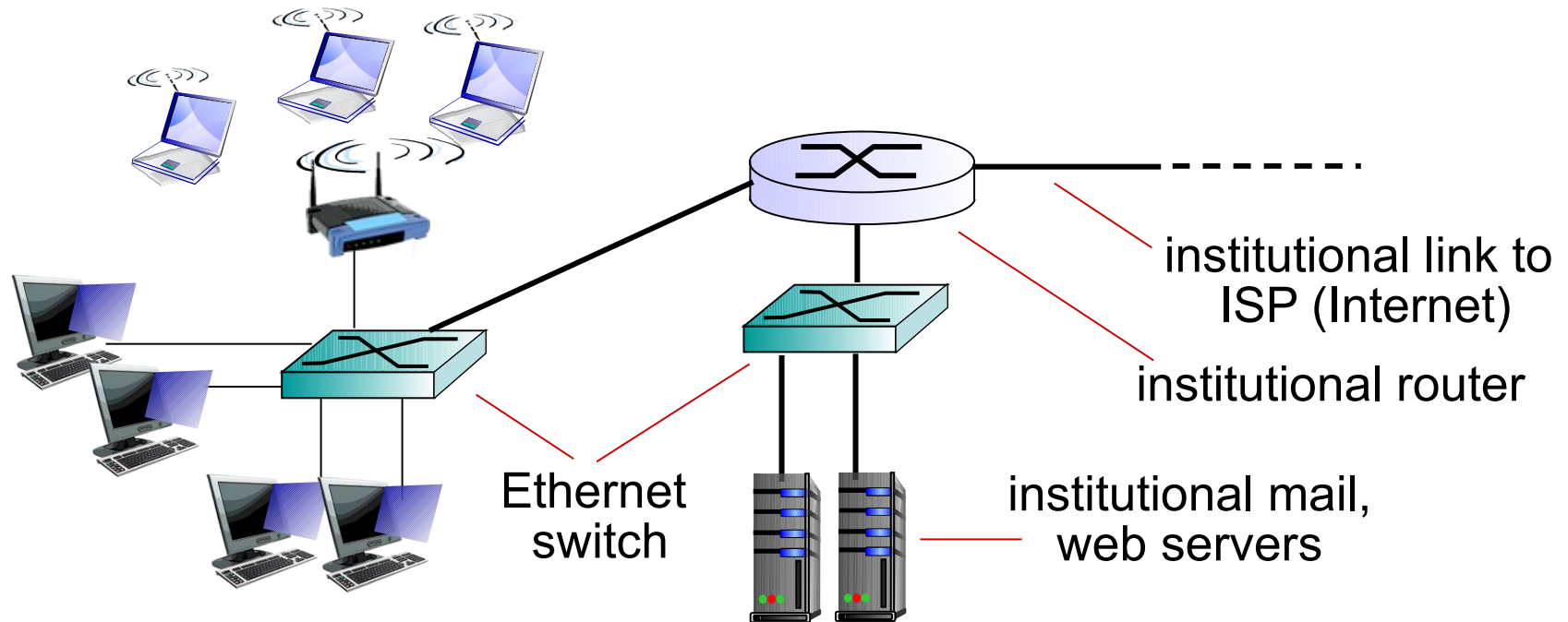
❖ typically used in companies, universities, etc
❖ 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
❖ today, end systems typically connect into Ethernet switch

Wireless access networks

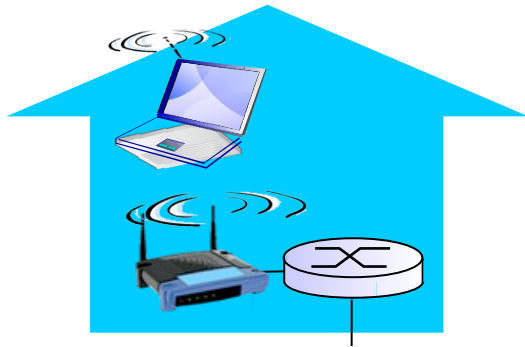❖ shared *wireless* access network connects end system to router
   · via base station aka "access point"
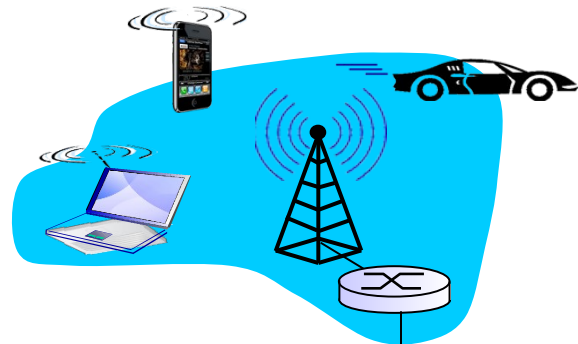
wireless LANs:
   · within building (100 ft)
   · 802.11b/g (WiFi): 11, 54
      Mbps transmission rate

to Internet

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

to Internet

Host: sends *packets* of data

host sending function:

❖ takes application message

❖ breaks into smaller chunks, known as *packets*, of length L bits

❖ transmits packet into access network at *transmission rate R*

• link transmission rate, aka link *capacity, aka link bandwidth*

two packets, L bits each

2 1

R: link transmission rate

host
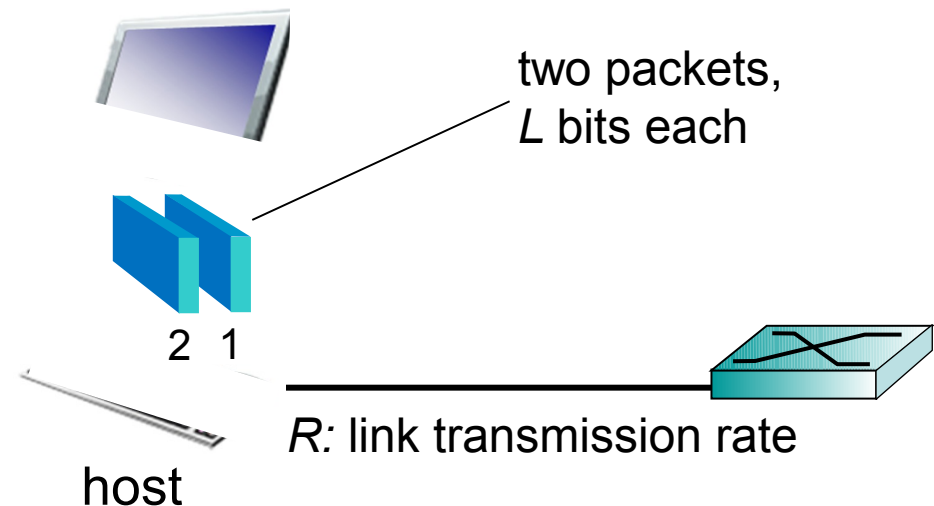
$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

Physical media

❖ **bit:** propagates between transmitter/receiver 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 5: 100 Mbps, 1 Gpbs Ethernet
  - Category 6: 10Gbps

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)
  - 3G cellular: ~ few Mbps

❖satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - geosynchronous versus low altitude

Chapter 1: roadmap

The network core

❖ mesh of interconnected routers

❖ packet-switching: hosts break application-layer messages into *packets*

• forward packets from one router to the next, across links on path from source to destination

• each packet transmitted at full link capacity

Packet-switching: store-and-forward

L bits per packet

3 2 1

source

R bps

R bps

destination

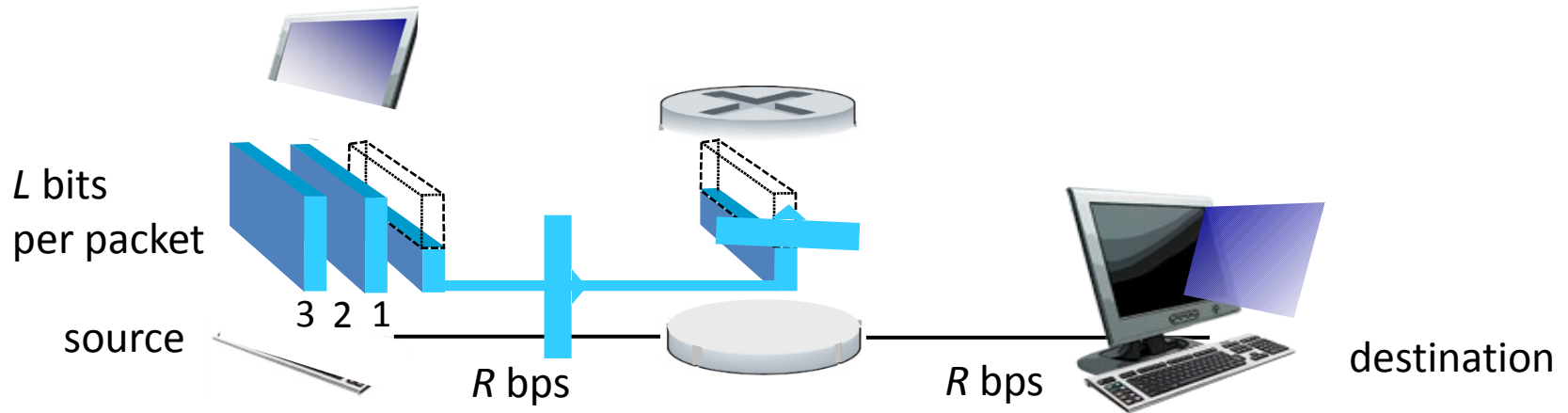❖ takes *L/R* seconds to transmit (push out) *L*-bit packet into link at *R* bps

