ECPE / COMP 177
Fall 2011

Computer Networking

Some slides from Kurose and Ross, *Computer Networking*, 5^{th} Edition

Logistics

- Instructor: Dr. Jeff Shafer
  - Email: jshafer at pacific dot edu
  - Office: Anderson 205
  - Office hours *(posted on my door)*
    - Mon: 10:30-11:30pm
    - Tue: 1:30-3:00pm
    - Wed: 1:30-3:00pm
    - *... plus whenever my office door is open*

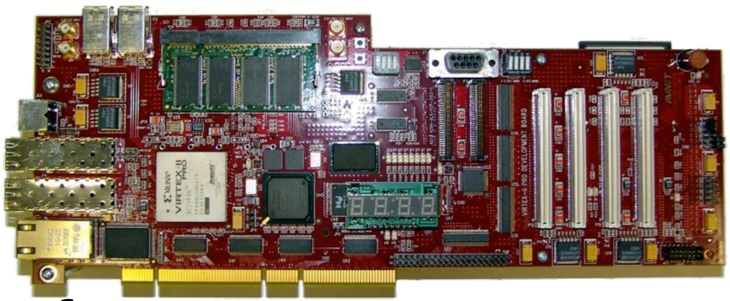
About Jeff

- Graduated from Rice University in May 2010
- Research areas
  - Cloud computing / storage architecture
  - Network Systems Architecture

Network Systems Architecture

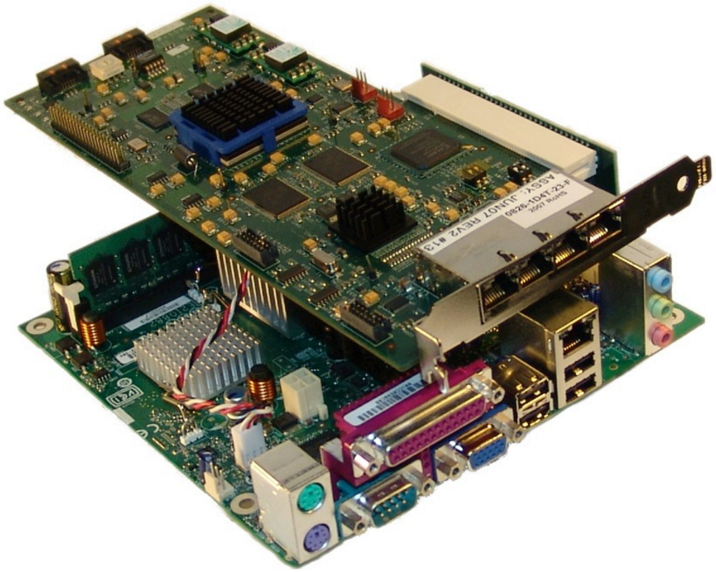
- Specialized systems for networking
  - Network interfaces
  - Switches
  - Routers
  - …
- Typically a combination of software control and special-purpose hardware for packet processing
  - Network processors
  - Lightweight microprocessors with custom hardware

RiceNIC – Prototyping NICs



- Platform for research and education
- Commercial FPGA board with custom VHDL / C
- Full control over hardware and software
  - Reconfigurable - FPGAs
  - Programmable - Embedded processors
- Integrated software and hardware debugging features
- High performance – Ethernet line rate
- Reference design is freely available, and in-use around the world

Overcome Network Switch Bottlenecks: Axon Network Device



- Datacenter network technology
- Replaces existing switches and routers
- Presents abstraction that entire network is one large Ethernet segment
  - Virtual machine migration
- Transparent source-routed Ethernet
  - Improves performance - No spanning tree
  - Improves scalability
    - Stores state at network edge

Logistics

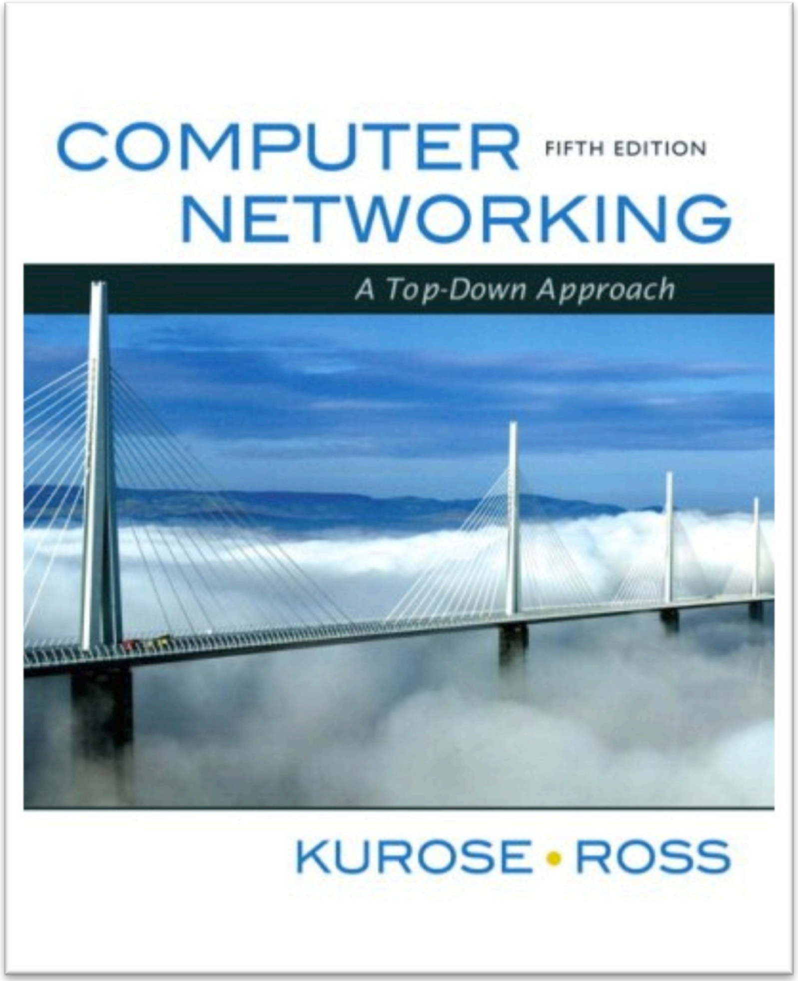
- Lecture
  - When: Tuesday / Thursday, 10am-11:45am
  - Where: Chambers 115
- Lab
  - When: Thursday, 2-4:50pm
  - Where: Baun 212
  - Lab start date: Thur Sept 8th
    - No lab this Thursday

Logistics

- Course websites:
  - http://ecs-network.serv.pacific.edu/ecpe-177
    - Slides, syllabus, schedule, assignments, and more
  - http://pacific.rsmart.com/
    - Sakai for assignment submission and emails only
    - Should auto-signup if enrolled in course

Logistics

- Book
  - *Computer Networking: A Top-Down Approach*
  - 5th edition
  - Kurose and Ross
- Not required



Pre-Requisites

- Official pre-req
  - COMP 53 – Data structures
    - High level language such as C
    - Basic data structures, arrays, **pointers**, functions, system calls, …
  - ECPE 170 – Computer Systems and Networks
- Unofficial pre-reqs
  - Number systems other than decimal (base-2, base-16)
  - Digital electronics
  - Basic computer skills
    - Windows at the command line
    - Linux at the command line

Course Format

- Homework – 18%
  - Six assignments throughout semester
- Labs – 12%
  - Eleven labs in Baun 212
  - Applying theoretical concepts to real-world network equipment (Cisco routers and switches)
- Exams
  - Mid-term written exam – 10%
  - Final written exam – 10%
  - Lab practical exam – 10%

Course Format

- Projects – 40%
  - 3 programming projects
  - Individually or groups of 2
  - Past projects include web (HTTP) download client and web proxy server
  - Implementation language: C with standard Unix sockets
  - Implementation platform: Linux
- Tutorials will be provided
  - Writing / compiling / debugging programs on Linux

Questions?

? ? ?

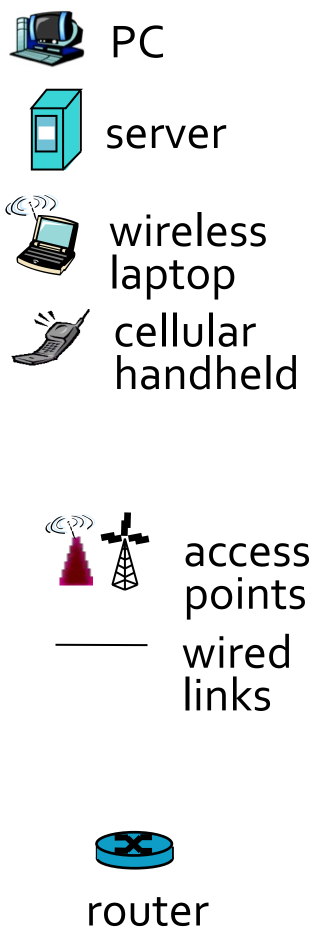
Intro to Networking

- What is the Internet?
- Network edge
  - End systems, access networks, links
- Network core
  - Circuit switching, packet switching, network structure
- Performance: Delay, loss and throughput in packet-switched networks
- Protocol layers, service models
- Networks under attack: security

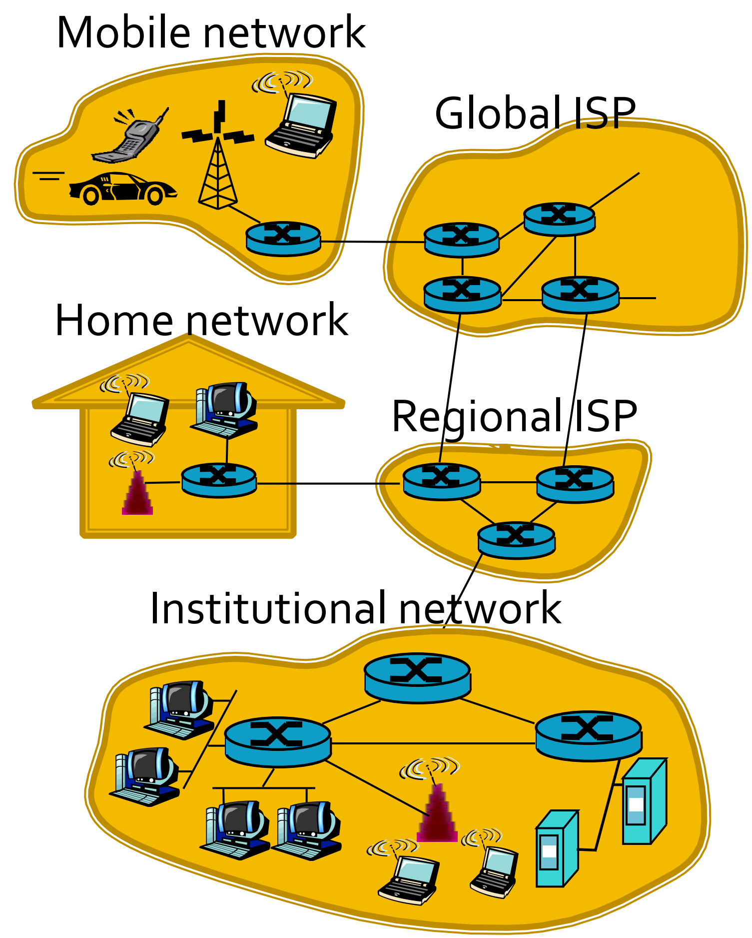
Networks are Ubiquitous

- What good is a computer when the network is down?
  - *I just keep hitting refresh on my web browser until something happens…*
- What good is my iPhone with no AT&T / Verizon service?
- What good is a TV without on-demand Netflix streaming?

What's the Internet: High Level View

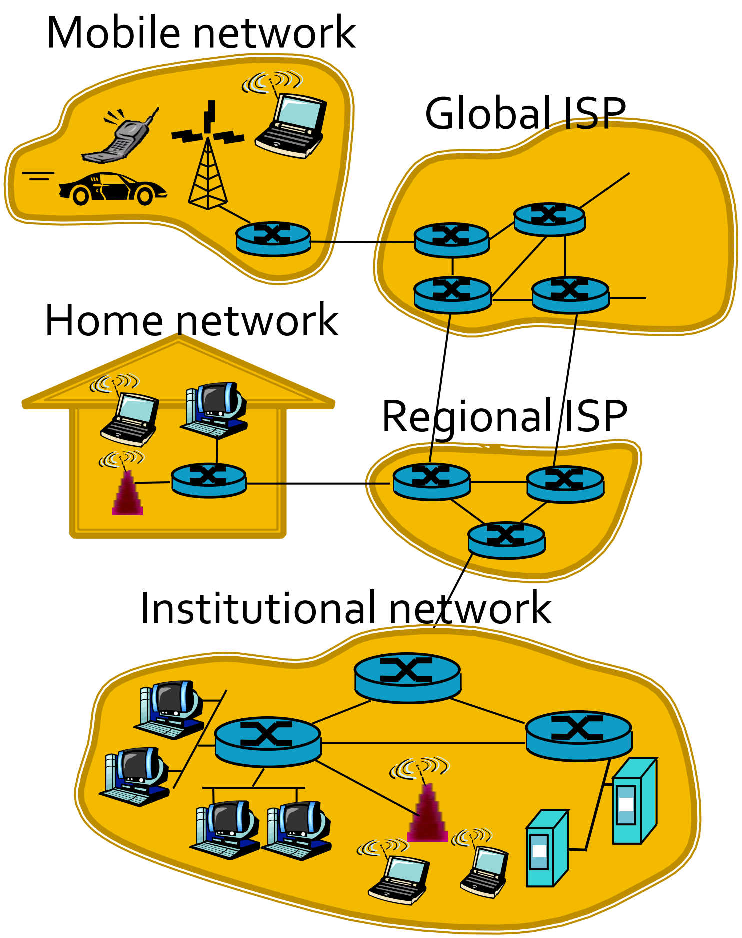


- **Hosts (end systems)**
  - Millions of connected computing devices
  - Running network apps
- **Communication links**
  - Fiber, copper, radio, satellite
  - Transmission rate = bandwidth
- **Routers**
  - Forward packets (chunks of data) between links



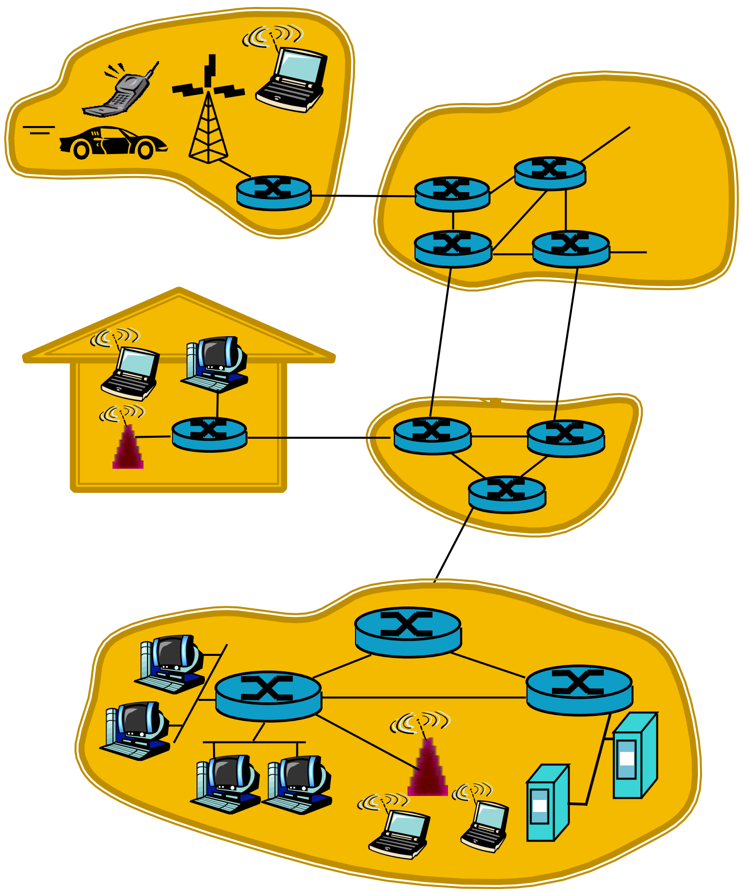
What's the Internet: High Level View

- **Protocols**
  - Control sending and receiving of messages
  - e.g., TCP, IP, HTTP, Skype, Ethernet
- Internet standards
  - Who makes (some of) the protocols?
  - IETF: Internet Engineering Task Force
  - RFC: Request for comments
- **Internet**: "network of networks"
  - Loosely hierarchical
  - Public *Internet* versus private *intranet*



What's the Internet: 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, *or*
  - "Best effort" (unreliable) data delivery



What's a Protocol?

