

# Chapter 1

## Introduction

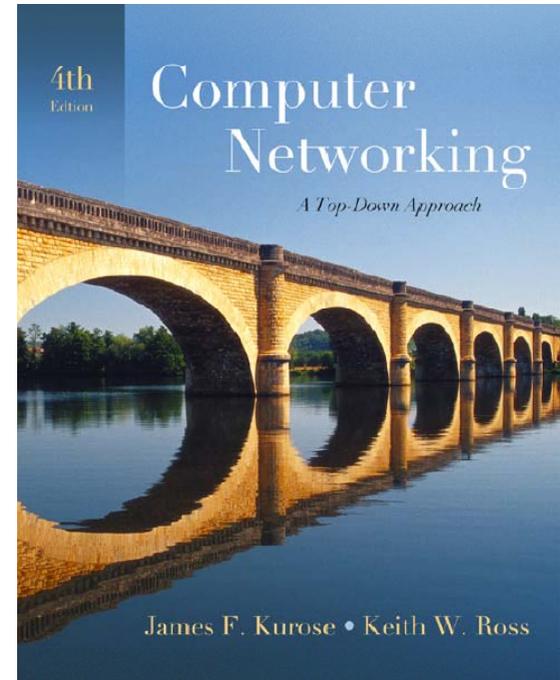
### A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2007  
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer Networking:  
A Top Down Approach ,  
4<sup>th</sup> edition.*

*Jim Kurose, Keith Ross  
Addison-Wesley, July  
2007.*

# 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; 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

## 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

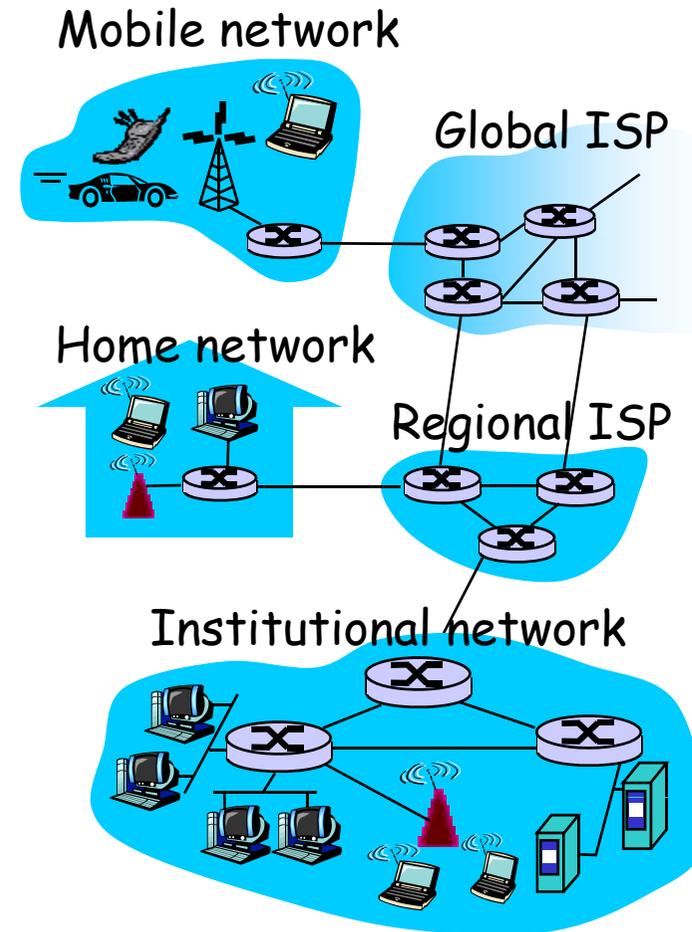
## 1.5 Protocol layers, service models

## 1.6 Networks under attack: security

## 1.7 History

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

-  PC
  -  server
  -  wireless laptop
  -  cellular handheld
  -  access points
  -  wired links
  -  router
- millions of connected computing devices: *hosts = end systems*
    - ❖ running *network apps*
  - *communication links*
    - ❖ fiber, copper, radio, satellite
    - ❖ transmission rate = *bandwidth*
  - *routers*: forward packets (chunks of data)



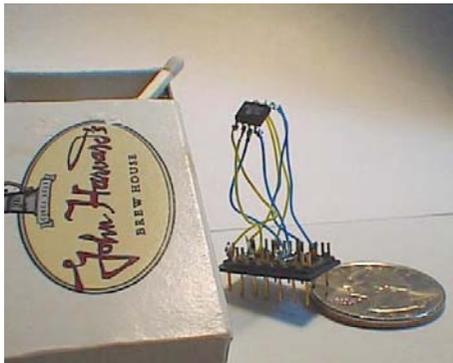
# “Cool” internet appliances



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



Web-enabled toaster +  
weather forecaster



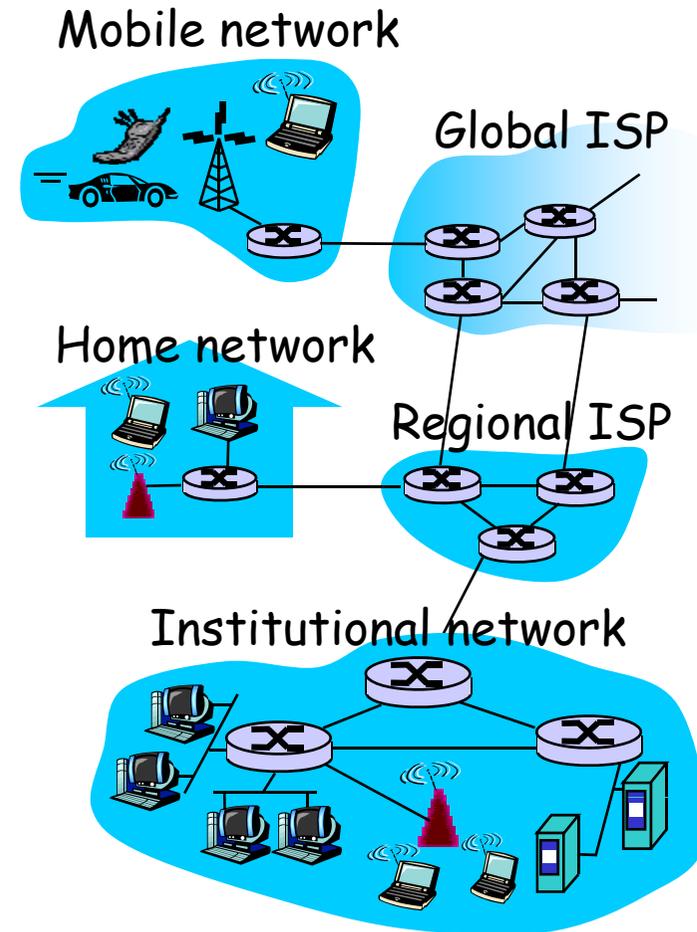
World's smallest web server  
<http://www-ccs.cs.umass.edu/~shri/iPic.html>



Internet phones

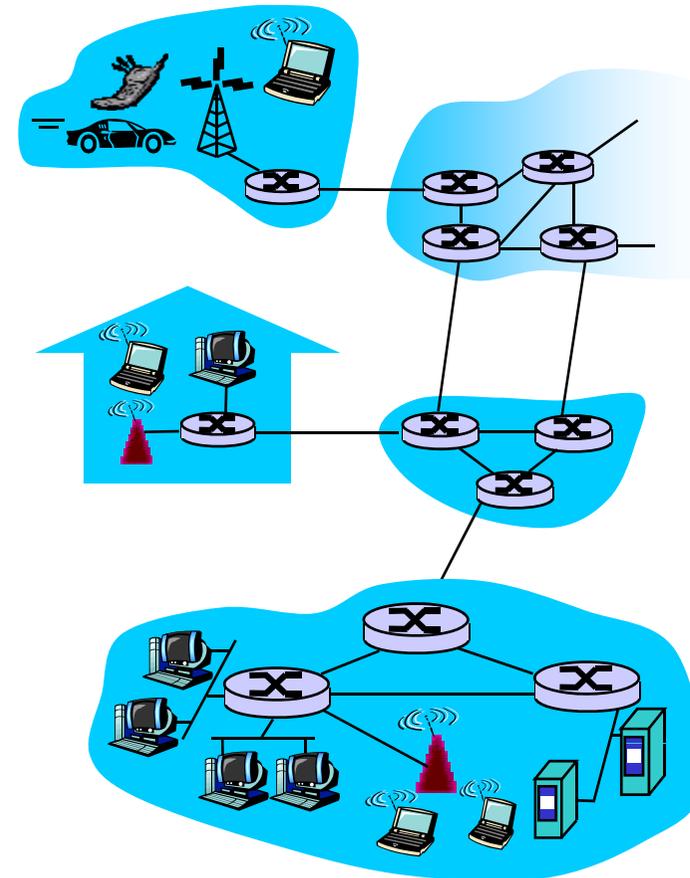
# What's the Internet: "nuts and bolts" view

- *protocols* control sending, receiving of msgs
  - ❖ e.g., TCP, IP, HTTP, Skype, Ethernet
- *Internet: "network of networks"*
  - ❖ loosely hierarchical
  - ❖ public Internet versus 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, VoIP, email, games, e-commerce, file sharing
- **communication services provided to apps:**
  - ❖ reliable data delivery from source to destination
  - ❖ "best effort" (unreliable) data delivery



# 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

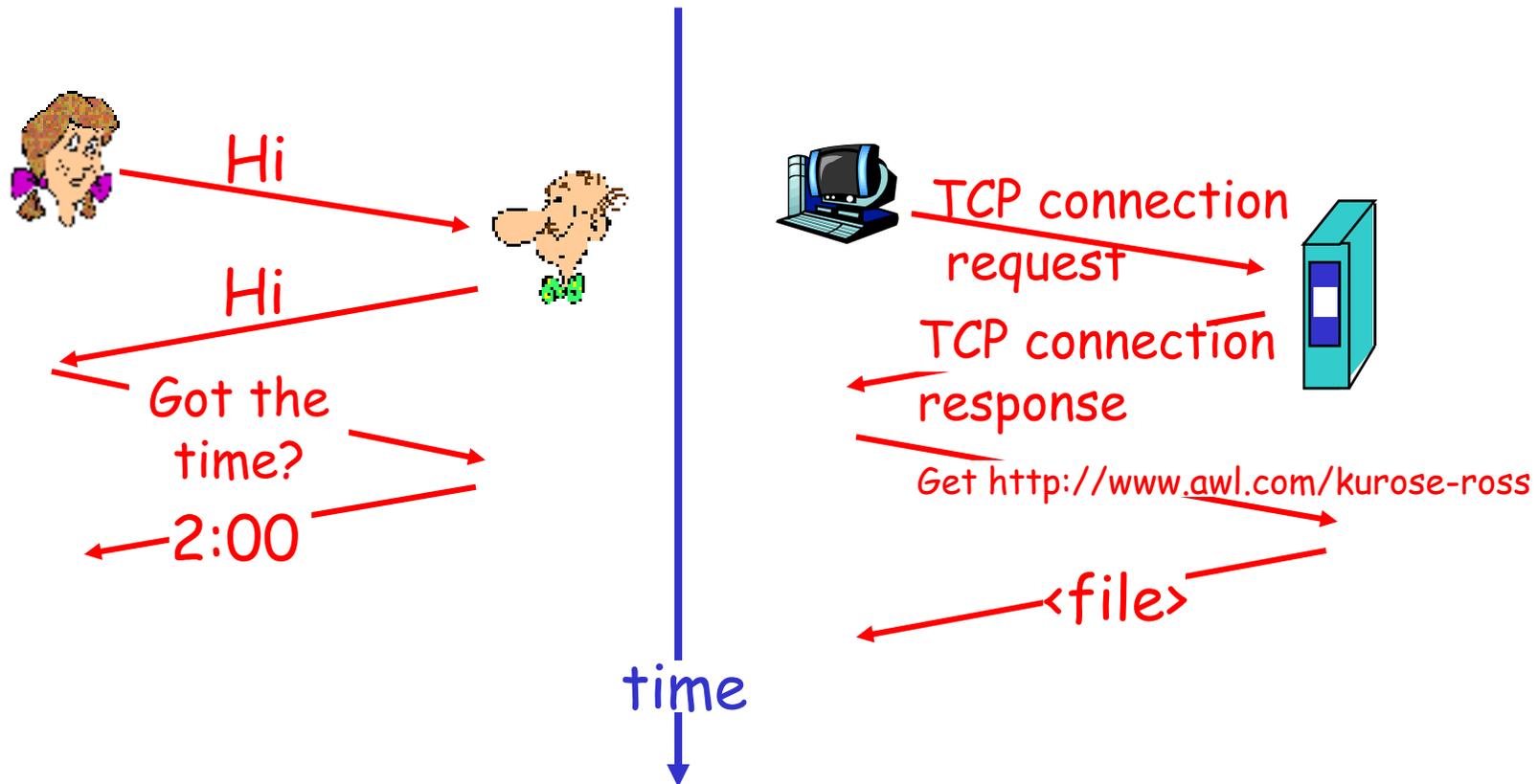
## network protocols:

- ❑ 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:



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

- circuit switching, packet switching, network structure

1.4 Delay, loss and throughput in packet-switched networks

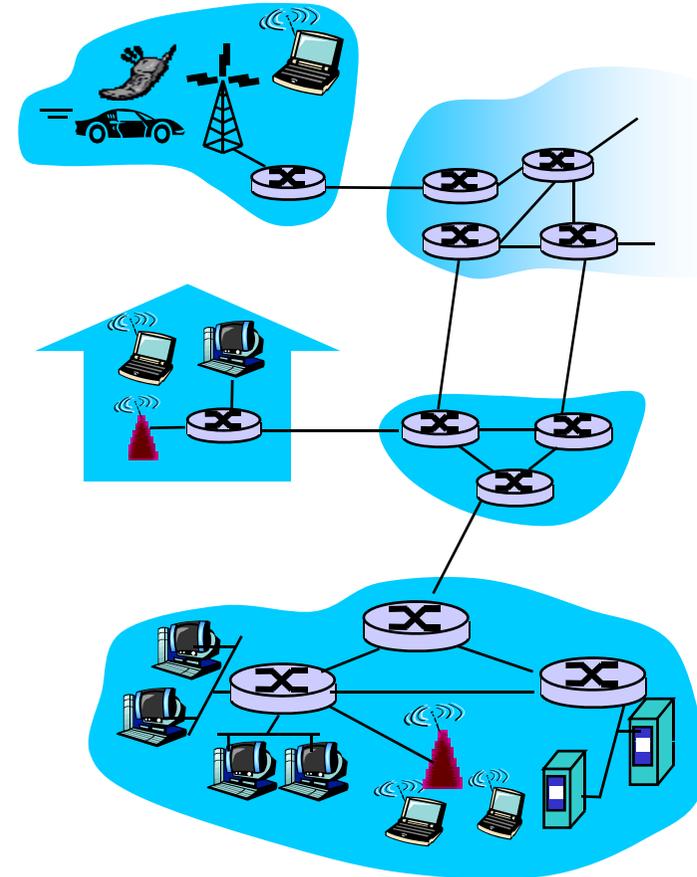
1.5 Protocol layers, service models

1.6 Networks under attack: security

1.7 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):

