ECPE / COMP 177
Fall 2015

Computer Networking

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

Logistics

- Instructor: Dr. Jeff Shafer
  - Email: jshafer at pacific dot edu
  - Office: Anderson 205
  - Office hours *(posted on my door)*
    - Monday: 2:00-3:00pm
    - Tuesday/Thursday: 1:00-2:30pm
    - *… plus whenever my office door is open*

Logistics

- Lecture
  - When: Tuesday / Thursday, 10-11:45am
  - Where: Chambers 115
- Lab
  - When: Monday, 3:30-6:30pm
  - Where: Baun 214
  - Lab start date: Next Monday (Aug 31st)

Logistics

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

Pre-Requisites

- COMP 53 – Data structures
  - Programming in high level language
  - Basic data structures, arrays, pointers, functions, system calls, …
- ECPE 170 – Computer Systems and Networks
  - Linux / command-line usage
  - C programming

Course Vision

*What do I, as an **application programmer**, need to understand about computer networks (including software and hardware both on your computer and elsewhere on the network) in order to write efficient, high-performing programs?*

Course Format

- Labs – 10%
  - Applying theoretical concepts to real-world network equipment (Cisco routers and switches)
- In-class Presentations – 10%
  - Two presentations

Course Format

- Exams
  - Midterm Exam – 10%
  - Final Exam – 10%
  - Lab Practical Exam – 10%

Course Format

- Projects – 50%
  - 5 programming projects using network sockets
  - Individual
  - Implementation platform: Linux
  - Python (3.4+)
- *Past projects:*
  - *Web server (basic) + web server (parallel)*
  - *Latency / bandwidth measurement tool*
  - *Instant messenger / file sharing client*

Survey

- Will have in-class project work days throughout the semester
- A laptop to bring to class would be ideal
  - Must be able to run Linux (either in a virtual machine, or dual boot)
- Do you have a laptop?
- Do we need an alternate plan? (USB key booting…)

Questions?

