

Computer Networks

Prof. Xiaoming Fu

Assistants: F. Tegeler and N. Neumann

Course Overview

- 28 Oct. 2010 Introduction & Layering
- 4 Nov. 2010 Link Layer I
- 11 Nov. 2010 Link Layer II
- 18 Nov. 2010 Network Layer I
- 25 Nov. 2010 Network Layer II; Routing I
- 2 Dec. 2010 Network Layer III; Routing II; Mobility
- 9 Dec. 2010 Transport Layer I
- 16 Dec. 2010 Transport Layer II
- 6 Jan. 2011 Networked Multimedia
- 13 Jan. 2011 Quality of Service
- 20 Jan. 2011 Network Security I
- 27 Jan. 2011 Network Security II
- 3 Feb. 2011 Questions & Answers Session
- 10 Feb. 2011 Written Examination

Excercises

- Contact e-mail:

tegeler@cs.uni-goettingen.de

- Homework exercises will be handed out regularly after class and are in the wiki.
- Students are encouraged to work on their own and solve the homework exercises to prepare for the final exam.
- Solutions will be presented one week later after class. Thursdays 12:00 – 13:00 in the lecture room.

Grading

- The grading is as follows:

100% Final exam!

- All important information (click on Computer Networks)

wiki.net.informatik.uni-goettingen.de

Chapter 1

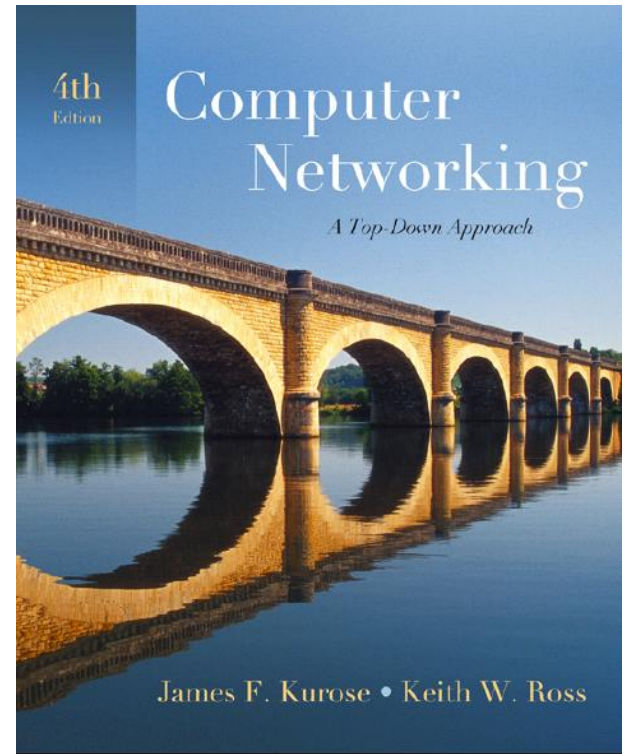
Introduction

This lecture is based on the book:

Computer Networking: A Top Down Approach
4th edition. Jim Kurose, Keith Ross, Addison-Wesley,
July 2007.

Alternative textbook:

- A. Tanenbaum, "Computer Networks", 4th edition, Prentice Hall, 2002
- D. Comer, "Computer Networks and Internets", 4th edition, Prentice Hall, 2002



Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models

1.6 History

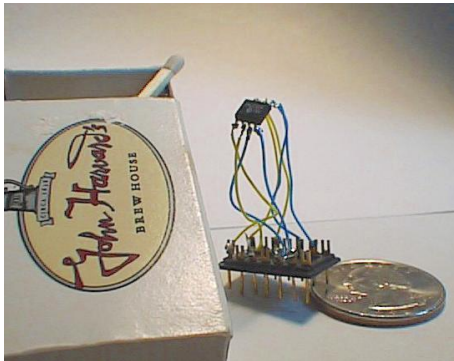
“Cool” internet appliances



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



Web-enabled toaster +
weather forecaster



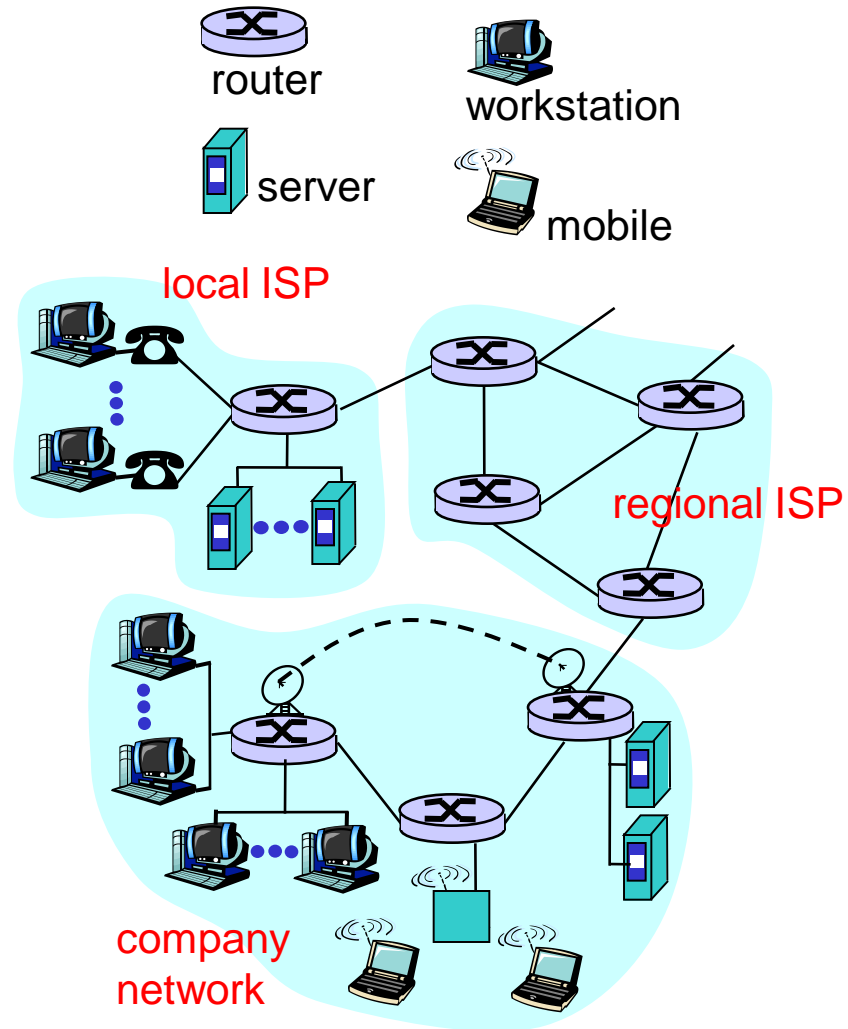
World's smallest web server
<http://www-ccs.cs.umass.edu/~shri/iPic.html>



Internet phones

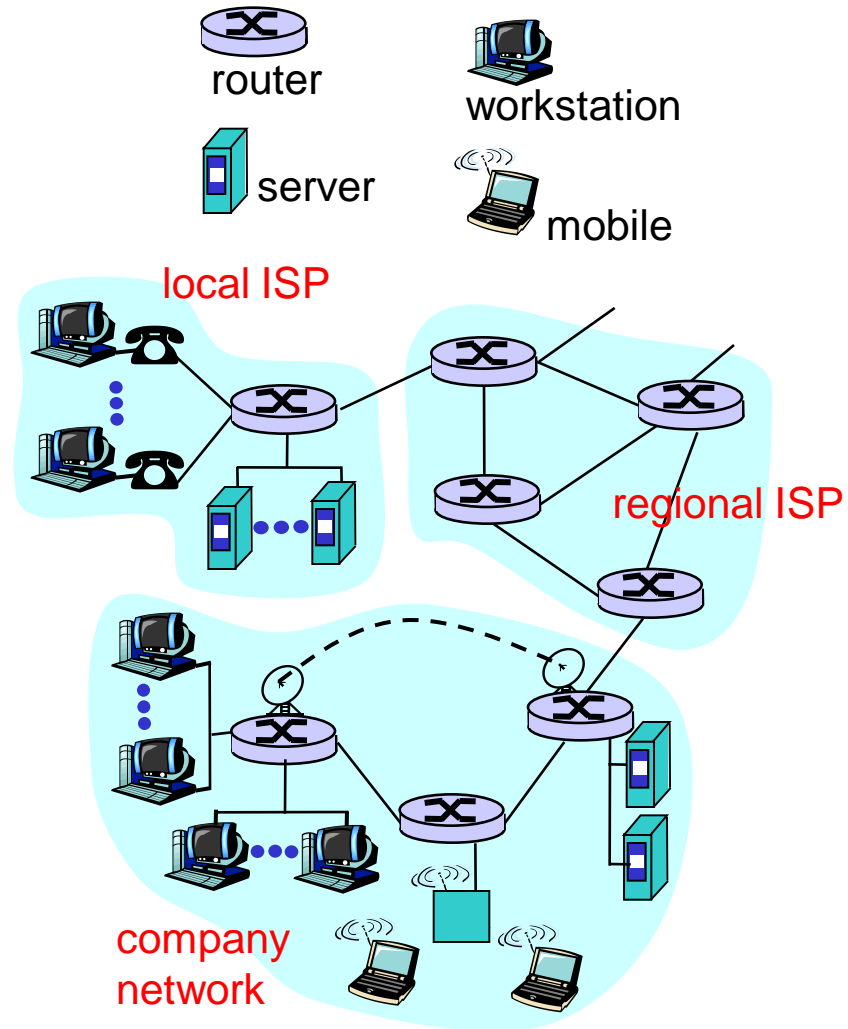
What's the internet? A close look...

