CS 456 – Computer Networks

- Instructor: Ian Goldberg
  http://www.cs.uwaterloo.ca/~iang/

- Classes: Tuesday and Thursday
  8:30 – 9:50am MC 4063 (section 1)
  2:30 – 3:50pm MC 2038 (section 2)

- You will need an account on the student.cs environment.
  - If you don't have a student.cs account for some reason, get one set up in MC 3017.

CS 456 – Computer Networks

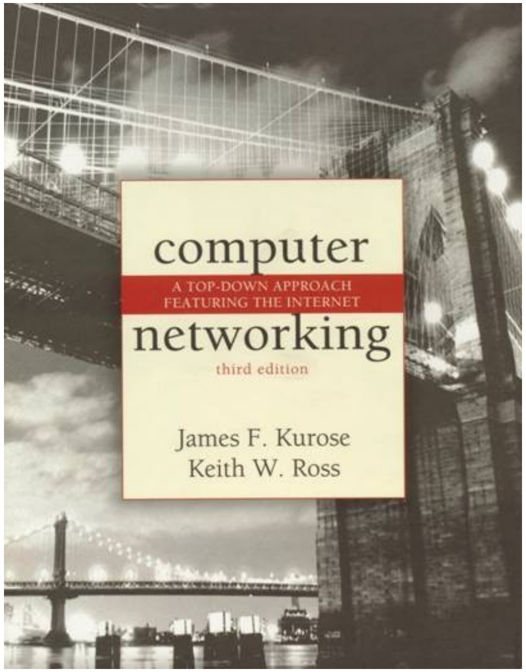
- This course will use UW-ACE (aka UWANGEL) extensively.
  - Syllabus, calendar, lecture notes, additional materials, assignments, discussion, communication, important announcements, etc.
- It is your responsibility to keep up with the information on that site.
  - But check your UW email as well; we may need to send emergency messages there.
  - Only use UW-ACE to send messages to course personnel.
- Feedback is encouraged!

Grading Policy

- Midterm (15%)
  - Around the end of October
- Final (35%)
- Three programming assignments (10% + 15% + 15%)
  - Work alone
  - Require CS student computing environment for submission
  - Additional tasks for CS 656 students
- Two labs (5% + 5%)
  - Lab 1: In October
  - Lab 2: In November
  - Groups of two
- Additional research survey paper for CS 656 students
  - Details on UW-ACE
- See UW-ACE for late and reappraisal policies, academic integrity policy, and other details.

Required Textbook

Computer Networking: A Top Down Approach Featuring the Internet,
3rd edition.
Jim Kurose, Keith Ross
Addison-Wesley, 2005.



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

Course Goals

- Learn how communication networks are put together
  - mechanisms, algorithms, technology components
- Our primary example will be the Internet.
  - but we'll touch on some others as well
- Understand fundamental challenges
- Learn about existing solutions
  - typically: no single dominant solution
- What problems still need solving?

This class and next

- Course mechanics (done)
- Overview and introduction to communications networks
  - In particular, the Internet

Chapter 1: Introduction

Our goal:

- get "feel" and terminology
- more depth, detail *later* in course
- approach:
  - use Internet as example

Overview:

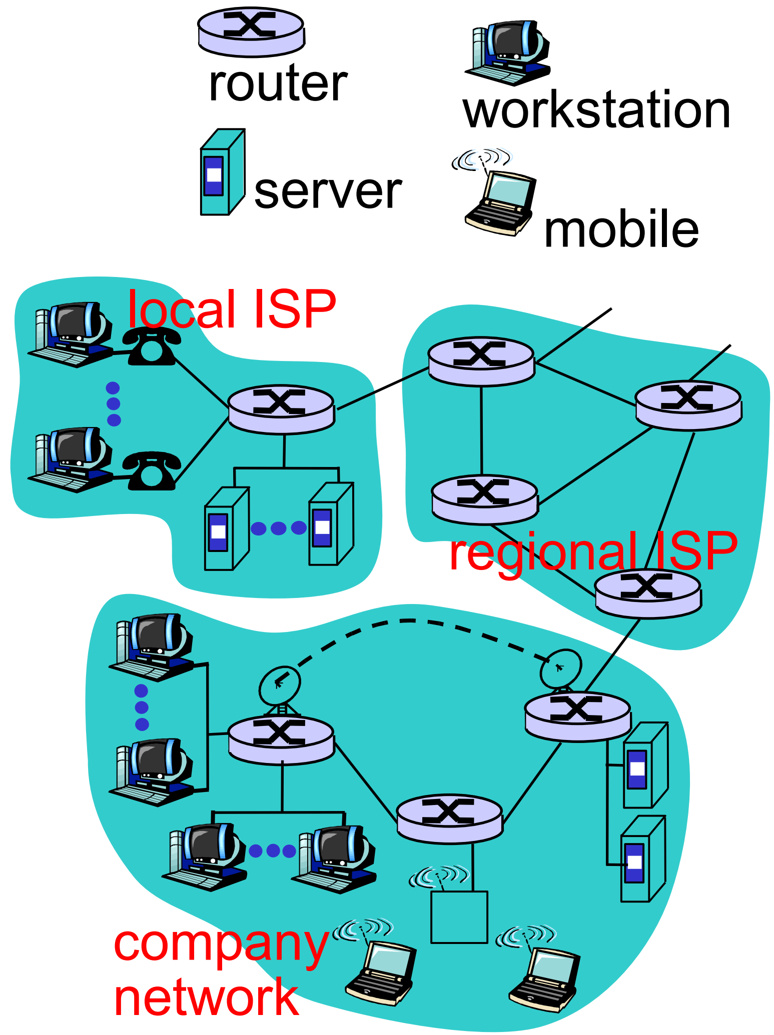
- what's the Internet?
- what's a protocol?
- network edge
- network core
- access net, physical media
- Internet/ISP structure
- performance: loss, delay
- protocol layers, service models
- network modeling

Chapter 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
1.3 Network core
1.4 Network access and physical media
1.5 Internet structure and ISPs
1.6 Delay & loss in packet-switched networks
1.7 Protocol layers, service models
1.8 History

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

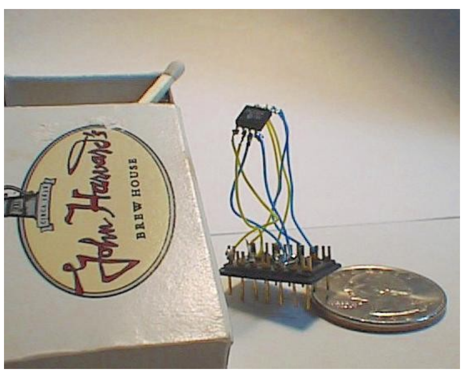
- millions of connected computing devices: *hosts = end systems*
- running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = ***bandwidth***
- *routers:* forward packets (chunks of data)