Intro to Networking

- What is the Internet?
- Network edge -vs- Network core
- Protocol layers

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


PC


server


wireless laptop

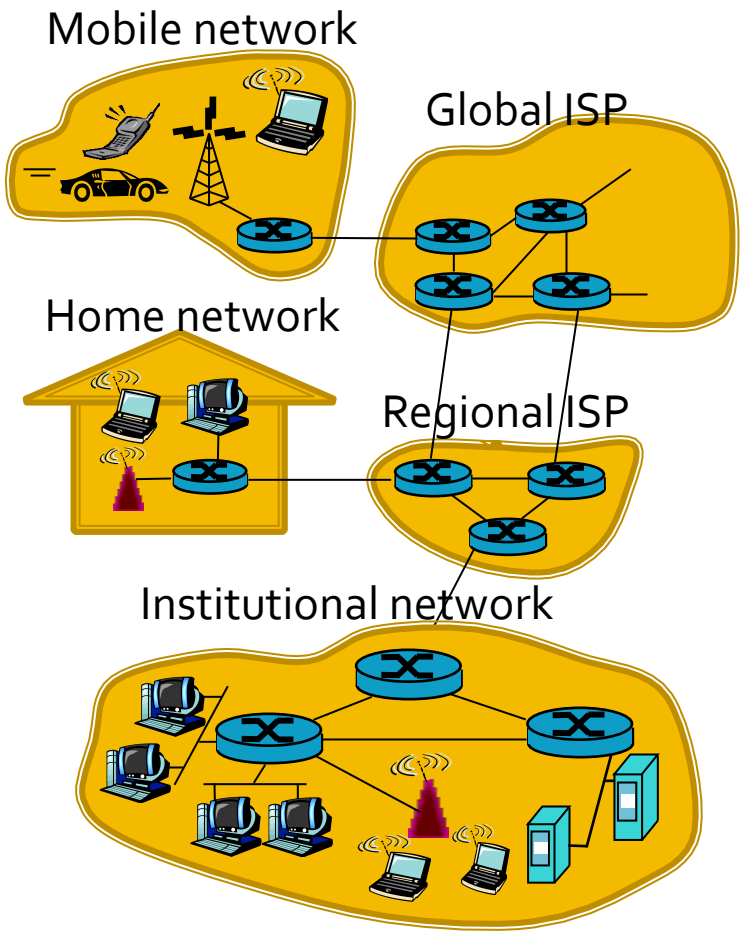

cellular handheld


access points

wired links

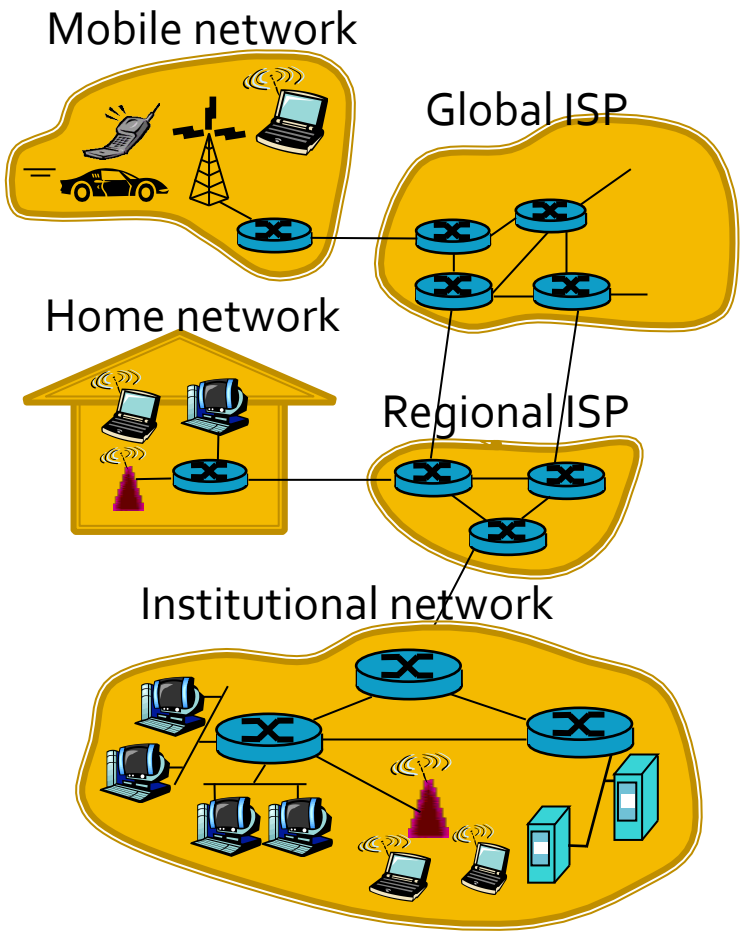

router

- **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*

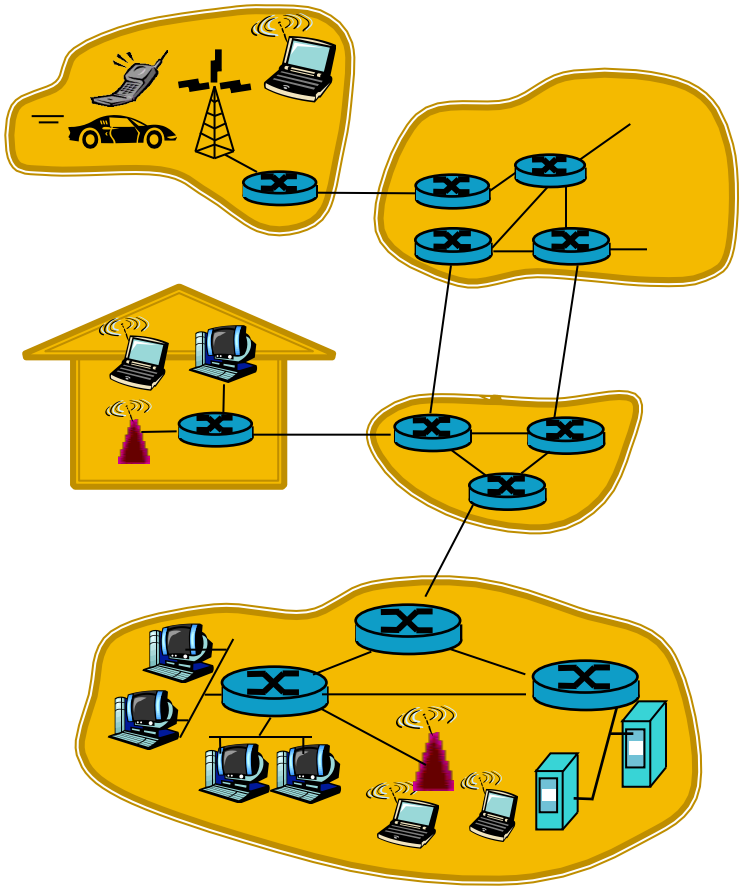


Intro to Networking

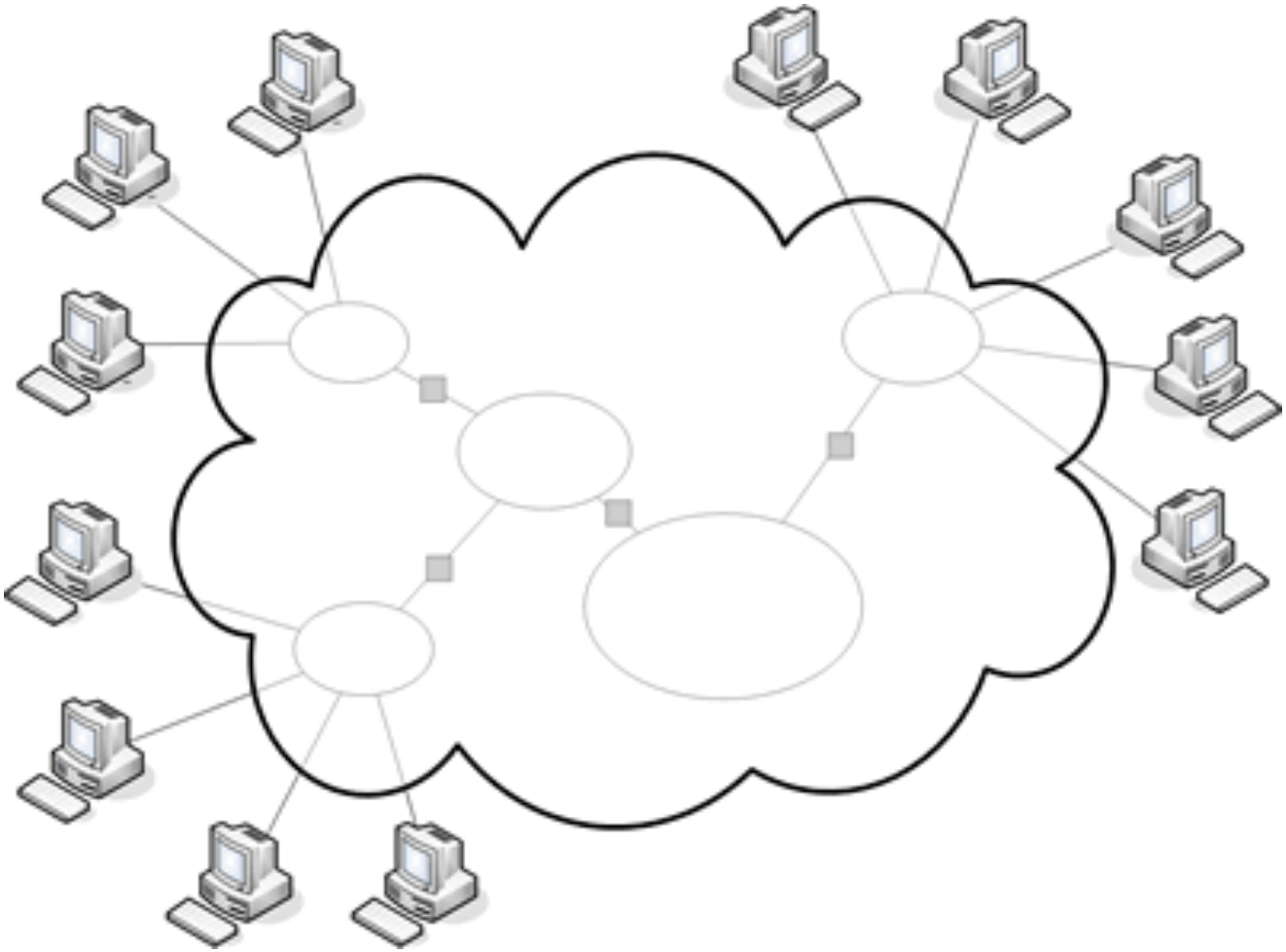
- What is the Internet?
- Network edge -vs- Network core
- Protocol layers