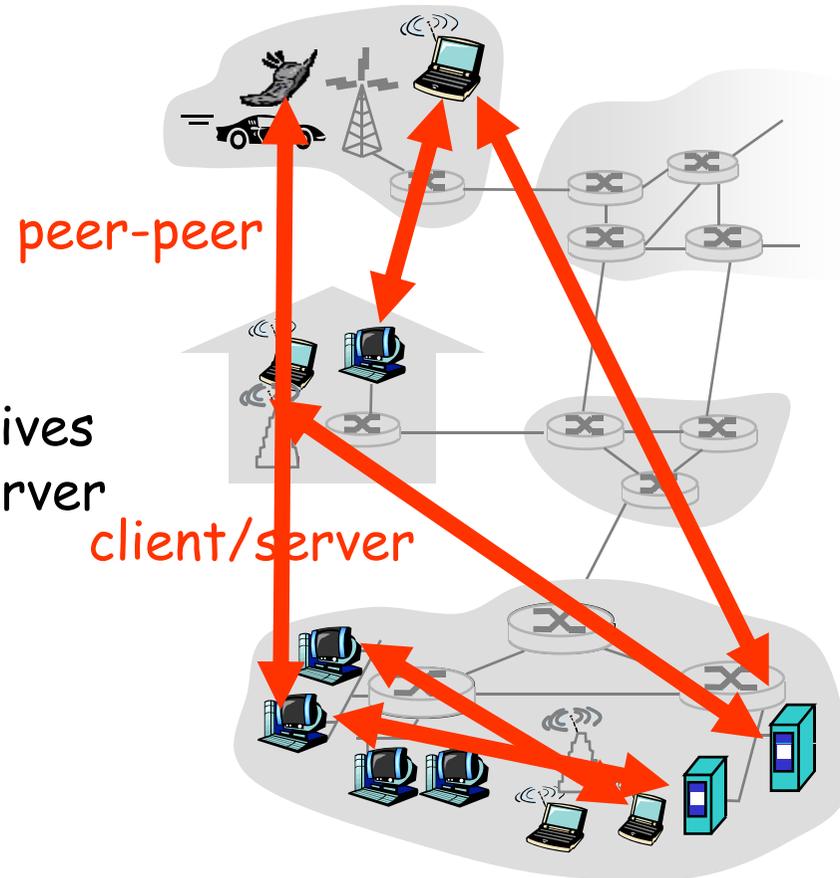
- ❖ run application programs
- ❖ e.g. Web, email
- ❖ at "edge of network"

## □ 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



# Network edge: reliable data transfer service

- Goal: data transfer between end systems
- ❑ *handshaking*: setup (prepare for) data transfer ahead of time
    - ❖ Hello, hello back human protocol
    - ❖ *set up "state"* in two communicating hosts
  - ❑ TCP - Transmission Control Protocol
    - ❖ Internet's reliable data transfer 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: best effort (unreliable) data transfer 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

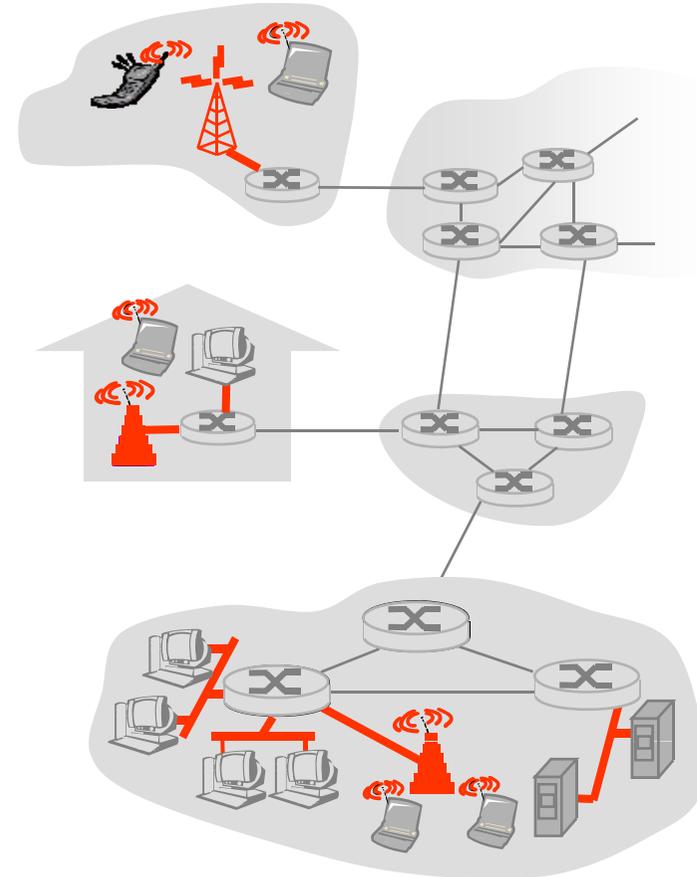
# Access networks and physical media

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

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

*Keep in mind:*

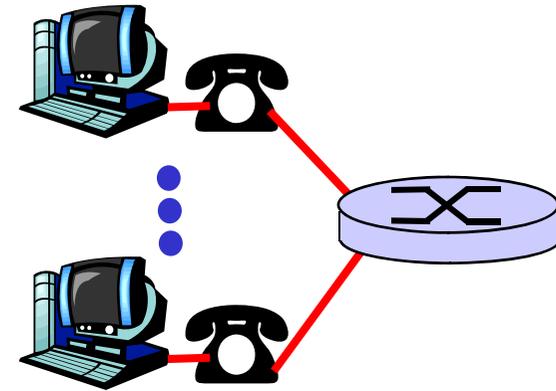
- ❑ bandwidth (bits per second) of access network?
- ❑ shared or dedicated?



## 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"



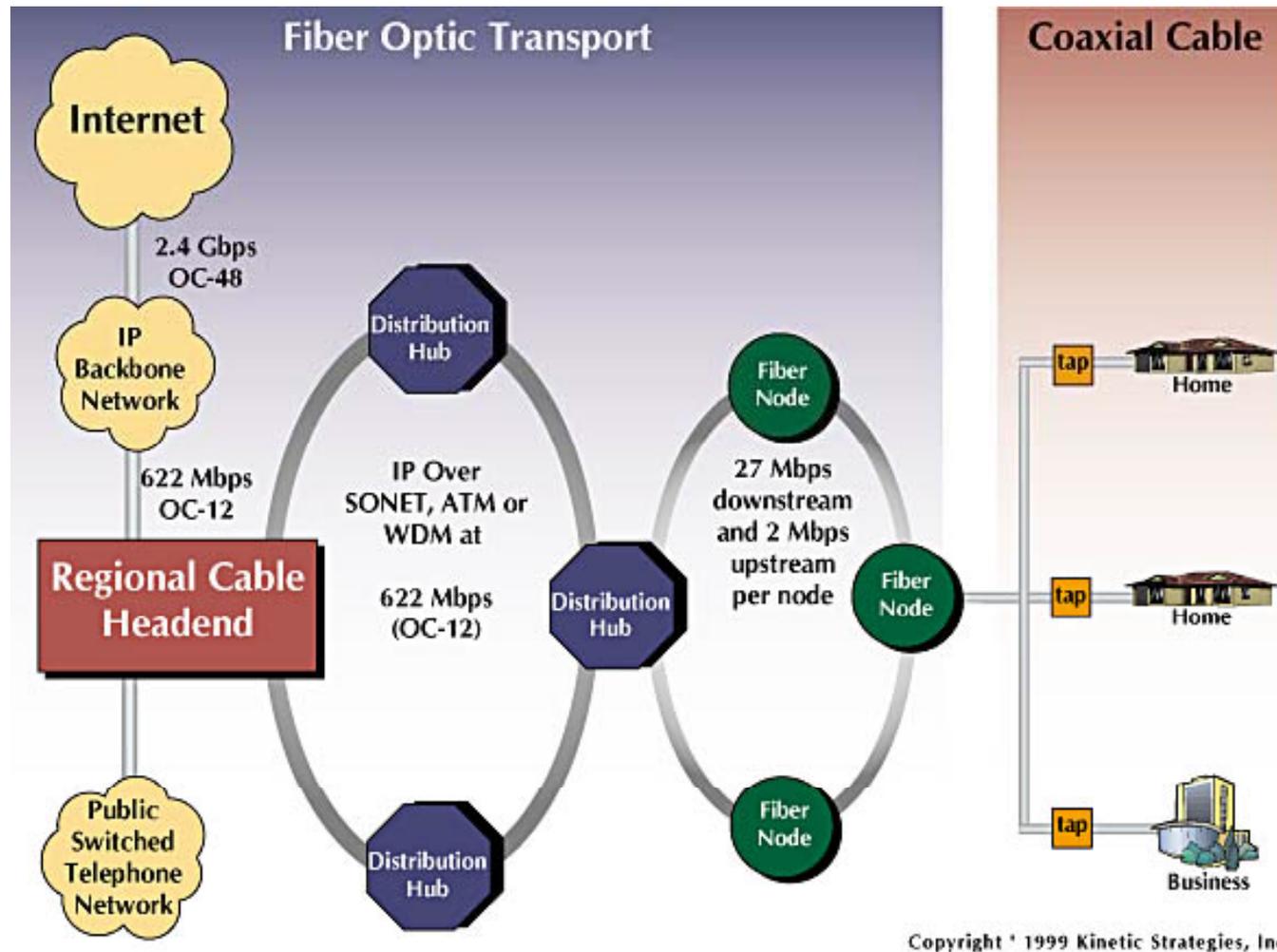
### □ DSL: digital subscriber line

- ❖ deployment: telephone company (typically)
- ❖ up to 1 Mbps upstream (today typically < 256 kbps)
- ❖ up to 8 Mbps downstream (today typically < 1 Mbps)
- ❖ dedicated physical line to telephone central office

## Residential access: cable modems

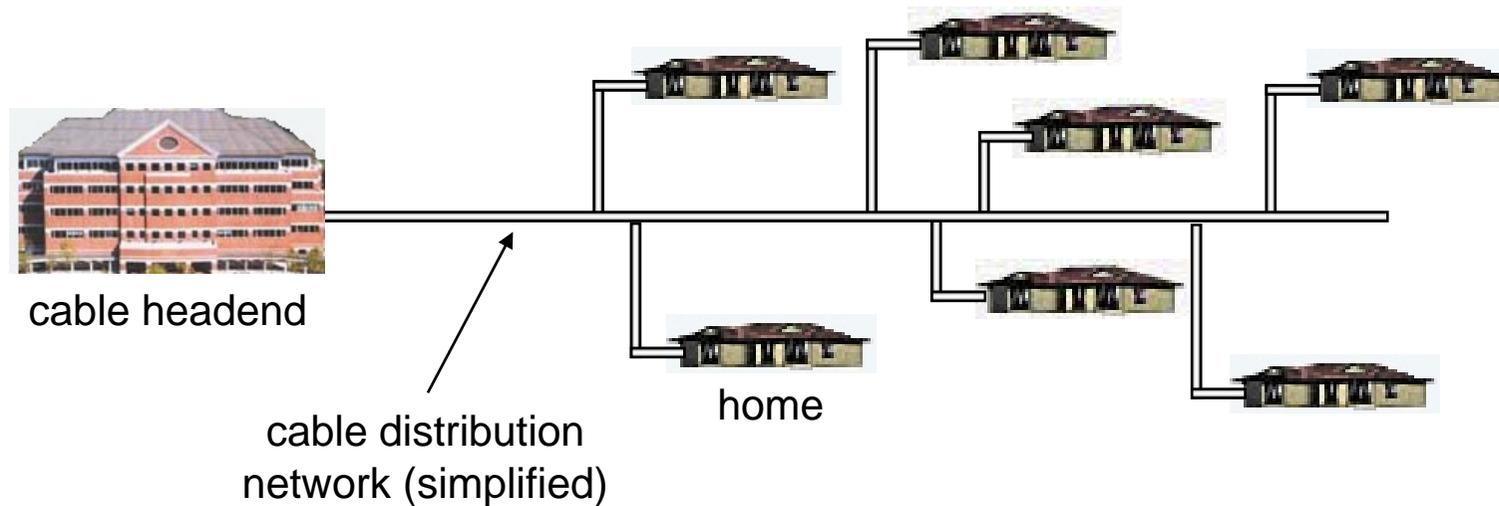
- ❑ **HFC: hybrid fiber coax**
  - ❖ asymmetric: up to 30Mbps downstream, 2 Mbps upstream
- ❑ **network** of cable and fiber attaches homes to ISP router
  - ❖ homes share access to router
- ❑ deployment: available via cable TV companies