“Cool” internet appliances

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

Web-enabled toaster +
weather forecaster

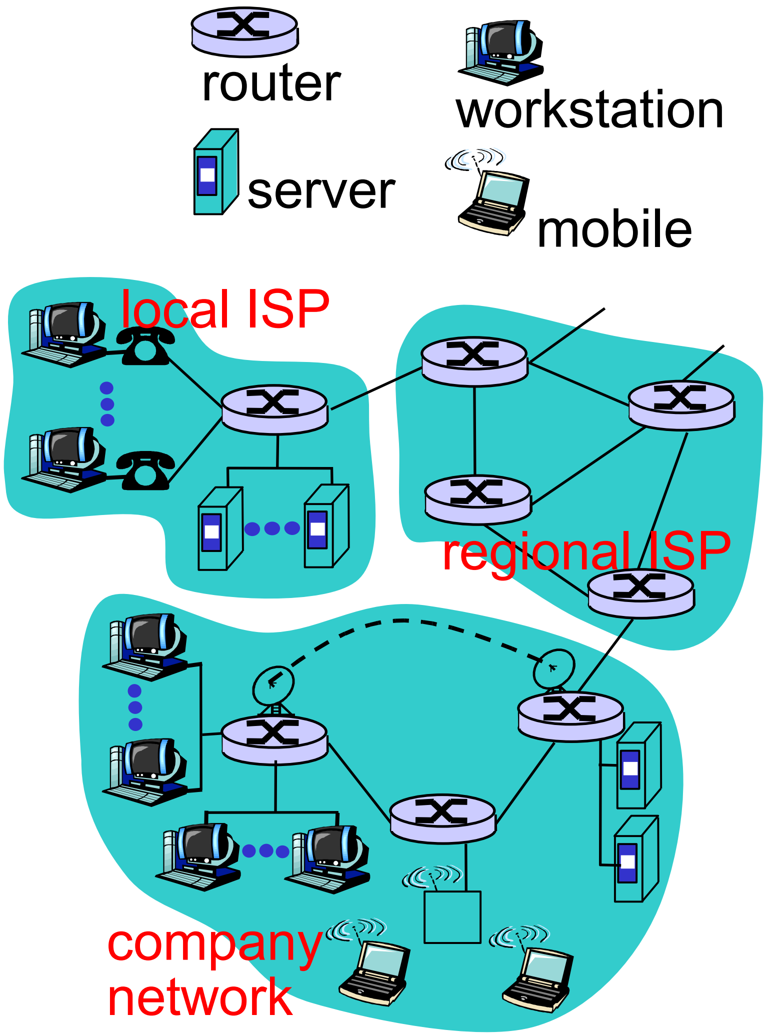


World’s smallest web server
http://www-ccs.cs.umass.edu/~shri/iPic.html

Internet phones

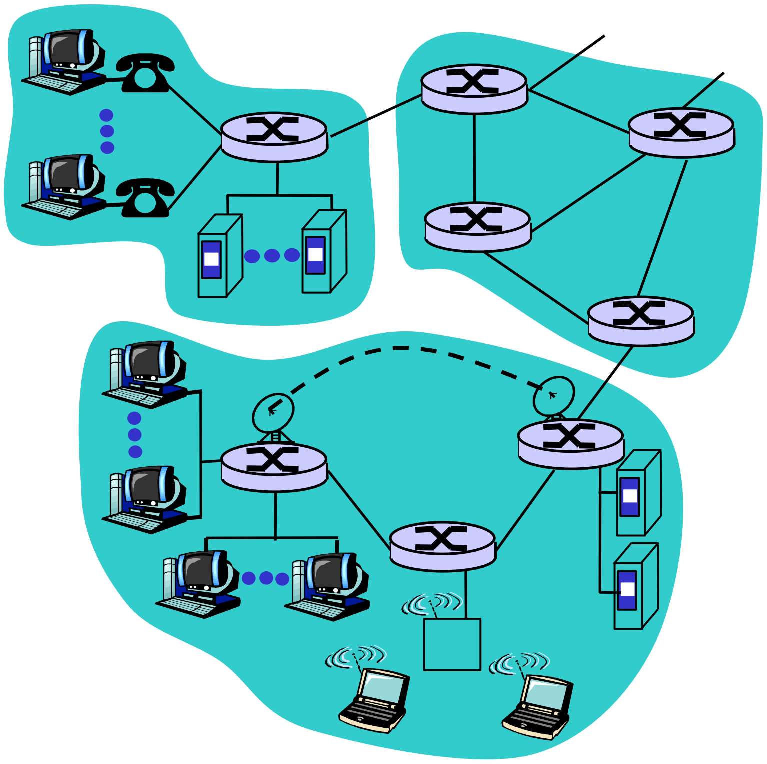
What's the Internet: a service view

- *Protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, FTP, PPP
- *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, email, e-commerce, file sharing, games
- Communication services provided to applications:
  - Connectionless unreliable
  - Connection-oriented reliable



What's a protocol?