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

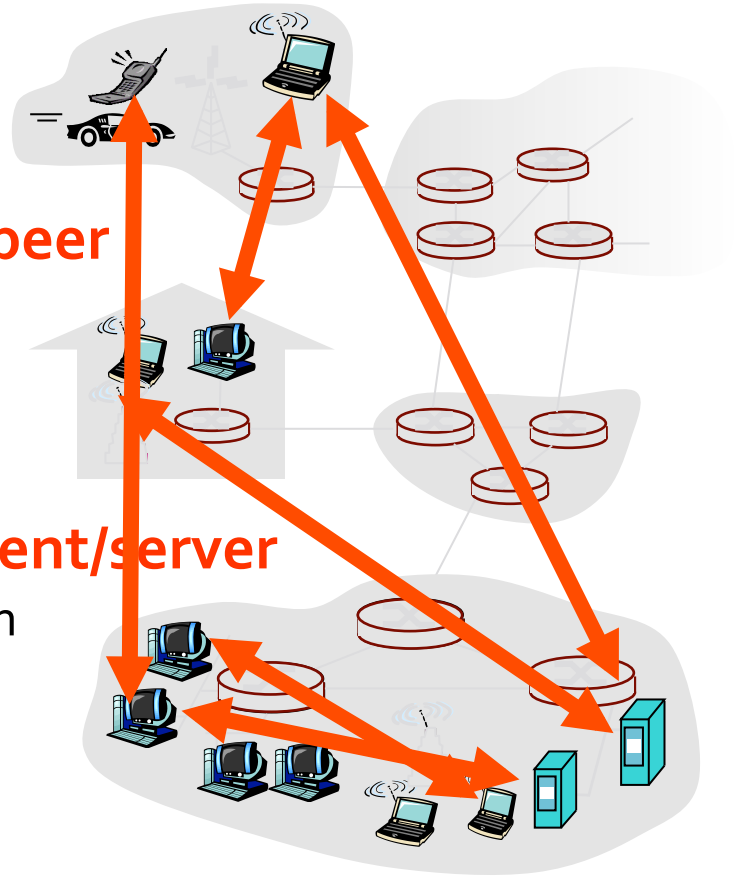


Why is it Called the Edge?



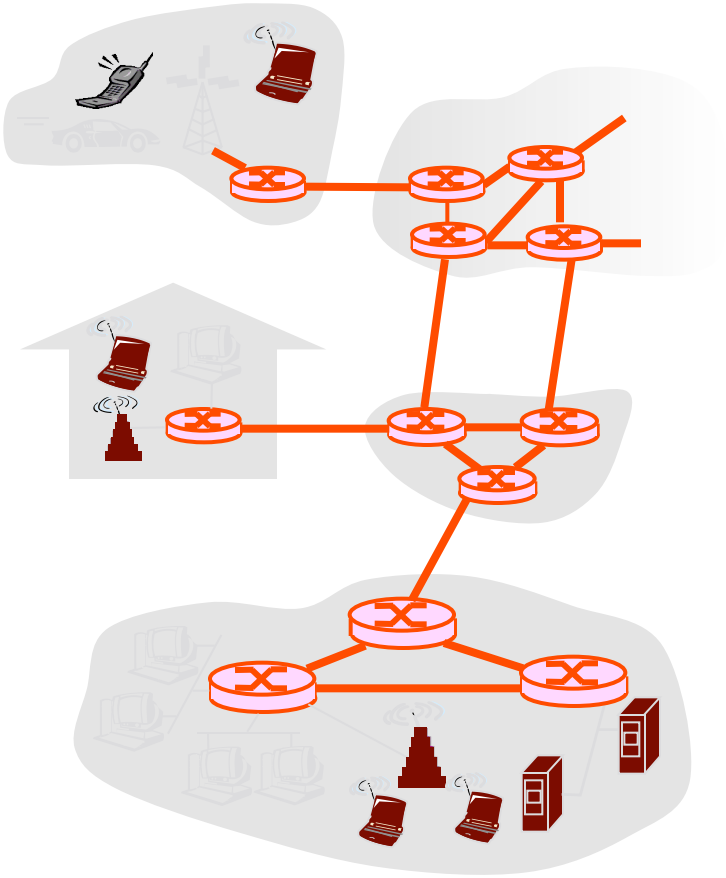
The Network Edge

- **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)



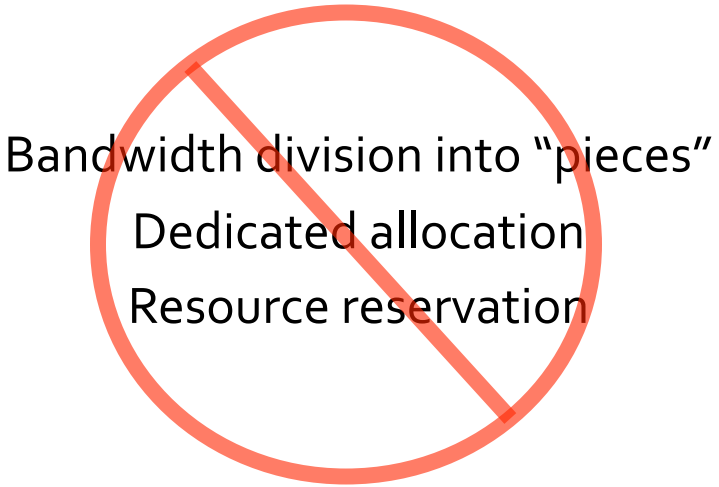
The Network Core

- 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"