# Residential access: cable modems

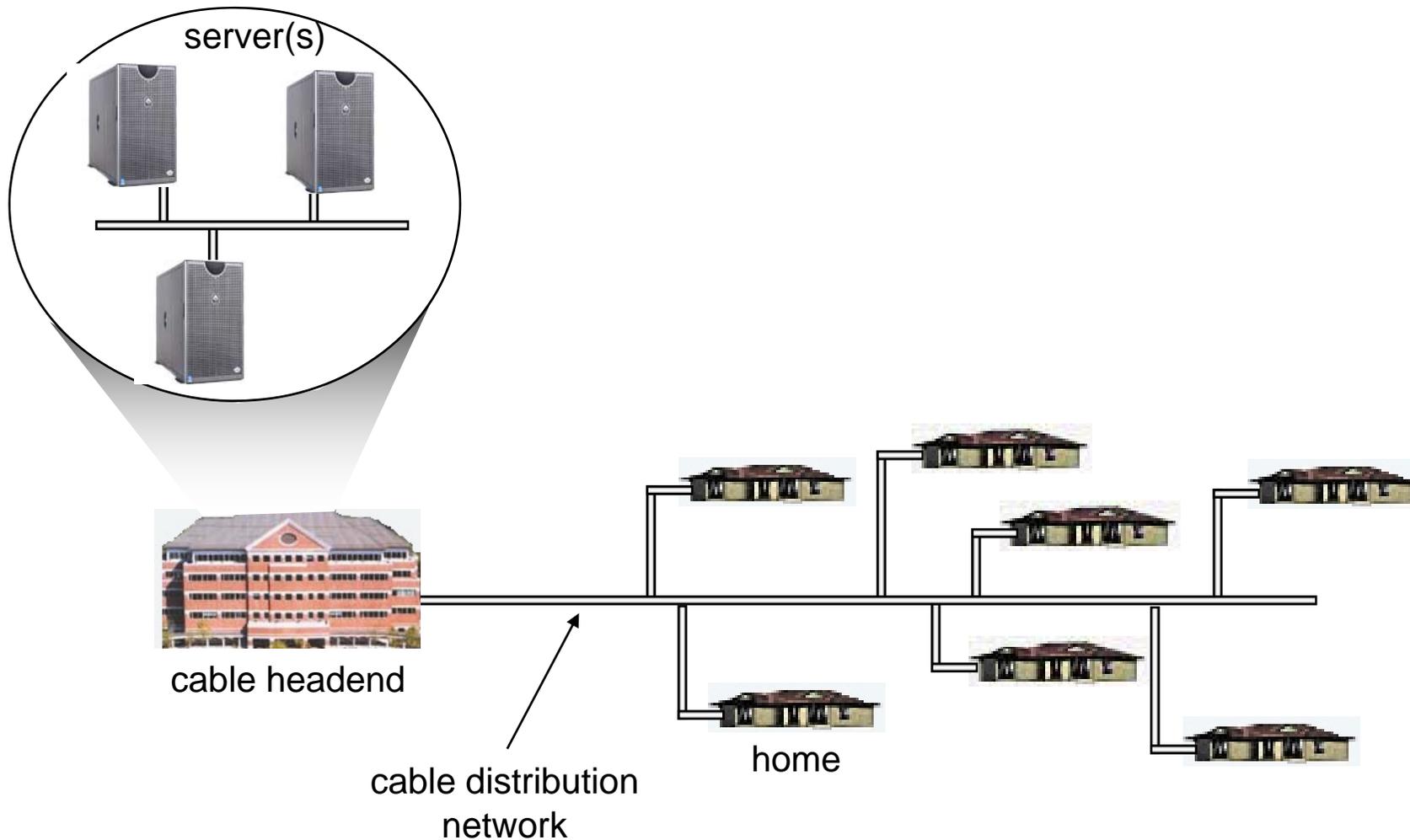


# Cable Network Architecture: Overview

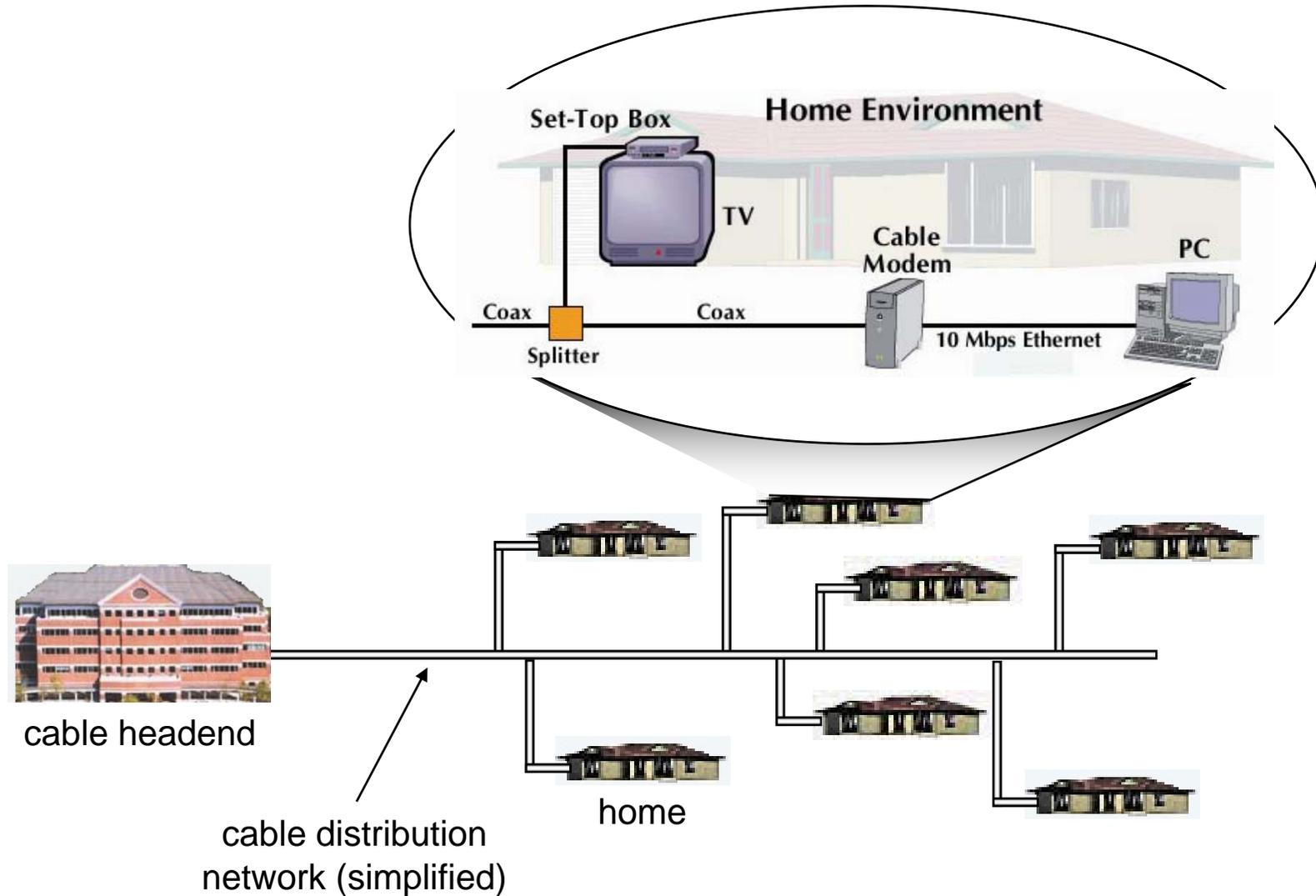
Typically 500 to 5,000 homes



# Cable Network Architecture: Overview



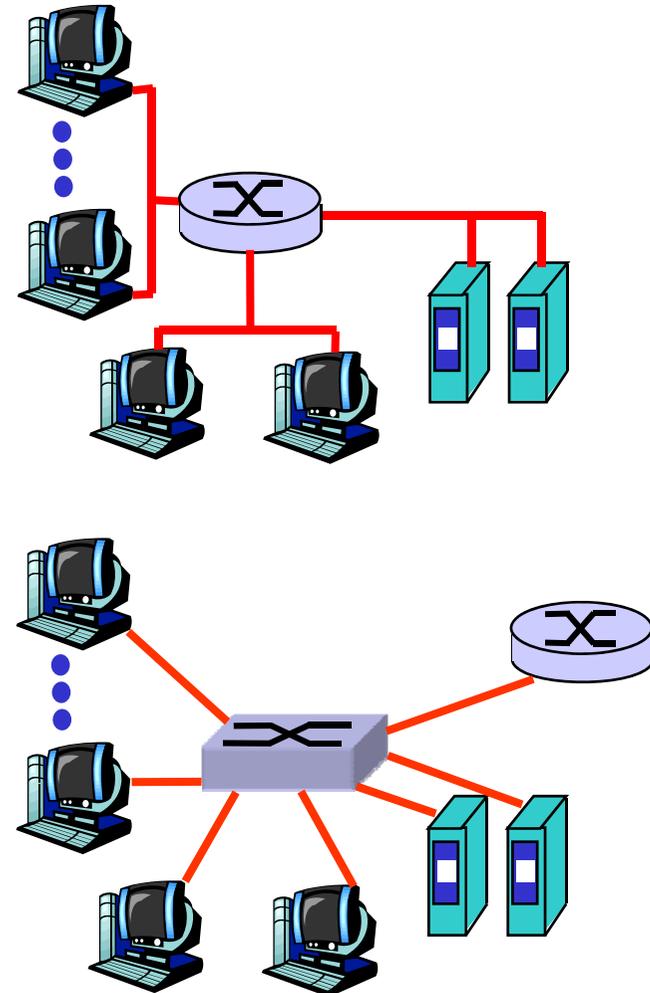
# Cable Network Architecture: Overview





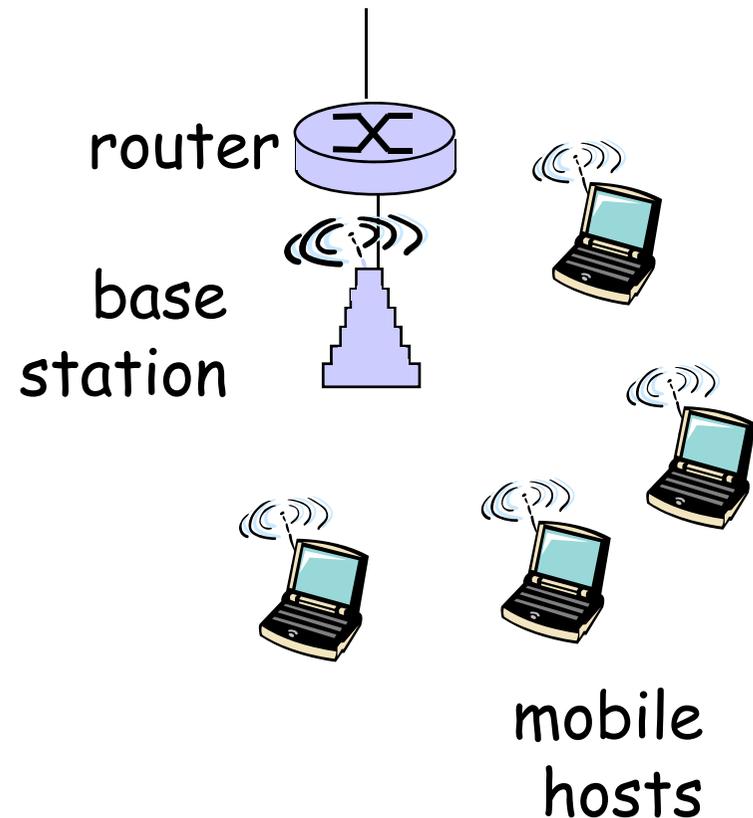
# Company access: local area networks

- ❑ company/univ **local area network** (LAN) connects end system to edge router
- ❑ **Ethernet:**
  - ❖ 10 Mbs, 100Mbps, 1Gbps, 10Gbps Ethernet
  - ❖ modern configuration: end systems connect into *Ethernet switch*
- ❑ LANs: chapter 5



# Wireless access networks

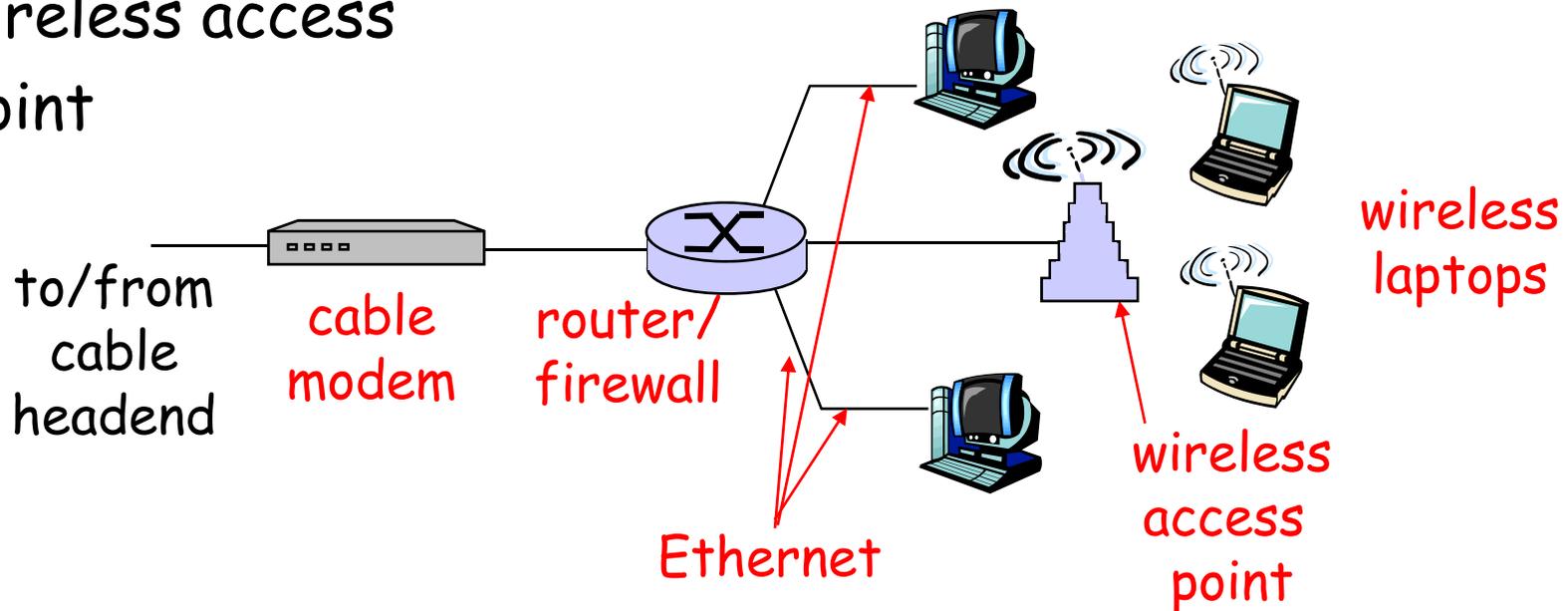
- shared *wireless* access network connects end system to router
  - ❖ via base station aka "access point"
- **wireless LANs:**
  - ❖ 802.11b/g (WiFi): 11 or 54 Mbps
- **wider-area wireless access**
  - ❖ provided by telco operator
  - ❖ ~1Mbps over cellular system
    - EVDO (Evolution, Data Only ), a standard for high speed wireless broadband
    - HSDPA (High Speed Download Packet Access )
  - ❖ next up (?): WiMAX (10's Mbps) over wide area



# Home networks

## Typical home network components:

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



# 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)
  - ❖ 3G cellular: ~ 1 Mbps
- ❑ **satellite**
  - ❖ Kbps to 45Mbps channel (or multiple smaller channels)
  - ❖ 270 msec end-end delay
  - ❖ geosynchronous versus low altitude