- millions of connected computing devices: *hosts, end-systems*
 - PCs, workstations, servers
 - PDAs, phones, toasters
 - running *network apps*
- *communication links*
 - fiber, copper, coax, radio, satellite
 - transmission rate = ***bandwidth***
- *routers*: forward packets (chunks of data)



What's the internet? ... and closer

- *protocols* define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt
- *Internet*: “network of networks”
 - loosely hierarchical
 - public Internet versus private intranet
- *Internet standards*
 - RFC: Request for Comments
 - IETF: Internet Engineering Task Force



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

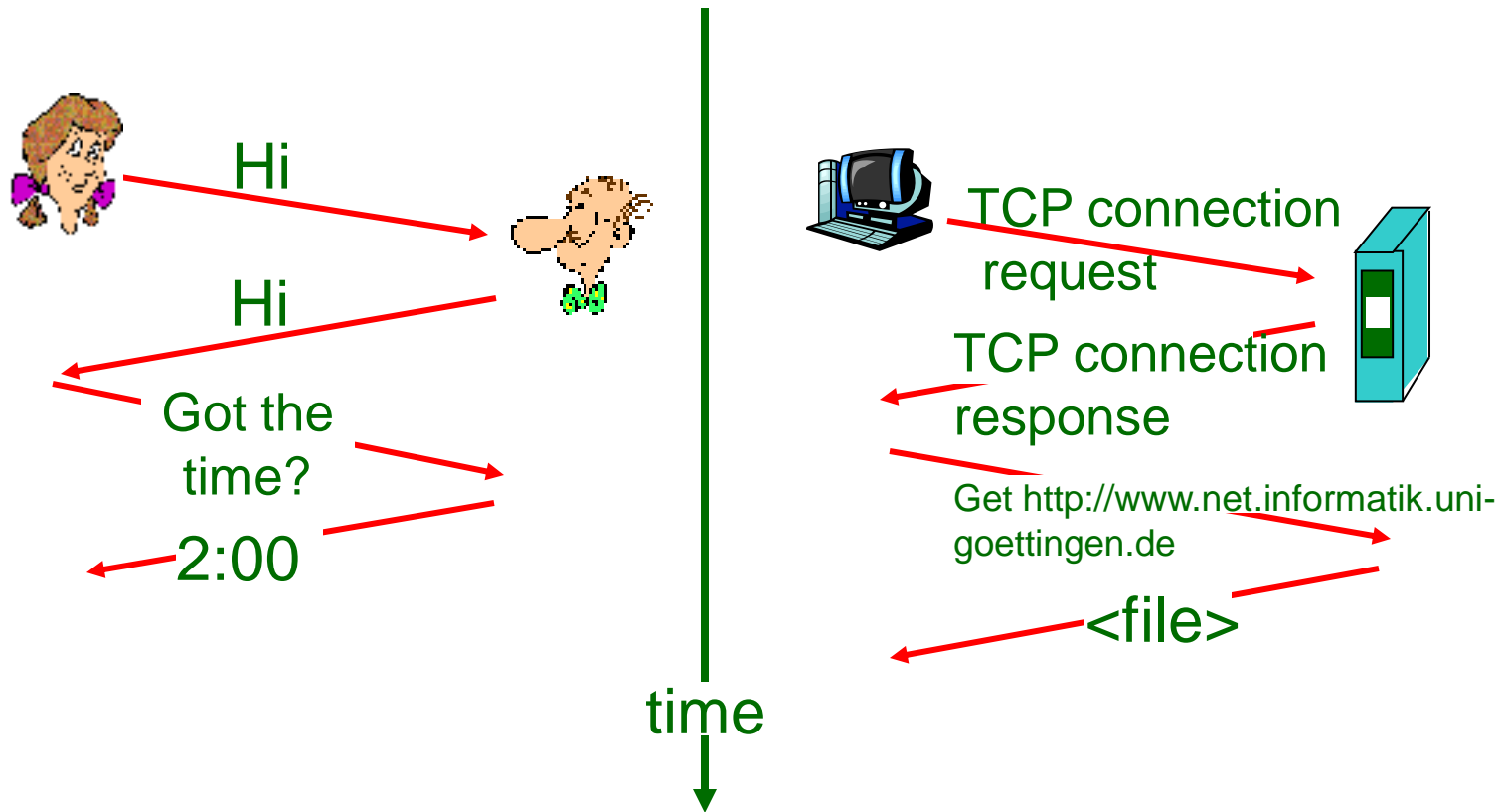
network protocols:

- 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:



Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

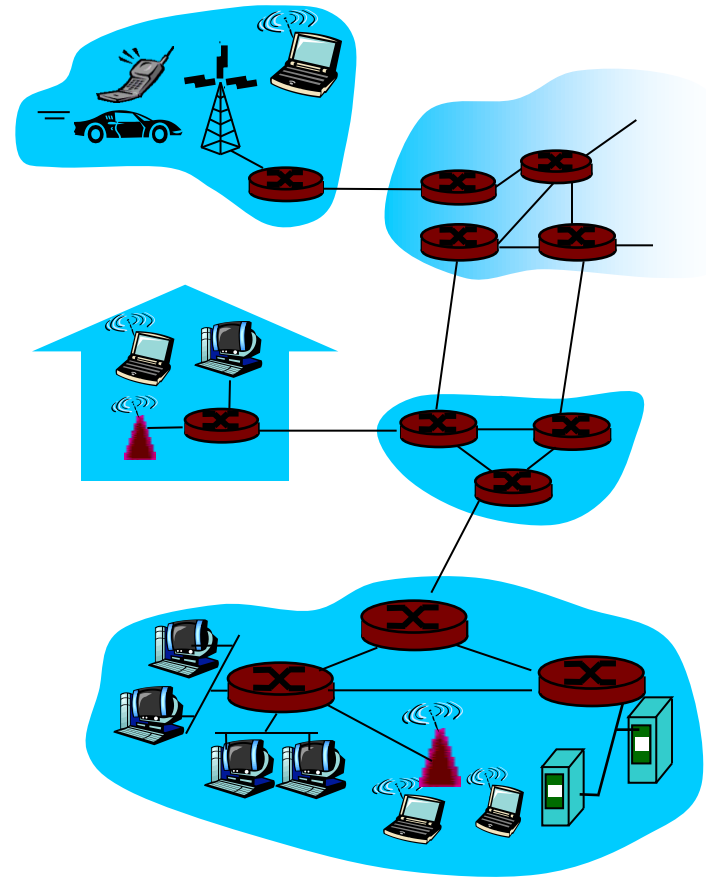
1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models

1.6 History

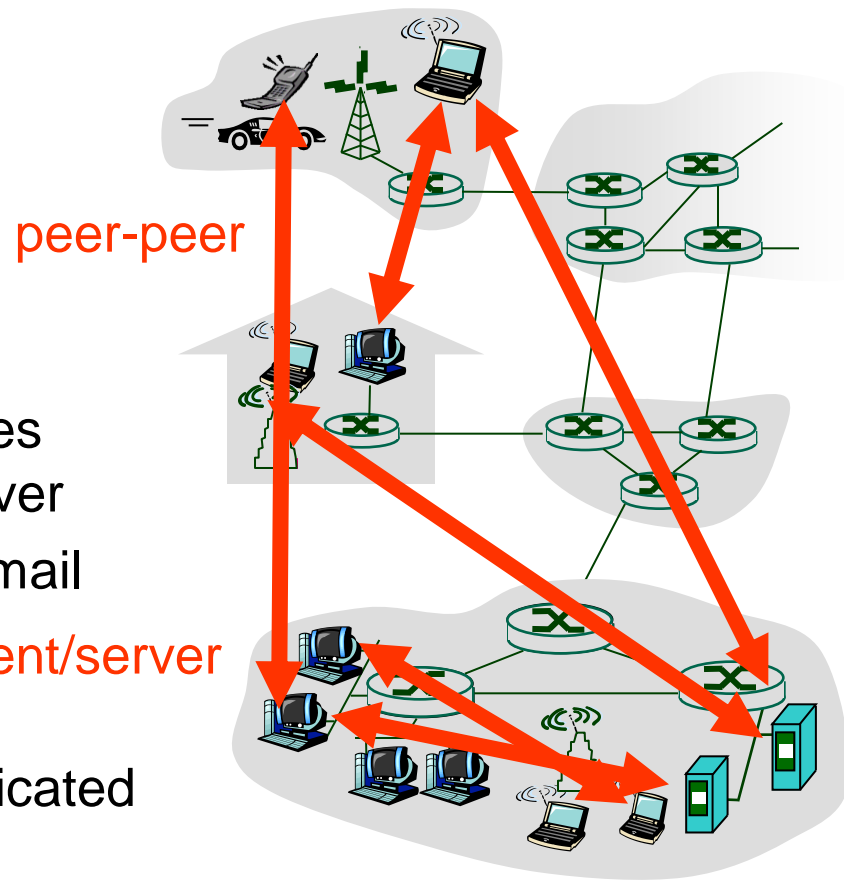
A closer look at network structure:

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



The network edge:

- **end systems (hosts):**
 - run application programs
 - e.g. web, email
 - at “edge of network”
- **client/server model**
 - client host requests, receives service from always-on server
 - e.g. web browser/server; email client/server
- **peer-peer model:**
 - minimal (or no) use of dedicated servers
 - e.g. Skype, BitTorrent



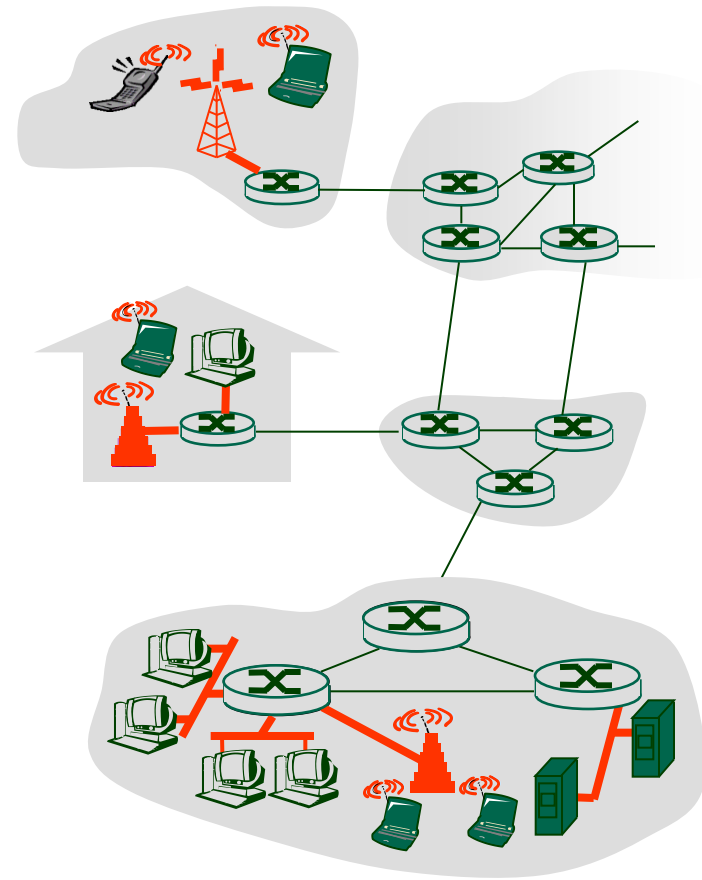
Access networks and physical media

Q: How to connect end systems to edge router?