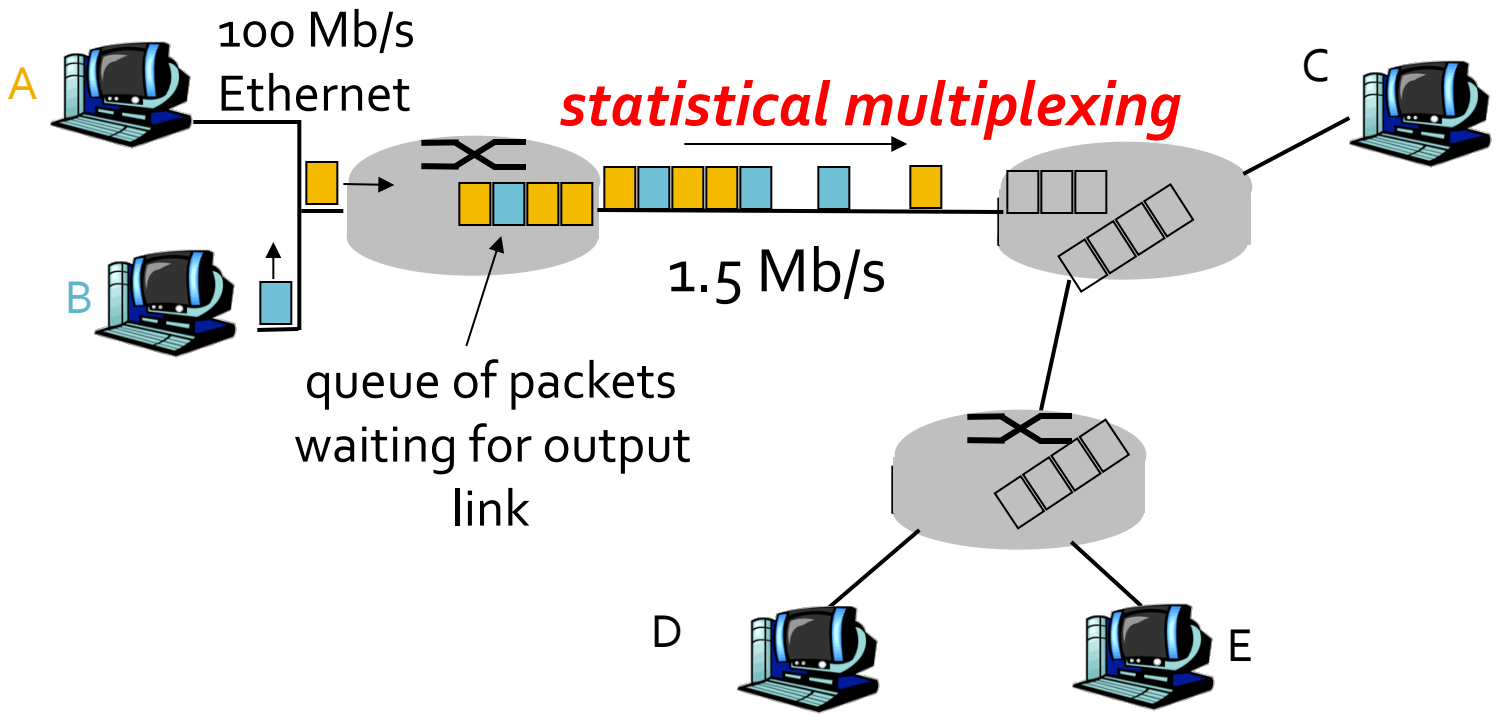
Network Core: Packet Switching

- Each end-end data stream divided into packets
  - User A, B packets share network resources
  - Each packet uses full link bandwidth
  - Resources used as needed



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

Packet Switching: Statistical Multiplexing

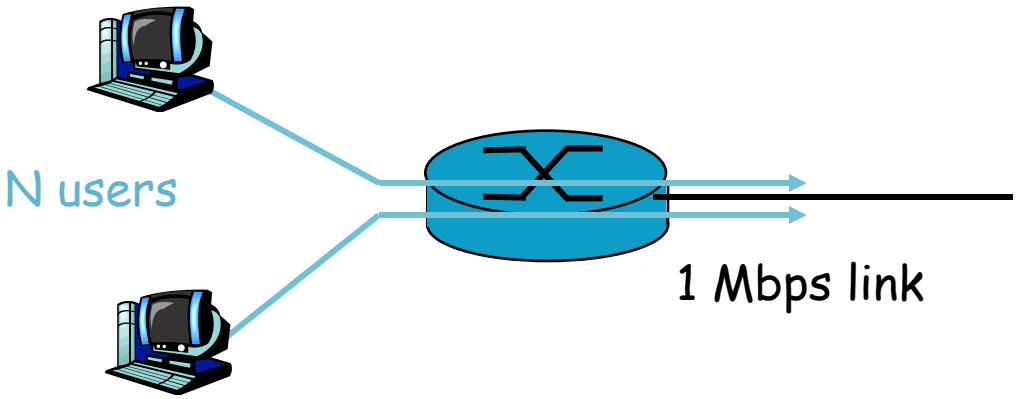


- Sequence of A & B packets does not have fixed pattern, bandwidth shared on demand ➡ ***statistical multiplexing***.
- Contrast against circuit switching / time-division multiplexing
  - Each host gets same slot (fixed pattern)