HUMAN PROTOCOLS

- "What's the time?"
- "I have a question"
- Introductions

NETWORK PROTOCOLS

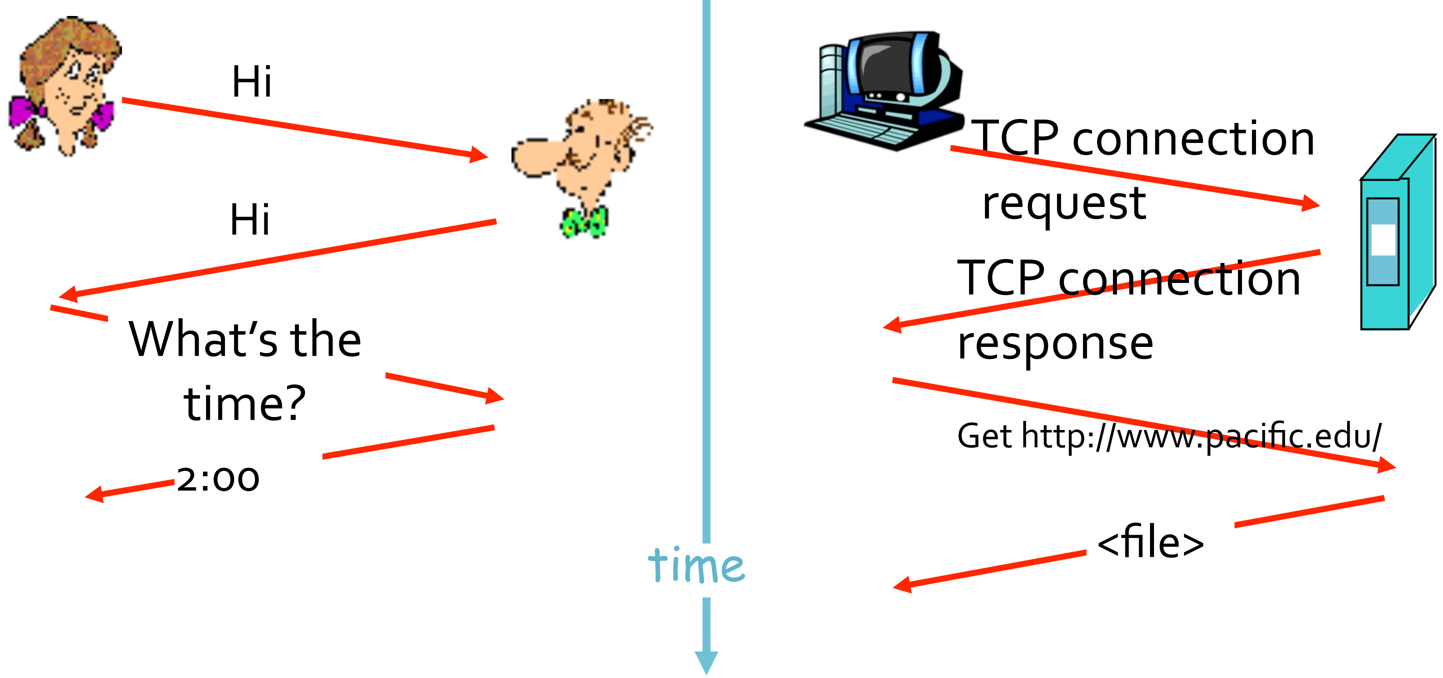
- Machines rather than humans
- All communication activity in Internet governed by protocols

- **Protocols (human and computer!) define**
  - Format of message
  - Order of messages sent/received on network
  - Actions taken after sending/receiving message

What's a Protocol?

HUMAN PROTOCOL

COMPUTER NETWORK PROTOCOL

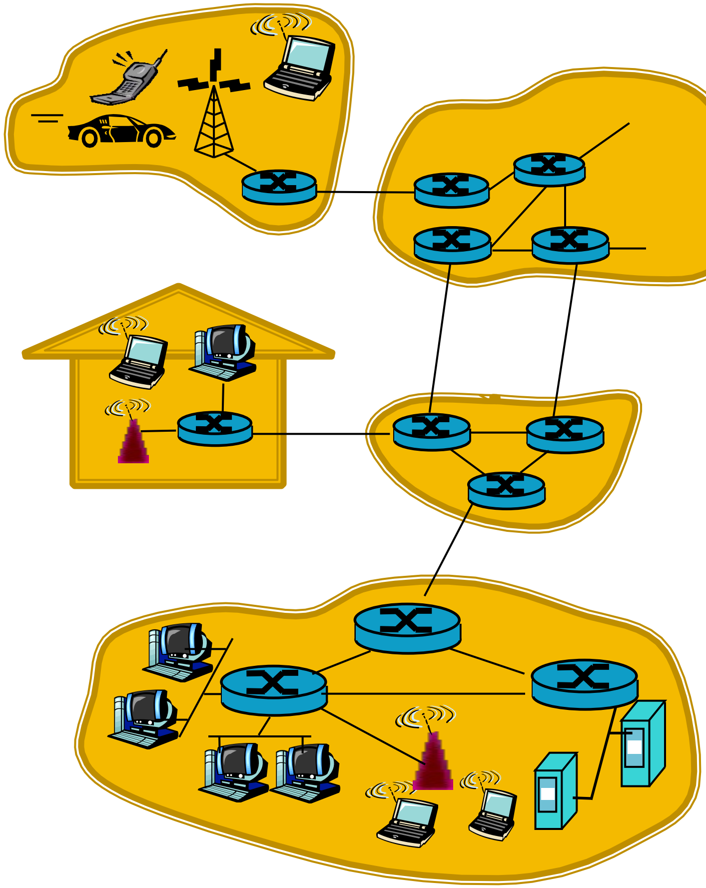


Intro to Networking

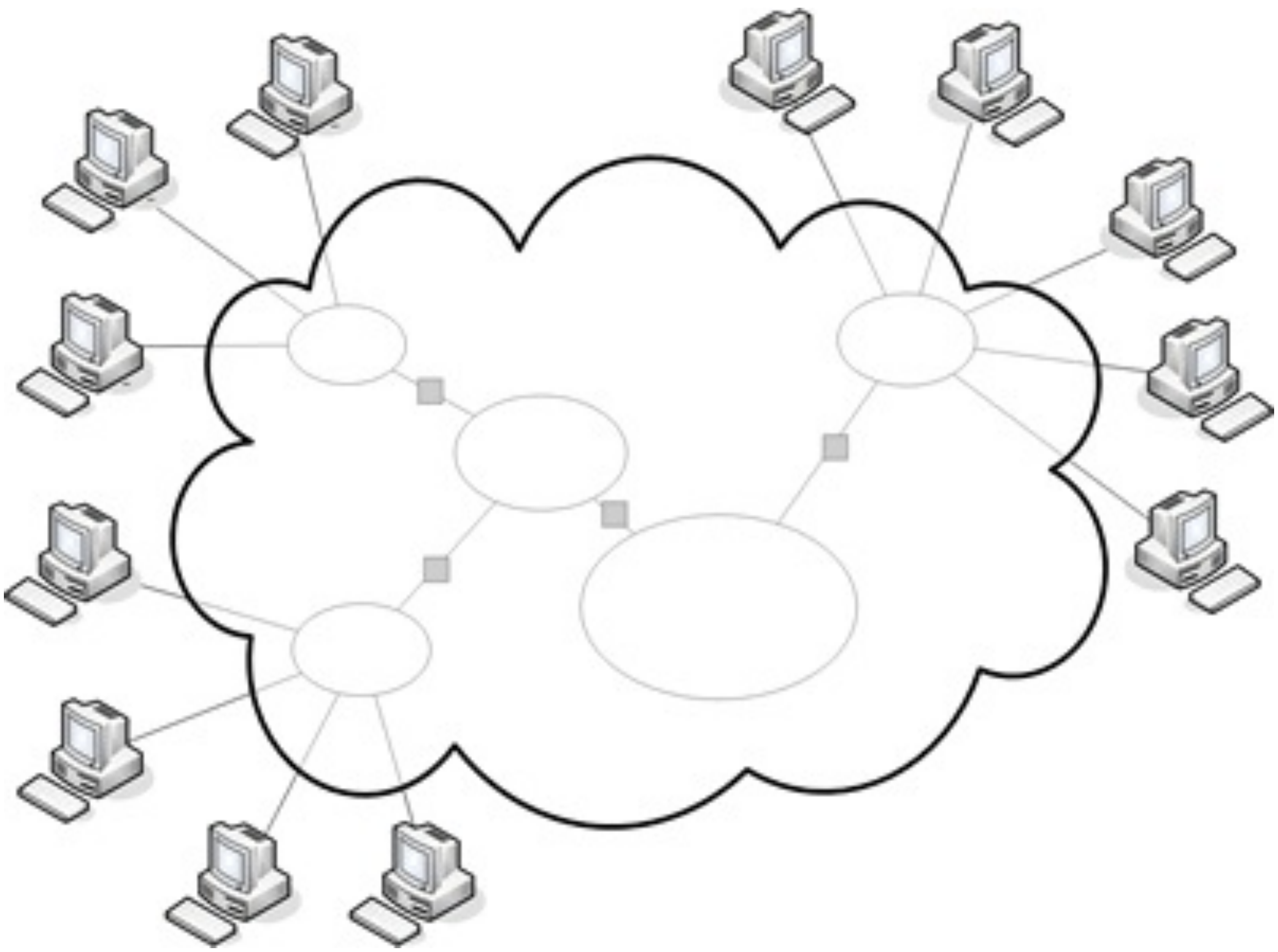
- What is the Internet?
- Network edge
  - End systems, access networks, links
- Network core
  - Circuit switching, packet switching, network structure
- Performance: Delay, loss and throughput in packet-switched networks
- Protocol layers, service models
- Networks under attack: security

A Closer Look at Network Structure

- **Network edge**
  - Applications and hosts
- **Access networks and physical media**
  - Wired, wireless communication links
- **Network core**
  - Interconnected routers
  - Network of networks

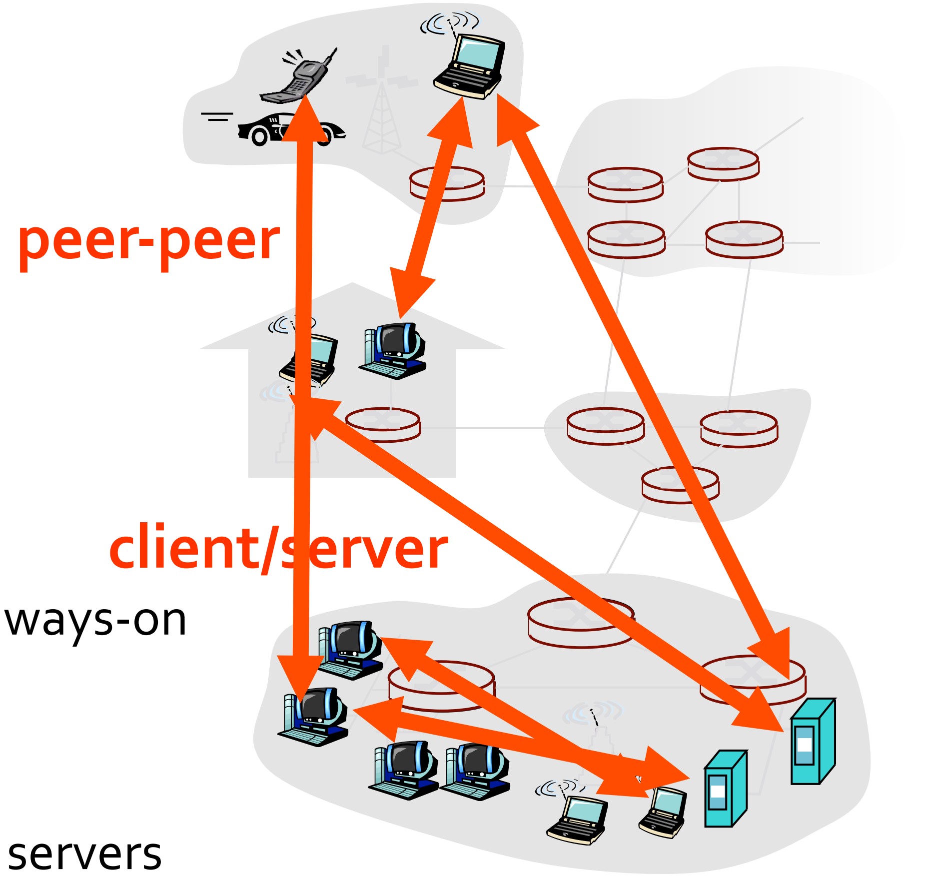


Why is it Called the Edge?