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

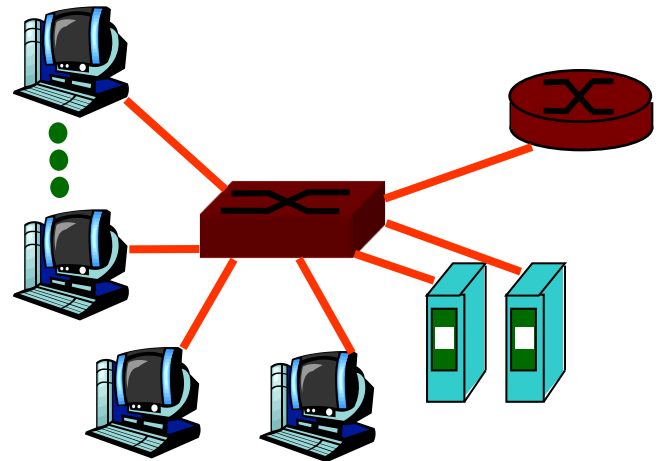
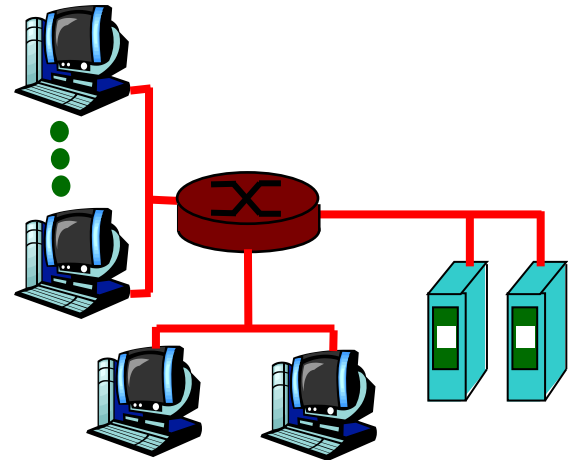
Keep in mind:

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



Example: Company access: local area networks

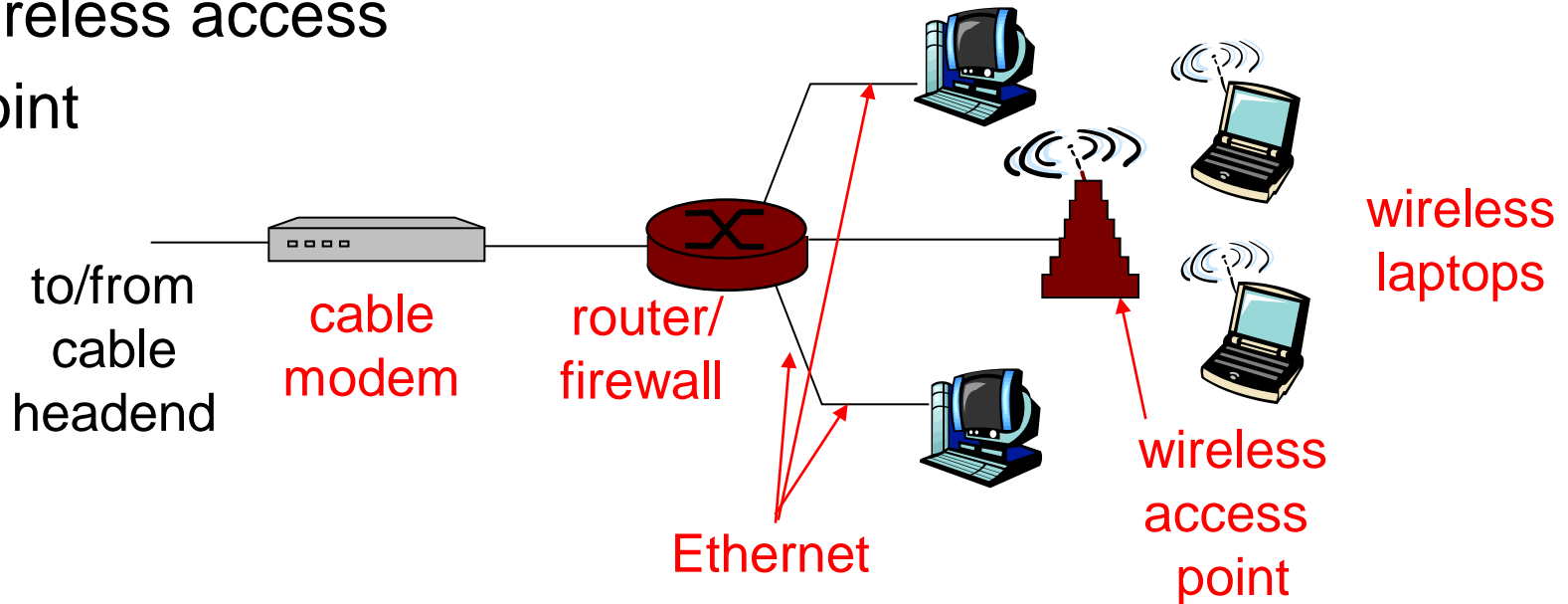
- company/univ **local area network** (LAN) connects end system to edge router (example: our GöNet)
- **Ethernet:**
 - 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
 - modern configuration: end systems connect into *Ethernet switch*
- LANs: will be discussed in detail throughout this lecture



Example: Home networks

Typical home network components:

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



Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

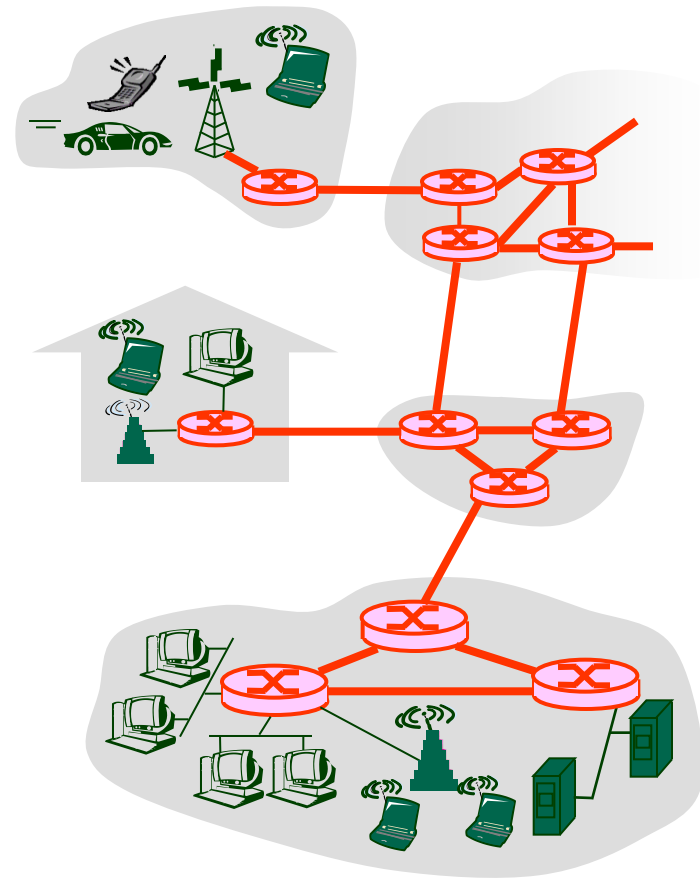
1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models

1.6 History

The Network Core

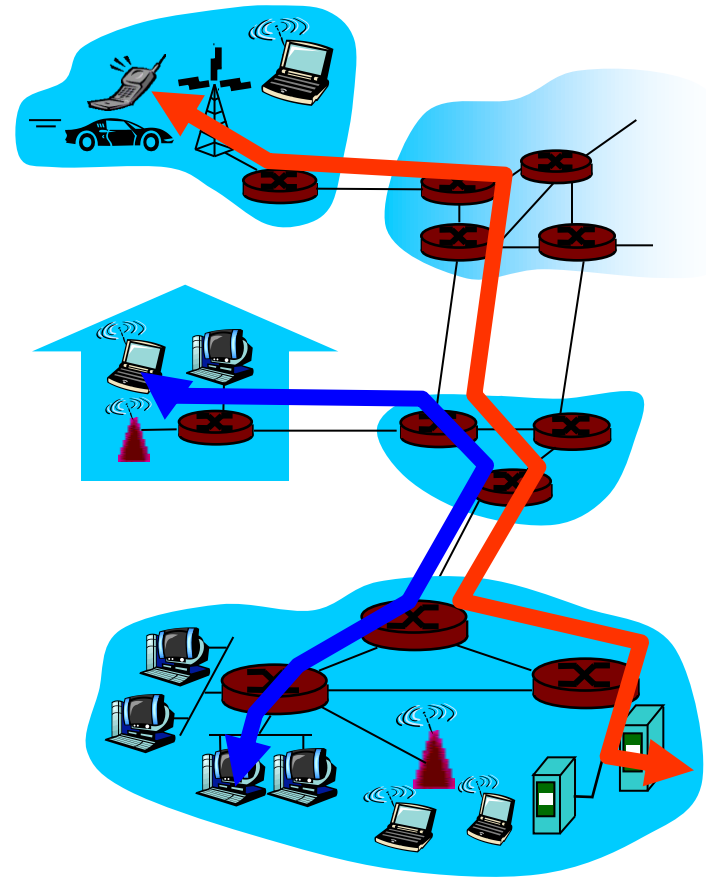
- mesh of interconnected routers
- ***the fundamental question:*** how is data transferred through net?
 - **circuit switching:** dedicated circuit per call: telephone network
 - **packet-switching:** data sent through a network in discrete “chunks”



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



Network Core: Circuit Switching

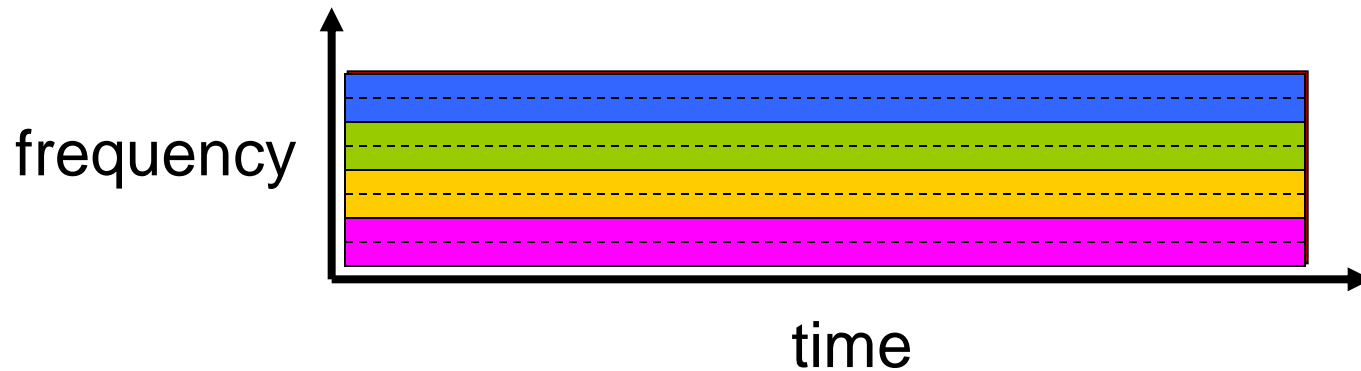
- network resources (e.g., bandwidth) **divided into “pieces”**
- pieces allocated to calls
 - resource piece **idle** if not used by owning call (*no sharing*)
 - dividing link bandwidth into “pieces”
 - frequency division
 - time division