Packet Switching 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

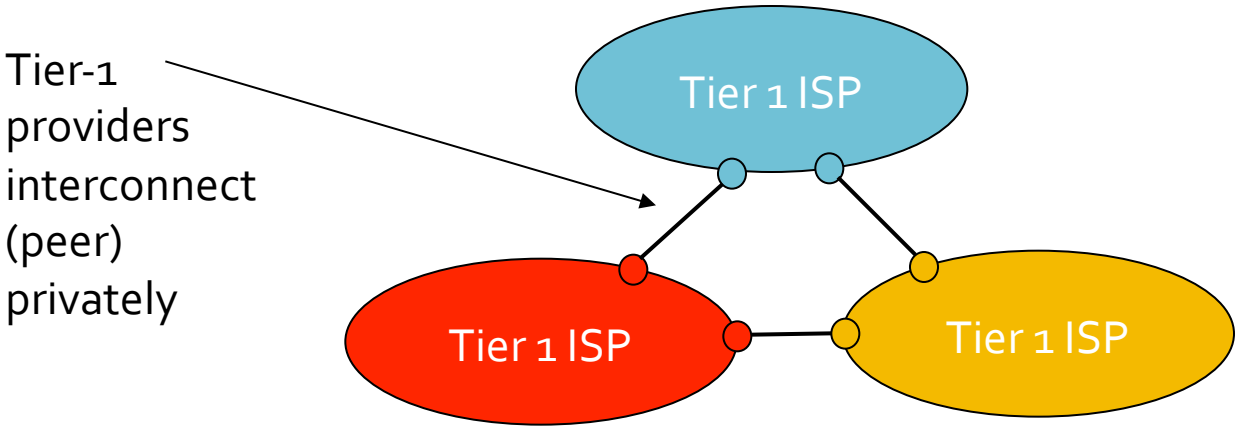


Packet Switching vs Circuit Switching

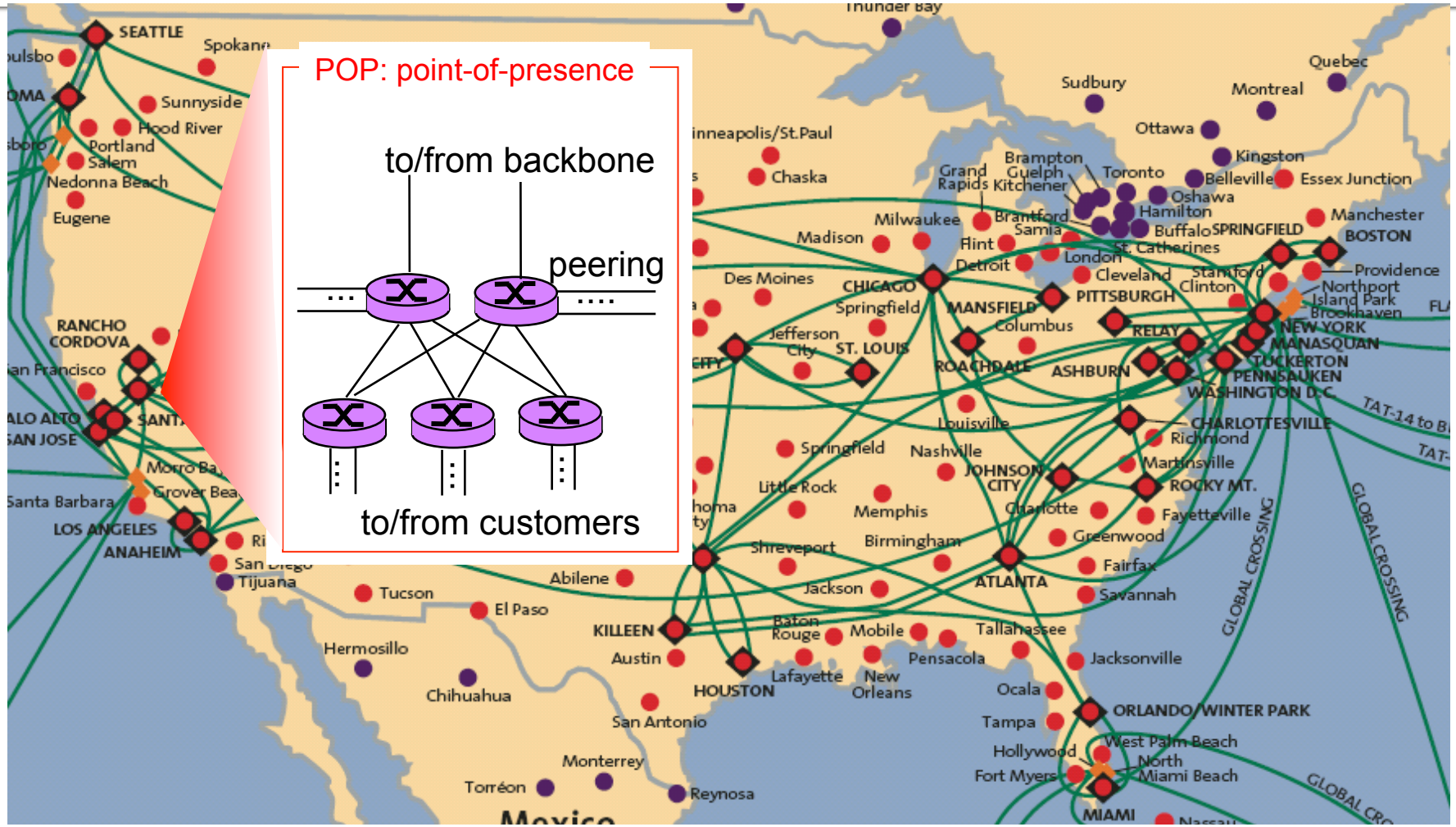
- **Is packet switching perfect in all situations?**
  - (Think about your own experiences)
- Great for bursty data
  - Resource sharing
  - Simpler, no call setup
- Less great during excessive congestion: packet delay / loss
  - Protocols needed for reliable data transfer and congestion control
- Some 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: Sprint, Cogent, NTT, L3, Verizon, AT&T…