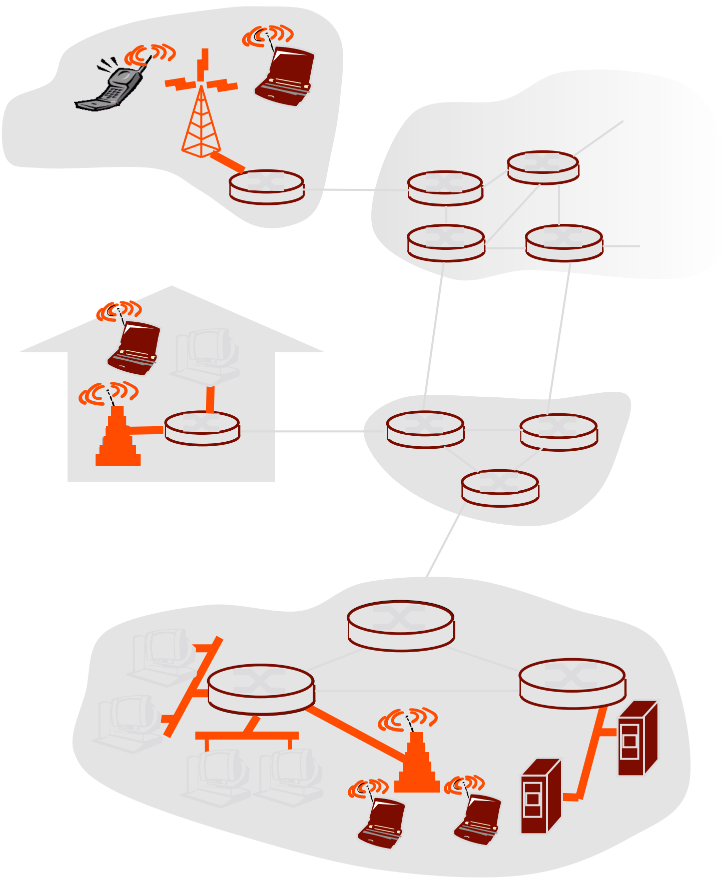
The Network Edge

- **End systems (hosts) at edge**
  - Run application programs
- Two models of applications
  - Client/server
  - Peer-to-Peer (P2P)
  - **What's the difference?**
- **Client/server model**
  - Client host requests data from always-on server (e.g. web, email, …)
- **Peer-to-peer model**
  - Minimal (or no) use of dedicated servers (e.g. Skype, BitTorrent)



Access Networks + Physical Links

- **How do you connect hosts to the nearest edge router?**
  - Residential access network
  - Institutional access networks (school, company)
  - Mobile access networks
- Concerns
  - Bandwidth (bits per second) of access network
  - **Other concerns?**
    - Shared or dedicated?
    - Cost?
    - Reliability?
    - Blocking / filtering?



Common Access Networks

- Dial-up modem
- Digital Subscriber Line (DSL)
- Cable Modem
- Fiber to the home
- Ethernet
- Wireless LAN
- Wide-area wireless

- **What do you use?**

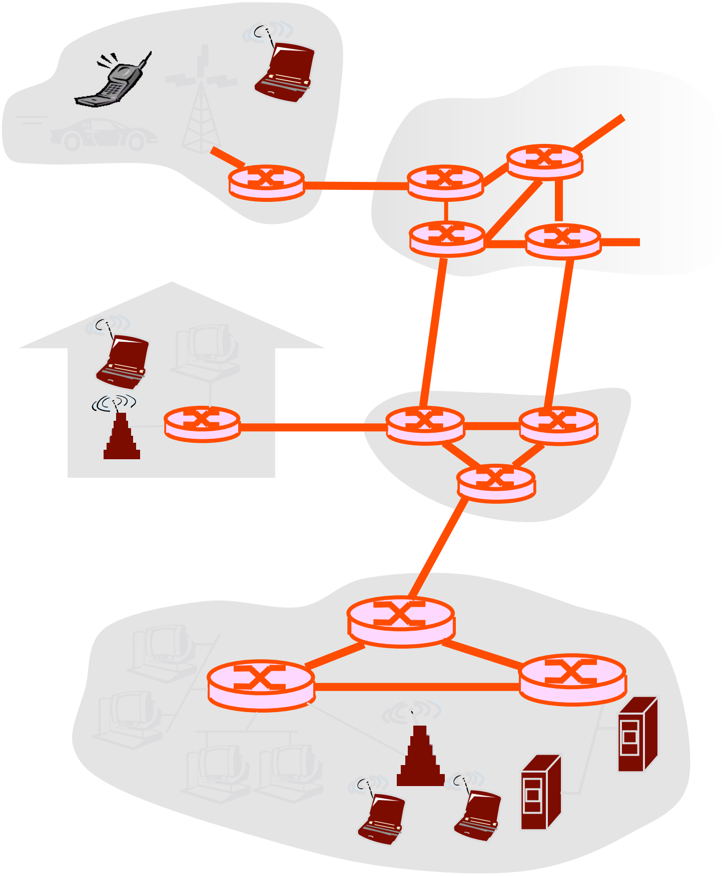
Intro to Networking

- What is the Internet?
- Network edge
  - End systems, access networks, links
- Network core
  - Circuit switching, packet switching, network structure
- Performance: Delay, loss and throughput in packet-switched networks
- Protocol layers, service models
- Networks under attack: security

The Network Core

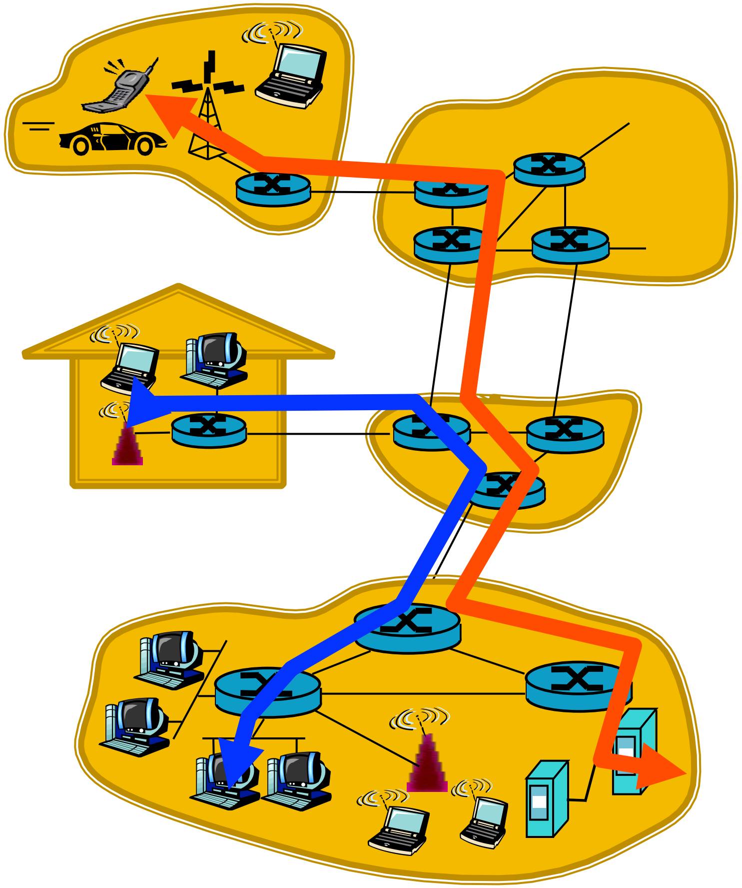
- Mesh of interconnected routers
- Fundamental question: **how is data transferred through mesh?**
  - **Circuit switching**
    - Dedicated circuit per call
    - "classic" telephone network
  - **Packet-switching:**
    - Data sent thru mesh in discrete "chunks"

- **Which method is used in the Internet?**
  - Packet switching at the highest layer
  - Some circuit switching (older) underneath…



Network Core: Circuit Switching

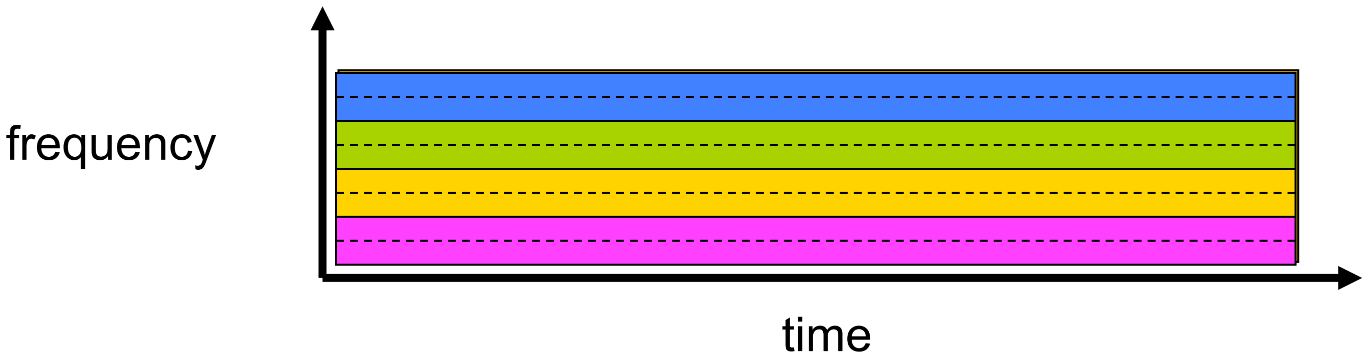
- End-end resources reserved
  - No sharing
  - Fixed link bandwidth and switch capacity
  - Guaranteed performance
  - Link must be setup before use
- Network resources (e.g., bandwidth) divided into "pieces"
  - Pieces allocated to use
  - Piece idle if not used by user (no sharing)
  - Frequency division or time division



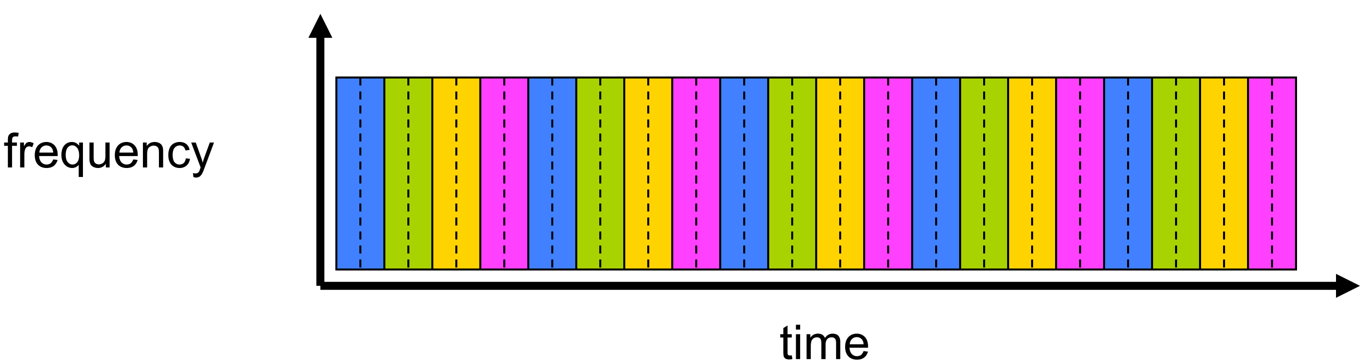
Circuit Switching: FDM and TDM

Example:
4 users

FDM



TDM



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

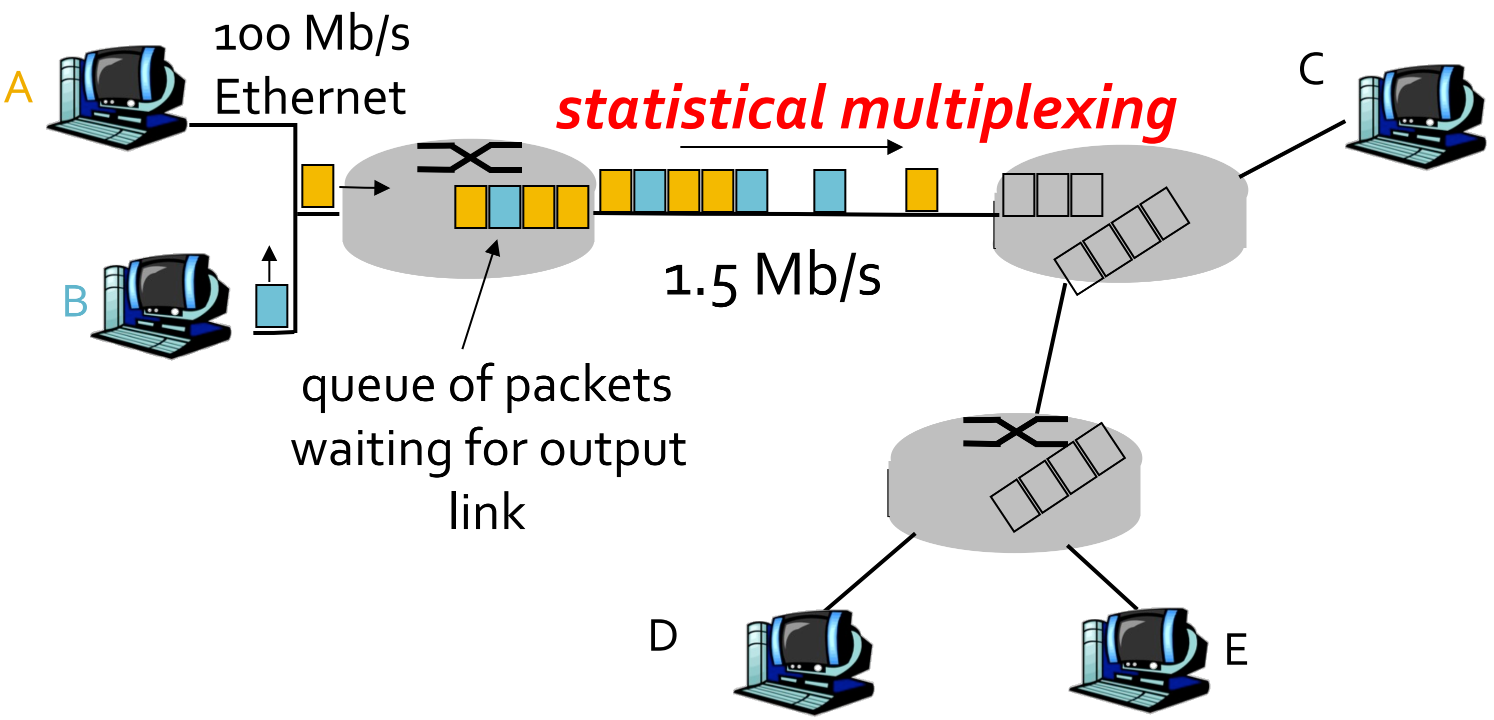
Bandwidth division into "pieces"