❖ *store and forward:* entire packet must arrive at router before it can be transmitted on next link

❖ end-end delay = 2*L/R* (assuming zero propagation delay)

one-hop numerical example:

- *L* = 7.5 Mbits

- *R* = 1.5 Mbps

- one-hop transmission delay = 5 sec

more on delay shortly …

Packet Switching: queueing delay, loss

A → R = 100 Mb/s

B

queue of packets
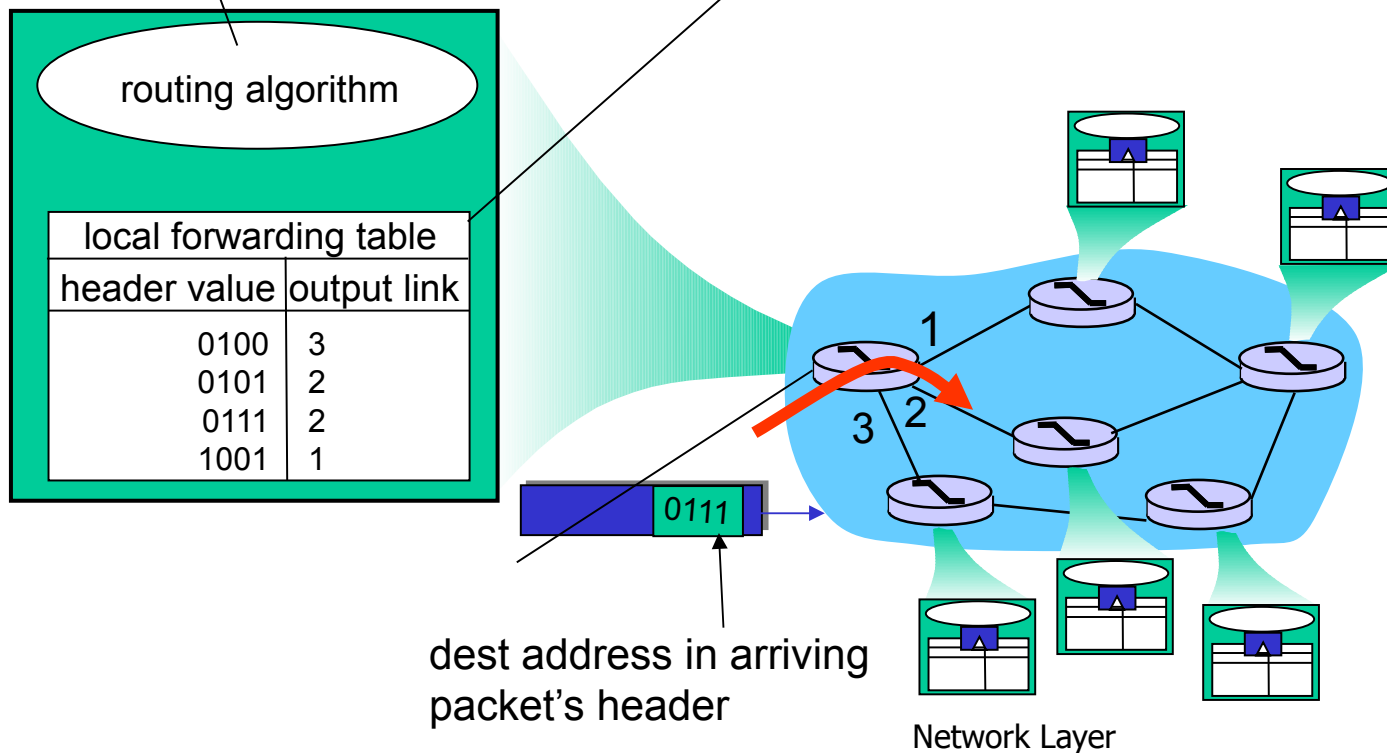waiting for output link

R = 1.5 Mb/s

C

D

E
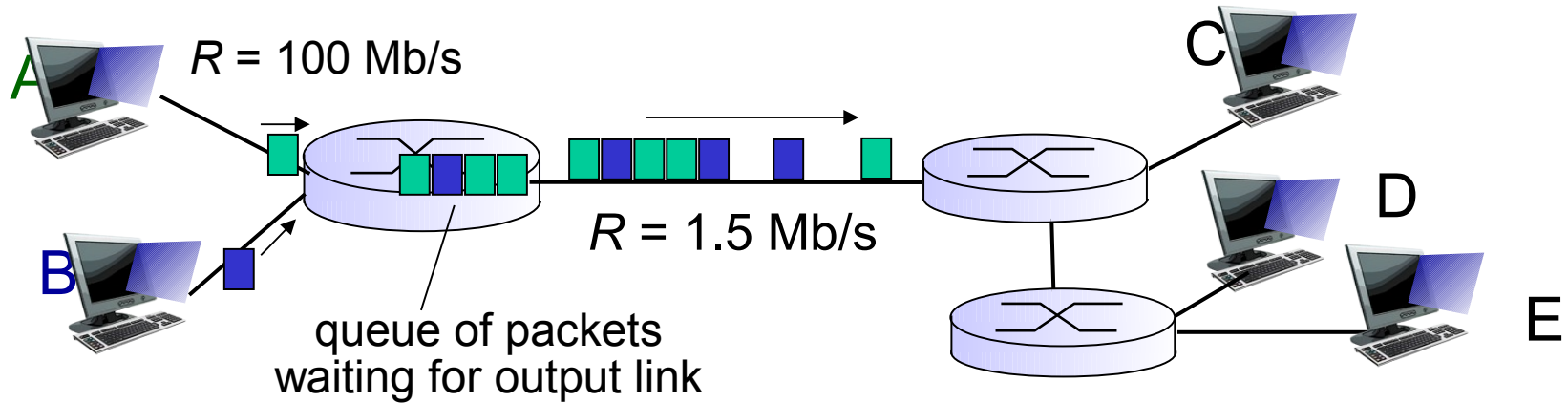
queuing and loss:

❖ If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
- packets will queue, wait to be transmitted on link
- packets can be dropped (lost) if memory (buffer) fills up

Two key network-core functions

routing: determines source-destination route taken by packets

- *routing algorithms*

*forwarding:* move packets from router's input to appropriate router output

routing algorithm

local forwarding table	
header value	output link
0100	3
0101	2
0111	2
1001	1

0111

dest address in arriving packet's header

Alternative core: circuit switching

end-end resources allocated to, reserved for "call" between source & dest:

❖ In diagram, each link has four circuits.

  ▪ call gets 2nd circuit in top link and 1st circuit in right link.