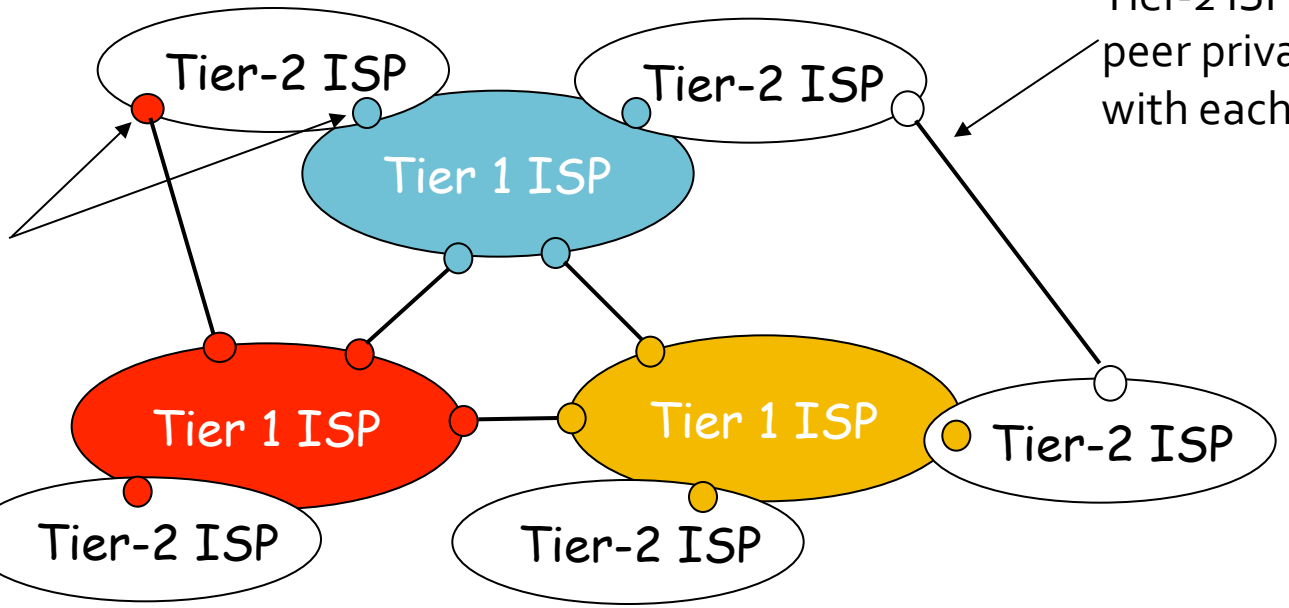
Tier-1 ISP: e.g., Sprint



Internet Structure: Network of Networks

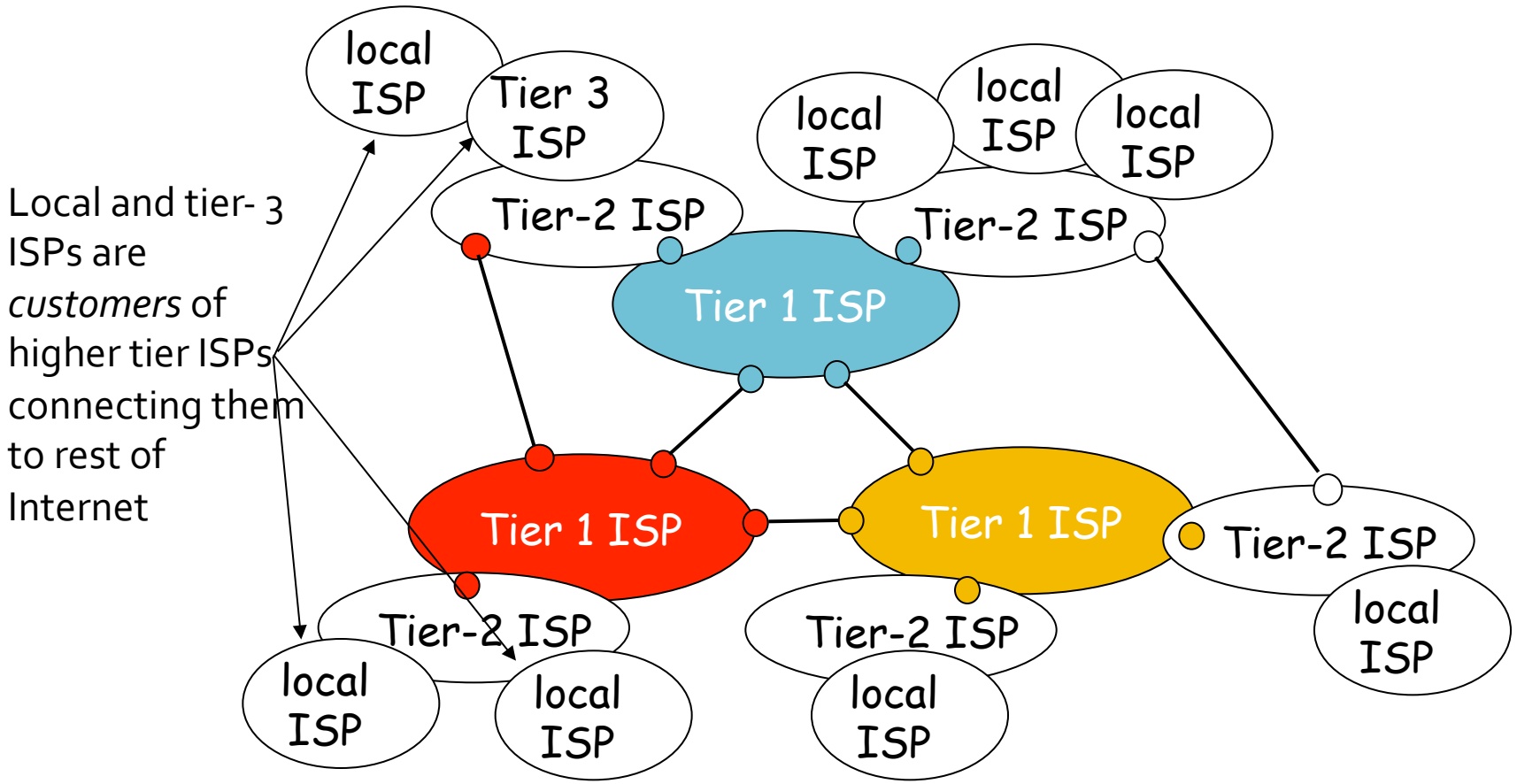
- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet (they are a *customer* of the Tier-1 provider)