# 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

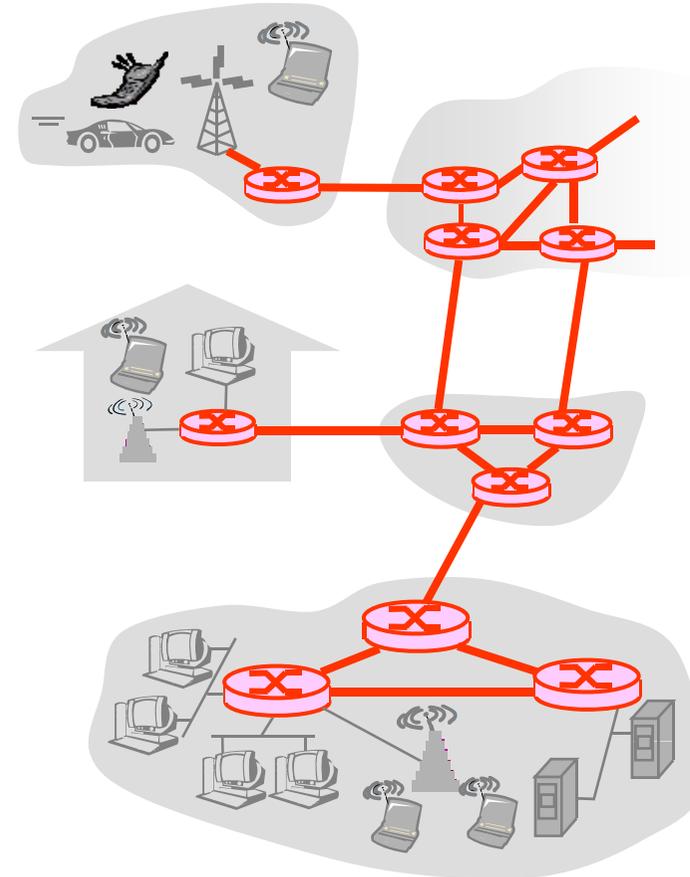
1.5 Protocol layers, service models

1.6 Networks under attack: security

1.7 History

# The Network Core

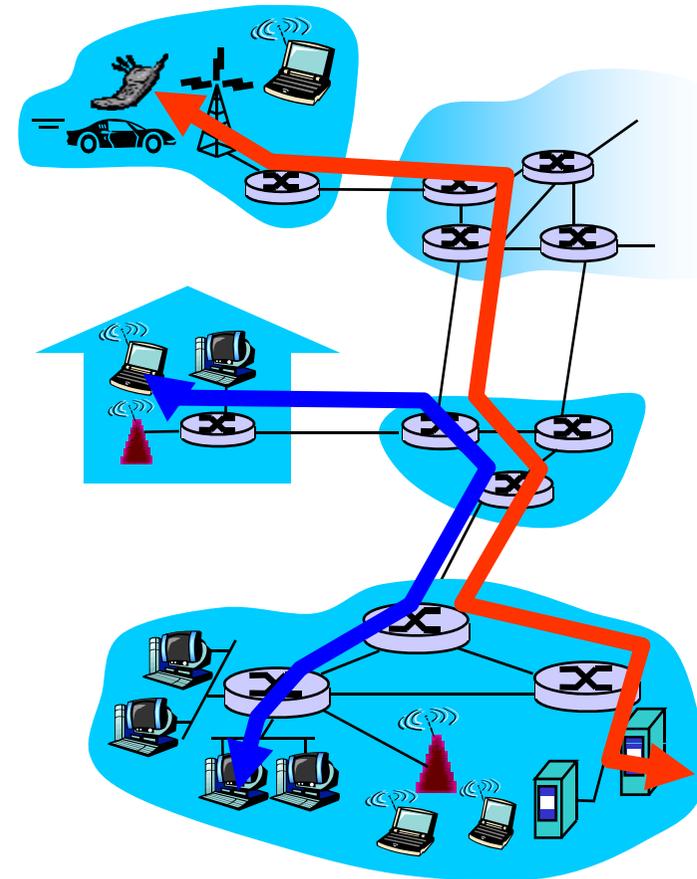
- mesh of interconnected routers
- the fundamental question: how is data transferred through net?
  - ❖ circuit switching: dedicated circuit per call: telephone net
  - ❖ packet-switching: data sent thru net in discrete "chunks"



# 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



# 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*)

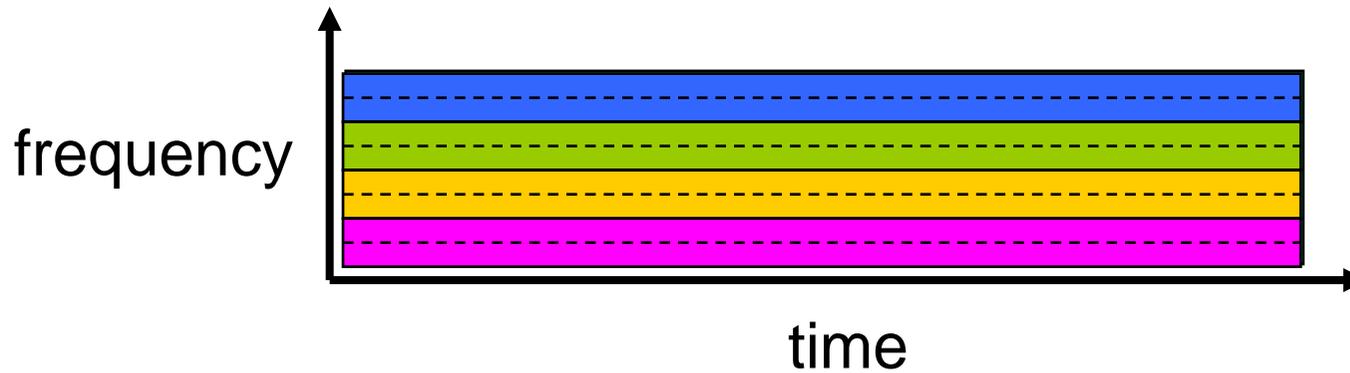
- dividing link bandwidth into "pieces"
  - ❖ frequency division
  - ❖ time division

# Circuit Switching: FDM and TDM

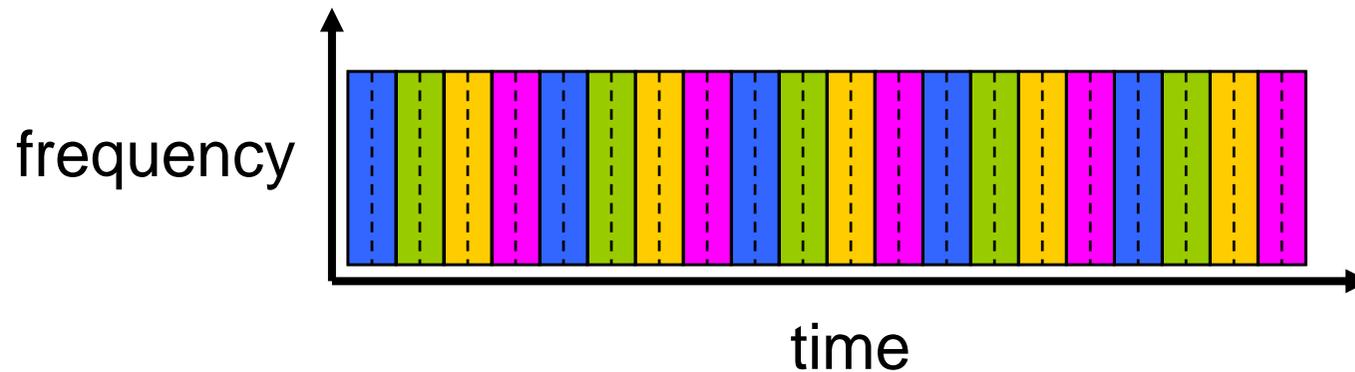
Example:

4 users

FDM



TDM



# 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!

$$\begin{aligned}\text{Total time} &= \text{call setup} + \text{transmission time} \\ &= 500 \text{ ms} + 640000 / (1.536\text{M} / 24)\end{aligned}$$

# 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*

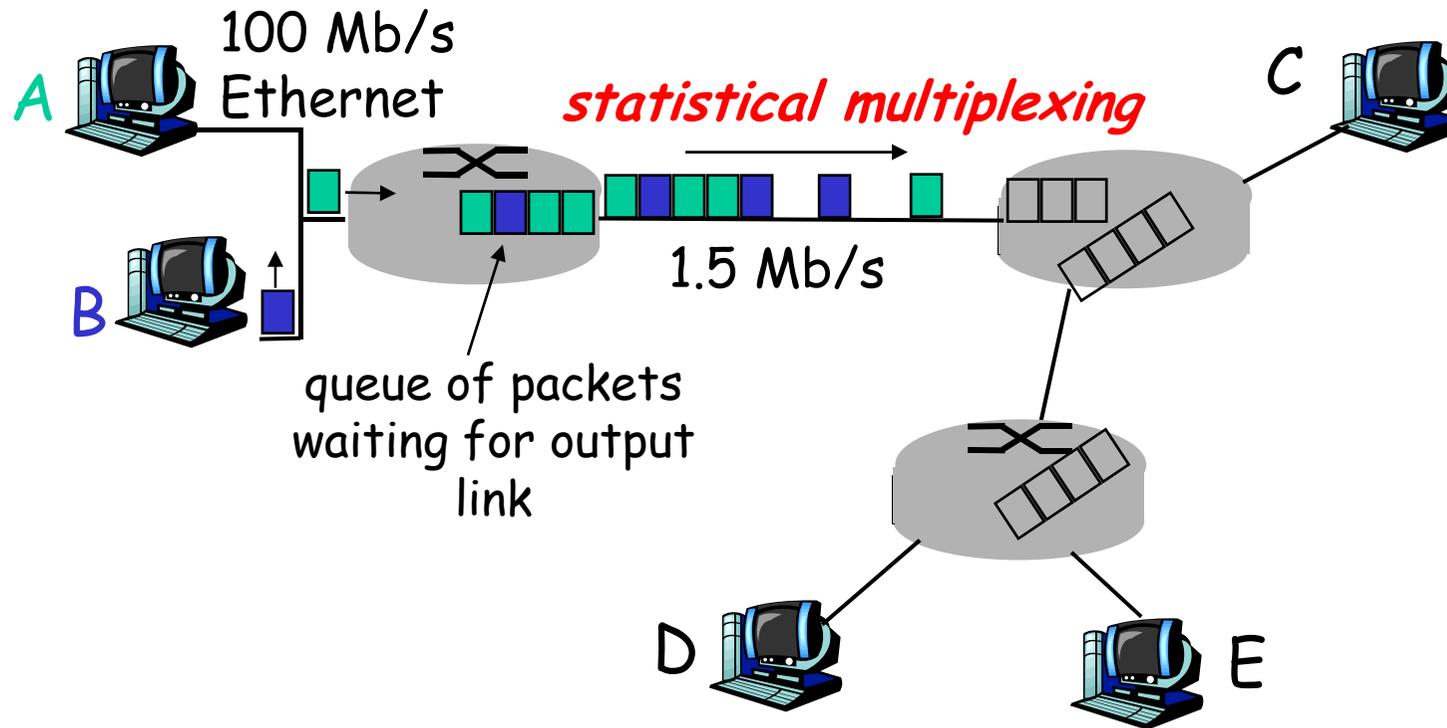
Bandwidth division into "pieces"  
Dedicated allocation  
Resource reservation



resource contention:

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

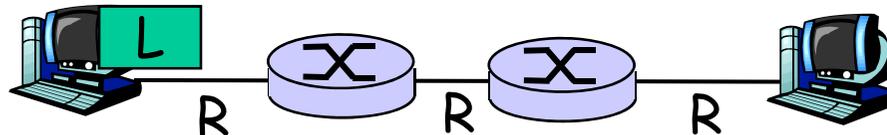
# Packet Switching: Statistical Multiplexing



Sequence of A & B packets does not have fixed pattern,  
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

} more on delay shortly ...

# Packet switching versus circuit switching

*Packet switching allows more users to use network!*

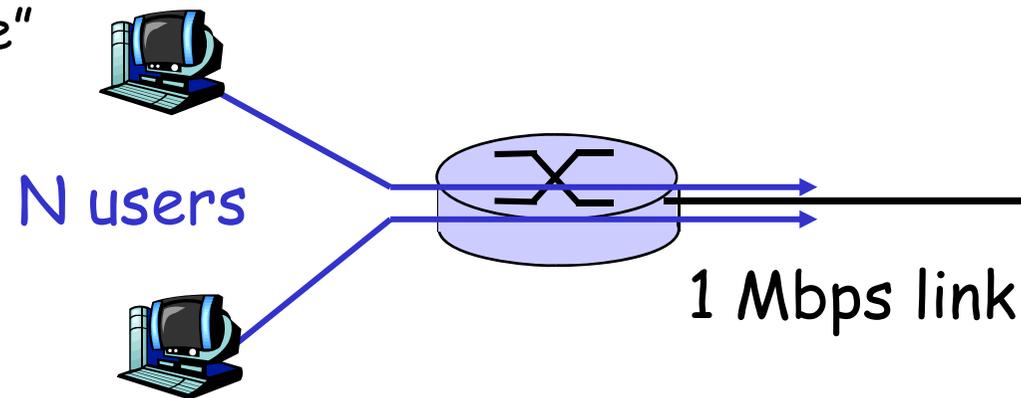
- ❑ 1 Mb/s link
- ❑ each user:
  - ❖ 100 kb/s when "active"
  - ❖ active 10% of time

- ❑ *circuit-switching:*

- ❖ 10 users

- ❑ *packet switching:*

- ❖ with 35 users, probability  $> 10$  active at same time is less than .0004



Q: how did we get value 0.0004?

# Packet switching versus circuit switching

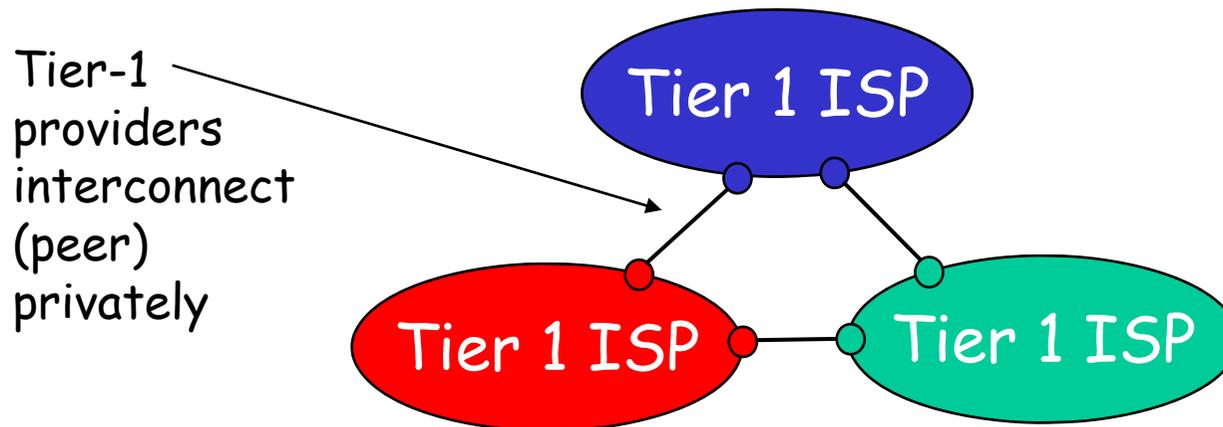
Is packet switching a "slam dunk winner?"