❖ dedicated resources: no sharing

  ▪ circuit-like (guaranteed) performance

❖ circuit segment idle if not used by call *(no sharing)*

❖ Commonly used in traditional telephone networks
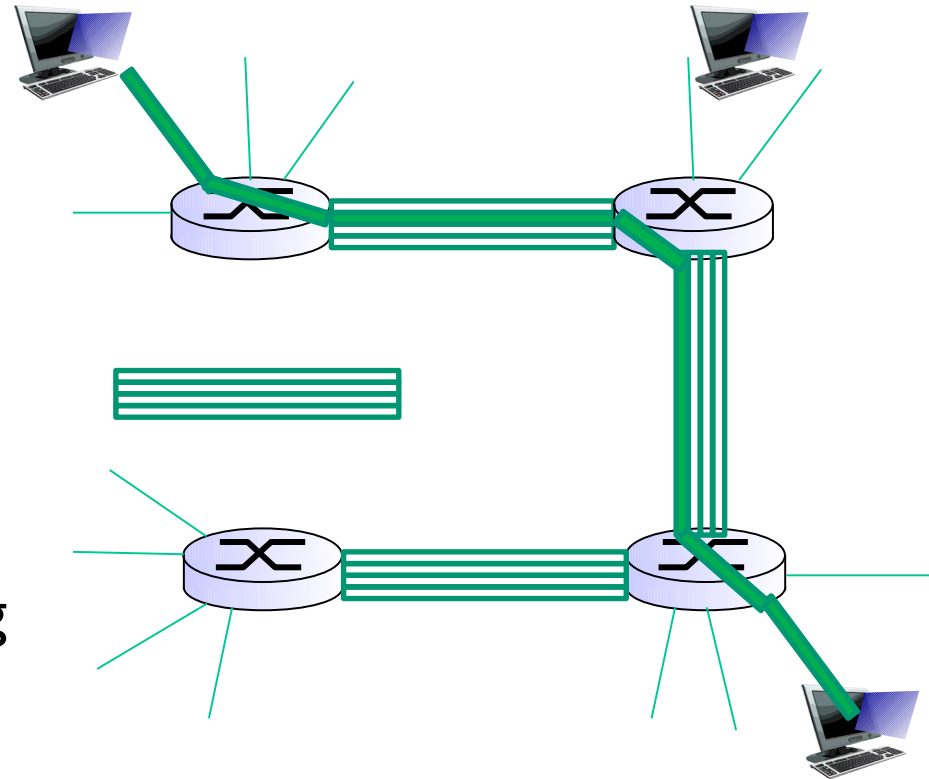
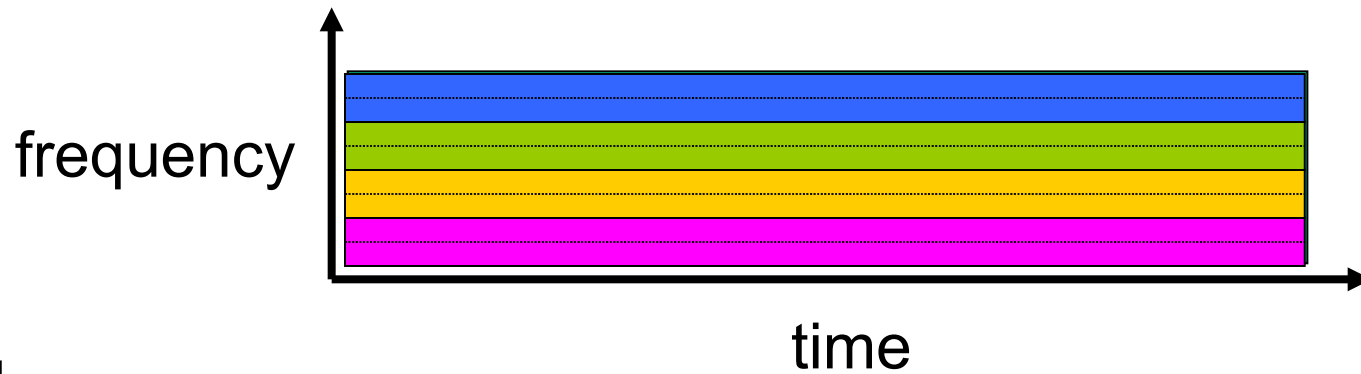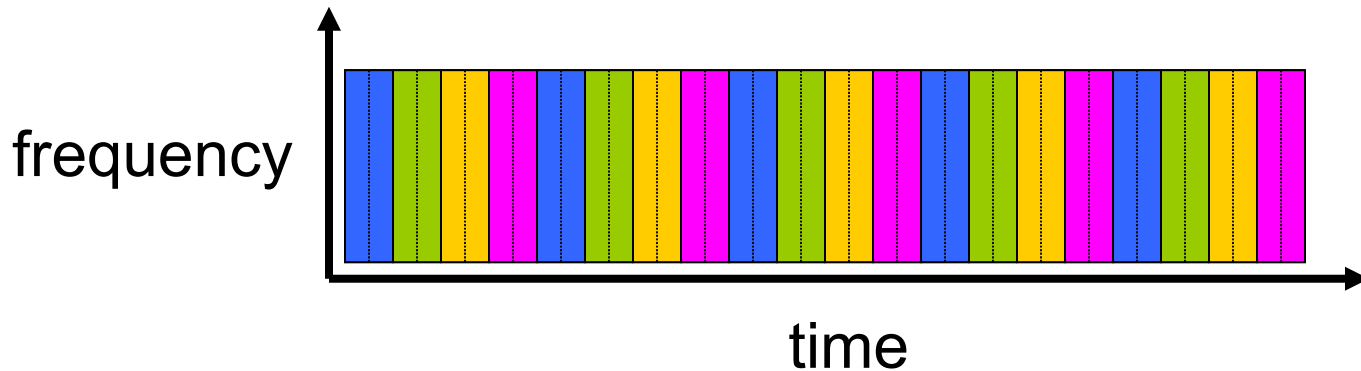# Circuit switching: FDM versus TDM

FDM

Example:

4 users

frequency

time

TDM

frequency

time

Packet switching versus circuit switching

is packet switching a "slam dunk winner?"

❖ 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)

Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

Internet structure: network of networks

- ❖ End systems connect to Internet via access ISPs (Internet Service Providers)
  - · Residential, company and university ISPs
- ❖ Access ISPs in turn must be interconnected.
  - ❖ So that any two hosts can send packets to each other
- ❖ Resulting network of networks is very complex
  - ❖ Evolution was driven by economics and national policies
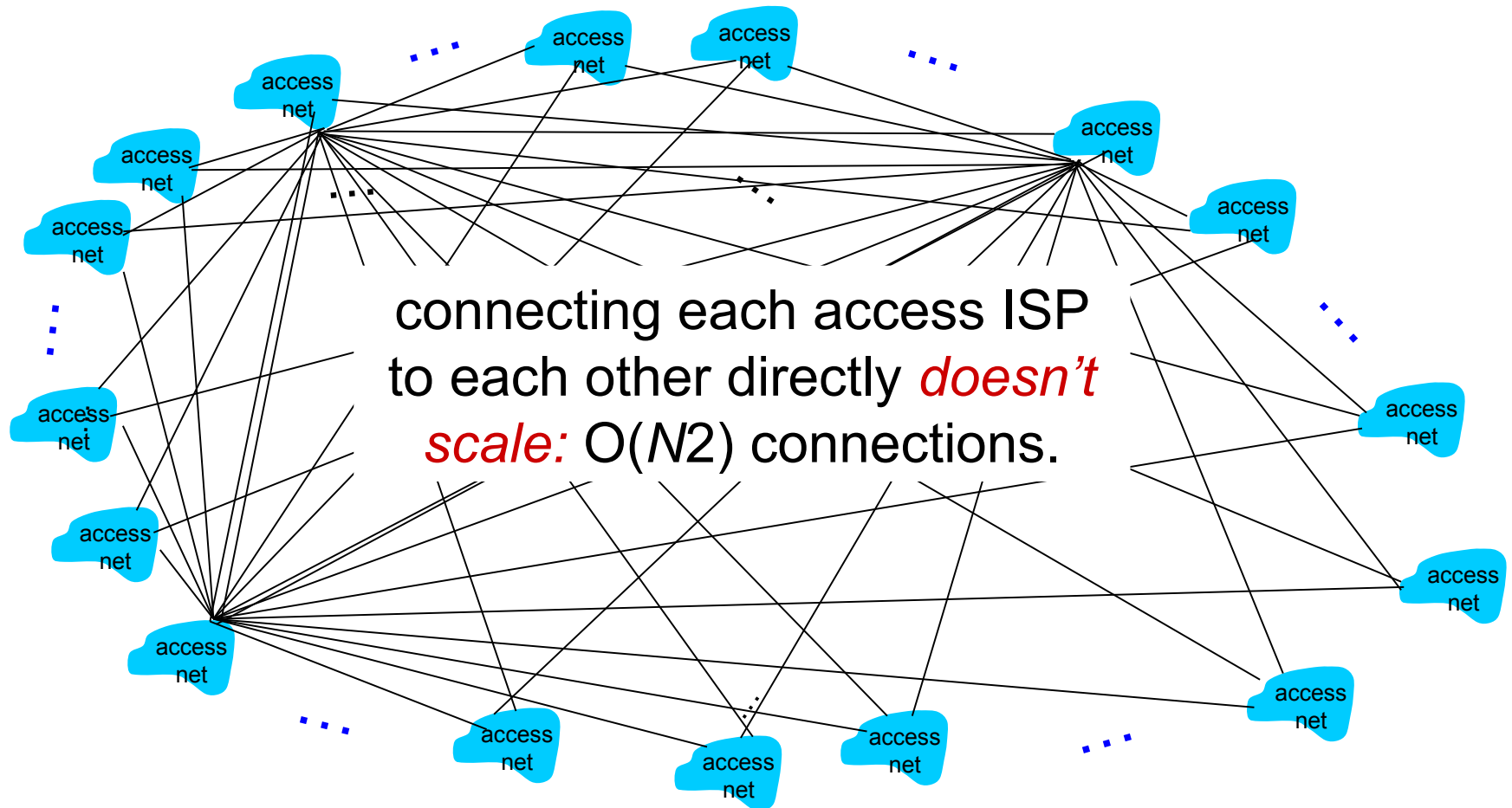- ❖ Let's take a stepwise approach to describe current Internet structure