Human protocols:

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

… specific messages sent

… specific actions taken when messages received, or other events

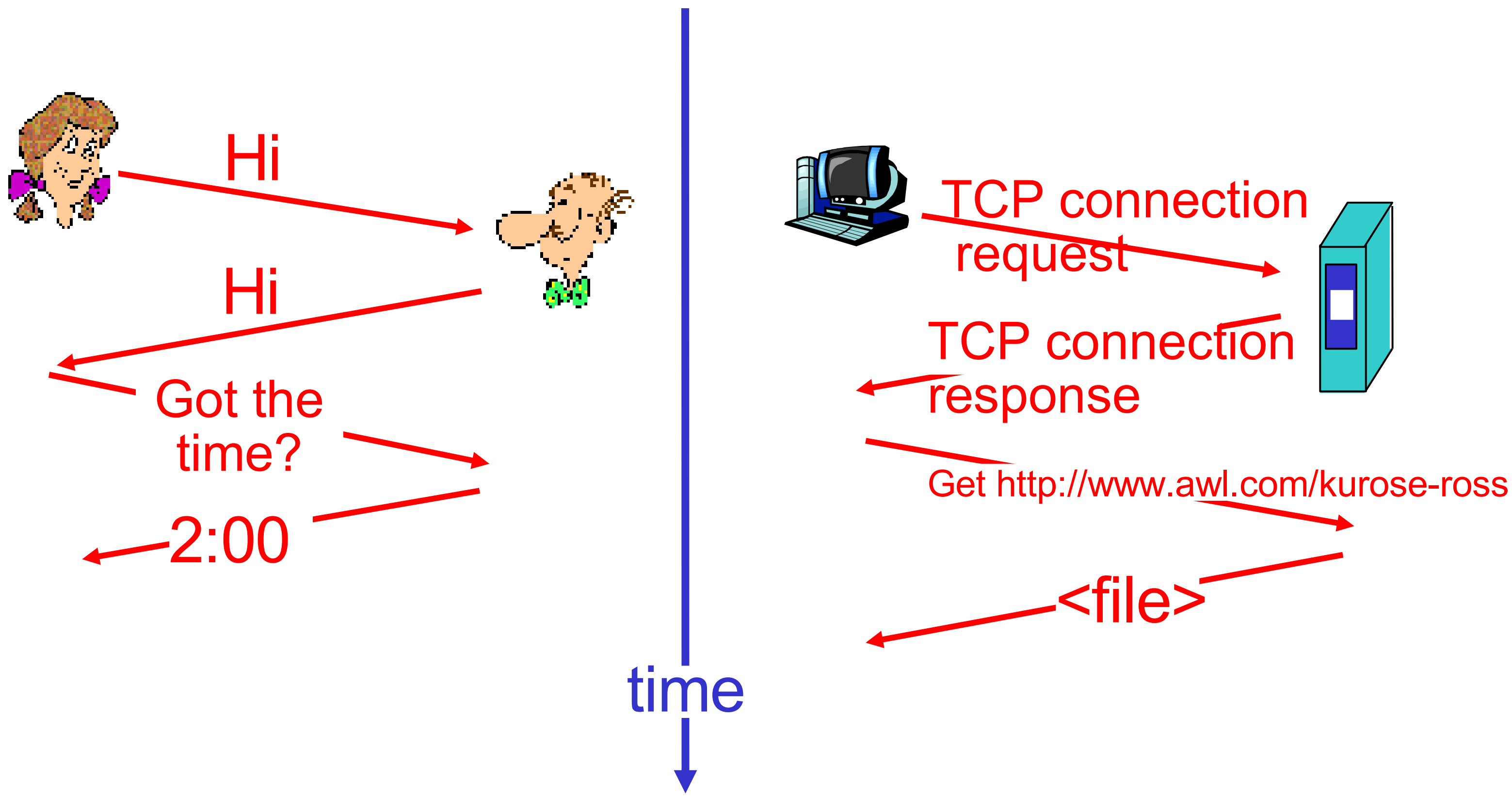
Network protocols:

- machines rather than humans
- all communication activity in Internet governed by protocols

Protocols define the format and order of messages sent and received among network entities, and actions taken on message transmission and receipt.