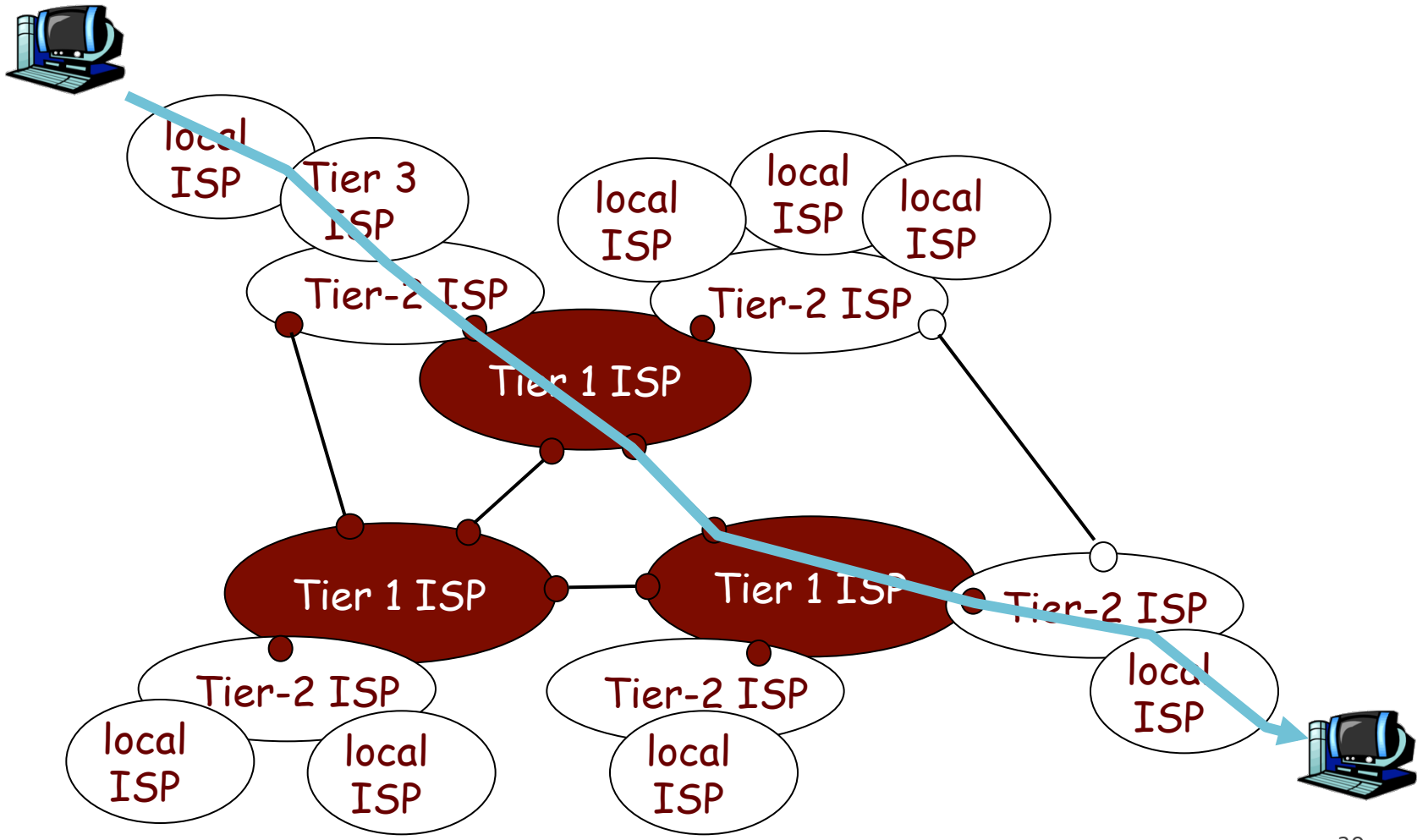
Internet Structure: Network of Networks

- "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



Intro to Networking

- What is the Internet?
- Network edge -vs- Network core
- Protocol layers

What's a Protocol?

HUMAN PROTOCOLS

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

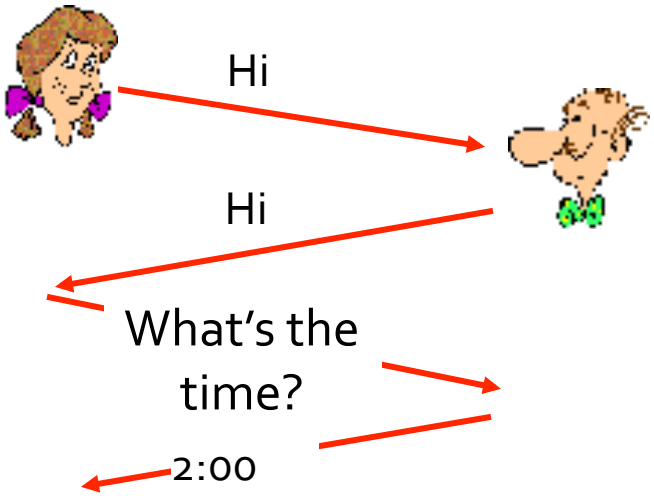
NETWORK PROTOCOLS

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

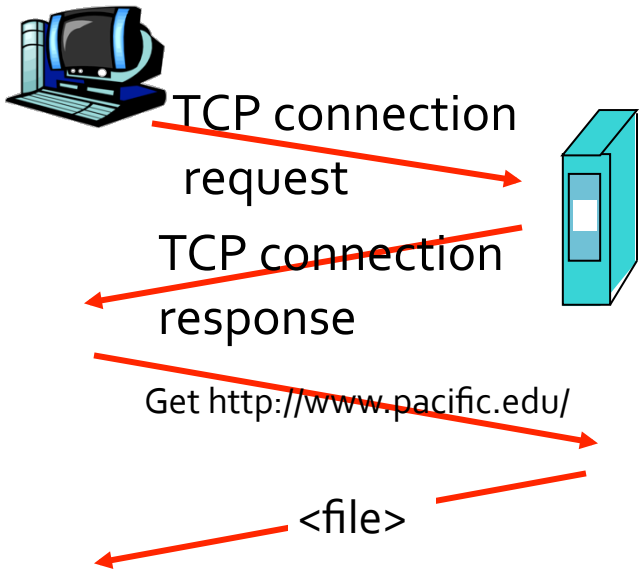
- **Protocols (human and computer!) define**
  - Format of message
  - Order of messages sent/received on network
  - Actions taken after sending/receiving message

What's a Protocol?

HUMAN PROTOCOL



COMPUTER NETWORK PROTOCOL



time

Layers of Protocols

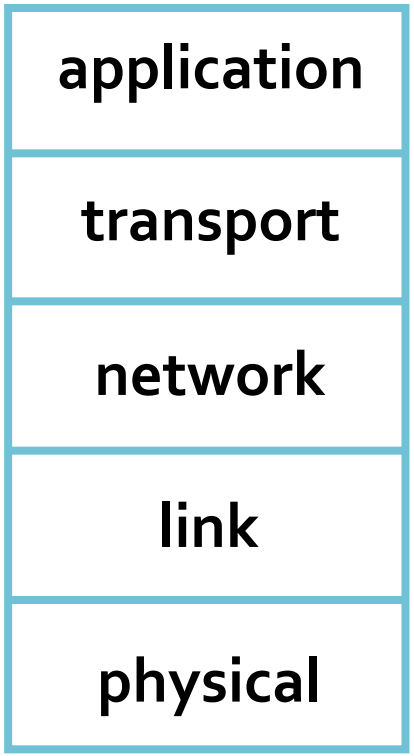
- Networks are complex with many pieces
  - Hosts
  - Routers
  - Links of various media
  - Applications
  - Protocols
  - Hardware, software
- We divide network functions into "layers"
  - Easier to understand and discuss role of various devices

Why Layering?