Internet structure: network of networks

Question: given *millions* of access ISPs, how to connect them together?
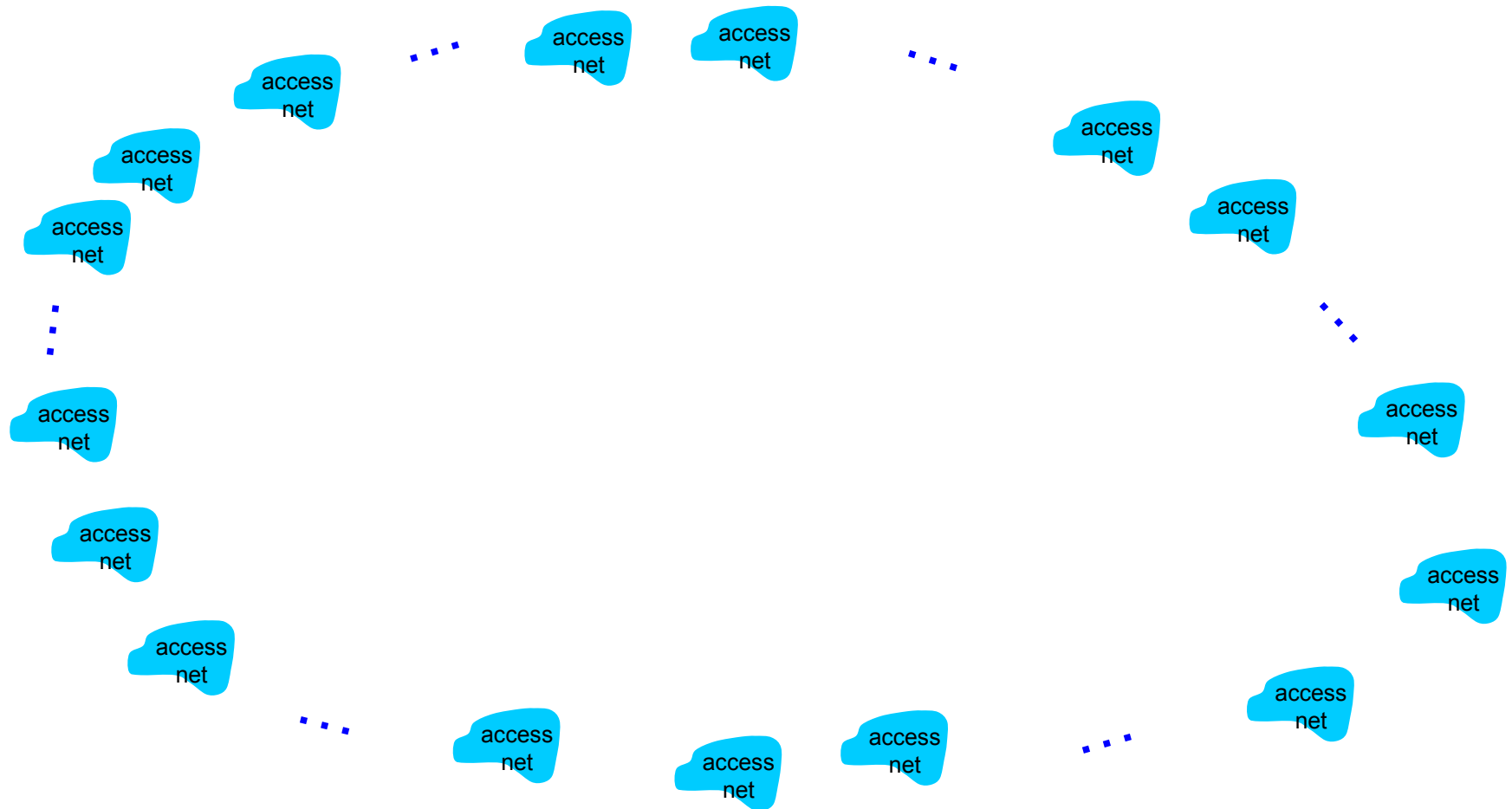
Internet structure: network of networks

Option: *connect each access ISP to every other access ISP?*

connecting each access ISP to each other directly *doesn't scale:* O(*N2*) connections.

Internet structure: network of networks

Option: connect each access ISP to a global transit ISP? *Customer* and *provider* ISPs have economic agreement.
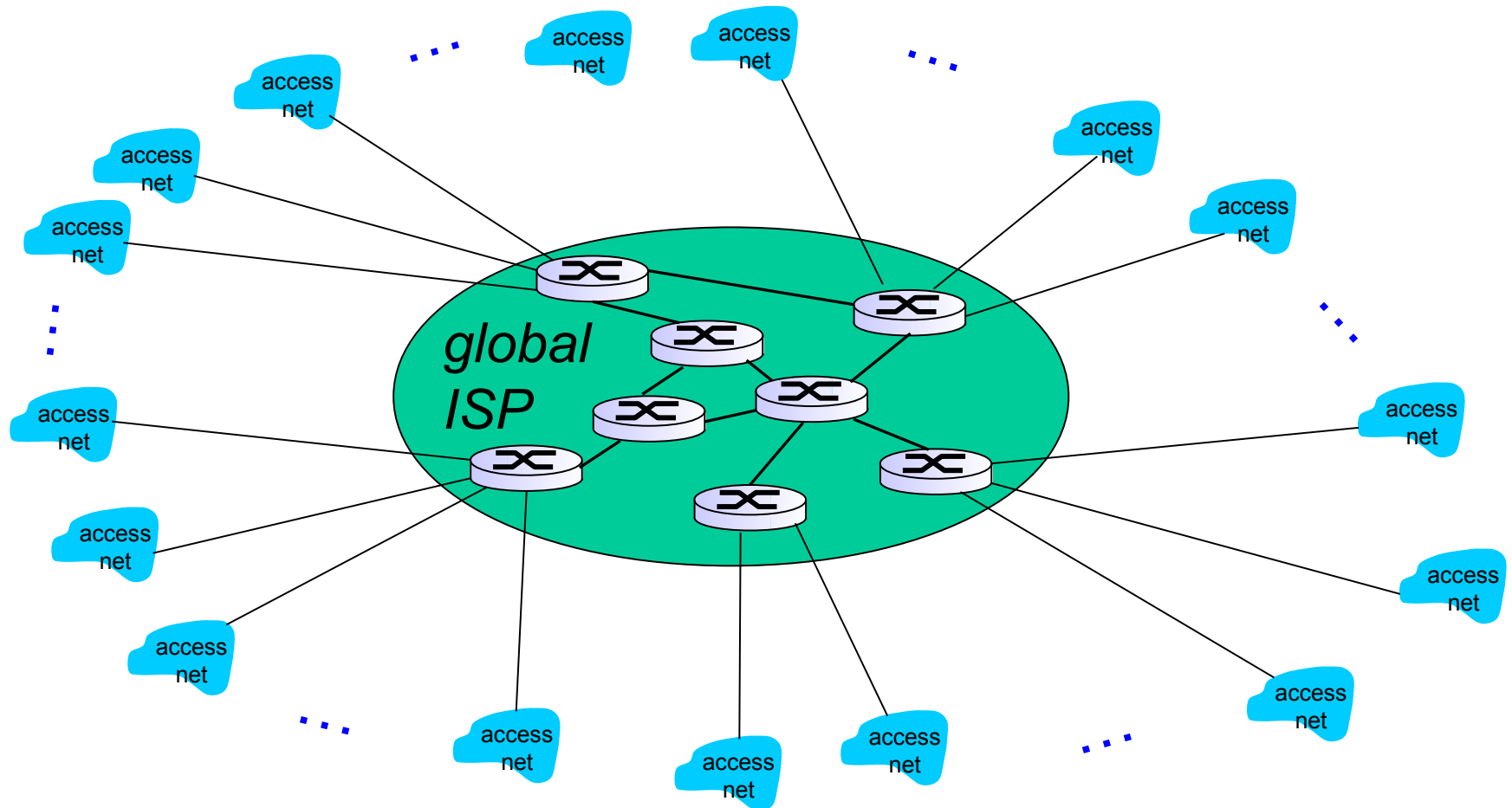
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors
….
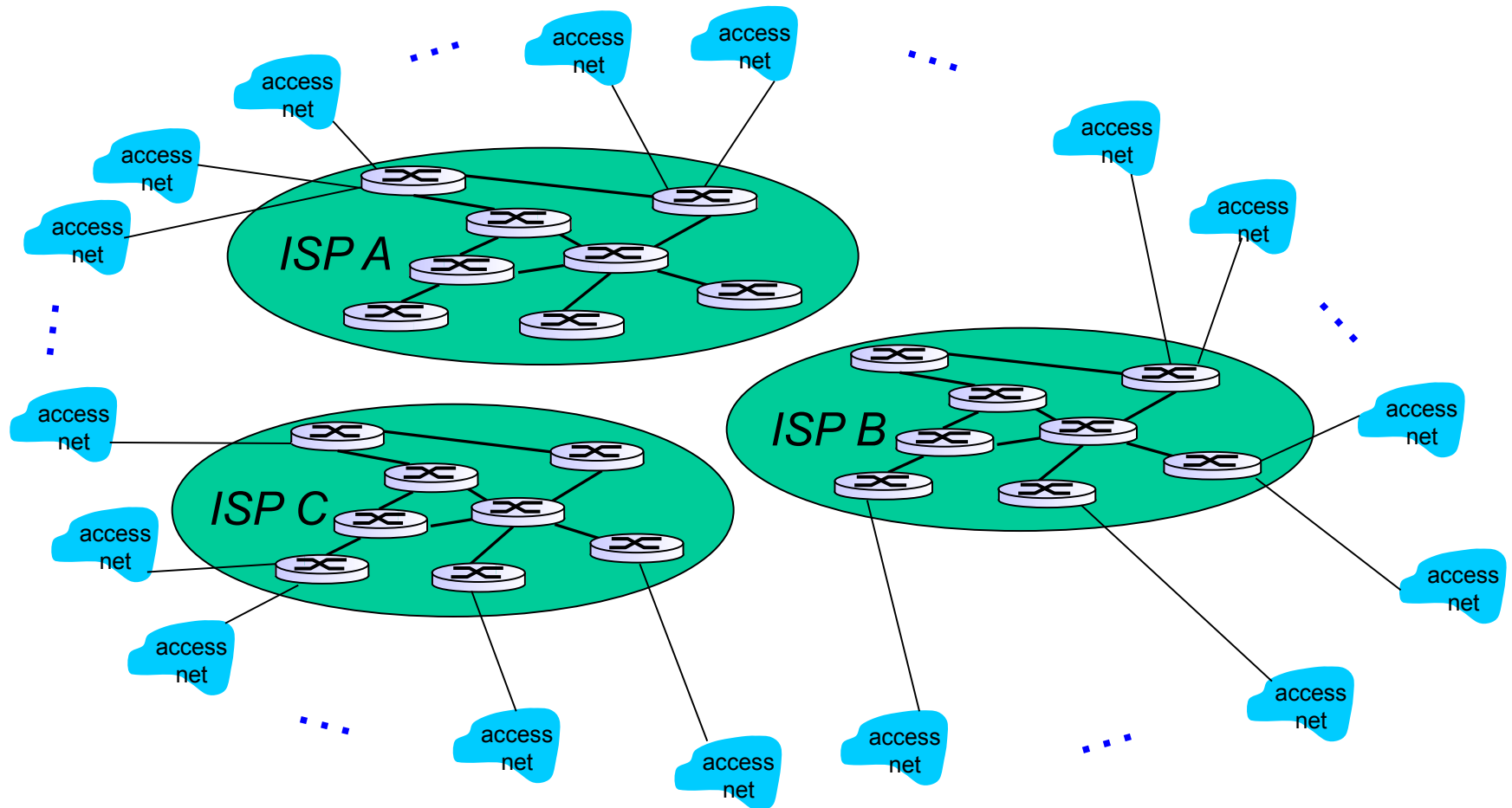
Internet structure: network of networks