Dedicated allocation

Resource reservation

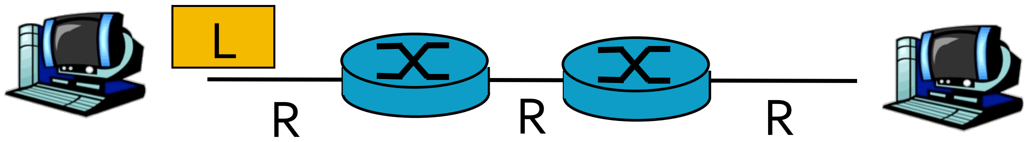
- Resource contention
  - Aggregate resource demand can exceed amount available
  - Congestion: packets must wait in queue
- Store and forward: packets move one hop at a time
  - Receive complete packet before forwarding

Packet Switching: Statistical Multiplexing



- Sequence of A & B packets does not have fixed pattern, bandwidth shared on demand ➡ ***statistical multiplexing***.
- Contrast against circuit switching / time-division multiplexing
  - Each host gets same slot (fixed pattern)

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
- Total delay to send message?
  - Delay = 3L/R
  - Assuming zero propagation delay

more on delay shortly …

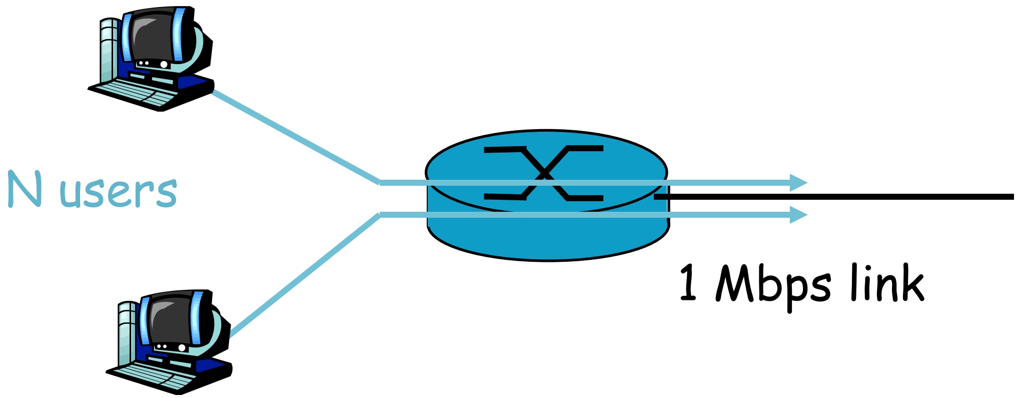
Example:

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

Packet Switching vs 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 max
- **Packet switching:**
  - With 35 users, probability > 10 active at same time is less than .0004



Packet Switching vs Circuit Switching

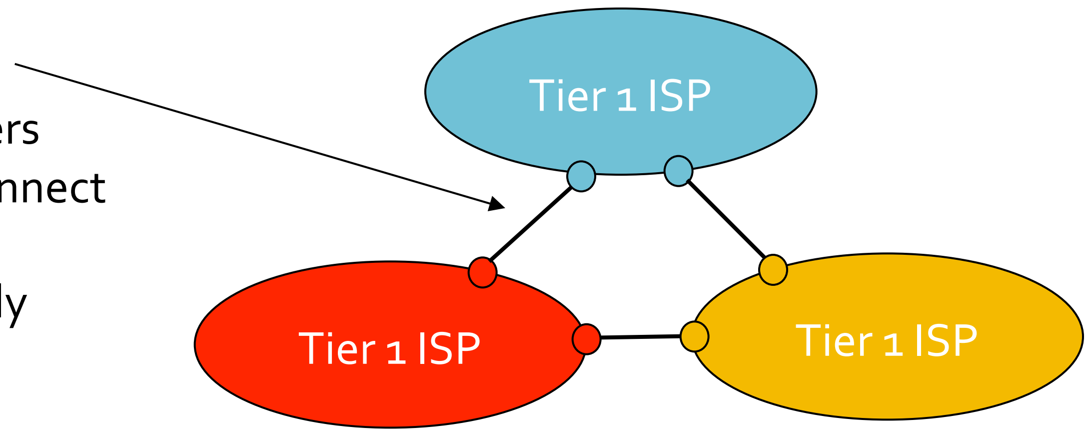
- **Is packet switching perfect in all situations?**
  - (Think about your own experiences)

- Great for bursty data
  - Resource sharing
  - Simpler, no call setup
- Less great excessive congestion: packet delay and loss
  - Protocols needed for reliable data transfer and congestion control
- Some applications really want circuit-like behavior
  - Streaming video, streaming audio, interactive games, …
    - If streaming video data arrives late, it is useless
  - Bandwidth / latency (delay) guarantees needed
    - Still an unsolved problem!

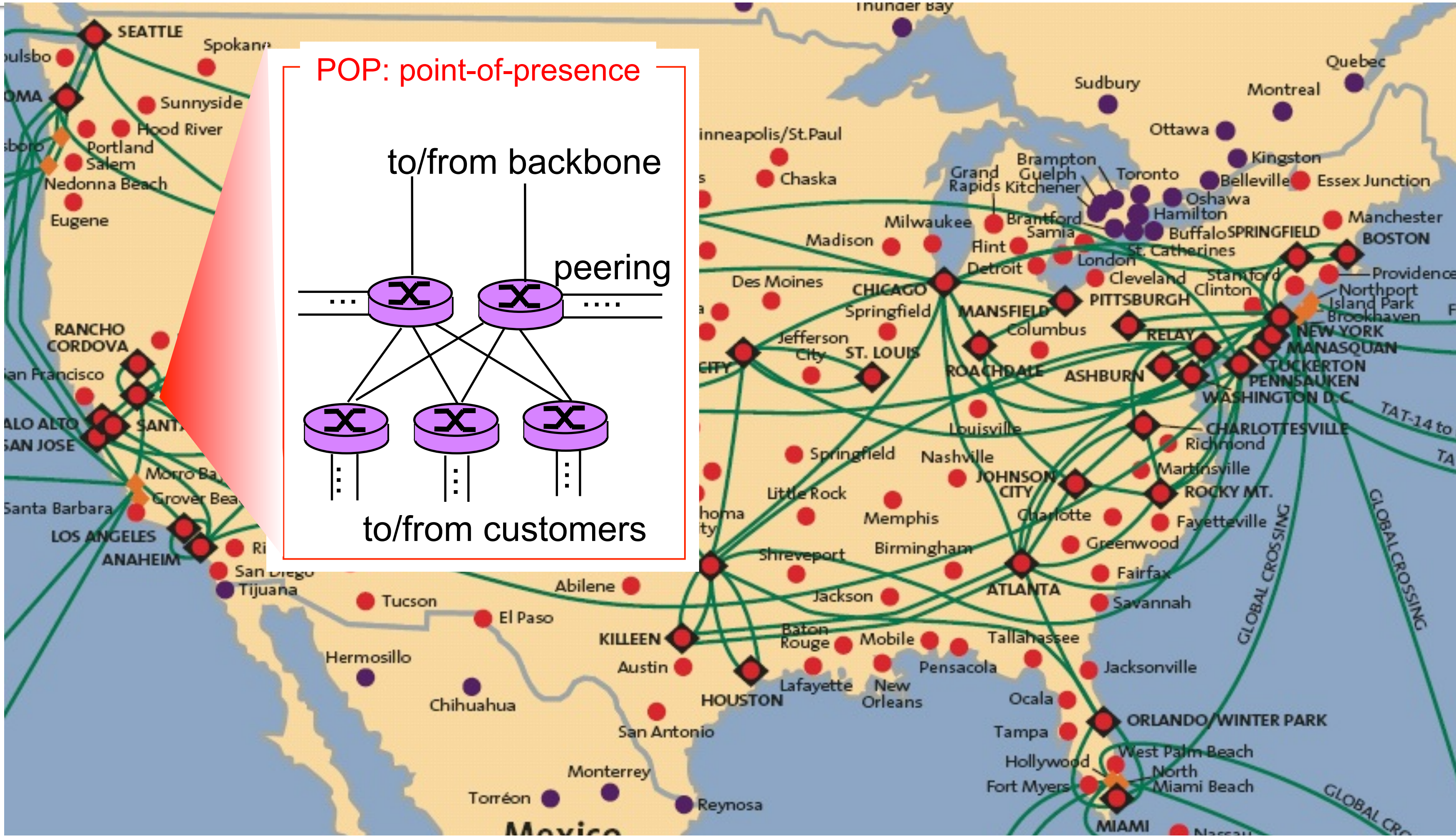
Internet Structure: Network of Networks

- Roughly hierarchical
- At center: **"tier-1" ISPs** with national/international coverage
  - Treat each other as equals
  - Examples: Qwest, Sprint, NTT, L3, AT&T…

Tier-1 providers interconnect (peer) privately



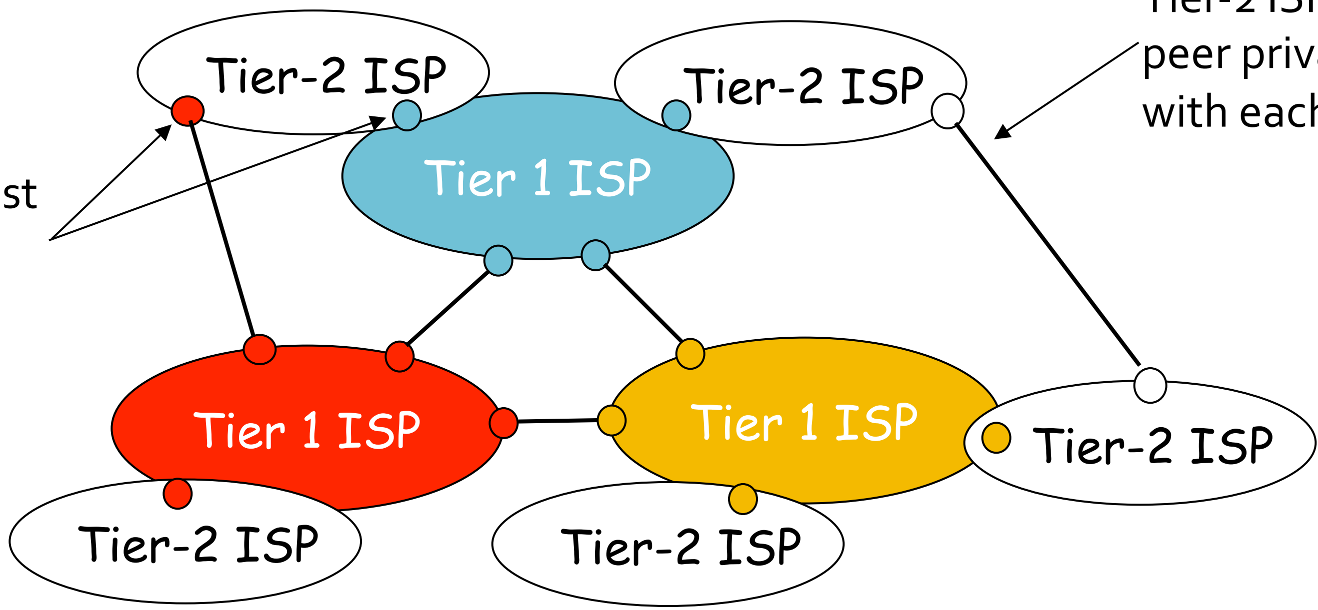
Tier-1 ISP: e.g., Sprint



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 (they are a *customer* of the Tier-1 provider

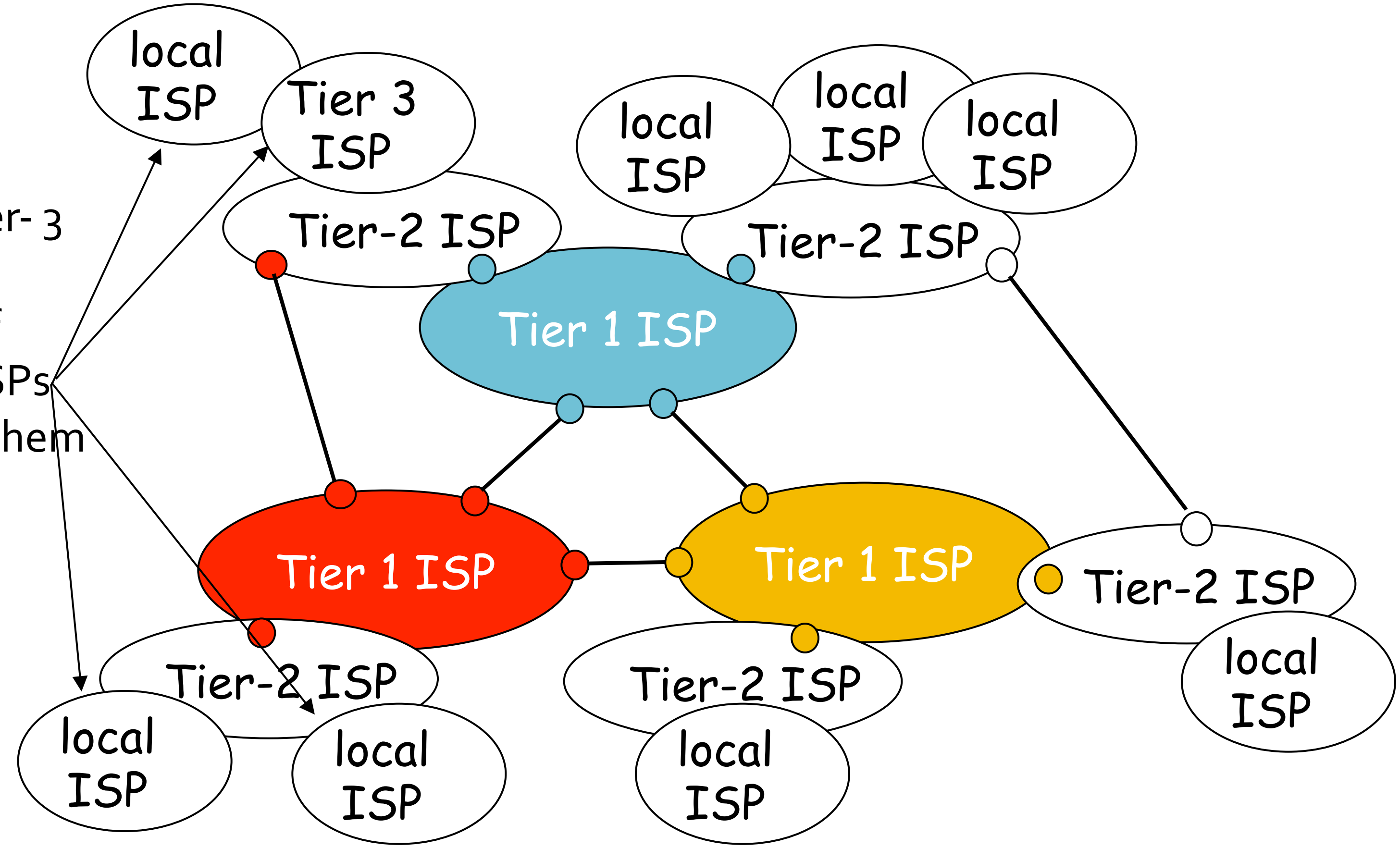


Tier-2 ISPs also peer privately with each other.