Circuit Switching: FDM and TDM

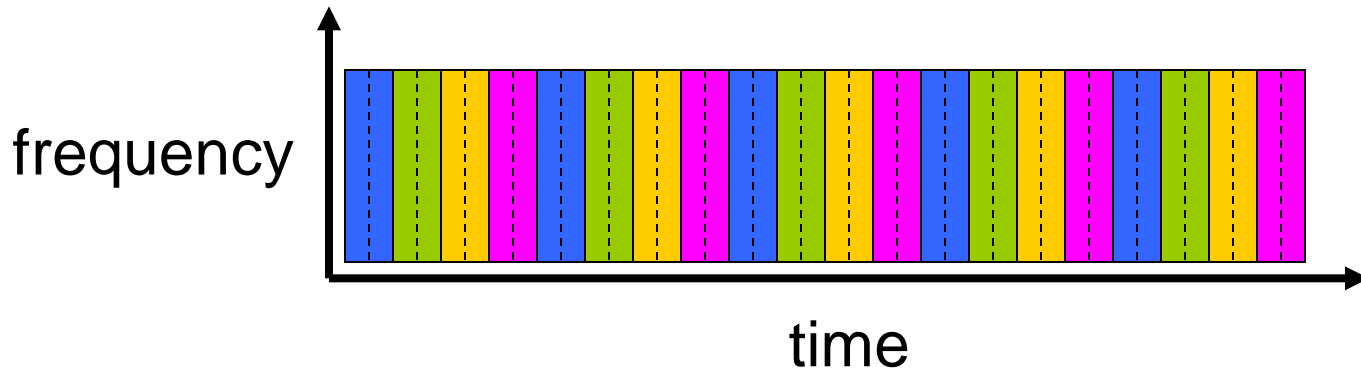
FDM

Example:

4 users



TDM



Network Core: Packet Switching

each end-end data stream
divided into *packets*

- user A, B packets *share* network resources
 - Sequence of sending packets does not have fixed pattern → **statistical multiplexing**
- each packet uses full link bandwidth
- resources used *as needed*

resource contention:

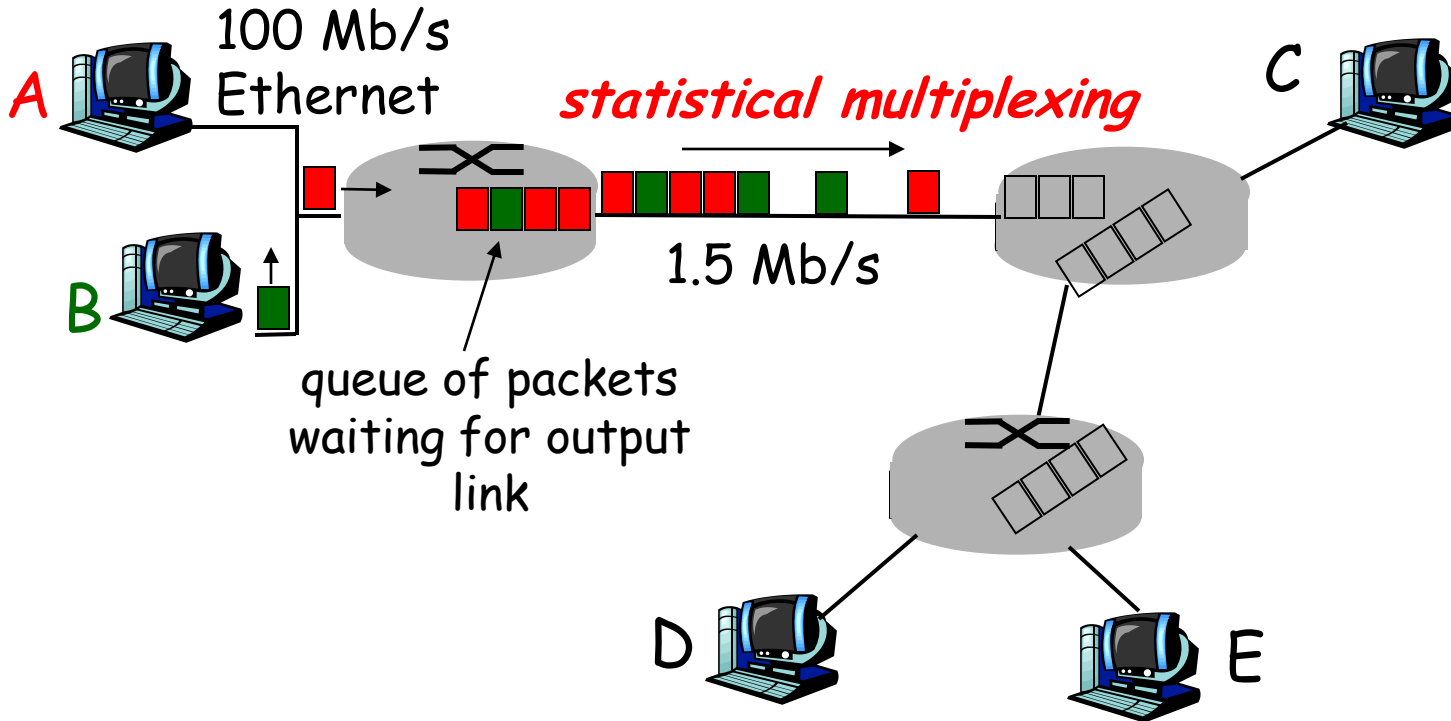
- aggregate resource demand can exceed amount available
- congestion: packets queue, wait for link use
- store and forward: packets move one hop at a time
 - Node receives complete packet before forwarding

Bandwidth division into “pieces”

Dedicated allocation

Resource reservation

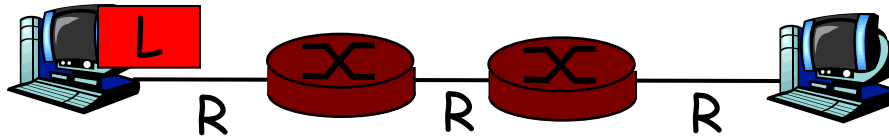
Packet Switching: Statistical Multiplexing



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

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

Packet-switching: store-and-forward



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

Example:

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

Note:

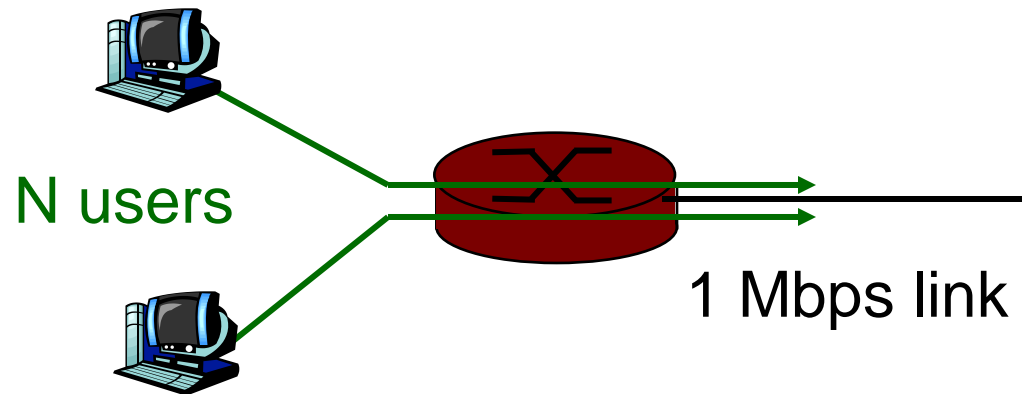
- In order to be more efficient, large packets are usually segmented into smaller packets

→ *Can you explain 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 at same time is low



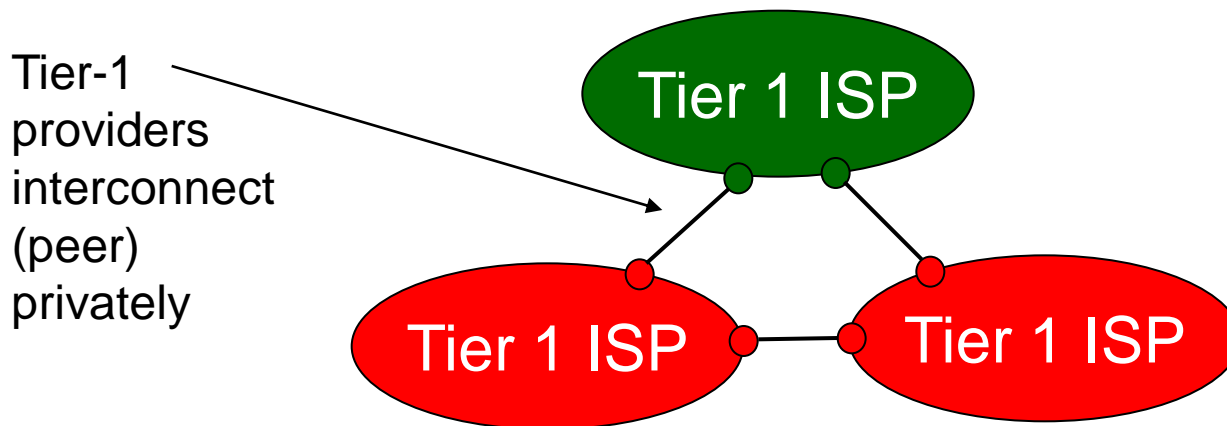
Packet switching versus circuit switching

Is packet switching better than circuit switching?