But if one global ISP is viable business, there will be competitors …. which must be interconnected
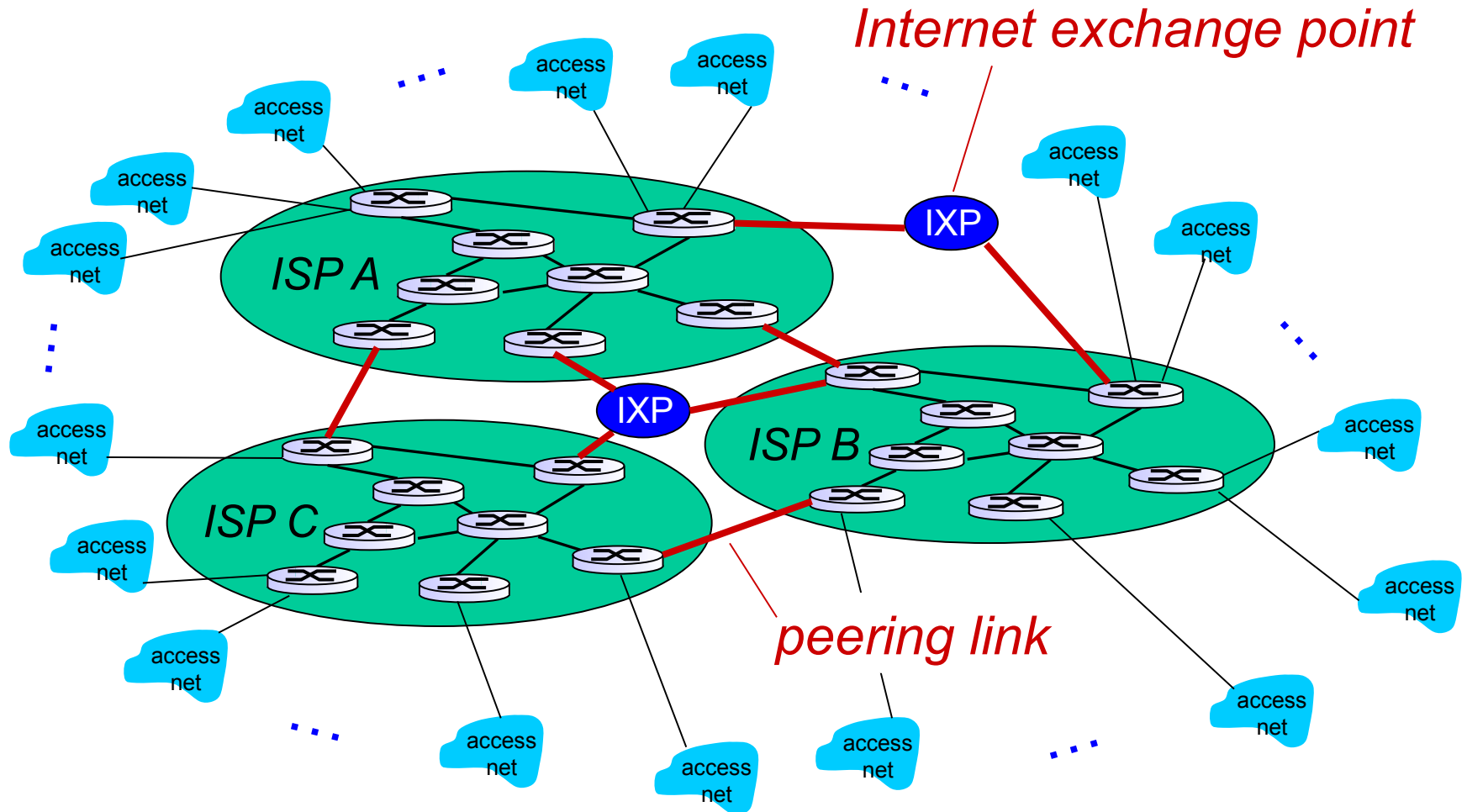
Internet structure: network of networks

… and regional networks may arise to connect access nets to ISPS
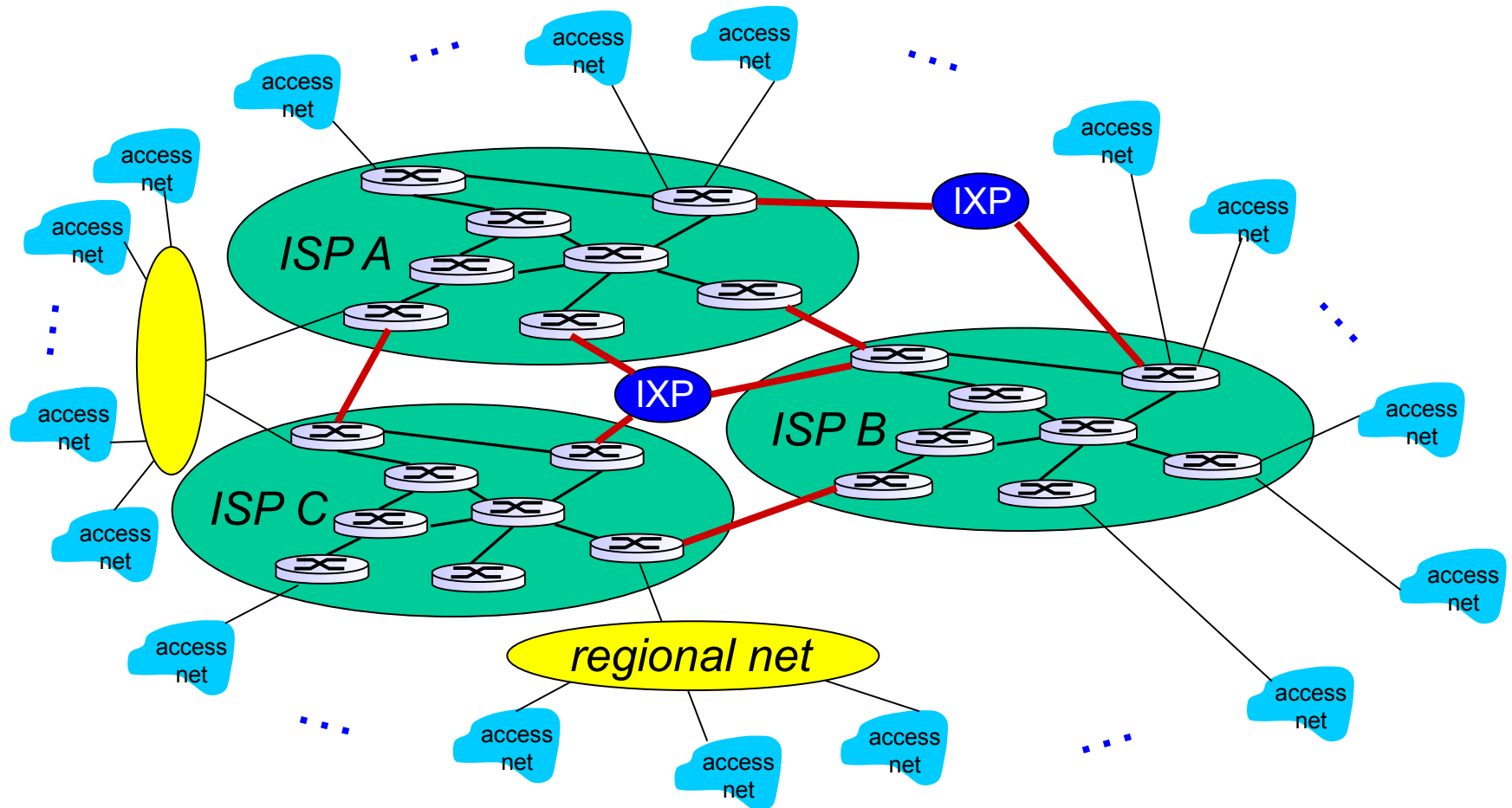
Internet structure: network of networks