- ❑ great for bursty data
  - ❖ resource sharing
  - ❖ simpler, no call setup
- ❑ **excessive congestion:** packet delay and loss
  - ❖ protocols needed for 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)

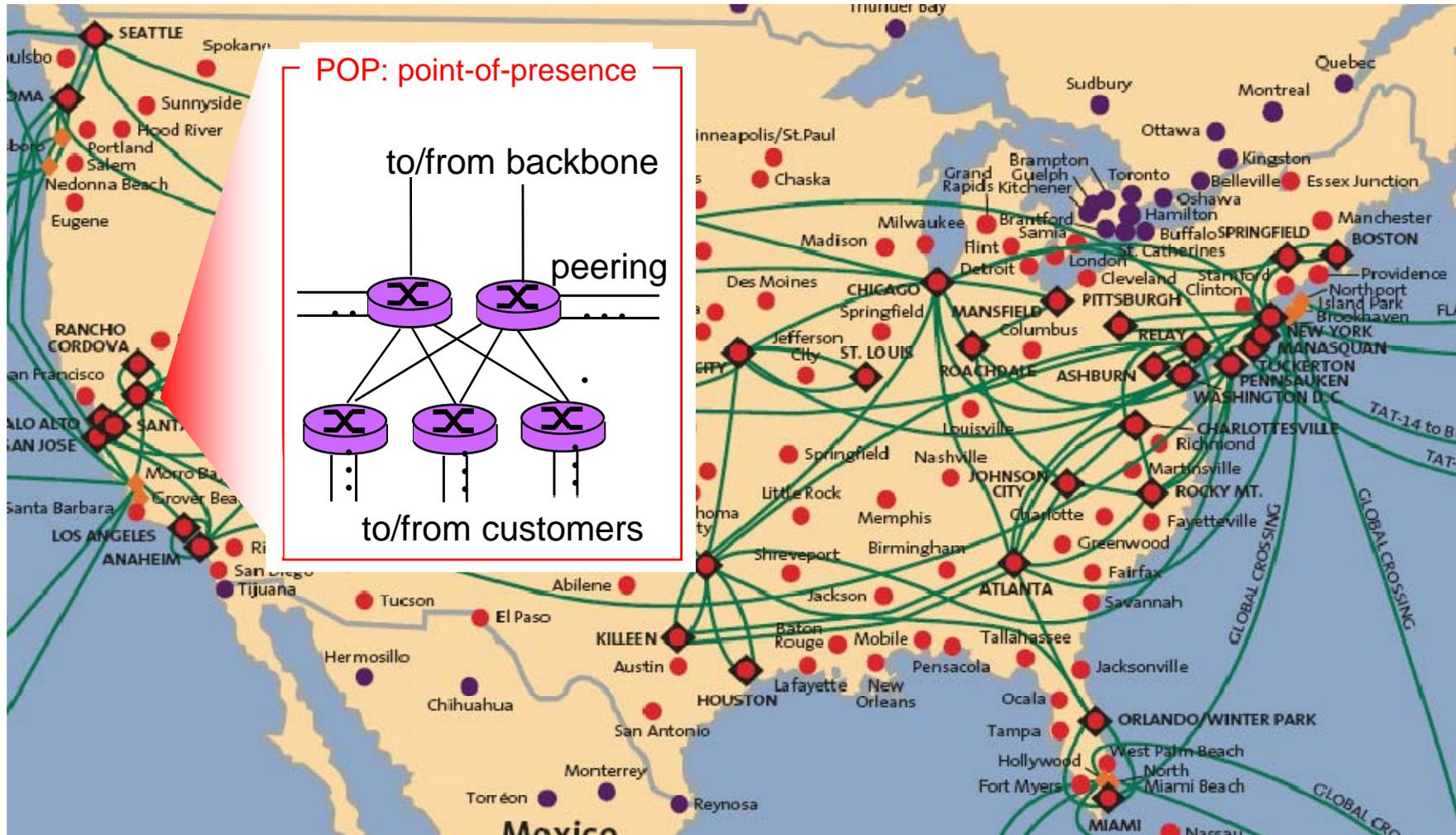
Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

# Internet structure: network of networks

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

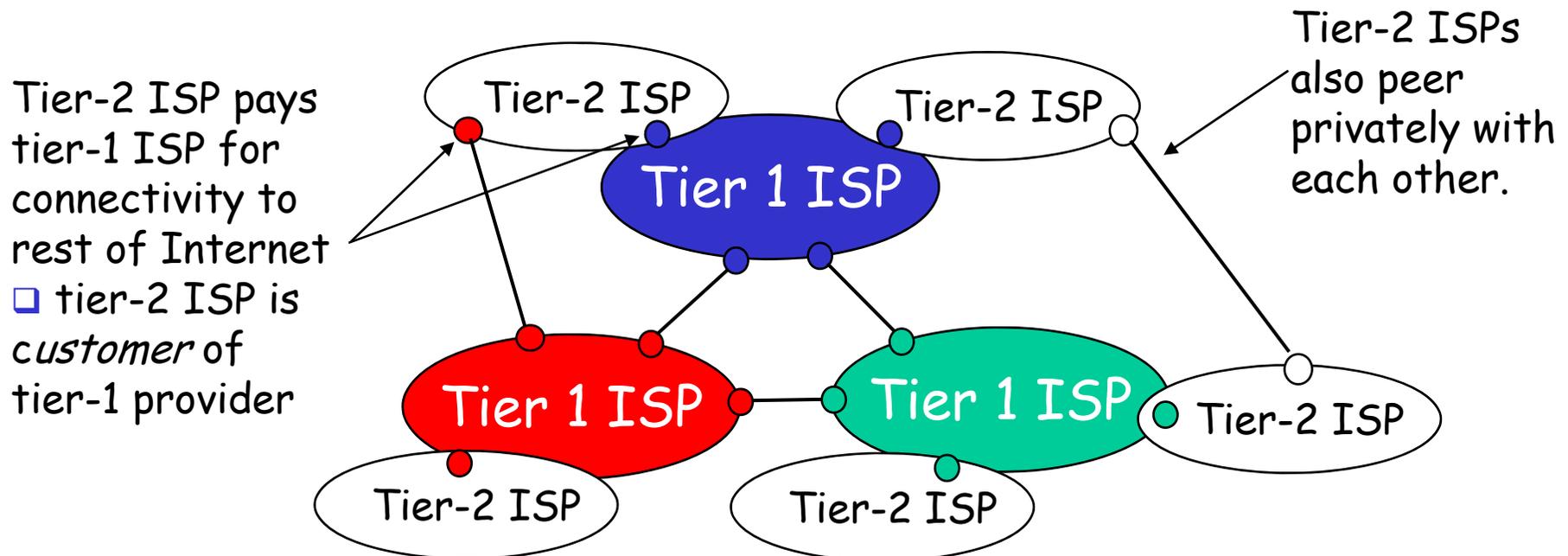


# Tier-1 ISP: e.g., Sprint



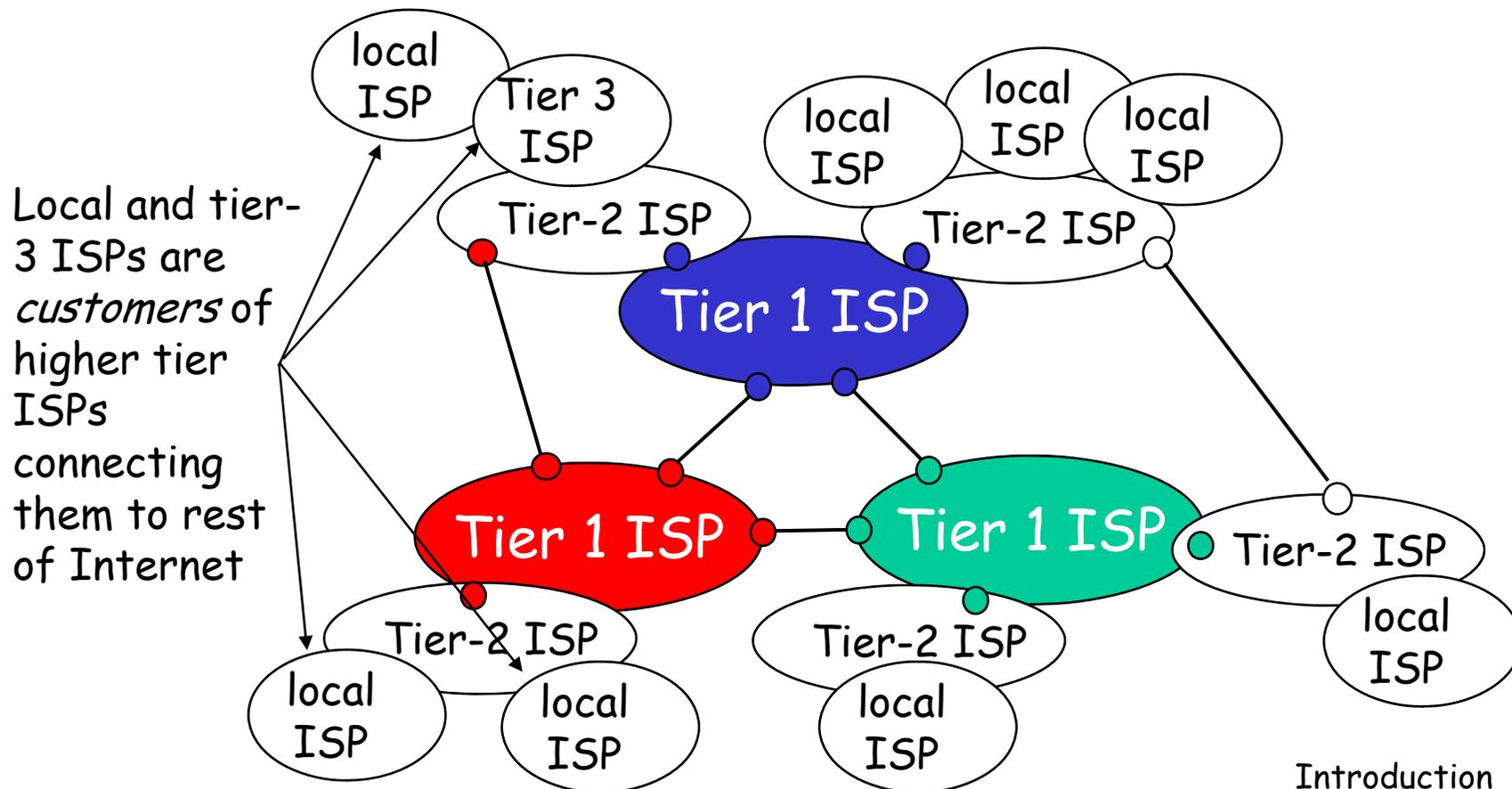
# Internet structure: network of networks

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



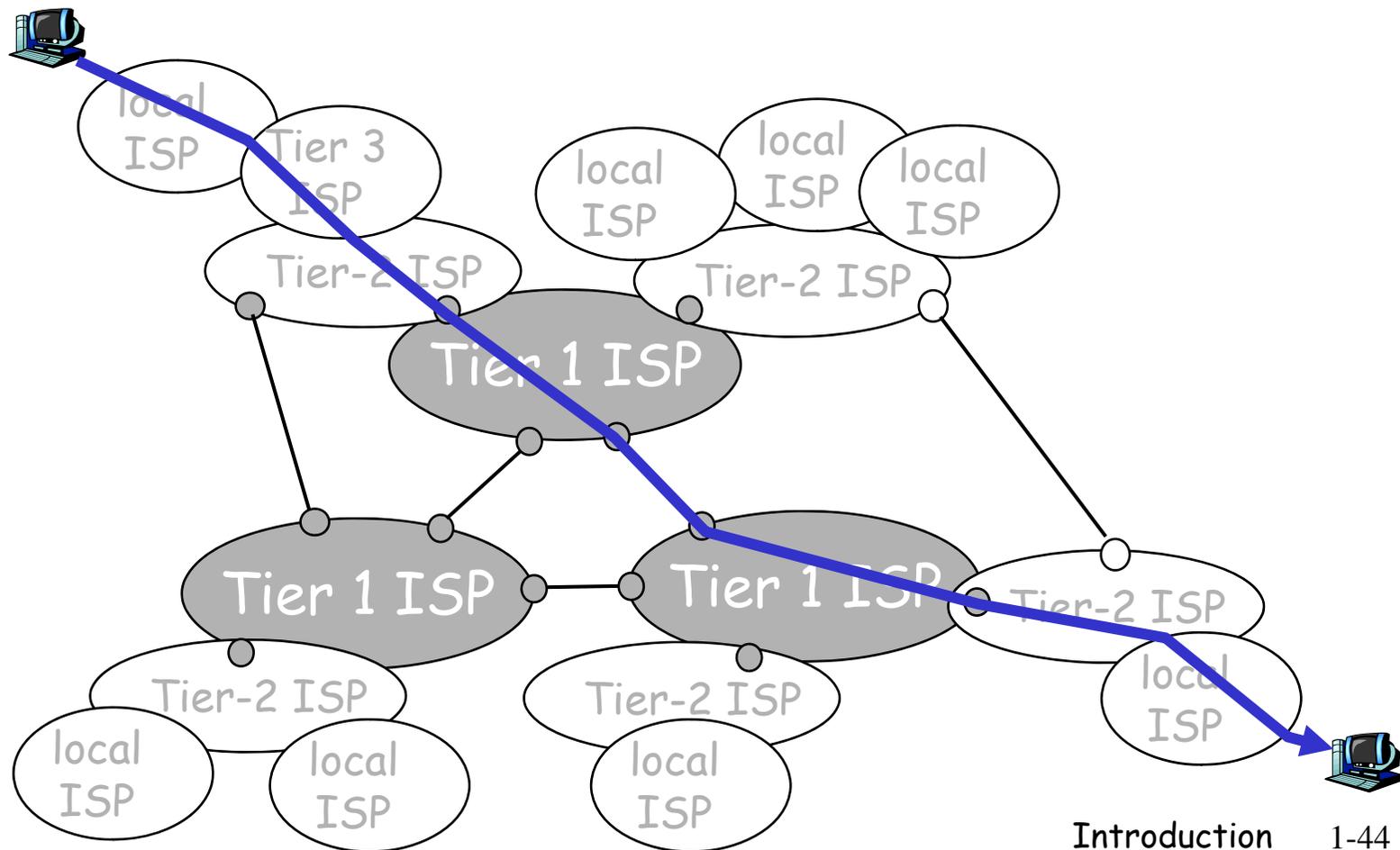
# Internet structure: network of networks