Internet Structure: Network of Networks

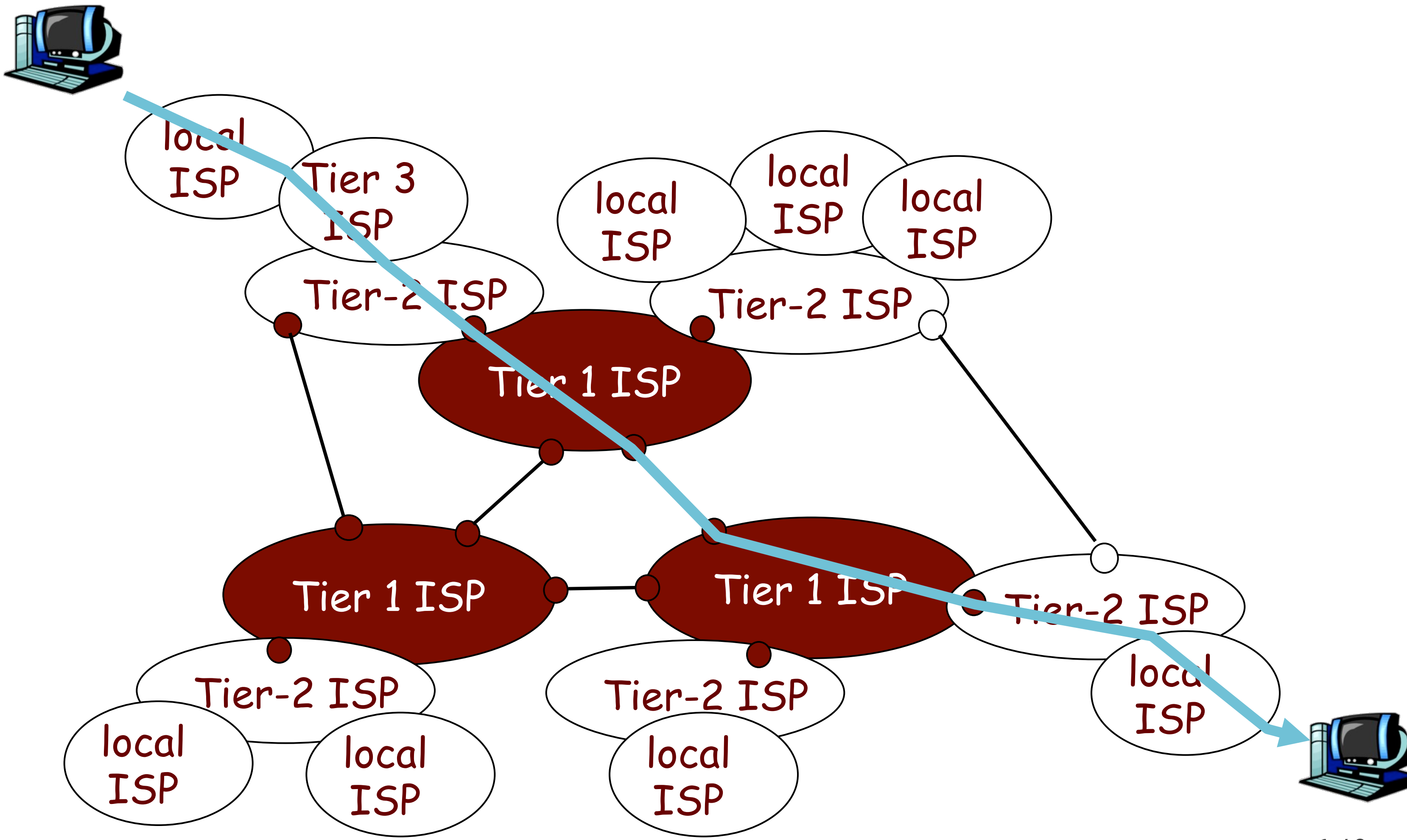
- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)

Local and tier- 3 ISPs are *customers* of higher tier ISPs connecting them to rest of Internet



Internet Structure: Network of Networks

- A packet passes through many networks



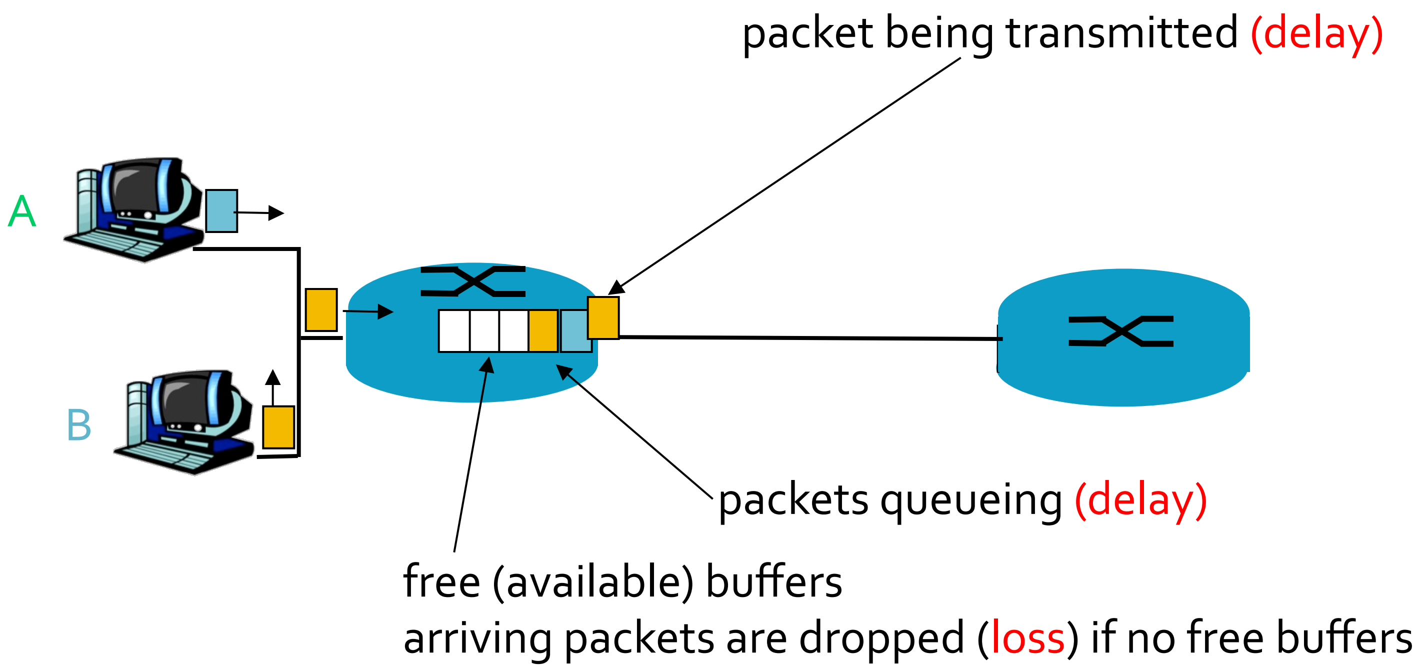
Intro to Networking

- What is the Internet?
- Network edge
  - End systems, access networks, links
- Network core
  - Circuit switching, packet switching, network structure
- Performance: Delay, loss and throughput in packet-switched networks
- Protocol layers, service models
- Networks under attack: security

How do Loss and Delay occur?

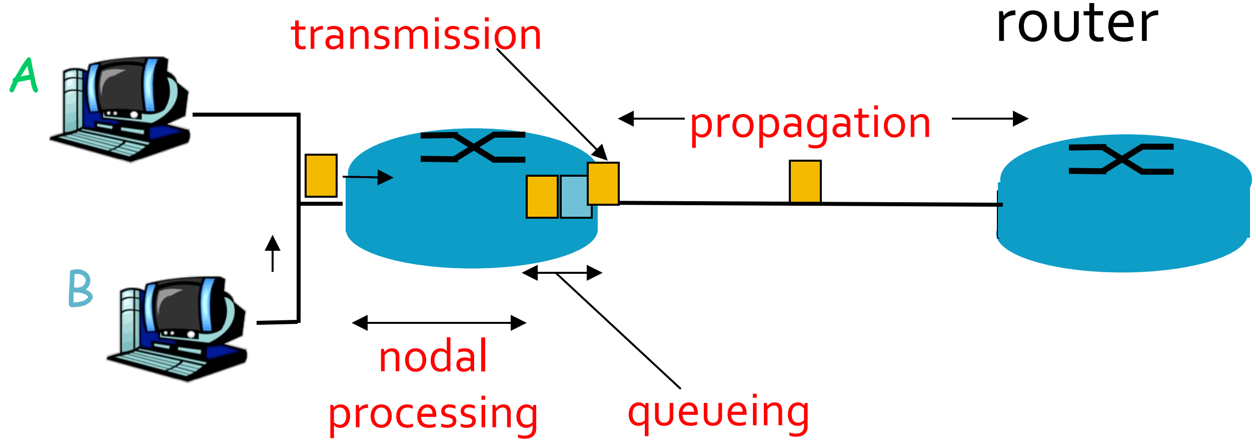
Packets *queue* in router buffers

- Packet arrival rate to link exceeds output link capacity
- Packets queue and wait for turn



Four Sources of Packet Delay

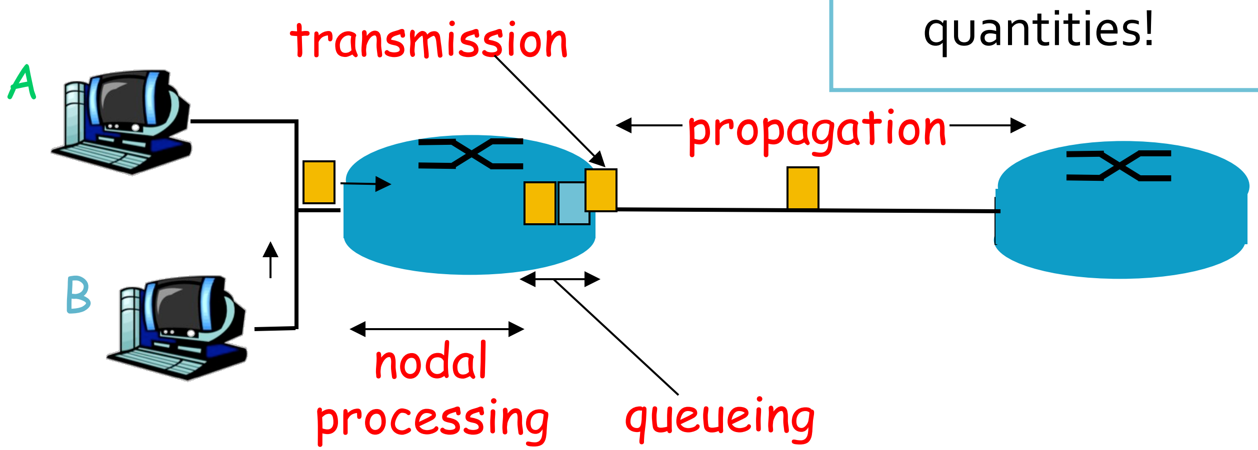
- 1. Node processing:
  - Check bit errors
  - Determine output link
  - Fixed time
- 2. Queueing
  - Time waiting at output link for transmission
  - Variable time: depends on congestion level of router



Four Sources of Packet Delay

- 3. Transmission delay:
  - L=packet length (bits)
  - R=link bandwidth (bps)
  - time to send bits into link = L/R
  - Time varies by packet size
- 4. Propagation delay:
  - d = length of physical link
  - s = propagation speed in medium (~2×10^8 m/sec)
  - propagation delay = d/s

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

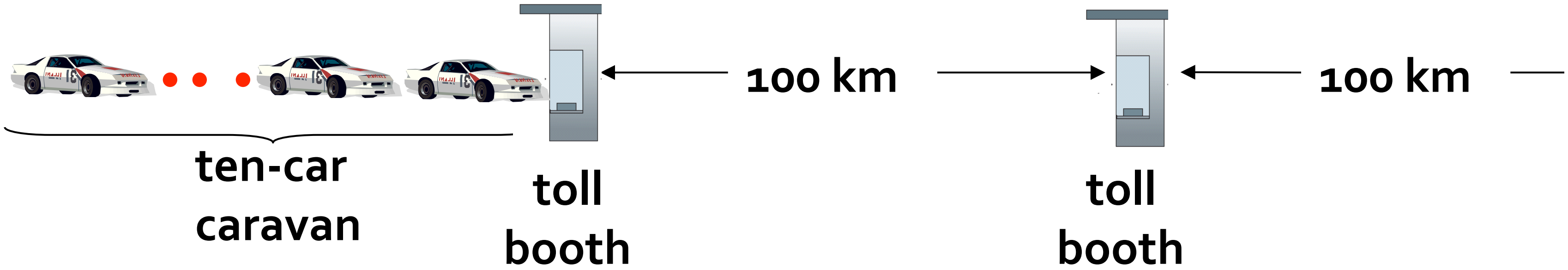


Nodal Delay

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

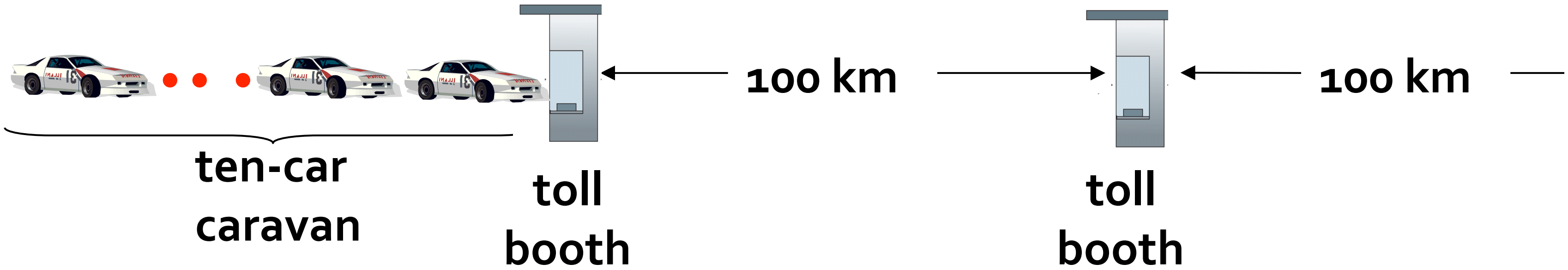
- d_{proc} = processing delay
  - typically a few microsecs or less
- d_{queue} = queuing delay
  - depends on congestion
- d_{trans} = transmission delay
  - = L/R, significant for low-speed links
- d_{prop} = propagation delay
  - a few microsecs to hundreds of msecs

Caravan Analogy



- Cars "propagate" at 100 km/hr
- Toll booth takes 12 sec to service car (transmission time)
- Car~bit; Caravan~packet
- **How long until end of caravan reaches 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
- **62 minutes**

Caravan Analogy (cont.)