… and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users

Internet structure: network of networks

- ❖ at center: small # of well-connected large networks
  - · "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
  - · content provider network (e.g, Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs

https://peering.google.com

Chapter 1: roadmap

How do loss and delay occur?

packets *queue* in router buffers

❖ packet arrival rate to link (temporarily) exceeds output link capacity

❖ packets queue, wait for turn

packet being transmitted (delay)

A

packets queueing (delay)

free (available) buffers: arriving packets dropped (loss) if no free buffers

Four sources of packet delay

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

d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of router

Four sources of packet delay
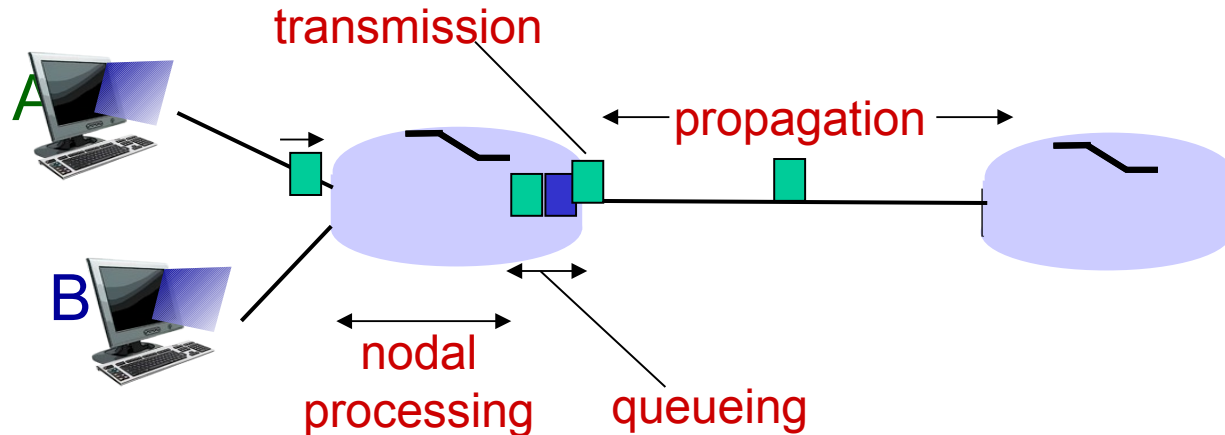
$$d\text{nodal} = d\text{proc} + d\text{queue} + d\text{trans} + d\text{prop}$$

*d*trans: transmission delay:
- *L*: packet length (bits)
- *R*: link *bandwidth (bps)*
- *d*trans = *L/R*

*d*prop: propagation delay:
- *d*: length of physical link
- *s*: propagation speed in medium (~2x108 m/sec)
- *d*prop = *d/s*

* Check out the Java applet for an interactive animation on trans vs. prop delay

Caravan analogy

ten-car caravan   toll booth   100 km   toll booth   100 km
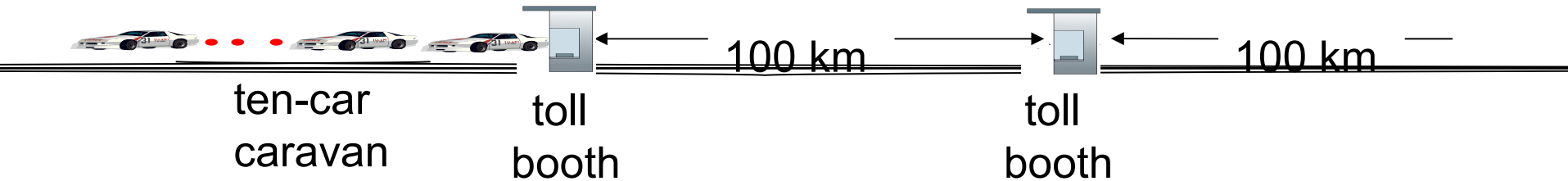
- ❖ cars "propagate" at 100 km/hr
- ❖ toll booth takes 12 sec to service car (bit transmission time)
- ❖ car~bit; caravan ~ packet
- ❖ *Q: How long until caravan is lined up before 2nd toll booth?*

- ▪ 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)

ten-car caravan   toll booth   100 km   toll booth   100 km
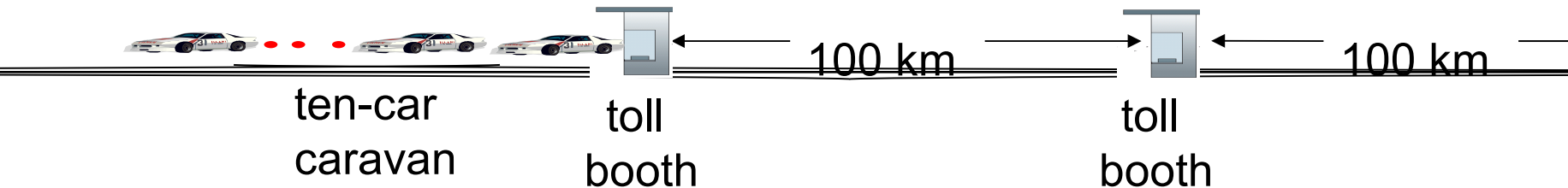
❖ suppose cars now "propagate" at 1000 km/hr
❖ and suppose toll booth now takes one min to service a car
❖ *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?

- *A:Yes!* after 7 min, 1st car arrives at second booth; three cars still at 1st booth.

- (100/1000) hours + 1 min = 7 min

Packet loss

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

buffer (waiting area)

packet being transmitted

A

B

packet arriving to full buffer is *lost*

* Check out the Java applet for an interactive animation on queuing and loss

Throughput

❖ *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

server sends bits
(fluid) into pipe

pipe that can carry
fluid at rate
R_S bits/sec)

pipe that can carry
fluid at rate
R_C bits/sec)

Throughput (more)

❖ *Rs < Rc* What is average end-end throughput?

RS bits/sec   RC bits/sec

❖ *Rs > Rc* What is average end-end throughput?

RS bits/sec   RC bits/sec

bottleneck link

link on end-end path that constrains end-end throughput

Chapter 1: roadmap

Protocol "layers"

Networks are complex, with 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?

Organization of air travel

ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing

airplane routing
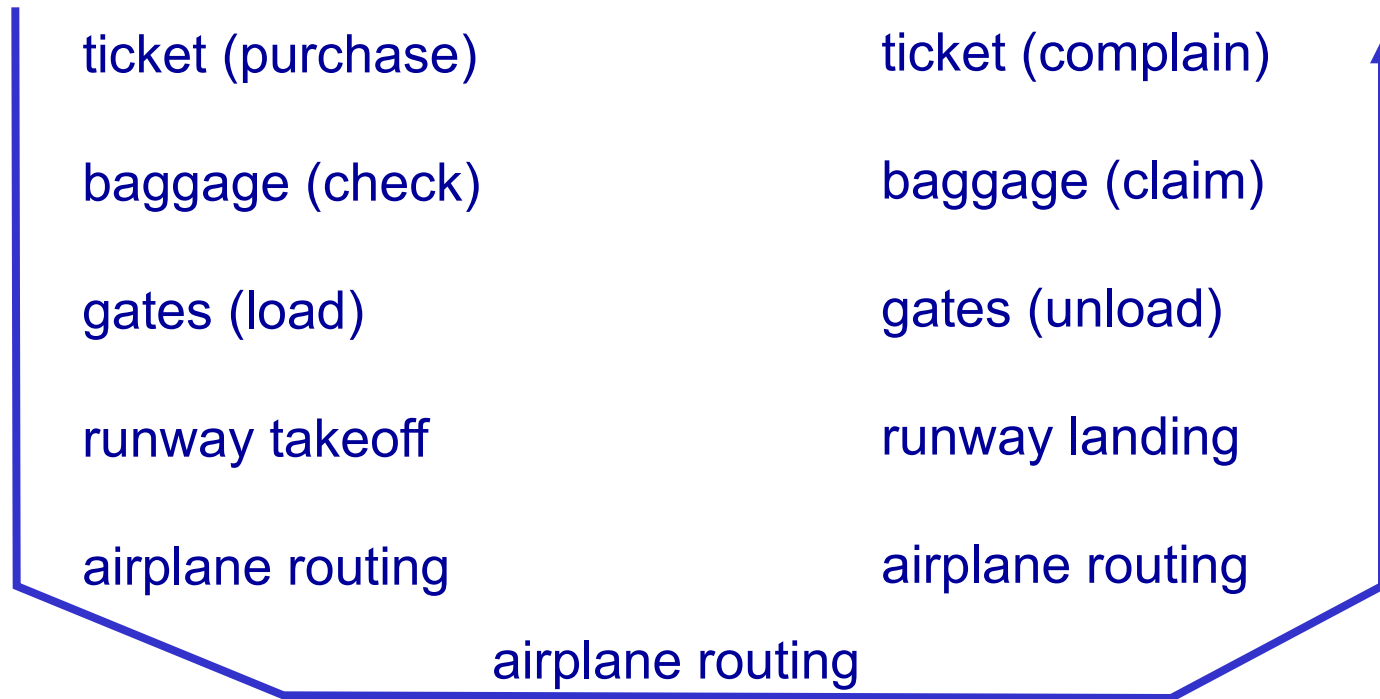
❖ a series of steps

Layering of airline functionality

ticket (purchase)			ticket (complain)
baggage (check)			baggage (claim
gates (load)			gates (unload)
runway (takeoff)			runway (land)
airplane routing	airplane routing	airplane routing	airplane routing