- “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

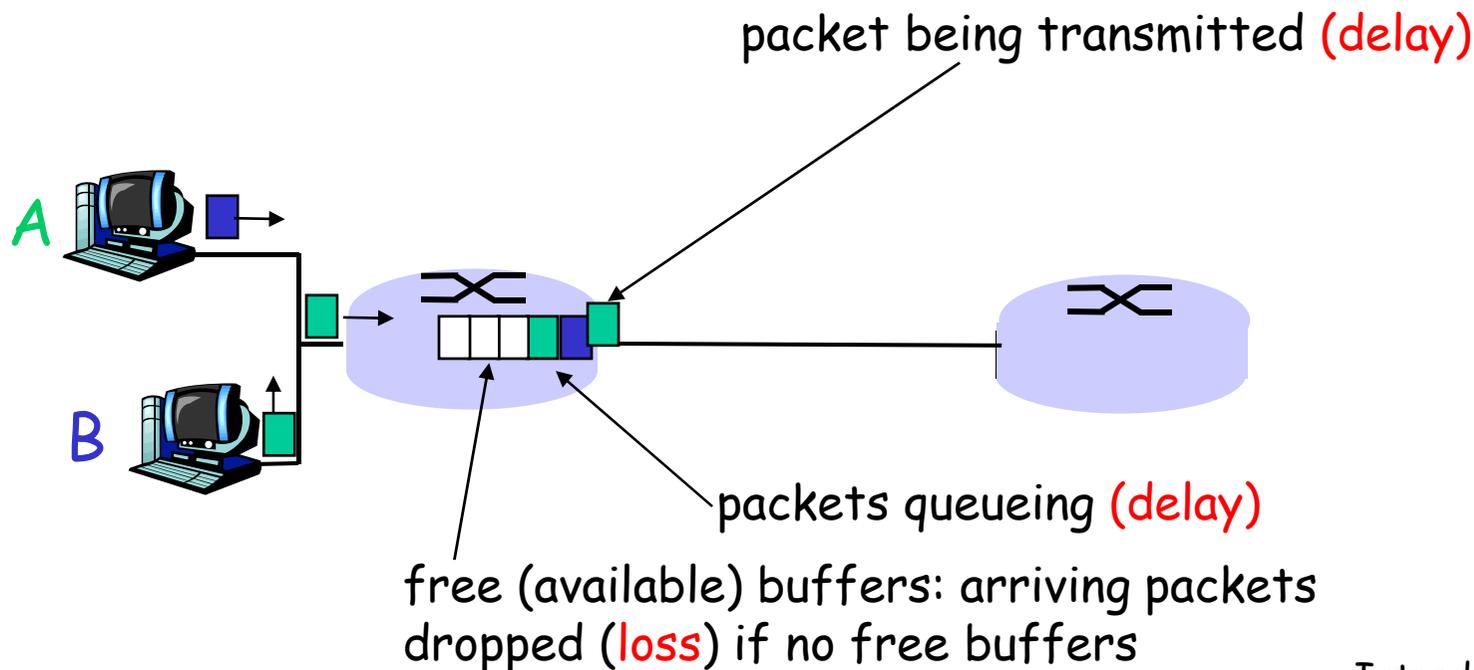
1.6 Networks under attack: security

1.7 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



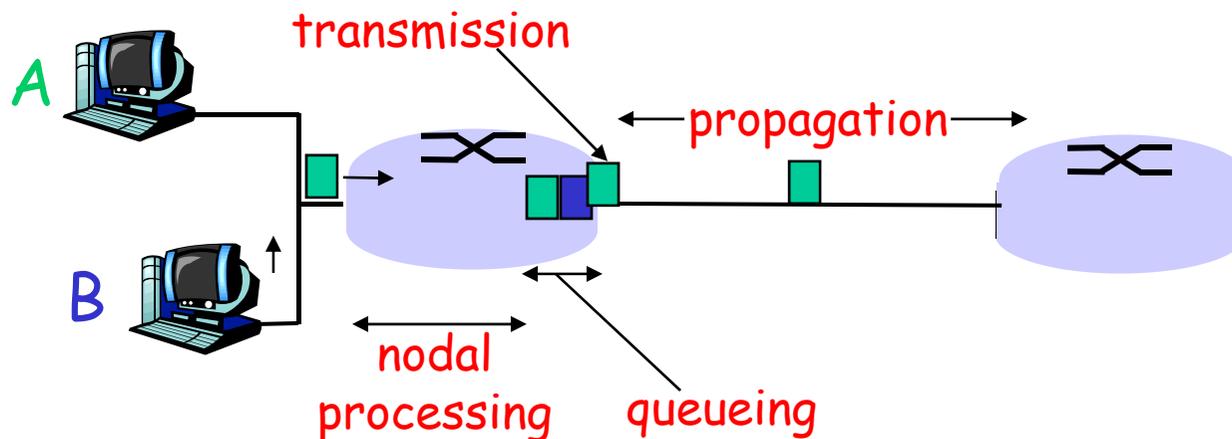
# Four sources of packet delay

## □ 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

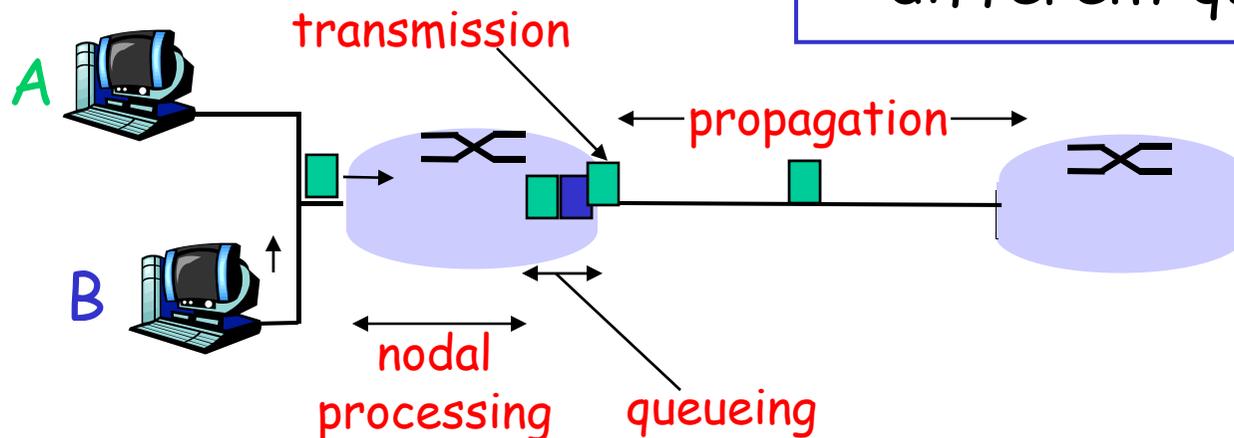
## 3. Transmission delay:

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

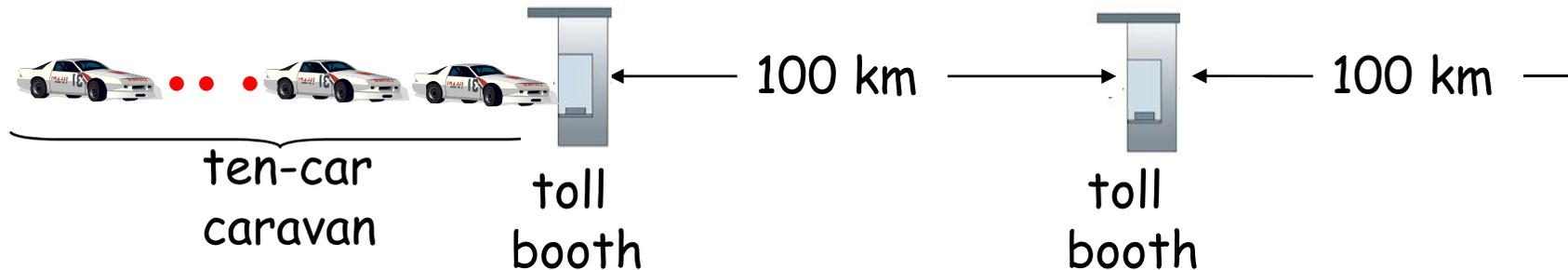
## 4. Propagation delay:

- $d$  = 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!

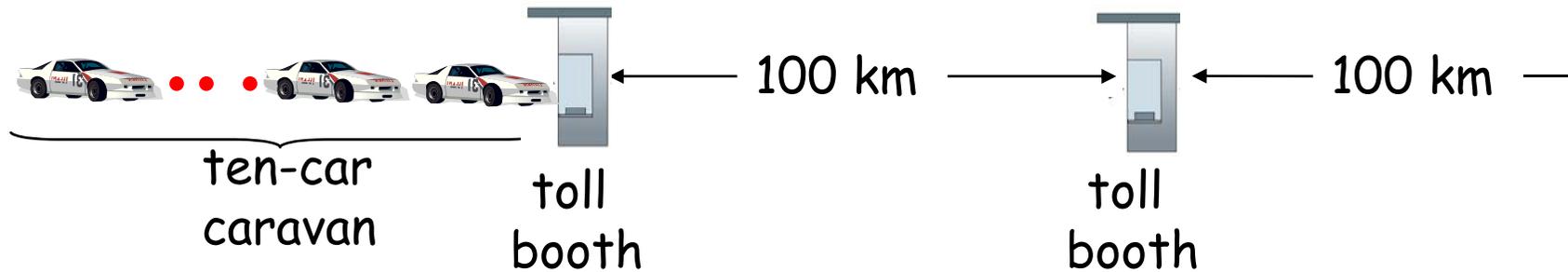


# Caravan analogy



- ❑ cars "propagate" at 100 km/hr
- ❑ toll booth takes 12 sec to service 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 booth:  $100 \text{ km} / (100 \text{ km/hr}) = 1$  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

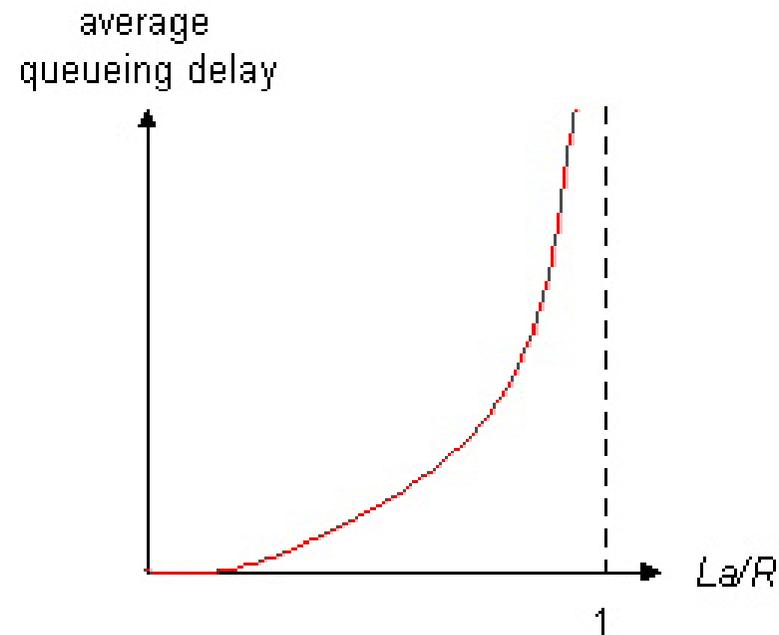
$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{queue}} + d_{\text{trans}} + d_{\text{prop}}$$

- $d_{\text{proc}}$  = processing delay
  - ❖ typically a few microsecs or less
- $d_{\text{queue}}$  = queuing delay
  - ❖ depends on congestion
- $d_{\text{trans}}$  = transmission delay
  - ❖ =  $L/R$ , significant for low-speed links
- $d_{\text{prop}}$  = 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