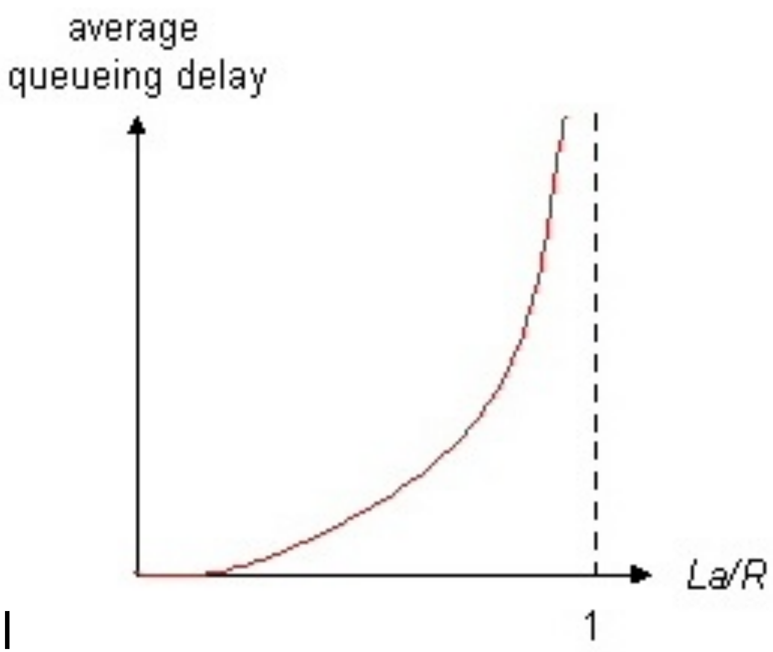


- Cars now "propagate" at 1000 km/hr
- Toll booth now takes 1 min to service a car
- **Will cars arrive to 2nd booth before all cars serviced at 1st booth?**

- 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!

Queueing Delay (revisited)

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

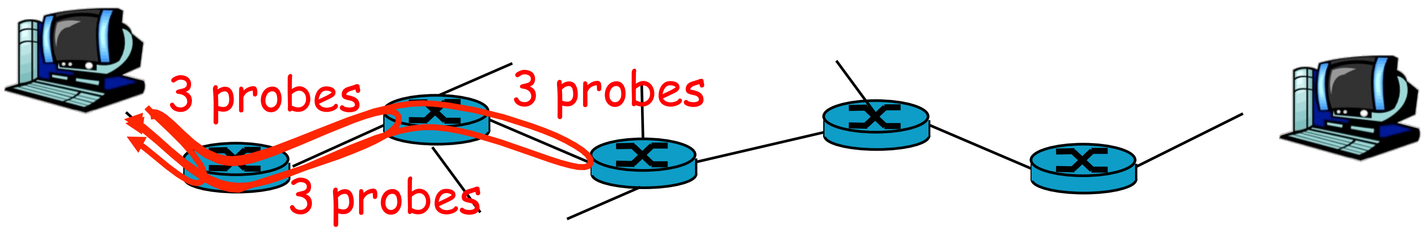


traffic intensity = La/R

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

"Real" Internet Delays and Routes

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



“Real” Internet Delays and Routes

traceroute: my laptop @ pacific to www.msu.ru

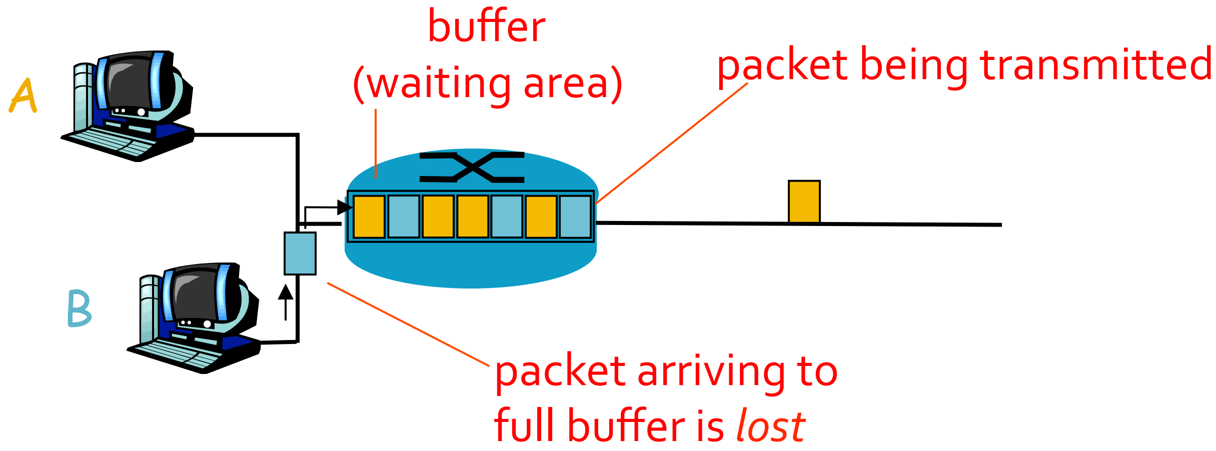
Three delay measurements

```
dhcp-10-10-207-20:~ shafer$ traceroute -a www.msu.ru
traceroute to www.msu.ru (193.232.113.151), 64 hops max, 52 byte packets
 1  [AS0] 138.9.253.252 (138.9.253.252)  0.740 ms  0.741 ms  1.290 ms
 2  [AS0] 74.202.6.5 (74.202.6.5)  5.245 ms  15.006 ms  5.142 ms
 3  [AS4323] sjc1-pr1-xe-0-0-0-0.us.twtelecom.net (66.192.251.170)  6.414 ms  6.640 ms  17.283 ms
 4  [AS6453] if-10-0-0-56.core3.sqn-sanjose.as6453.net (209.58.116.50)  6.628 ms *
    [AS6453] if-13-0-0-55.core3.sqn-sanjose.as6453.net (66.198.97.9)  7.056 ms
 5  [AS6453] if-9-0-0.mcore4.pdi-paloalto.as6453.net (216.6.33.6)  68.184 ms
    [AS6453] if-6-0-0-1145.mcore4.pdi-paloalto.as6453.net (216.6.86.45)  8.120 ms
    [AS6453] if-9-0-0.mcore4.pdi-paloalto.as6453.net (216.6.33.6)  491.007 ms
 6  [AS11029] if-0-0-0-892.mcore3.njy-newark.as6453.net (209.58.124.25)  78.807 ms  109.426 ms
78.890 ms
 7  [AS15706] if-4-0-0.core1.fv0-frankfurt.as6453.net (195.219.69.29)  167.206 ms  167.461 ms
167.002 ms
 8  [AS15706] if-0-0-0.core1.fr1-frankfurt.as6453.net (195.219.69.54)  171.256 ms  171.844 ms
174.118 ms
 9  [AS6453] if-7-1-0-1310.core1.stk-stockholm.as6453.net (195.219.131.45)  1180.587 ms  437.592 ms
586.125 ms
10  [AS6453] ix-4-0-1.core1.stk-stockholm.as6453.net (195.219.131.22)  200.475 ms  200.301 ms
201.106 ms
11  [AS3267] b57-1-gw.spb.runnet.ru (194.85.40.129)  216.199 ms  216.117 ms  214.311 ms
12  [AS3267] bl16-1-gw.spb.runnet.ru (194.85.40.78)  214.723 ms  214.463 ms  214.494 ms
13  [AS3267] bm18-1-gw.spb.runnet.ru (194.85.40.169)  214.608 ms  214.504 ms  214.493 ms
14  [AS3267] tv11-1-gw.msk.runnet.ru (194.85.40.137)  214.260 ms  214.360 ms  214.478 ms
15  [AS3267] m9-2-gw.msk.runnet.ru (194.85.40.53)  214.752 ms  214.496 ms  214.882 ms
16   [AS3267] msu.msk.runnet.ru (194.190.255.234)  214.197 ms  214.907 ms  214.656 ms
17  [AS2848] 193.232.127.12 (193.232.127.12)  214.501 ms  214.166 ms  214.531 ms
18  [AS2848] 193.232.113.151 (193.232.113.151)  214.864 ms !Z  214.666 ms !Z  214.522 ms !Z
```

trans-oceanic link

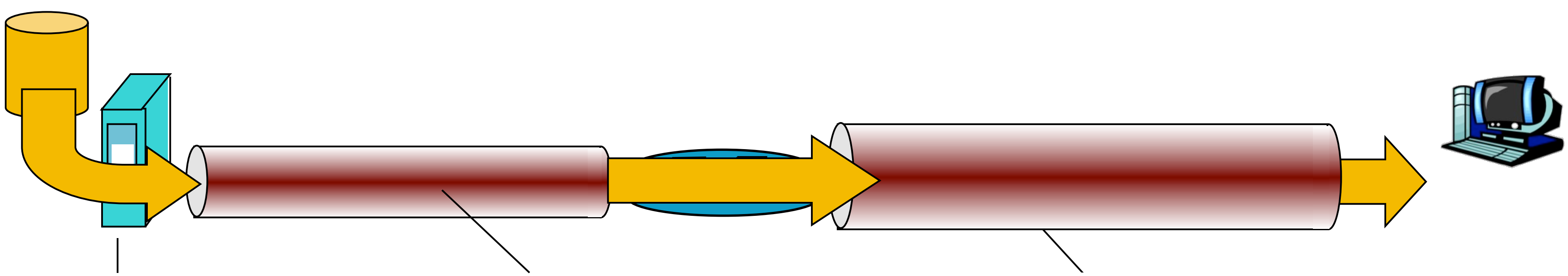
Packet Loss

- 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!**



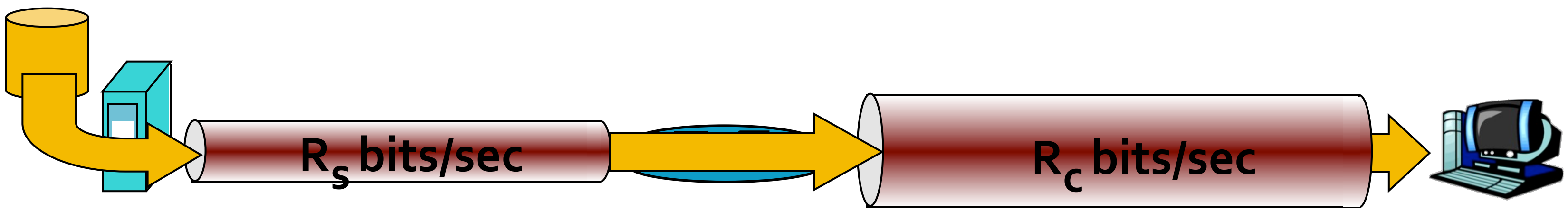
Throughput

- ***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

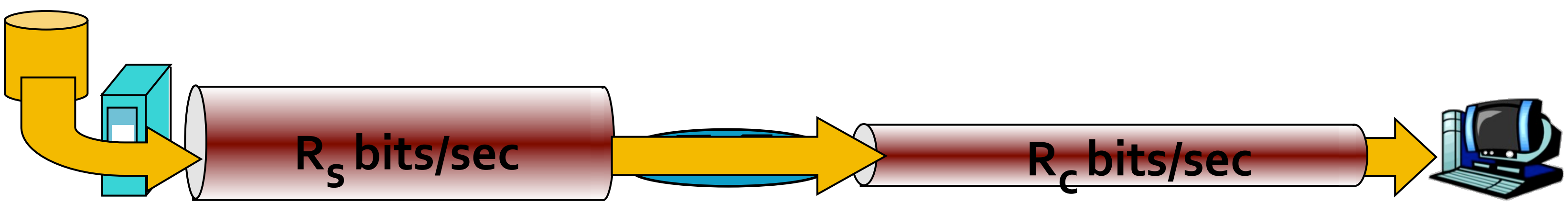


Throughput (more)