departure
airport

intermediate air-traffic
control centers

arrival
airport
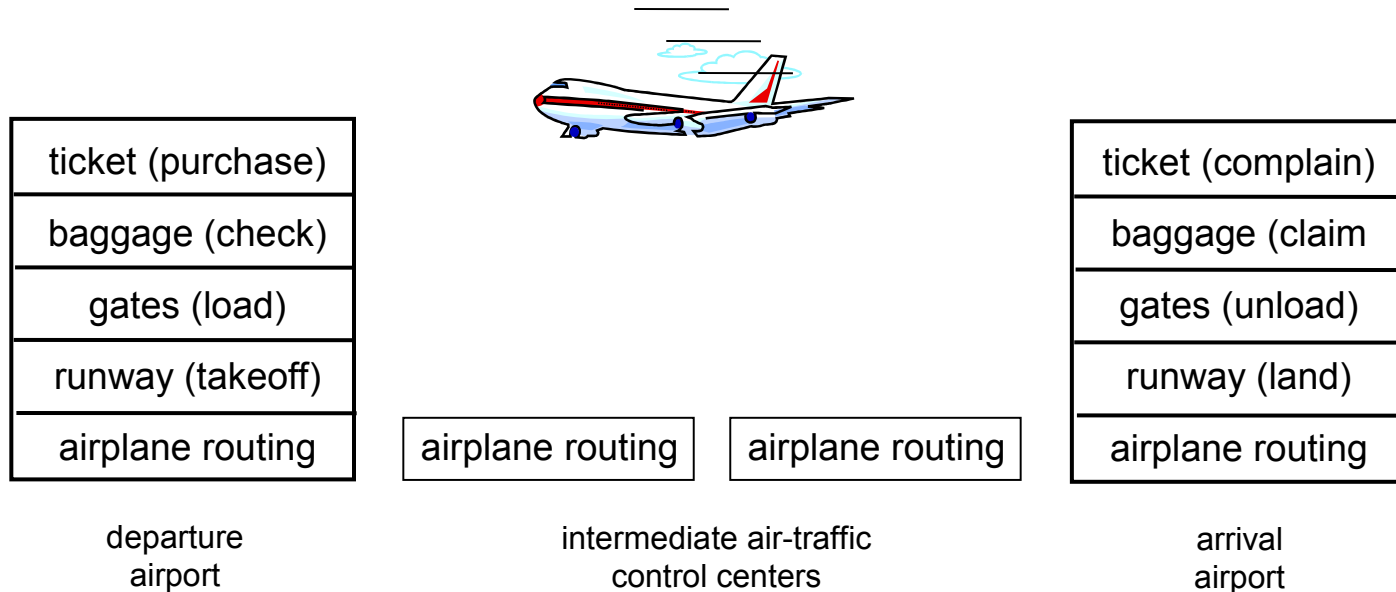
layers: each layer implements a service
- via its own internal-layer actions
- relying on services provided by layer below

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
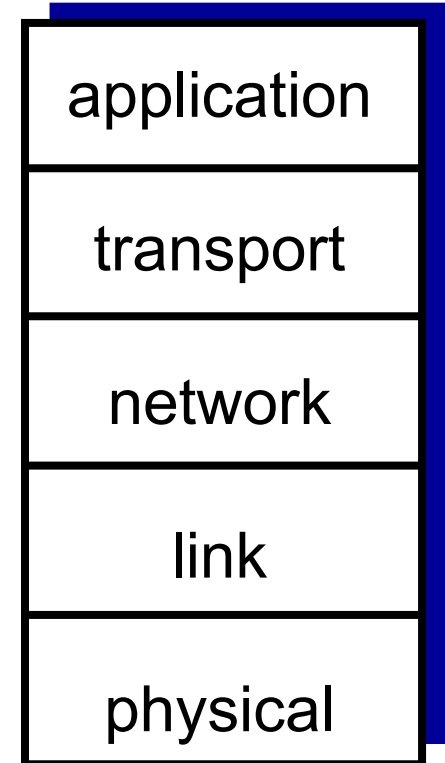 · FTP, SMTP, HTTP
❖ *transport:* process-process data transfer
  · TCP, UDP
❖ *network:* routing of datagrams from source to destination
  · IP, routing protocols
❖ *link:* data transfer between neighboring network elements
  · Ethernet, 802.111 (WiFi), PPP
❖ *physical:* bits "on the wire"

application
transport
network
link
physical

ISO/OSI reference model

❖ *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions

❖ *session:* synchronization, checkpointing, recovery of data exchange

❖ Internet stack "missing" these layers!

- these services, *if needed,* must be implemented in application
- needed?
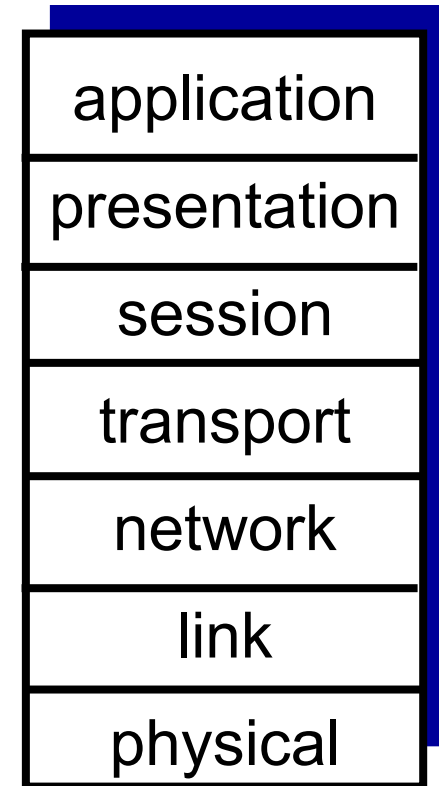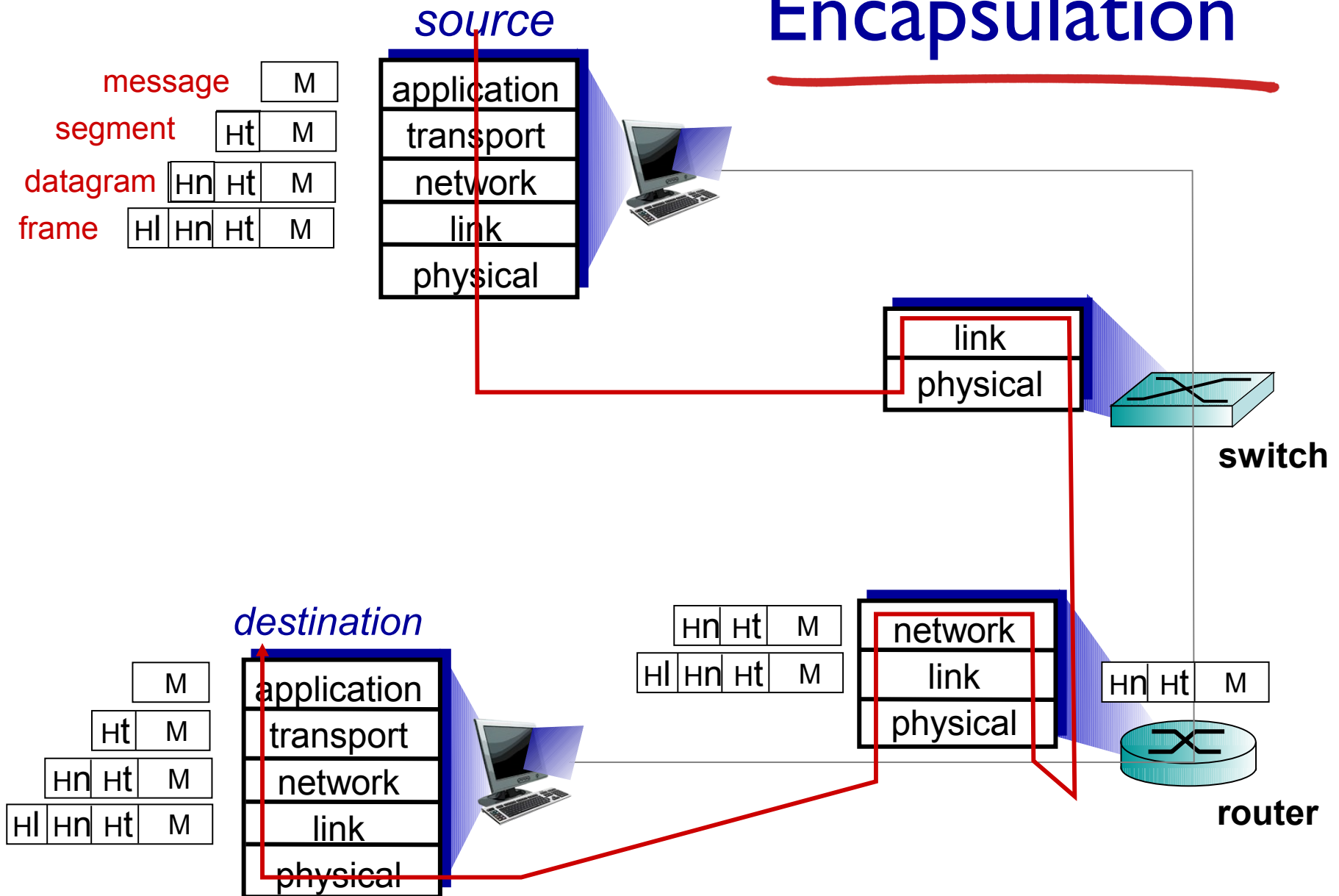
application
presentation
session
transport
network
link
physical

Encapsulation

message   M

segment   Ht | M

datagram   Hn | Ht | M

frame   Hl | Hn | Ht | M

source

| application |
| transport |
| network |
| link |
| physical |

| link |
| physical |

switch

destination

| application |
| transport |
| network |
| link |
| physical |

M

Ht | M

Hn | Ht | M

Hl | Hn | Ht | M

Hn | Ht | M

Hl | Hn | Ht | M

| network |
| link |
| physical |

Hn | Ht | M

router

Chapter 1: roadmap

Network security

❖ **field of network security:**

- how bad guys can attack computer networks
- how we can defend networks against attacks
- how to design architectures that are immune to attacks

❖ **Internet not originally designed with (much) security in mind**

- *original vision:* "a group of mutually trusting users attached to a transparent network" J
- Internet protocol designers playing "catch-up"
- security considerations in all layers!