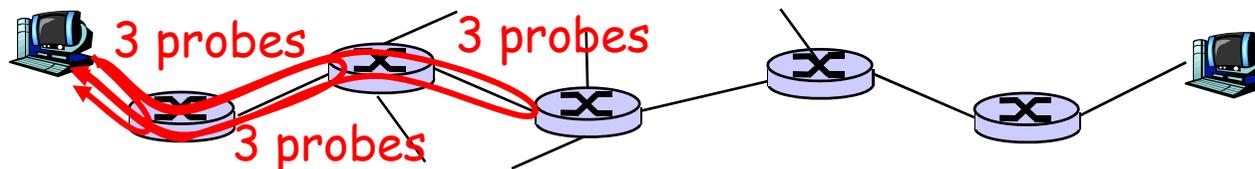
traffic intensity =  $La/R$



- $La/R \sim 0$ : average queueing delay small
- $La/R \rightarrow 1$ : 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 measurements from  
gaia.cs.umass.edu to cs-gw.cs.umass.edu



```

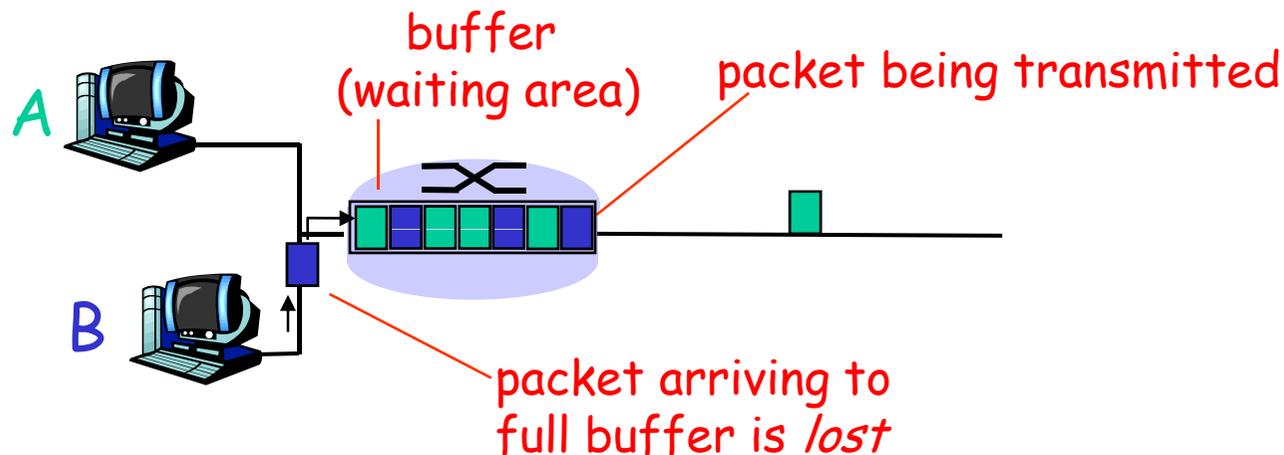
1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms
6 abilene-vbns.abilene.ucaid.edu (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)

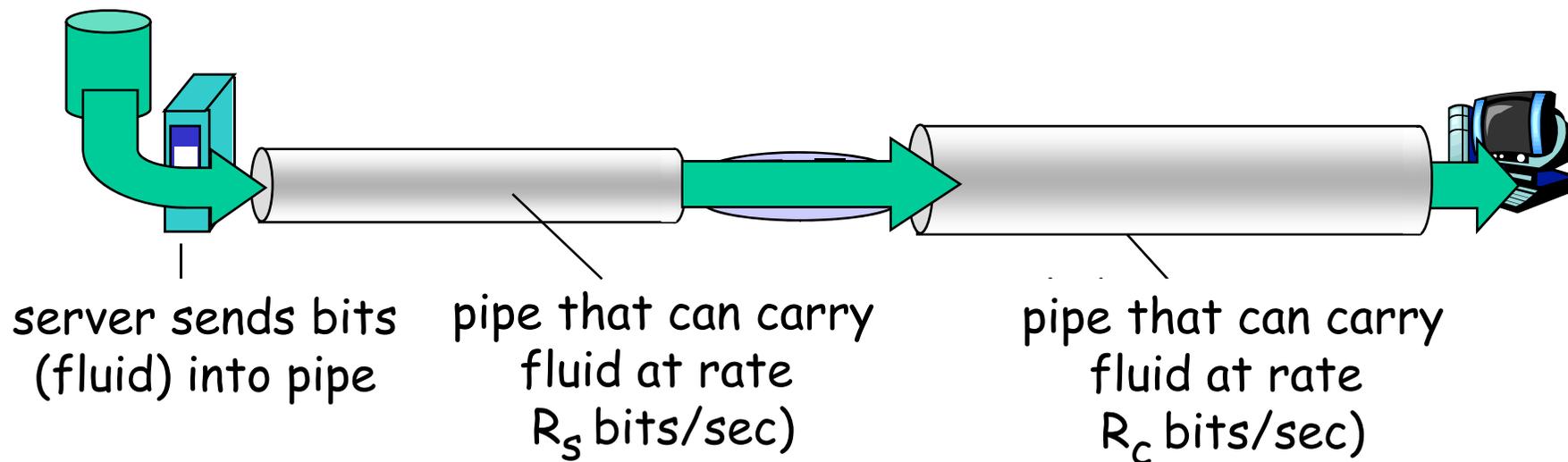
# Packet loss

- ❑ queue (aka buffer) preceding link 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



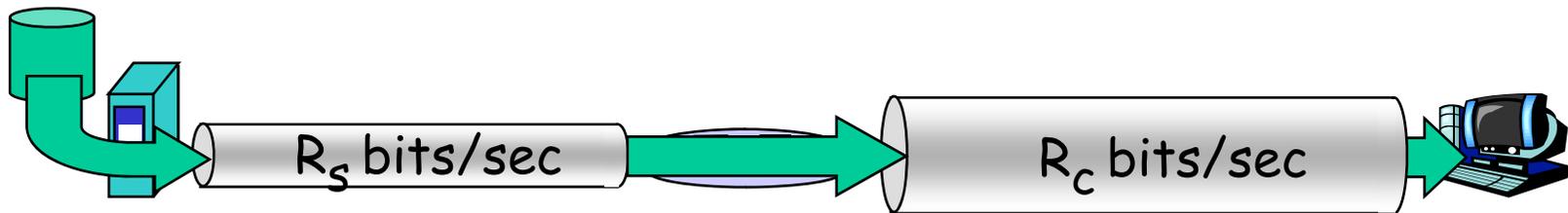
# Throughput

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

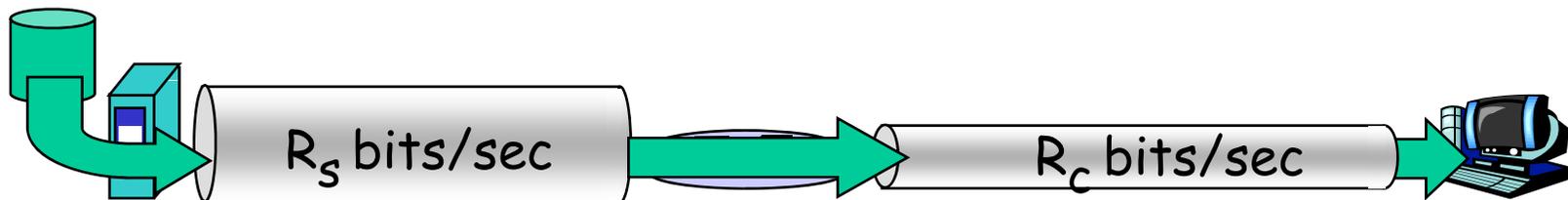


# Throughput (more)

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



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

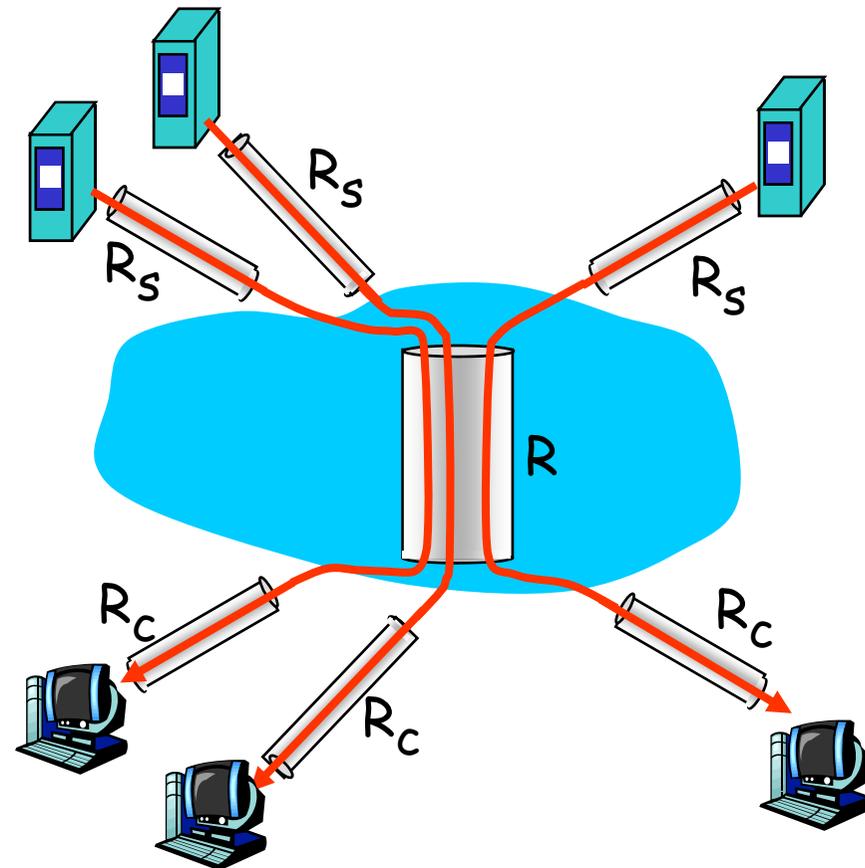


*bottleneck link*

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 Networks under attack: security

1.7 History

# Protocol "Layers"

## Networks are complex!

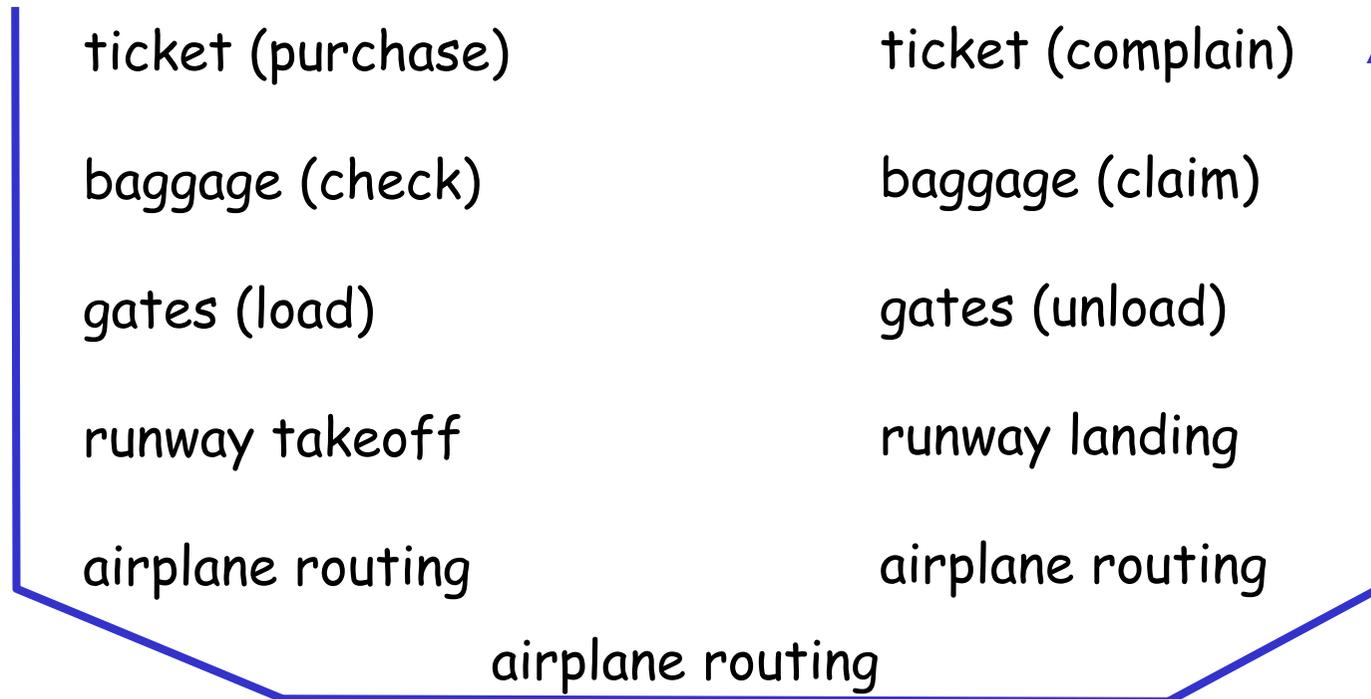
- many "pieces":
  - ❖ hosts
  - ❖ routers
  - ❖ links of various media
  - ❖ applications
  - ❖ protocols
  - ❖ hardware, software

## Question:

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

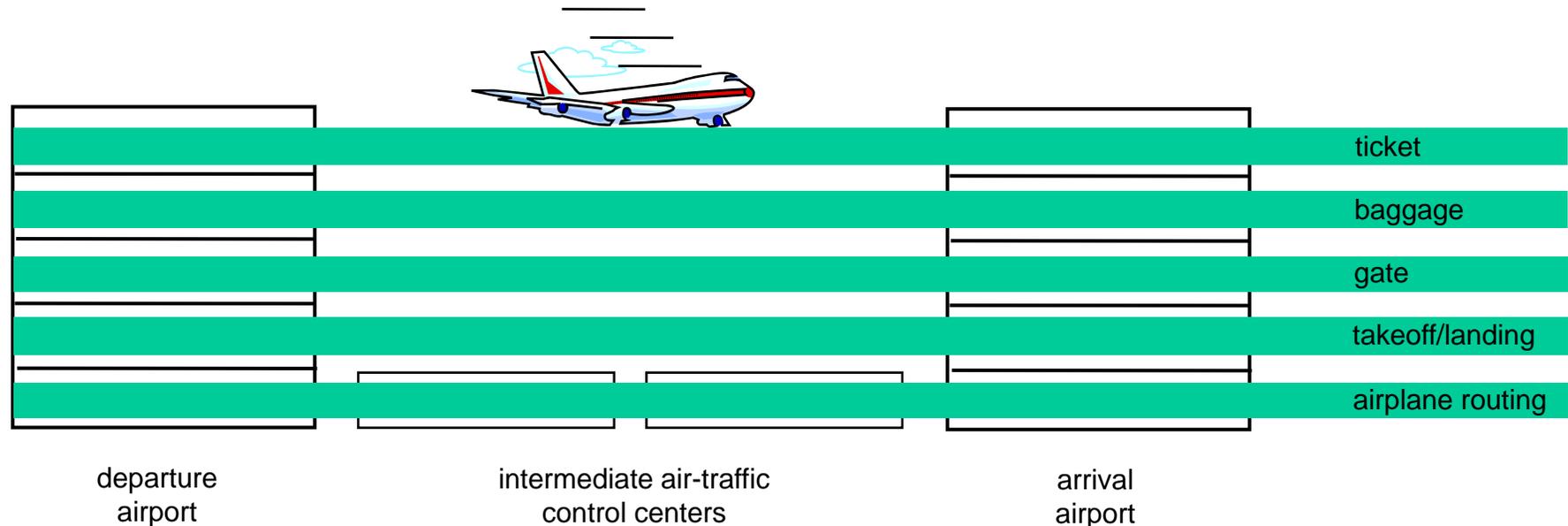
Or at least our discussion  
of networks?

# 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

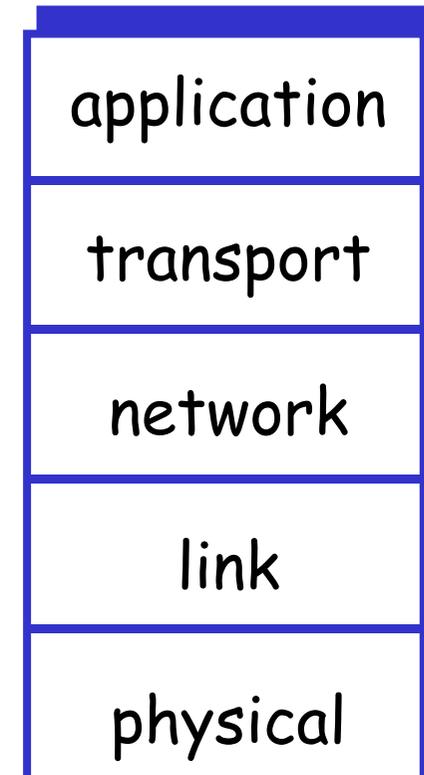
# Why layering?

## 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?

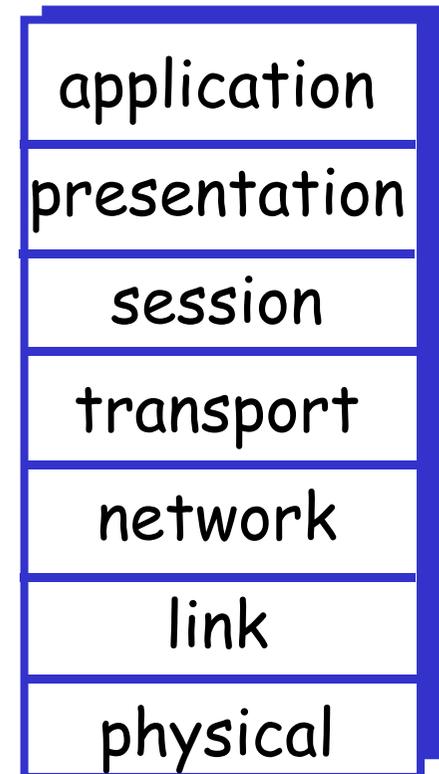
# Internet protocol stack

- **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"

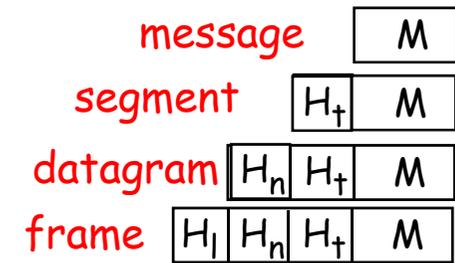


# ISO/OSI reference model

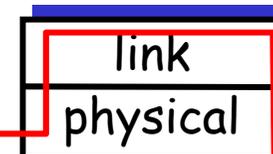
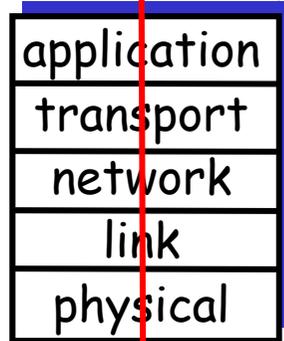
- **presentation**: allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- **session**: synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  - ❖ these services, *if needed*, must be implemented in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



# Encapsulation

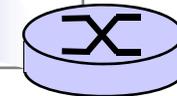
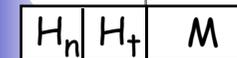
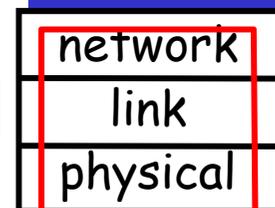
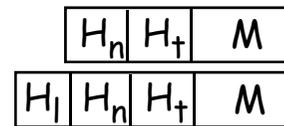
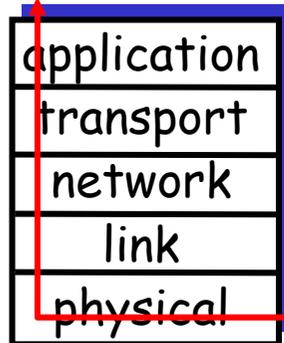
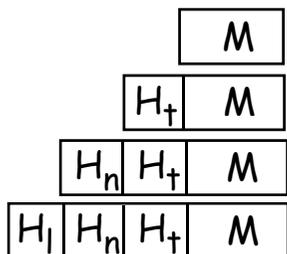


source



switch

destination



router

# 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 Networks under attack: security

1.7 History

# Network Security

- attacks on Internet infrastructure:
  - ❖ infecting/attacking hosts: malware, spyware, worms, unauthorized access (data stealing, user accounts)
  - ❖ denial of service: deny access to resources (servers, link bandwidth)