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



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

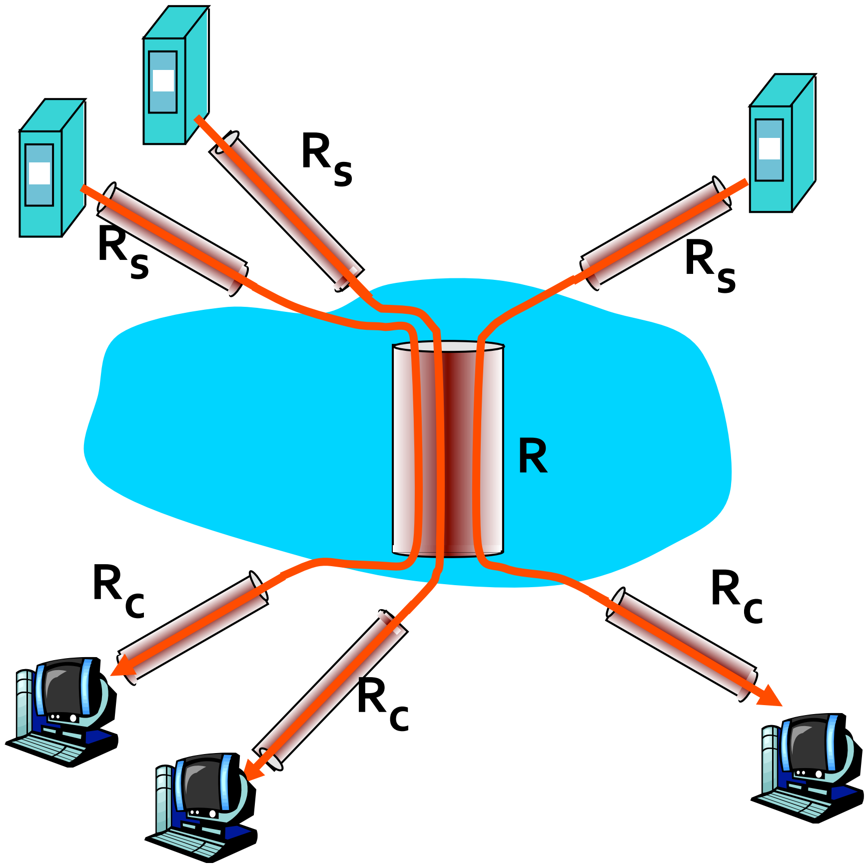


Bottleneck link

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

Throughput: Internet Scenario

- 10 connections (fairly) share backbone bottleneck link at R bits/sec
- Per-connection end-end throughput: min $(R_c, R_s, R/10)$
- In practice: R_c or R_s is often bottleneck



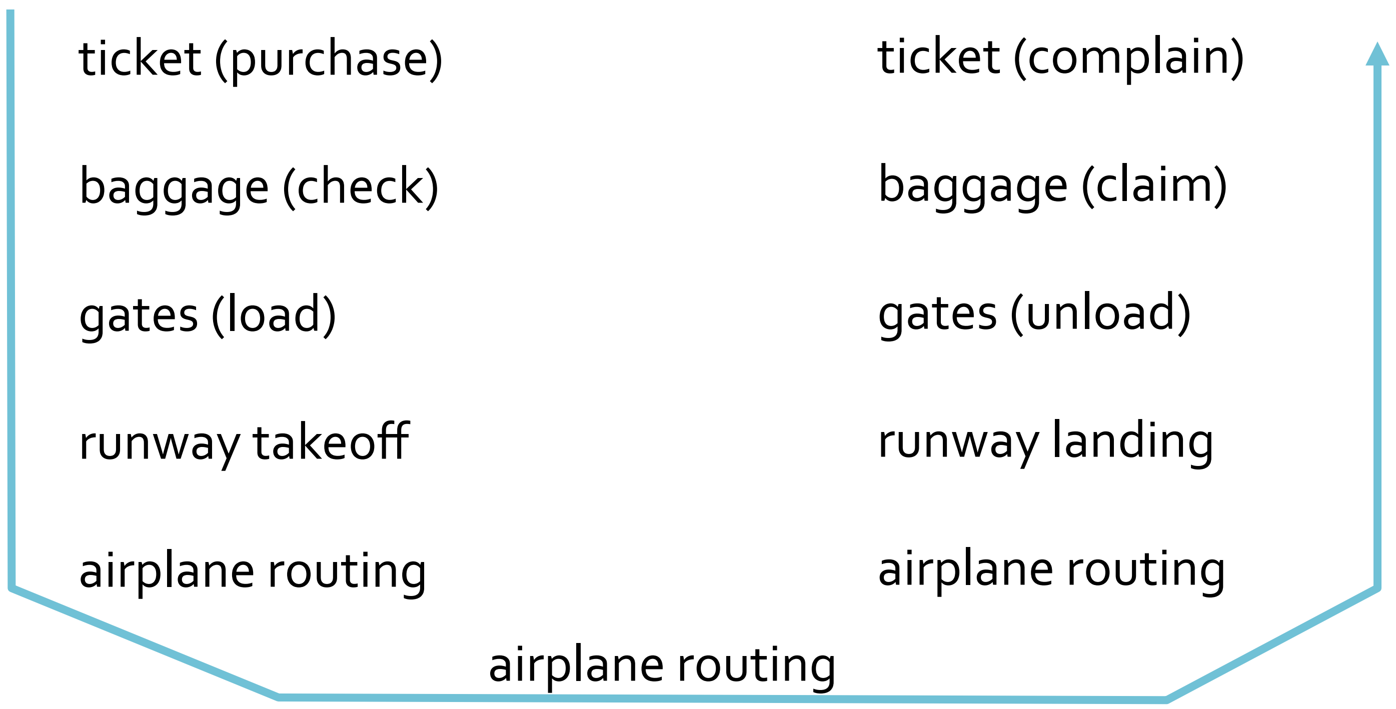
Intro to Networking

- What is the Internet?
- Network edge
  - End systems, access networks, links
- Network core
  - Circuit switching, packet switching, network structure
- Performance: Delay, loss and throughput in packet-switched networks
- Protocol layers, service models
- Networks under attack: security

Layers of Protocols

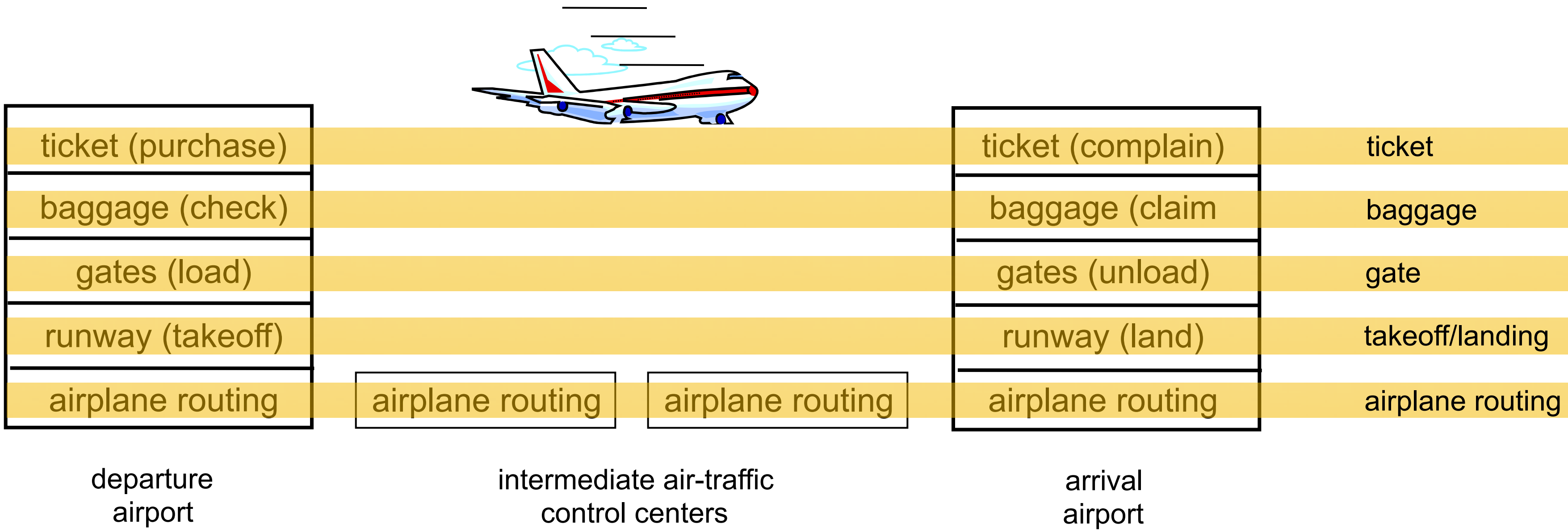
- Networks are complex with many pieces
  - Hosts
  - Routers
  - Links of various media
  - Applications
  - Protocols
  - Hardware, software
- We divide network functions into "layers"
  - Easier to understand and discuss role of various devices

Organization of Air Travel



- A series of steps

Layering of airline functionality



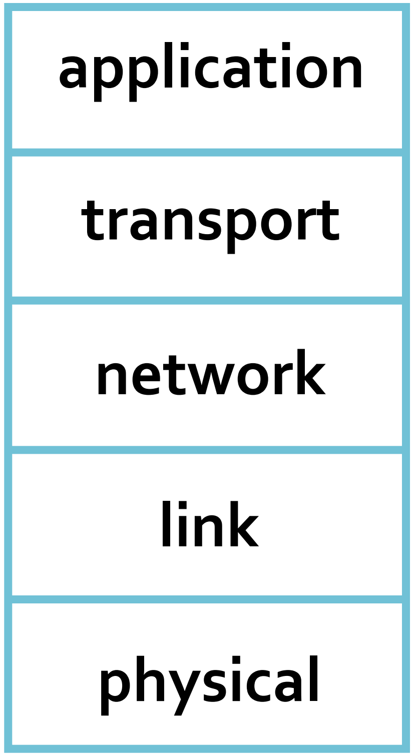
- Layers: Each layers implements a service
  - via its own internal-layer actions
  - relying on services provided by layer below

Why Layering?

- **Human Understanding / Discussion**
  - Dealing with complex systems
  - Explicit structure show relationship of between components
- Modularization eases **maintenance** and system **updates**
  - Can change how a layer is implemented without modifying other layers (change is transparent)
  - e.g., change in gate procedure doesn't affect rest of system

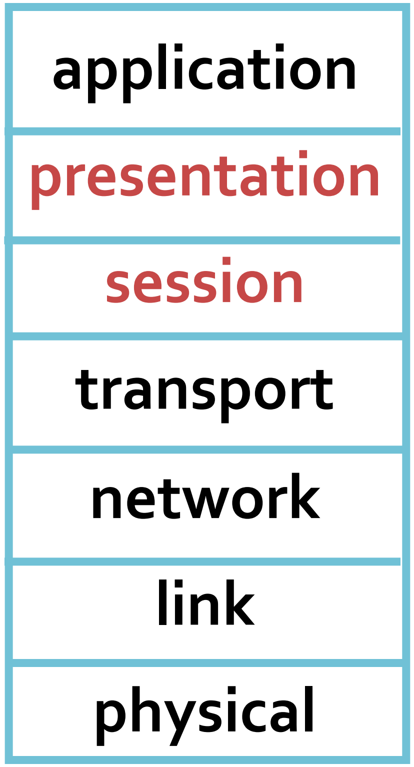
Internet Protocol Stack

- 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
- Physical: bits “on the wire”

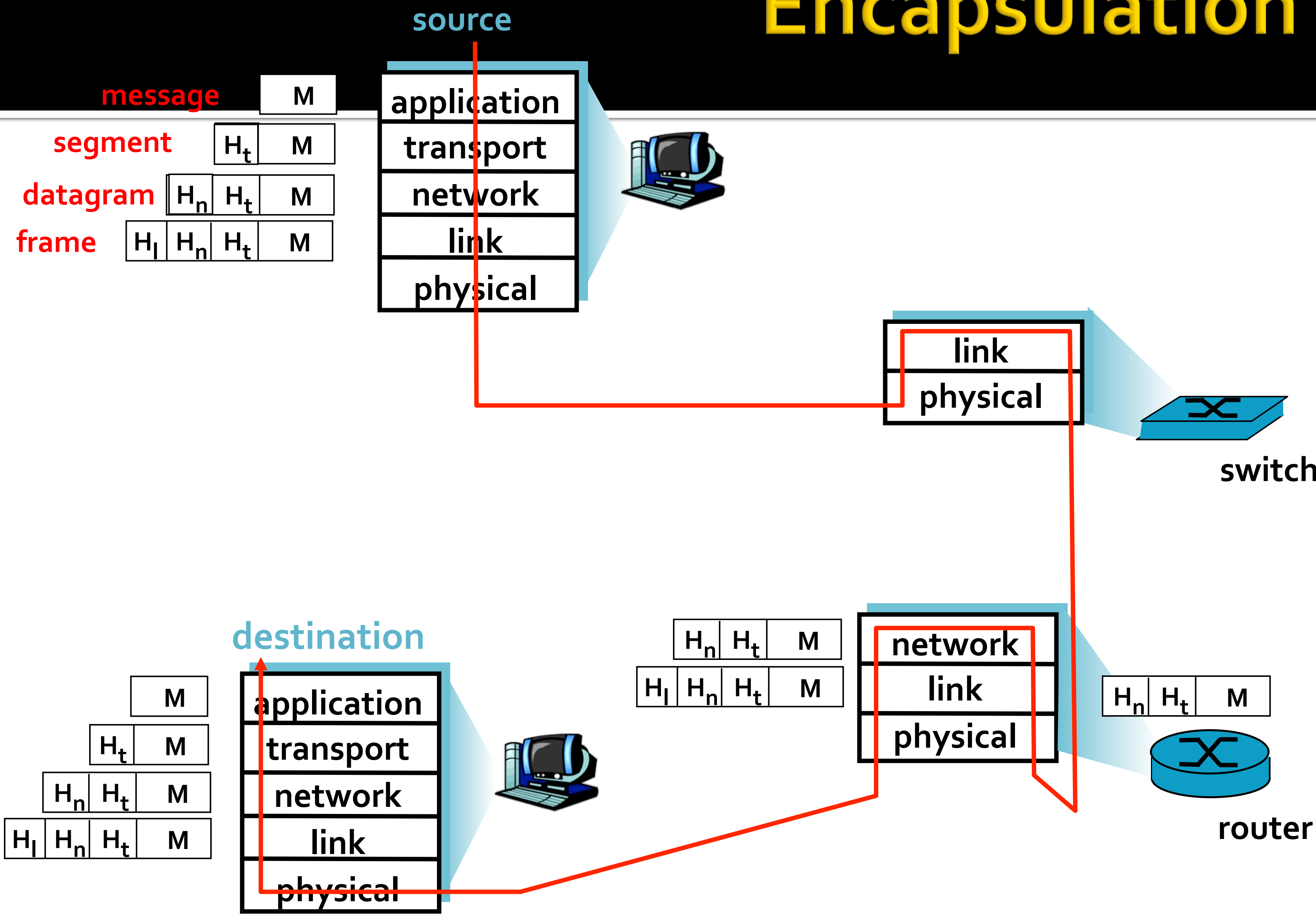


ISO/OSI reference model

- Two added layers!
  - 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



"Magic" of the Internet

- TCP: Reliable, in-order delivery
- IP: Un-reliable, order not guaranteed
- Magic
  - TCP is built on top of IP!

- Great clown analogy by Joel Spolsky
  http://www.joelonsoftware.com/articles/LeakyAbstractions.html

Clown Delivery

Need to move clowns from Broadway to Hollywood for a new job

Broadway, NYC

Clown Delivery – Problems?