Bad guys: put malware into hosts via Internet

❖ malware can get in host from:

  ▪ *virus:* self-replicating infection by receiving/executing object (e.g., e-mail attachment)

  ▪ *worm:* self-replicating infection by passively receiving object that gets itself executed

❖ spyware malware can record keystrokes, web sites visited, upload info to collection site

❖ infected host can be enrolled in botnet, used for spam. DDoS attacks

Bad guys: attack server, network infrastructure

Denial of Service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic
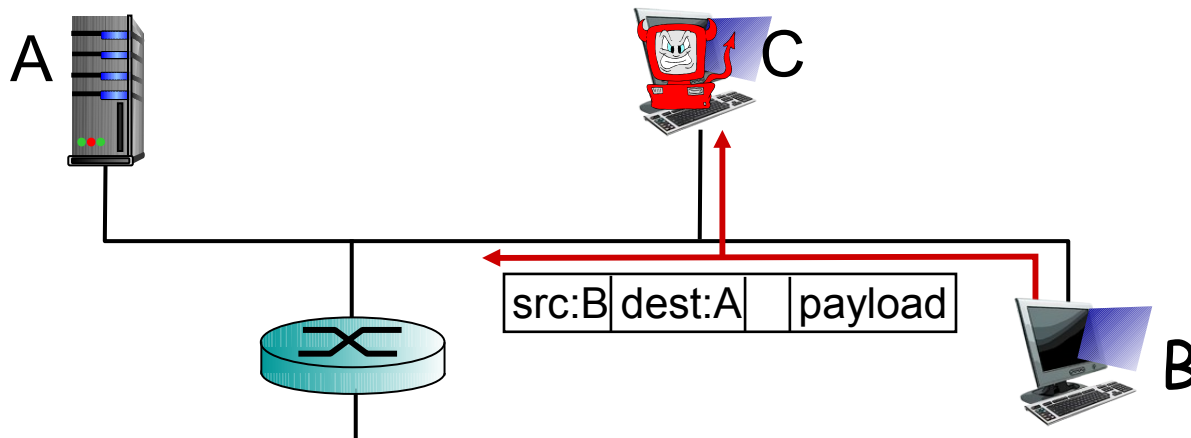
1. select target

2. break into hosts around the network (see botnet)

3. send packets to target from compromised hosts

target
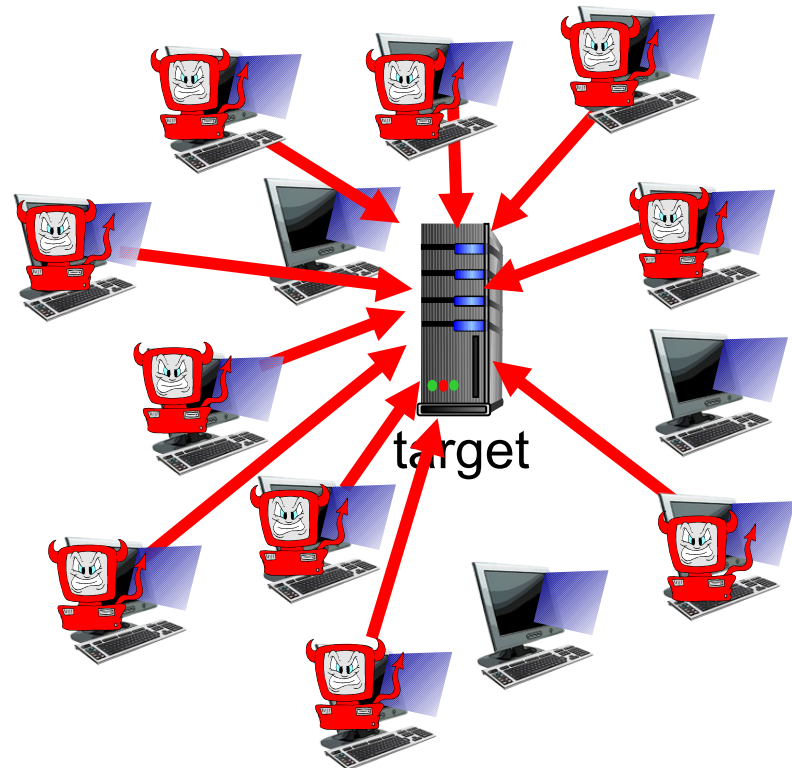
Bad guys can sniff packets

packet "sniffing":

- broadcast media (shared ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by

| src:B | dest:A | | payload |

❖ wireshark software used for end-of-chapter labs is a (free) packet-sniffer

Bad guys can use fake addresses

IP spoofing: send packet with false source address

A |src:B| dest:A | payload | C

B
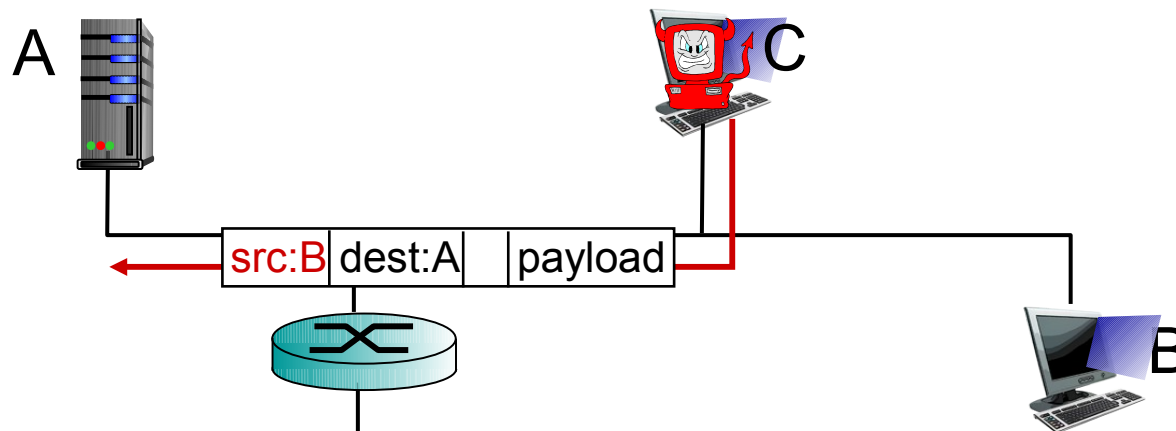
... lots more on security (throughout, Chapter 8)

Chapter 1: roadmap

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 demo
 · NCP (Network Control Protocol) first host-host protocol
 · first e-mail program
 · ARPAnet has 15 nodes
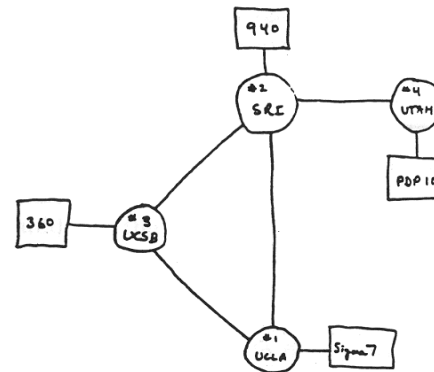
THE ARPA NETWORK

DEC 1969

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

Internet history

2005-present

❖ ~750 million hosts
  ▪ Smartphones and tablets
❖ Aggressive deployment of broadband access
❖ Increasing ubiquity of high-speed wireless access
❖ Emergence of online social networks:
  ▪ Facebook: soon one billion users
❖ Service providers (Google, Microsoft) create their own networks
  ▪ Bypass Internet, providing "instantaneous" access to search, emai, etc.
❖ E-commerce, universities, enterprises running their services in "cloud" (eg, Amazon EC2)

Introduction: summary

covered a "ton" of material!

❖ Internet overview

❖ what's a protocol?

❖ network edge, core, access network

  ▪ packet-switching versus circuit-switching

  ▪ Internet structure

❖ performance: loss, delay, throughput

❖ layering, service models

❖ security

❖ history

you now have:

❖ context, overview, "feel" of networking

❖ more depth, detail *to follow!*