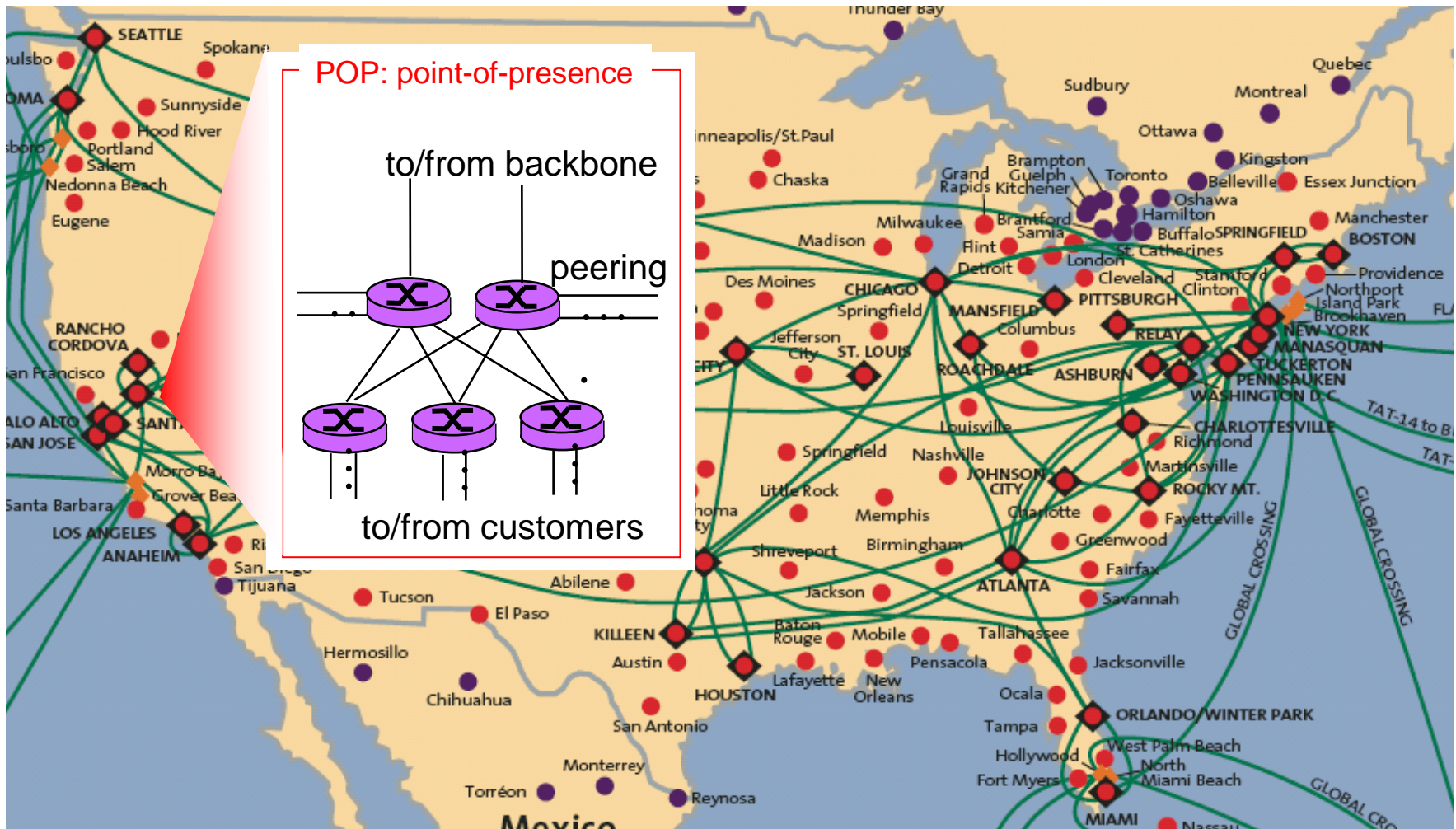
- great for bursty data
 - resource sharing
 - simpler, no call setup
- **excessive congestion**: packet delay and loss
 - protocols needed for reliable data transfer, congestion control
- **Q: How to provide circuit-like behavior?**
 - bandwidth guarantees needed for audio/video apps
 - still an unsolved problem

Internet structure: network of networks

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



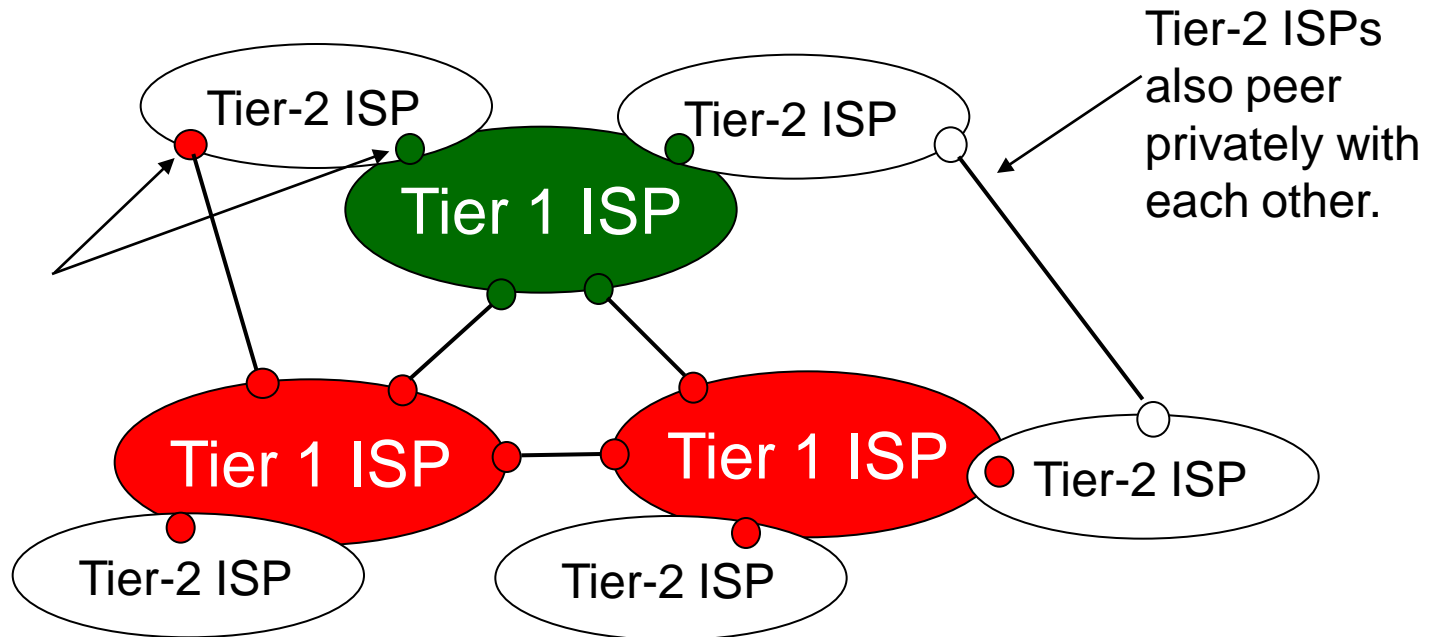
Tier-1 ISP: e.g., Sprint



Internet structure: network of networks

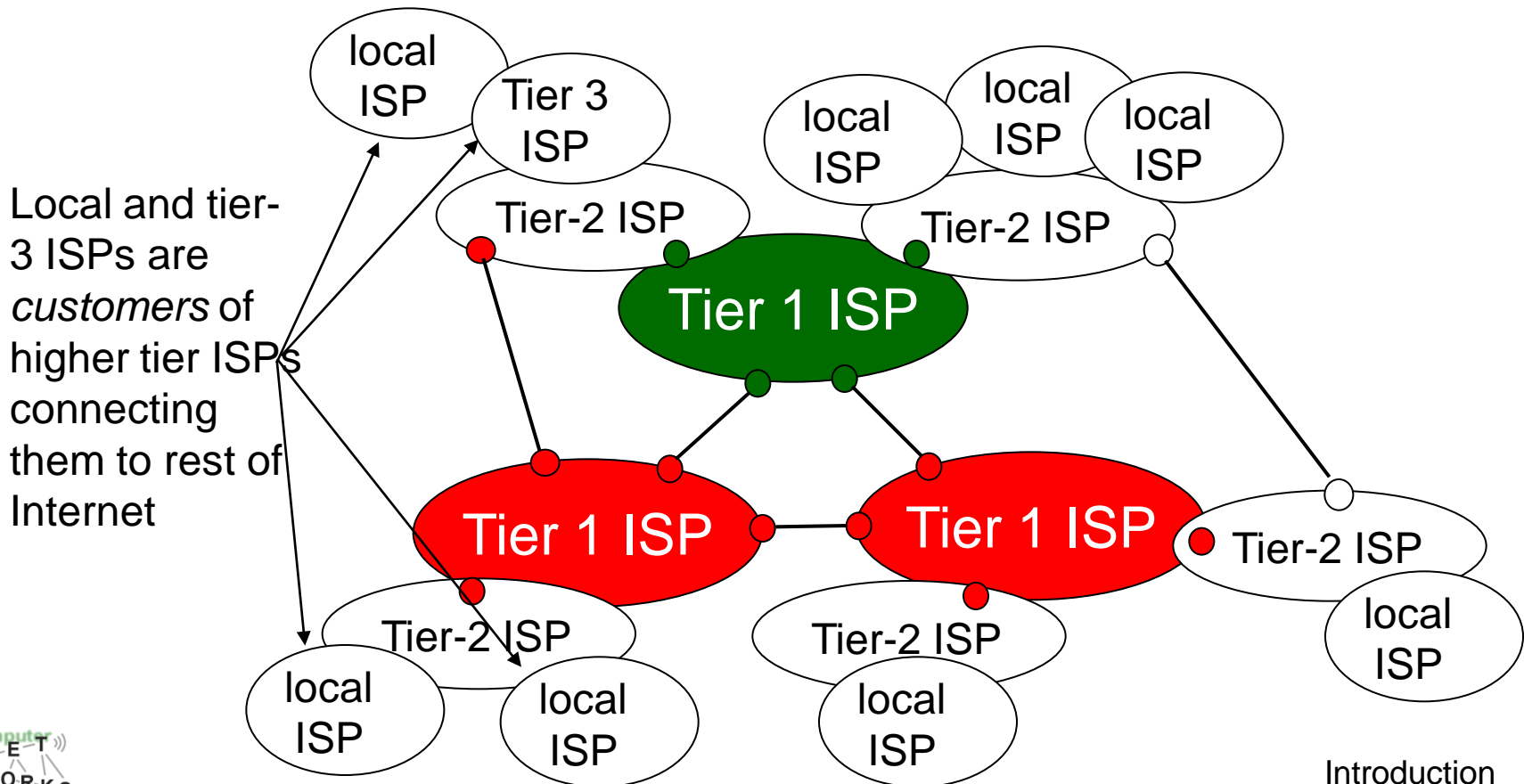
- “Tier-2” ISPs: smaller (often regional) ISPs
 - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet
□ tier-2 ISP is customer of tier-1 provider