- Internet not originally designed with (much) security in mind
  - ❖ *original vision*: "a group of mutually trusting users attached to a transparent network" 😊
  - ❖ Internet protocol designers playing "catch-up"
  - ❖ Security considerations in all layers!

# What can bad guys do: malware?

## □ Spyware:

- ❖ infection by downloading web page with spyware
- ❖ records keystrokes, web sites visited, upload info to collection site

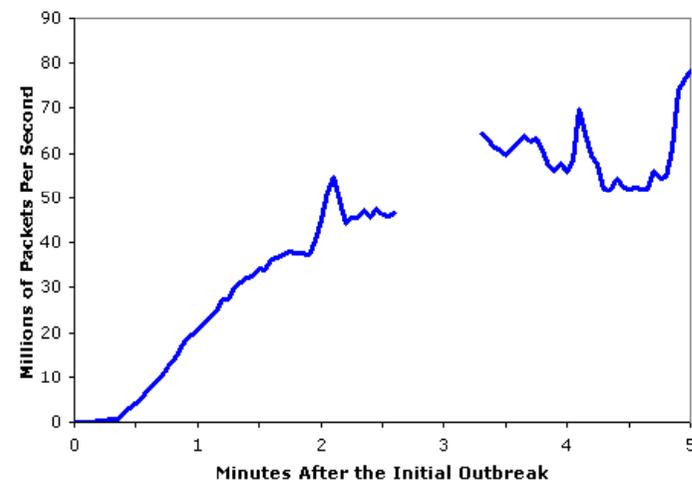
## □ Virus

- ❖ infection by receiving object (e.g., e-mail attachment), actively executing
- ❖ self-replicating: propagate itself to other hosts, users

## □ Worm:

- ❖ infection by passively receiving object that gets itself executed
- ❖ self-replicating: propagates to other hosts, users

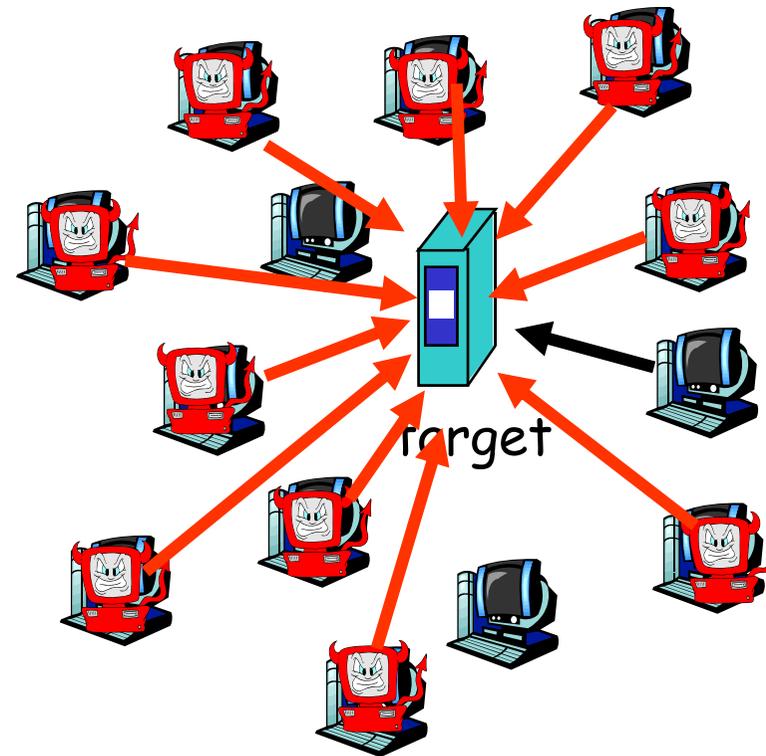
Sapphire Worm: aggregate scans/sec in first 5 minutes of outbreak (CAIDA, UWisc data)



# Denial of service attacks

- ❑ attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

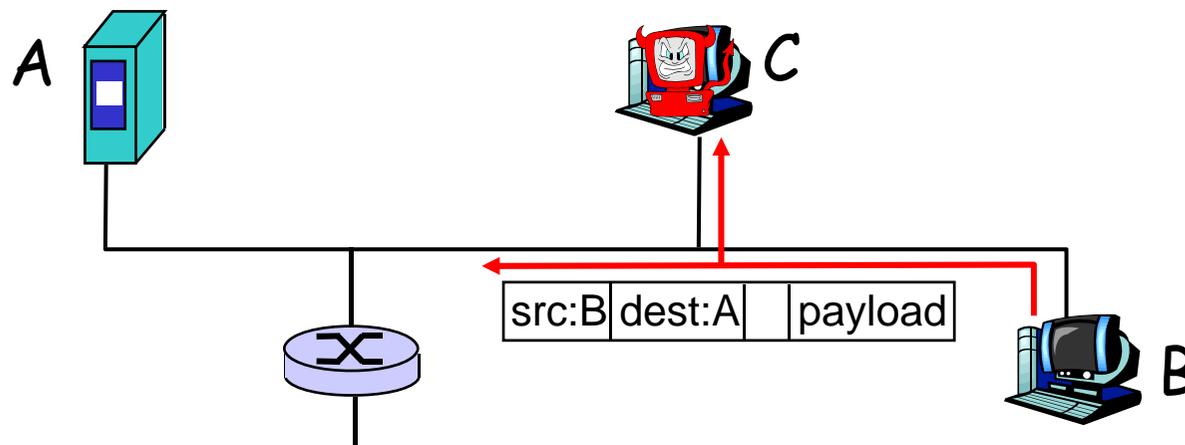
1. select target
2. break into hosts around the network (see malware)
3. send packets toward target from compromised hosts



# Sniff, modify, delete your packets

## *Packet sniffing:*

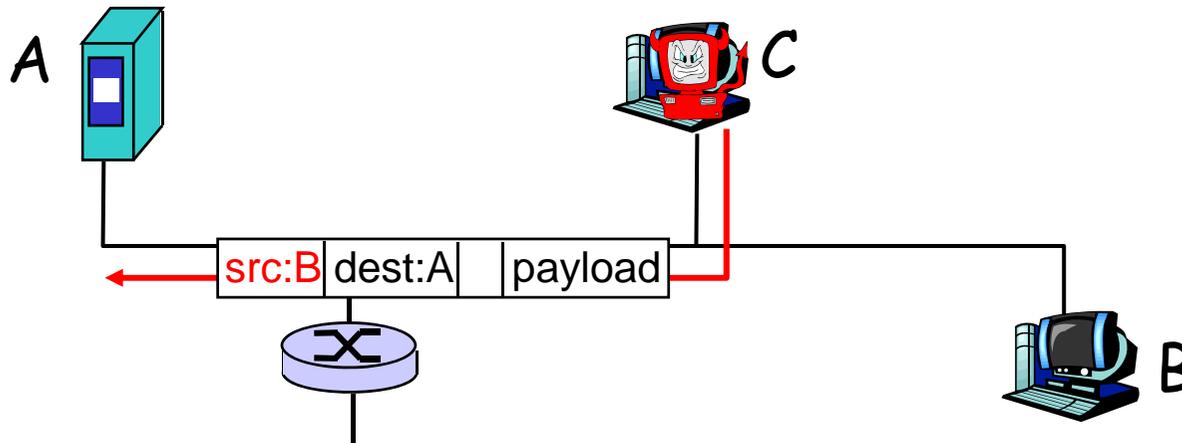
- ❖ broadcast media (shared Ethernet, wireless)
- ❖ promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- ❖ Ethereal software used for end-of-chapter labs is a (free) packet-sniffer
- ❖ more on modification, deletion later

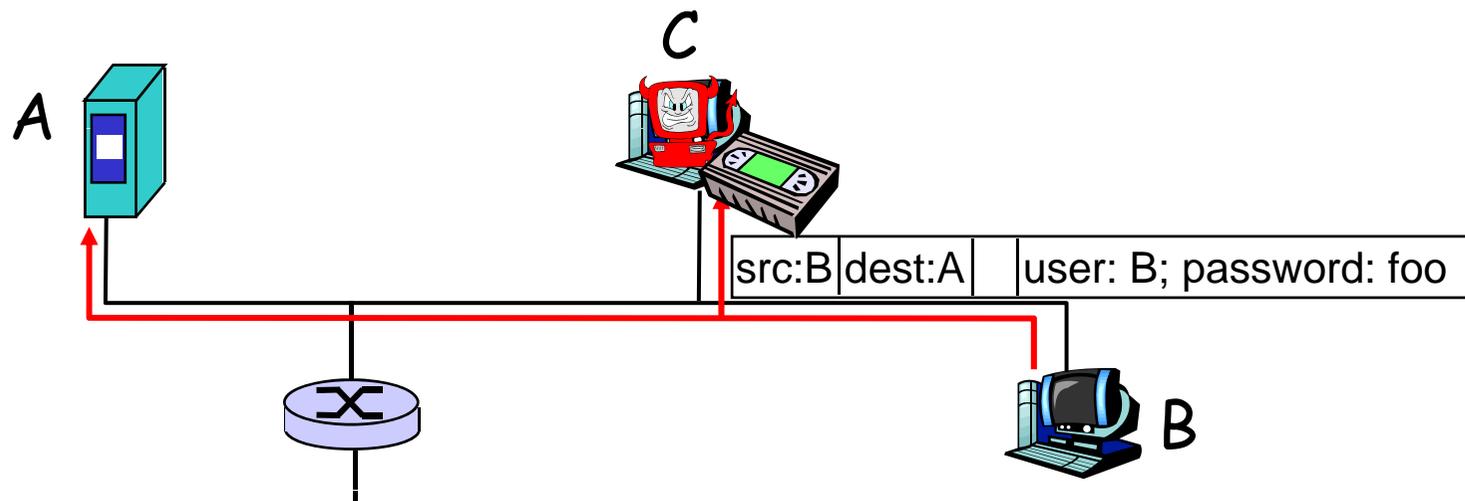
# Masquerade as you

- *IP spoofing*: send packet with false source address



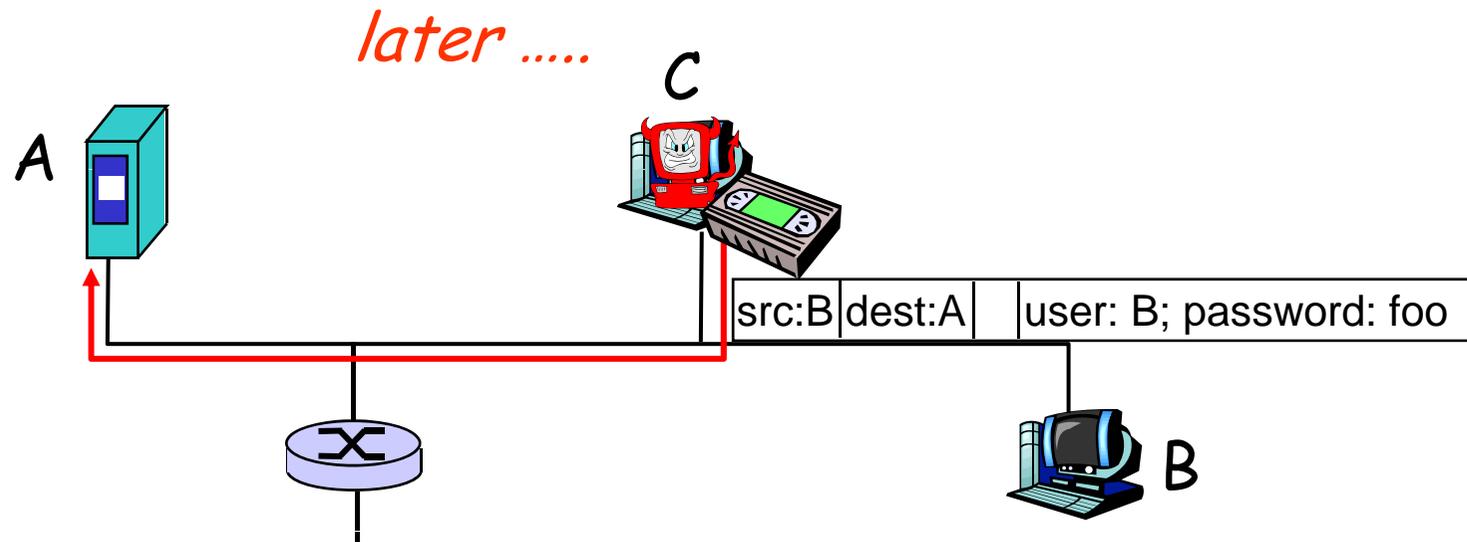
# Masquerade as you

- ❑ *IP spoofing*: send packet with false source address
- ❑ *record-and-playback*: sniff sensitive info (e.g., password), and use later
  - ❖ password holder *is* that user from system point of view



# Masquerade as you

- ❑ *IP spoofing*: send packet with false source address
- ❑ *record-and-playback*: sniff sensitive info (e.g., password), and use later
  - ❖ password holder *is* that user from system point of view



# Network Security

- more throughout this course
- chapter 8: focus on security
- cryptographic techniques: obvious uses and not so obvious uses

# 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

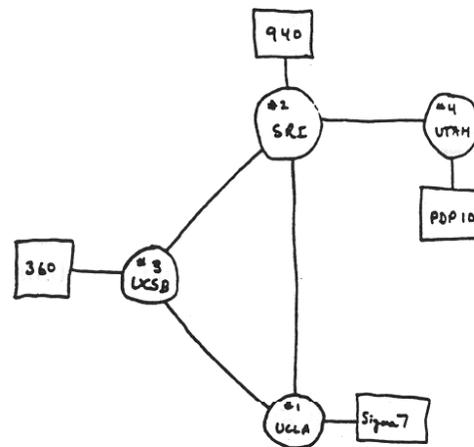
1.6 Networks under attack: security

1.7 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: Cset, 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

2007:

- ❑ ~500 million hosts
- ❑ Voice, Video over IP
- ❑ P2P applications: BitTorrent (file sharing) Skype (VoIP), PPLive (video)
- ❑ more applications: YouTube, gaming
- ❑ wireless, mobility

# 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!*