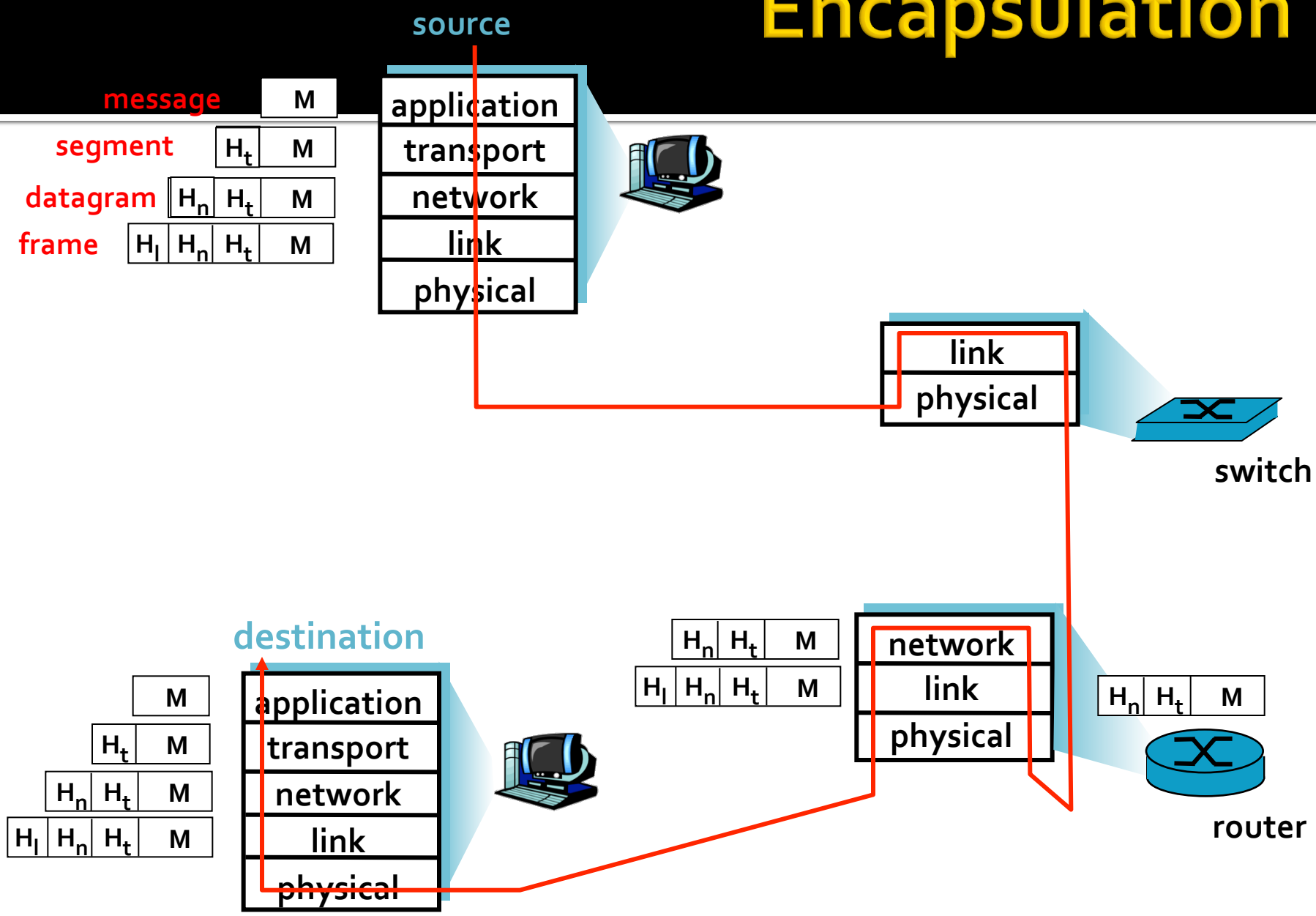
- **Human Understanding / Discussion**
  - Dealing with complex systems
  - Explicit structure show relationship of between components
- Modularization eases **maintenance** and system **updates**
  - Can change how a layer is implemented without modifying other layers (change is transparent)
  - e.g., 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"



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

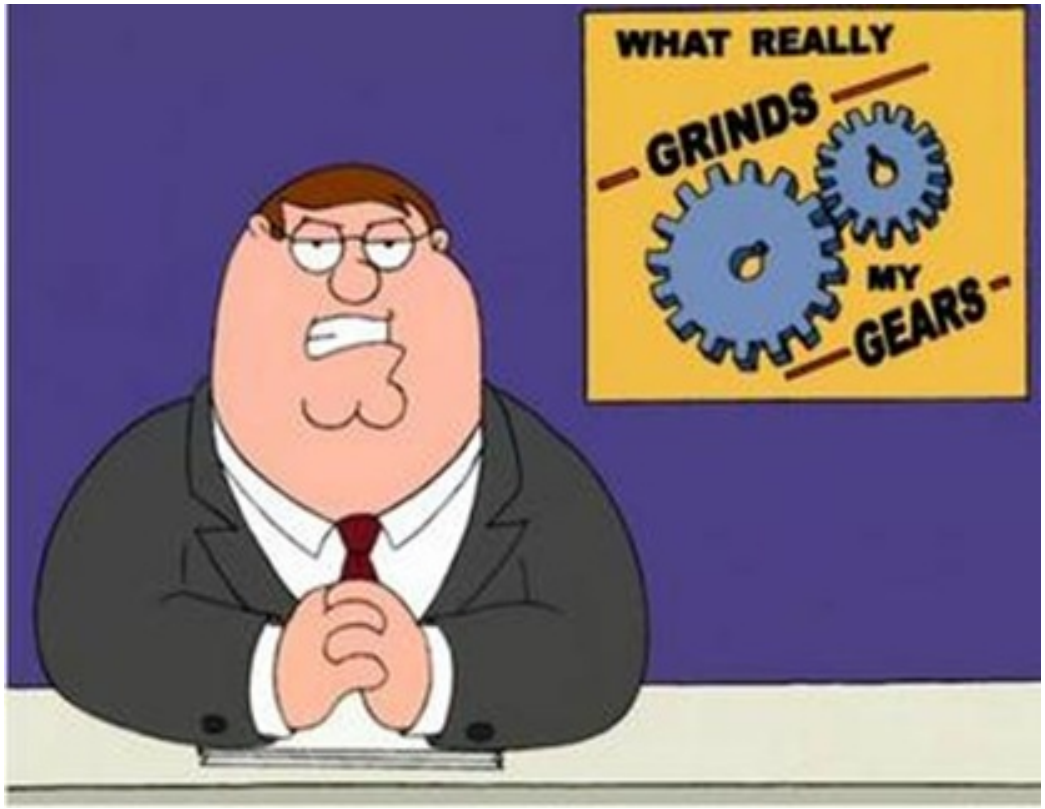
Shaved head / too ugly to work!

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

Introduction: Summary

- Today's brief overview
  - Internet overview
  - What's a protocol?
  - Network edge vs Network core
  - Protocol layers
- Rest of the semester: **more depth**!