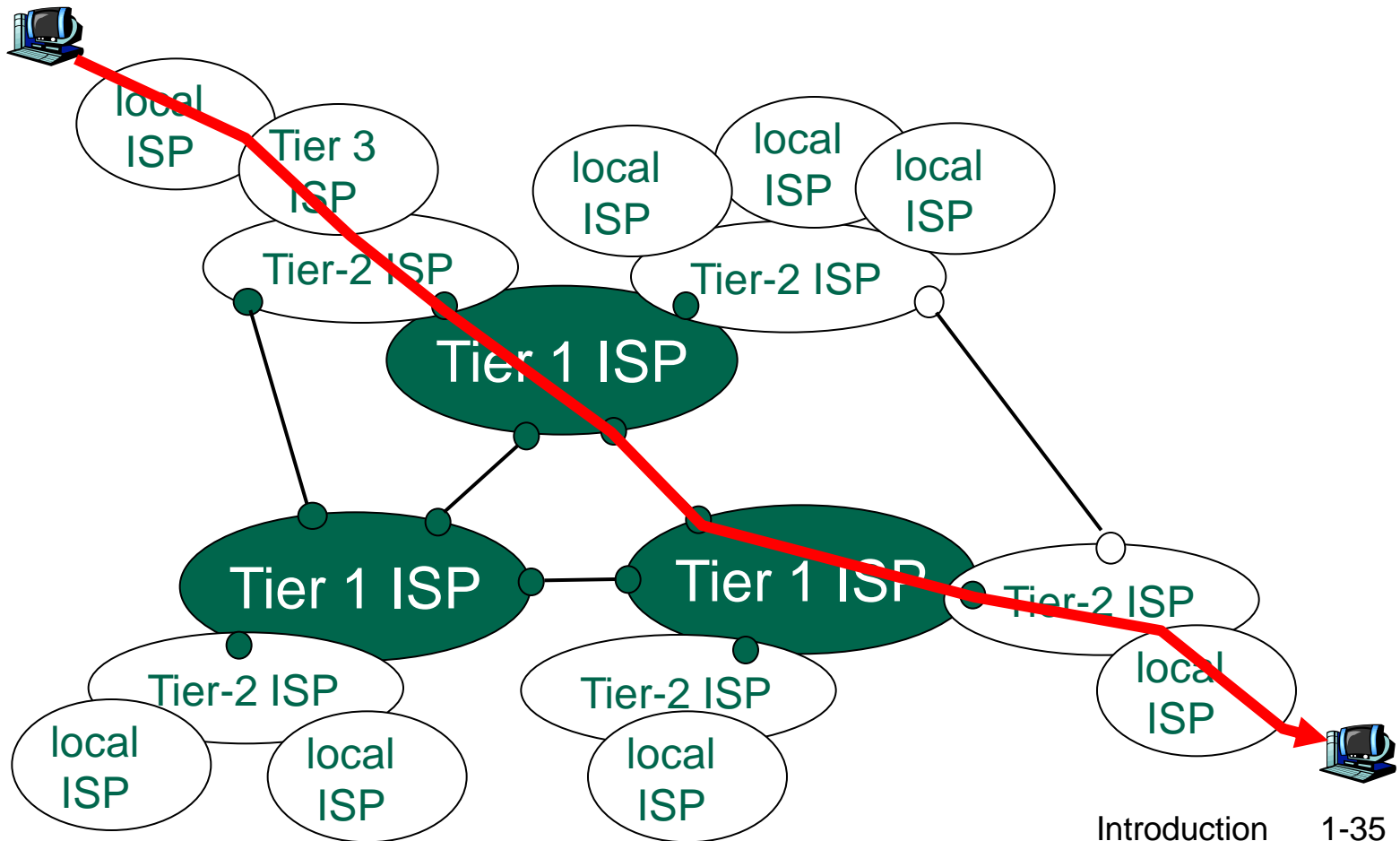
Internet structure: network of networks

- “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!



Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

1.4 Delay, loss and throughput in packet-switched networks

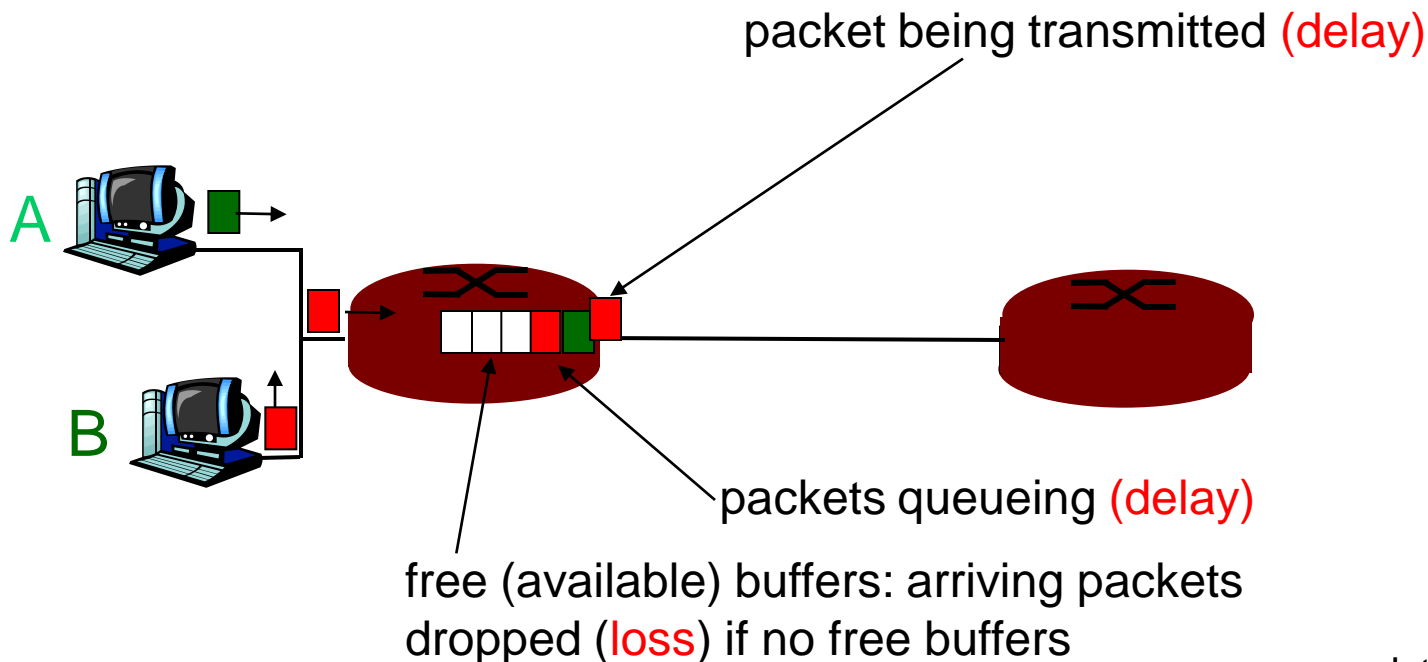
1.5 Protocol layers, service models

1.6 History

How do loss and delay occur?

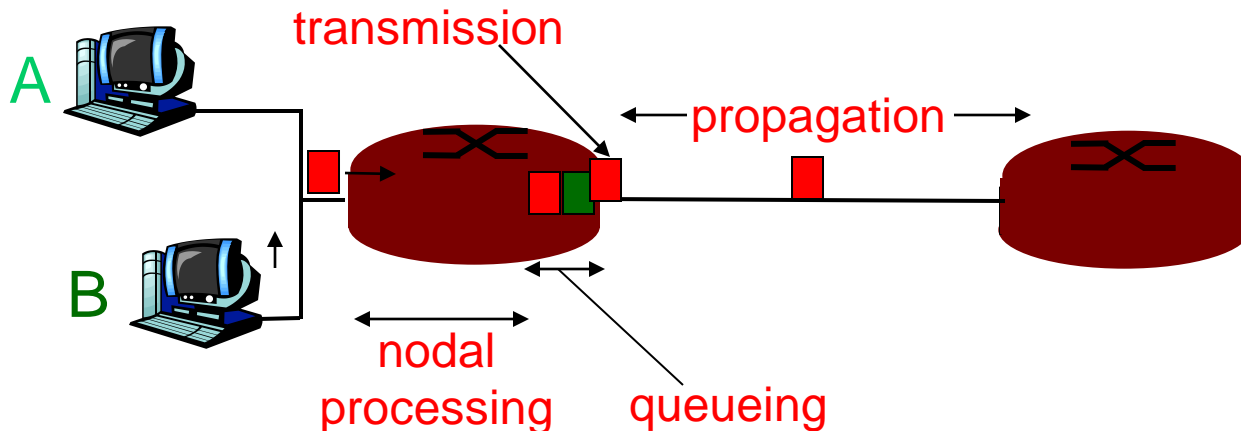
packets *queue* in router buffers

- 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



Delay in packet-switched networks

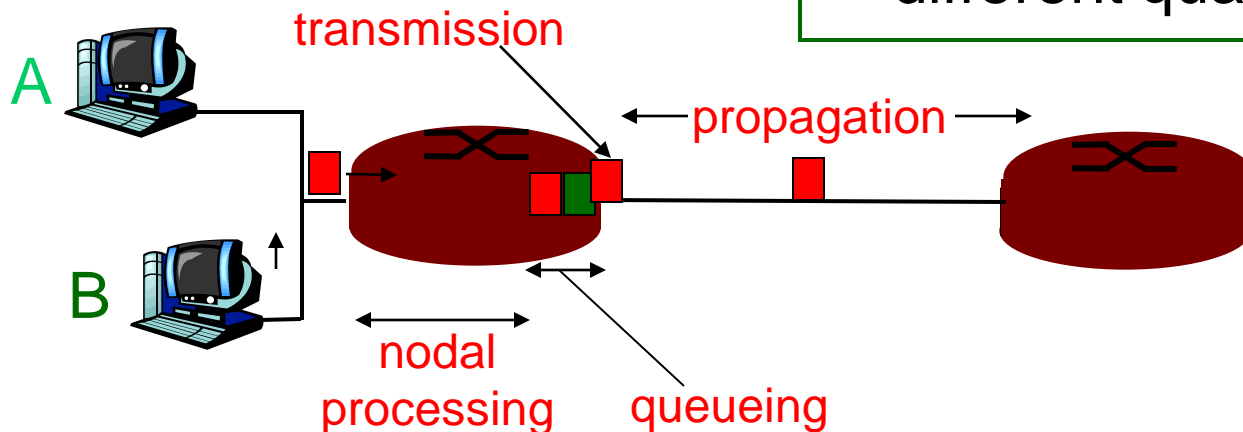
3. Transmission delay:

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

4. Propagation delay:

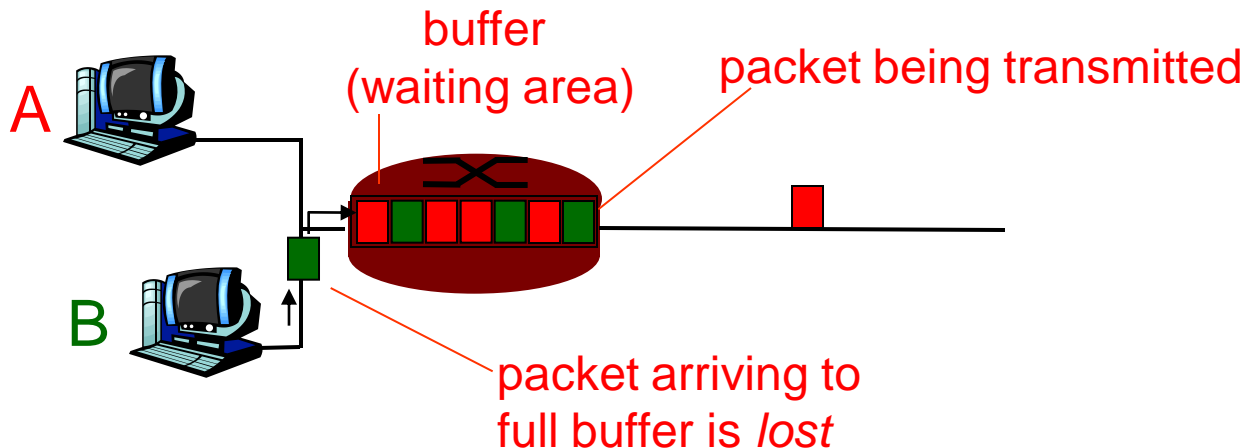
- d = length of physical link
- s = propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
- propagation delay = d/s

Note: s and R are very different quantities!



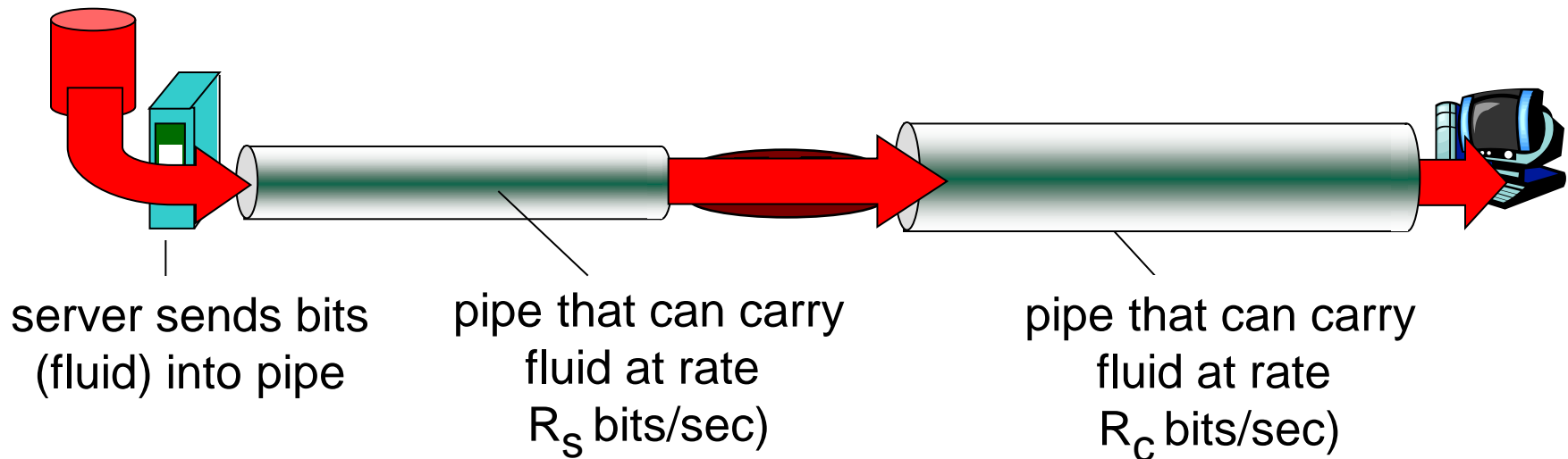
Packet loss

- 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



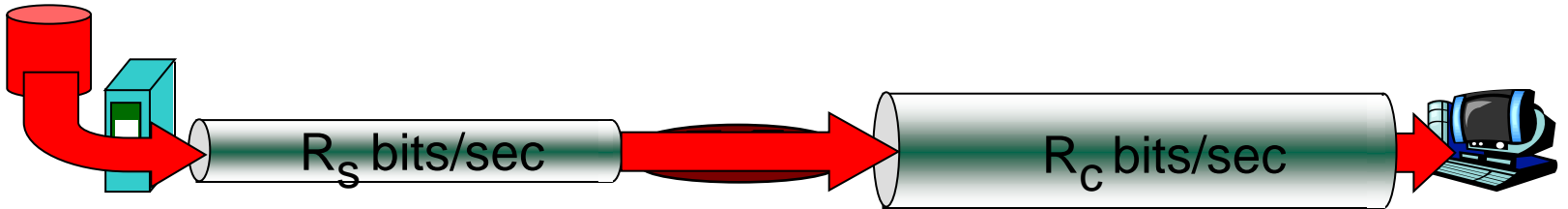
Throughput

- *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

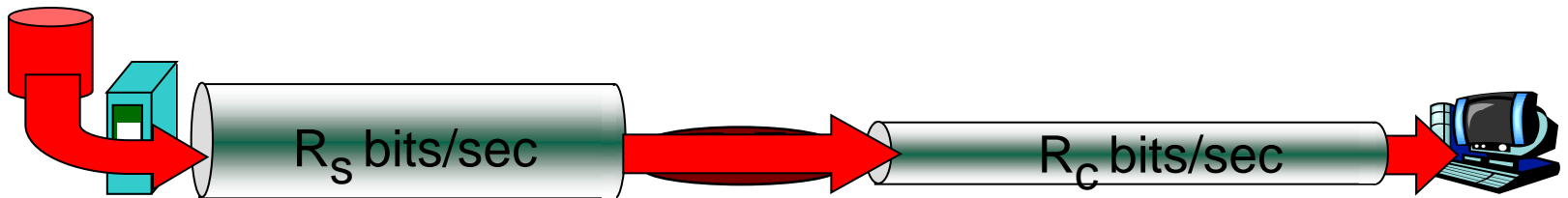


Throughput (more)

- $R_s < R_c$ What is average end-end throughput?



- $R_s > R_c$ What is average end-end throughput?

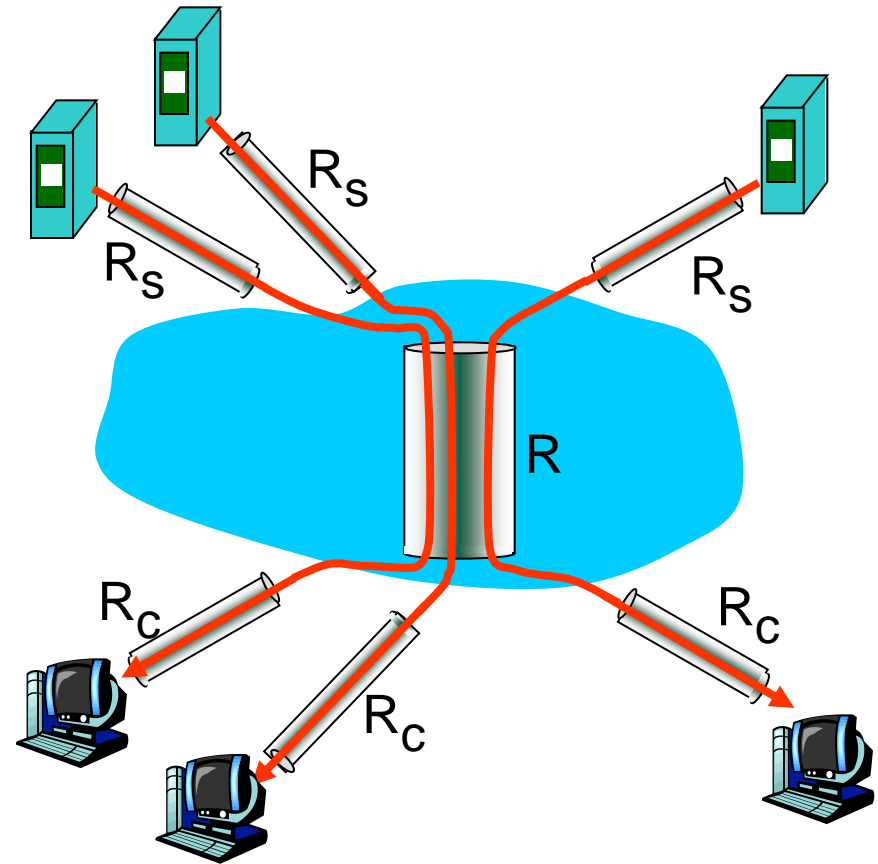


bottleneck link

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

Throughput: Internet scenario

- per-connection end-end throughput:
 $\min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often bottleneck



10 connections (fairly) share
backbone bottleneck link R bits/sec

Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models

1.6 History

Protocol “Layers”

Networks are complex!