Many cars, many clowns
Bad things are guaranteed to happen to at least *some* of them

Car crash / lost

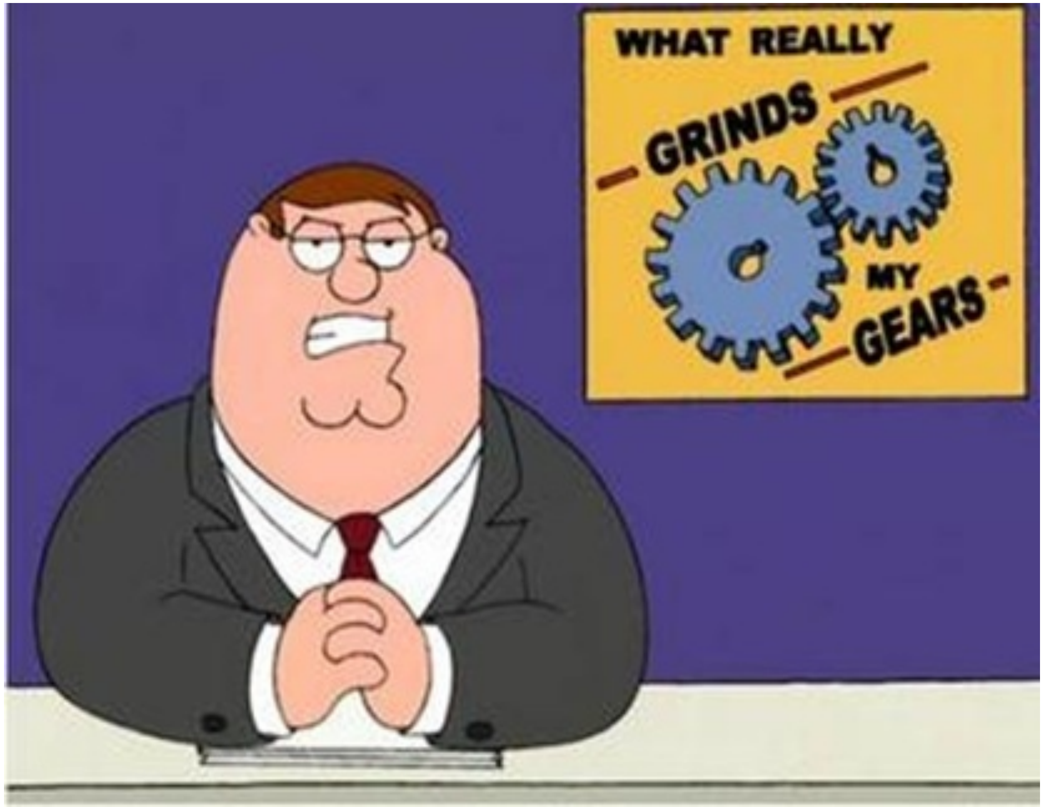
Shaved head / too ugly to work!

Different routes

Clown Delivery – Problems?

People in Hollywood get frustrated –

It's hard to make movies with clowns in this condition!



Clown Delivery - Solution

- New company
  - **Hollywood Express**
- Guarantees that all clowns
  - (1) Arrive
  - (2) In Order
  - (3) In Perfect Condition
- Mishap? Call and request clown's twin brother be sent immediately

- UFO crash in Nevada blocks highway?

- Clowns re-routed via Arizona
  - Director never even *hears* about the UFO crash
  - Clowns arrive a little more slowly

Networking Abstraction

- TCP provides a similar reliable delivery service for IP
- Abstraction has its limits
  - Ethernet cable chewed through by cat?
  - No useful error message for that problem!
  - The abstraction is "leaky" – it couldn't save the user from learning about the chewed cable

Intro to Networking

- What is the Internet?
- Network edge
  - End systems, access networks, links
- Network core
  - Circuit switching, packet switching, network structure
- Performance: Delay, loss and throughput in packet-switched networks
- Protocol layers, service models
- Networks under attack: security

Network Security

- The field of network security is about:
  - How bad guys can **attack** computer networks
  - How we can **defend** networks against attacks
  - How to **design** architectures that are resistant to attacks
- Internet not originally designed with 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!

Bad guys can put malware into hosts via Internet

- Malware can get in host from a virus, worm, or trojan horse.

- Spyware and malware can record keystrokes, web sites visited, upload info to collection site.

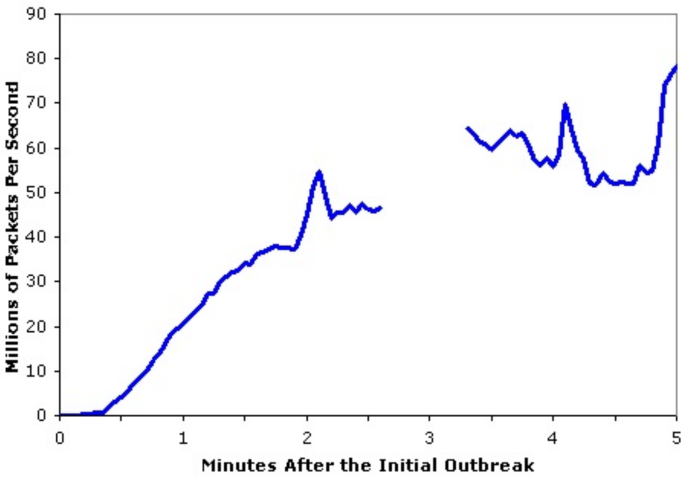
- Infected host can be enrolled in a botnet, used for spam and DDoS attacks.

- Malware is often self-replicating: from an infected host, seeks entry into other hosts

Bad guys can put malware into hosts via Internet

- Trojan horse
  - **Hidden** part of some otherwise useful software
  - Today often on a Web page
- Virus
  - Infection by receiving object (e.g., e-mail attachment) and **user actively executes it**
  - 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

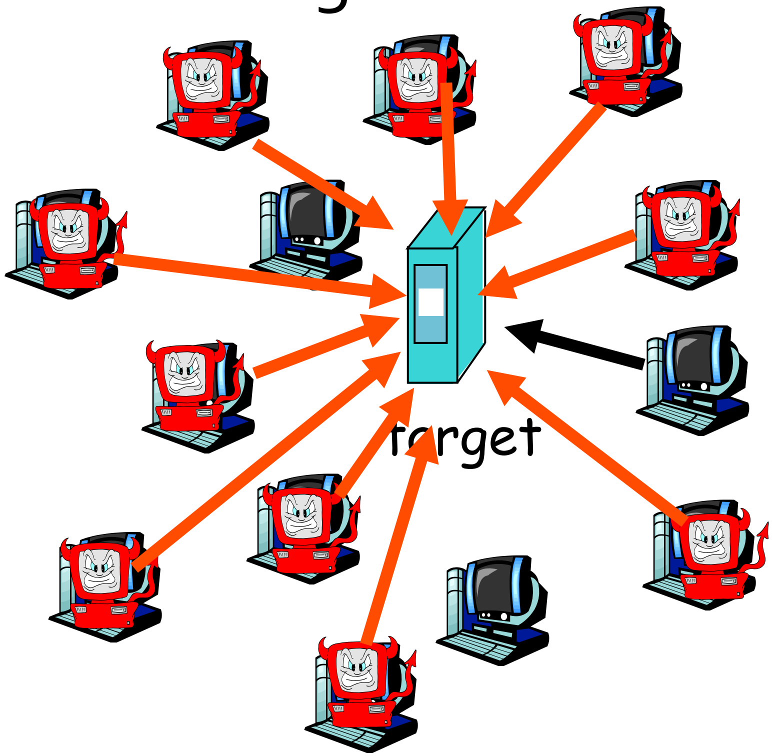
Sapphire Worm: aggregate scans/sec
in first 5 minutes of outbreak (CAIDA, UWisc data)



Bad guys can attack servers and network infrastructure

- Denial of service (DoS): attackers make resources (server, bandwidth) unavailable to legitimate traffic by overwhelming resource with bogus traffic

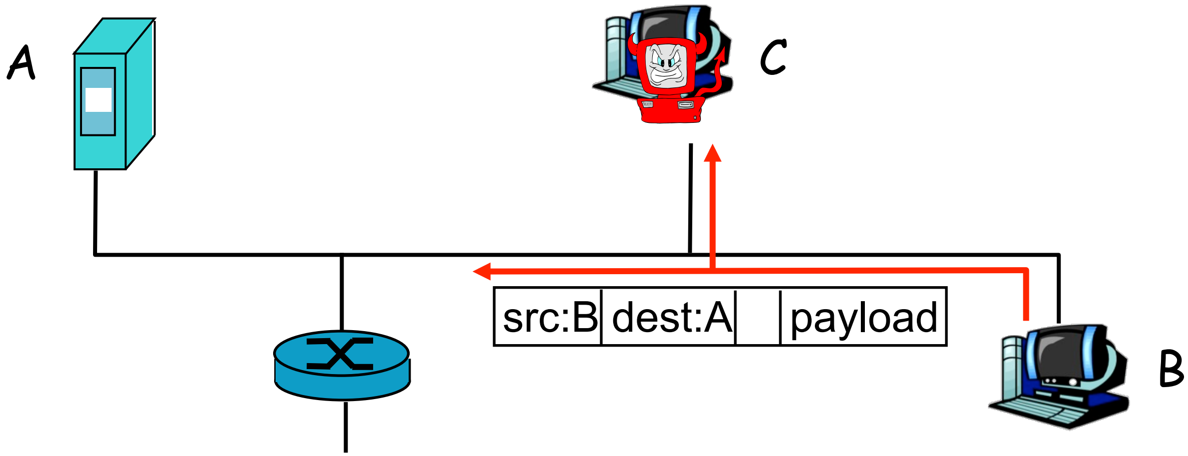
1. Select target
2. Break into hosts around the network
3. Send packets toward target from compromised hosts



The bad guys can sniff packets

Packet sniffing:

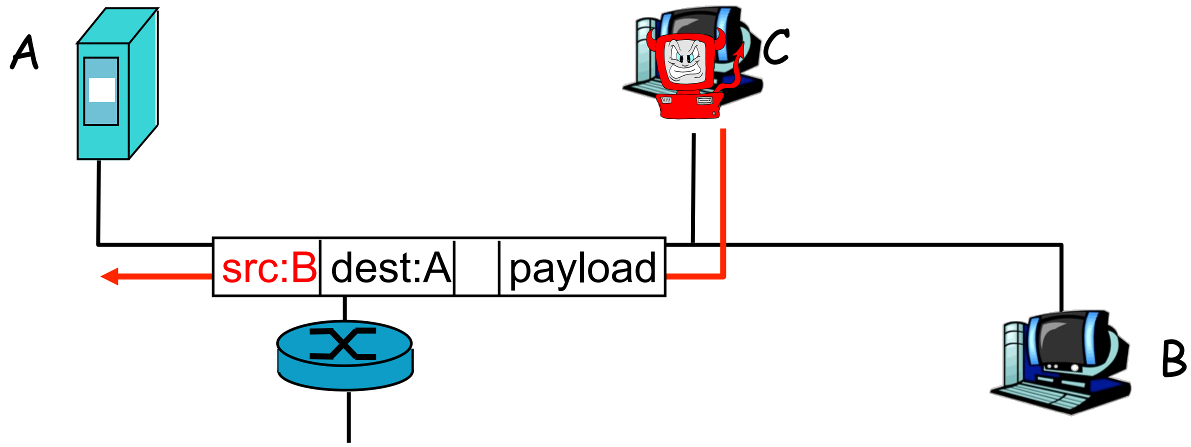
- **Broadcast media** (shared Ethernet, wireless)
- *Promiscuous* network interface reads/records all packets (including passwords, credit card numbers, etc) passing by



- Wireshark software used in labs is a free packet-sniffer
- Tools like this have legitimate and illegitimate uses

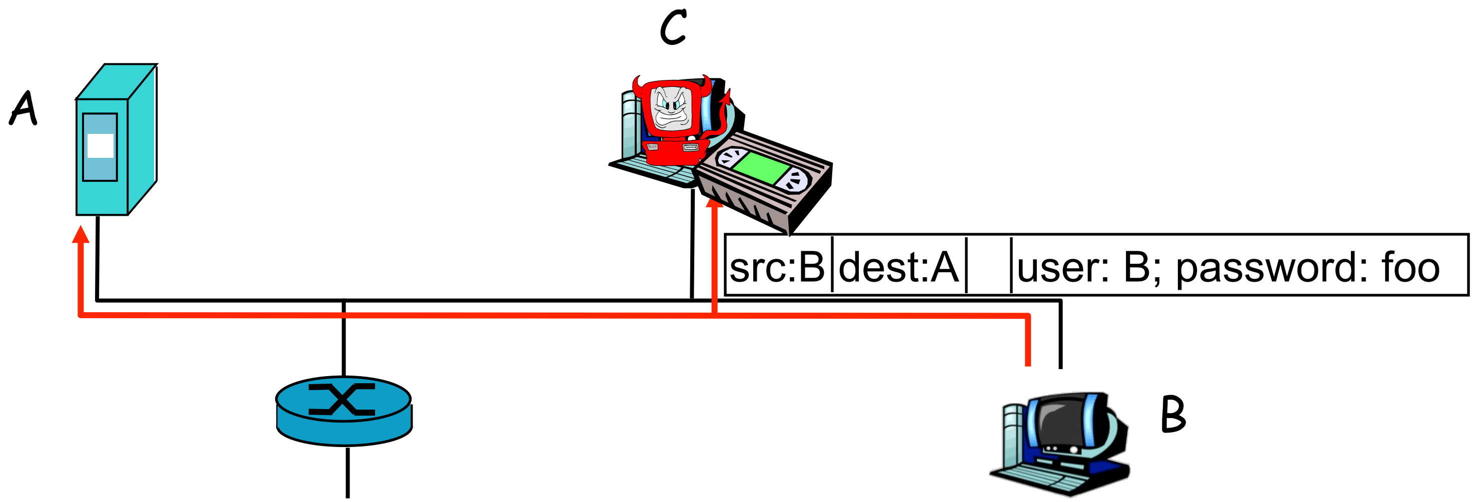
The bad guys can use false source addresses

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



The bad guys can record and playback

- *record-and-playback*: sniff sensitive info (e.g., password), and use later
  - password holder *is* that user from system point of view



Network Security

- Much more in ECPE / COMP 178 in the spring

Course Organization

- Two ways to organize course:

Top-Down

Applications
Transport Layer (e.g. TCP, UDP)
Network Layer (e.g., IP)
Data Link Layer (e.g. Ethernet)

Bottom-Up

- Chose top-down
  - Faster start to programming projects

Introduction: Summary

- This week's brief overview
  - 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
- Rest of the semester: **more depth**!