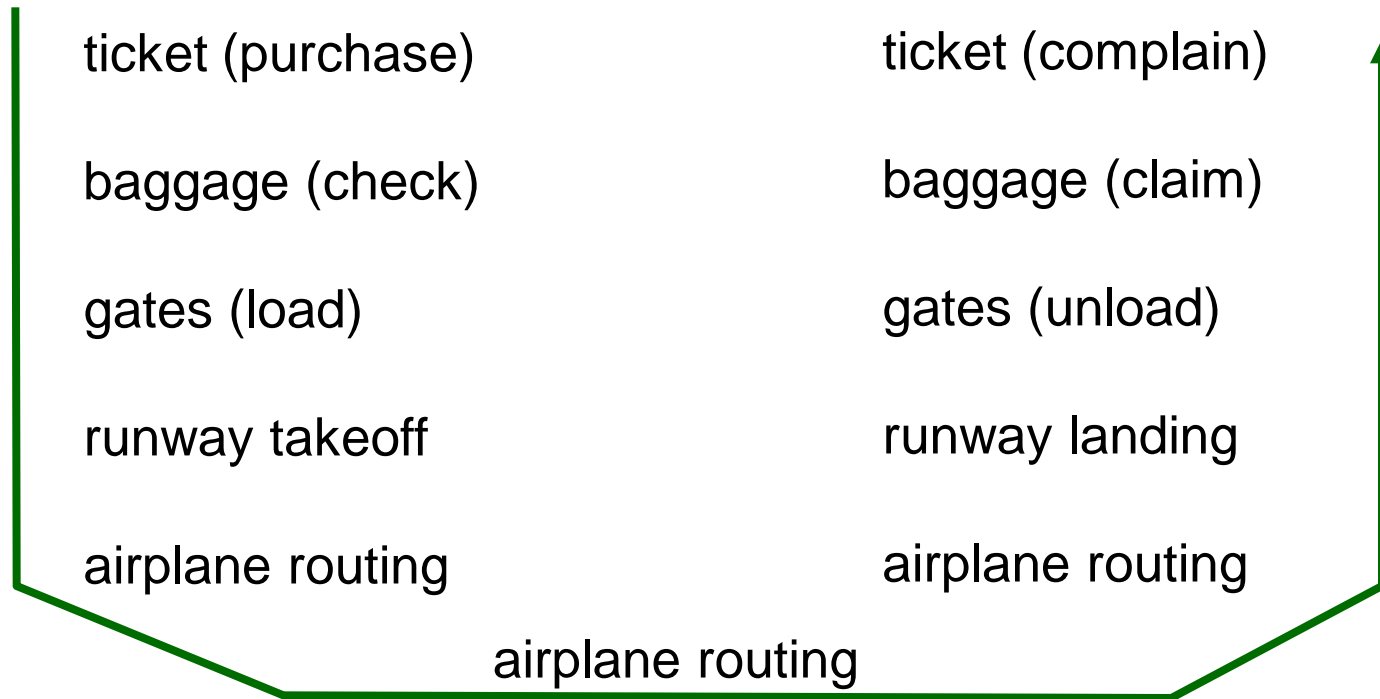
- many “pieces”:
 - hosts
 - routers
 - links of various media
 - applications
 - protocols
 - hardware, software

Question:

Is there any hope of
organizing structure of
network?

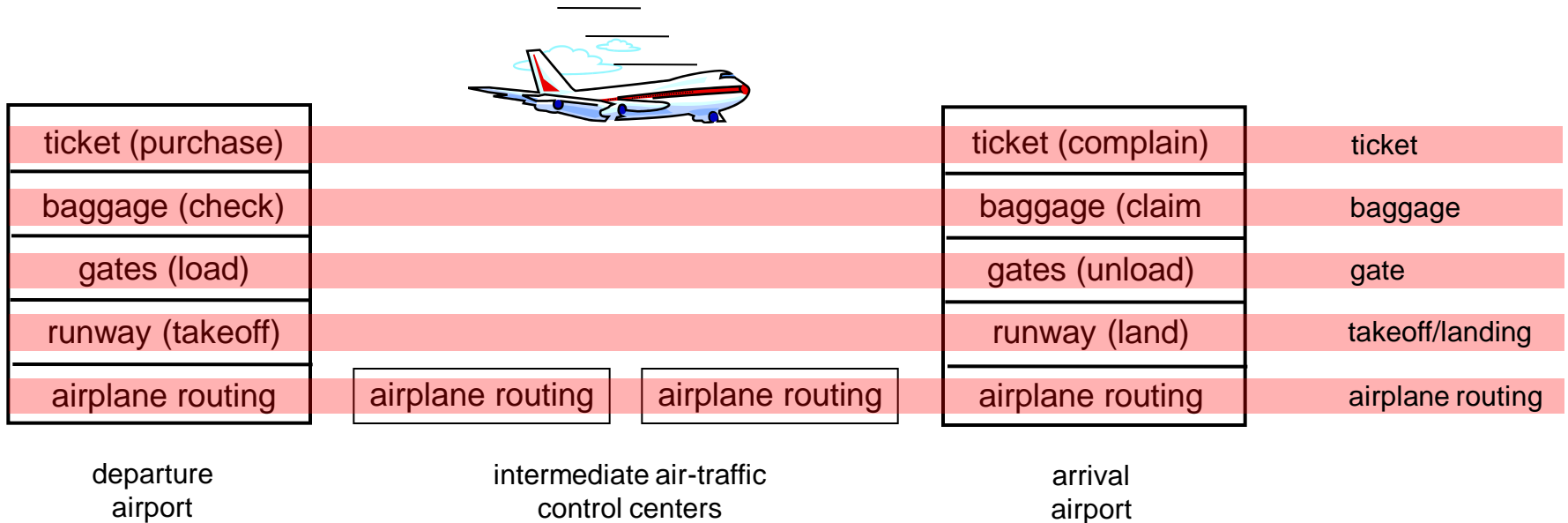
Or at least our discussion
of networks?

Organization of air travel



- a series of steps

Layering of airline functionality



Layers: each layer implements a service

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

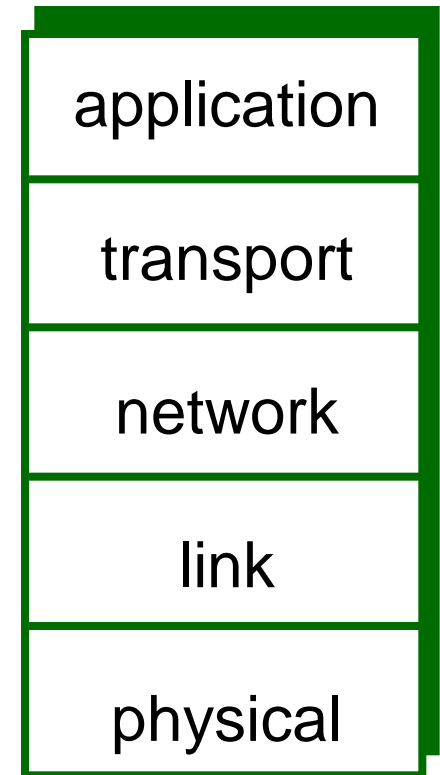
Why layering?

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?

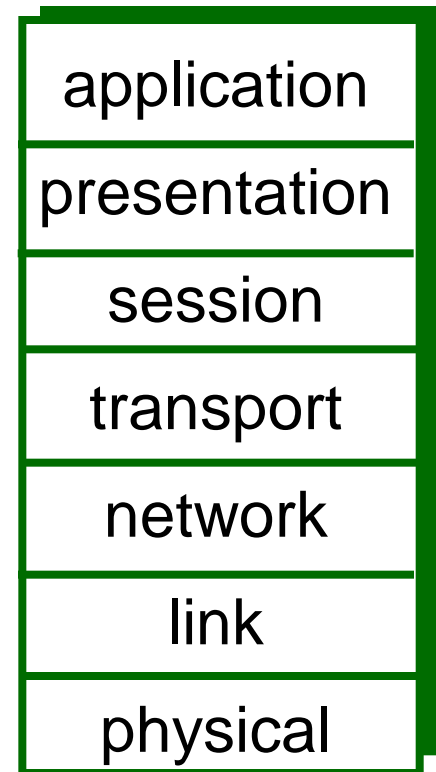
Internet protocol stack

- **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
 - PPP, Ethernet
- **physical:** bits “on the wire”

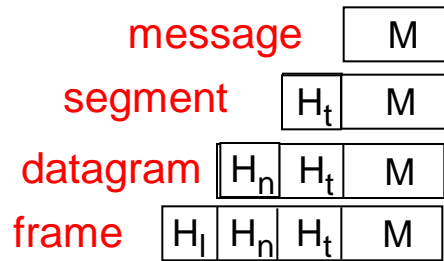


ISO/OSI reference model

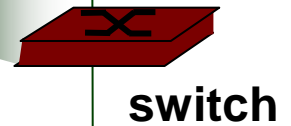
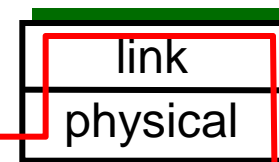
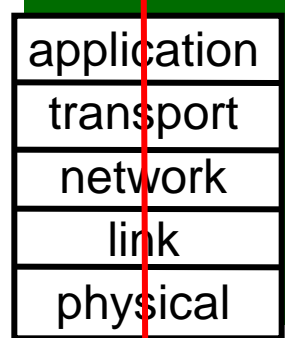
- **presentation**: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- **session**: synchronization, checkpointing, recovery of data exchange
- Internet stack “missing” these layers!
 - these services, *if needed*, must be implemented in application
 - needed?



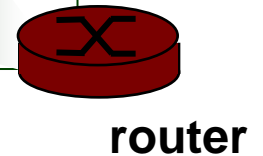
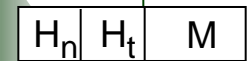
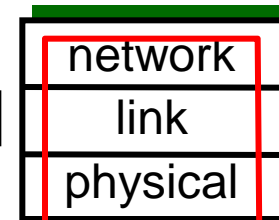
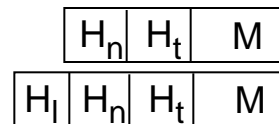
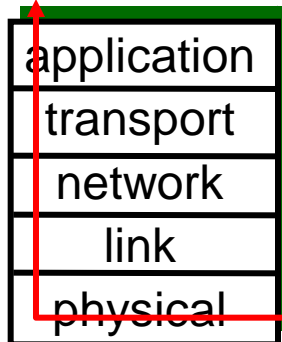
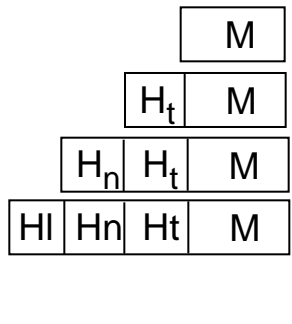
Encapsulation



source



destination



Chapter 1: roadmap

1.1 What *is* the Internet?

1.2 Network edge

- end systems, access networks, links

1.3 Network core

- circuit switching, packet switching, network structure

1.4 Delay, loss and throughput in packet-switched networks

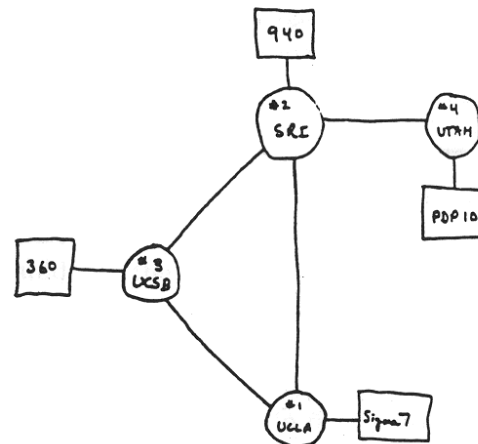
1.5 Protocol layers, service models

1.6 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
- **late 70'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

Introduction: Summary

Covered a “ton” of material!

- Internet overview
 - Incl. Internet / ISP structure
- what’s a protocol?
- network edge, core, access network
 - packet-switching versus circuit-switching
 - Internet structure
- performance: loss, delay, throughput
- layering, service models
- history

You (should ;) now have:

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