Protocol diagrams

A human protocol and a computer network protocol:

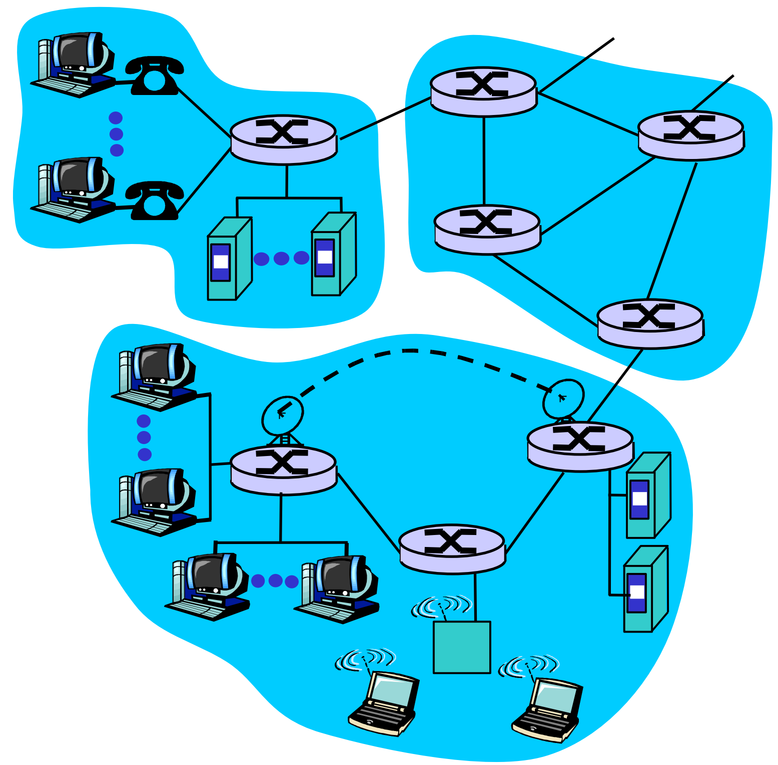


Chapter 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
1.3 Network core
1.4 Network access and physical media
1.5 Internet structure and ISPs
1.6 Delay & loss in packet-switched networks
1.7 Protocol layers, service models
1.8 History

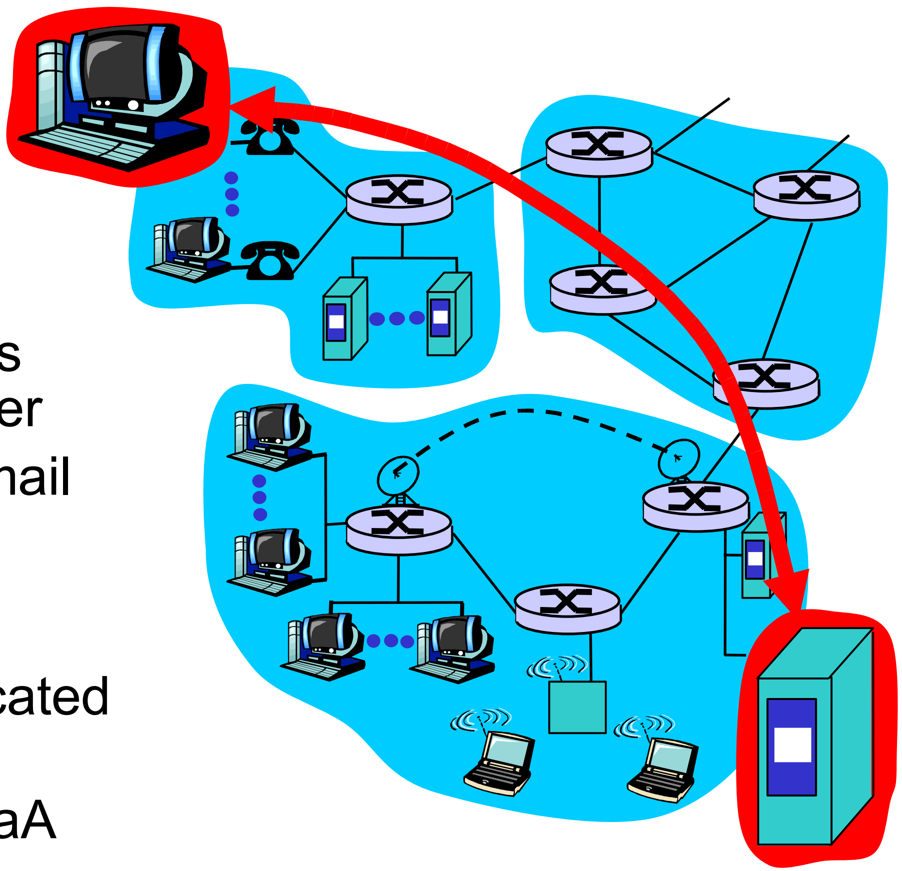
A closer look at network structure:

- **Network edge:** applications and hosts
- **Network core:**
  - routers
  - network of networks
- **Access networks, physical media:** communication links



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-to-peer model:
  - minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent, KaZaA



Network edge: connection-oriented 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 connection-oriented 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: connectionless 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

Some apps using TCP:

- HTTP (Web)
- FTP (file transfer)
- ssh (remote login)
- SMTP (email)

Some apps using UDP:

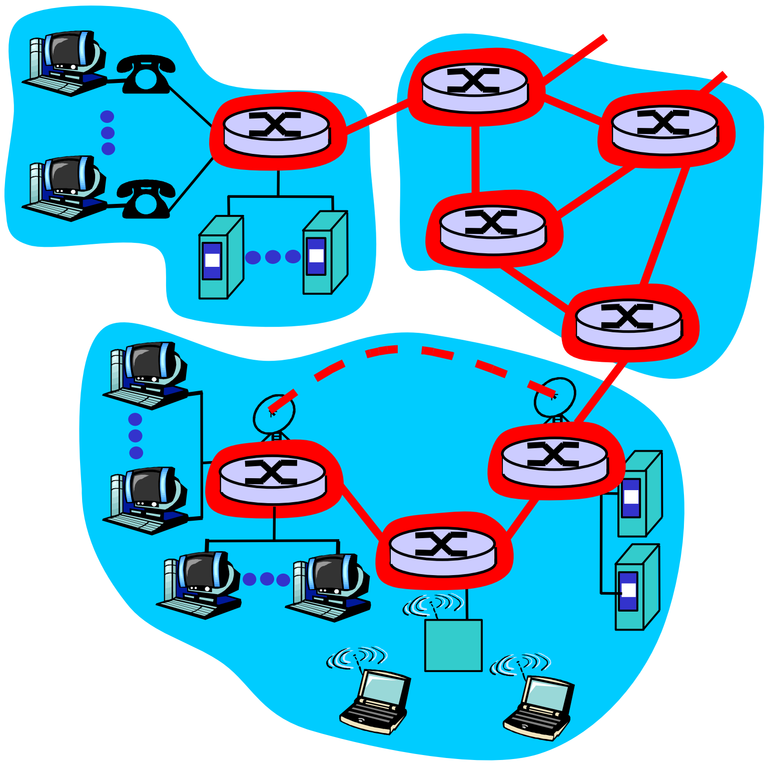
- streaming media
- teleconferencing
- DNS
- Internet telephony

Chapter 1: roadmap

1.1 What *is* the Internet?
1.2 Network edge
1.3 Network core
1.4 Network access and physical media
1.5 Internet structure and ISPs
1.6 Delay & loss in packet-switched networks
1.7 Protocol layers, service models
1.8 History

The Network Core

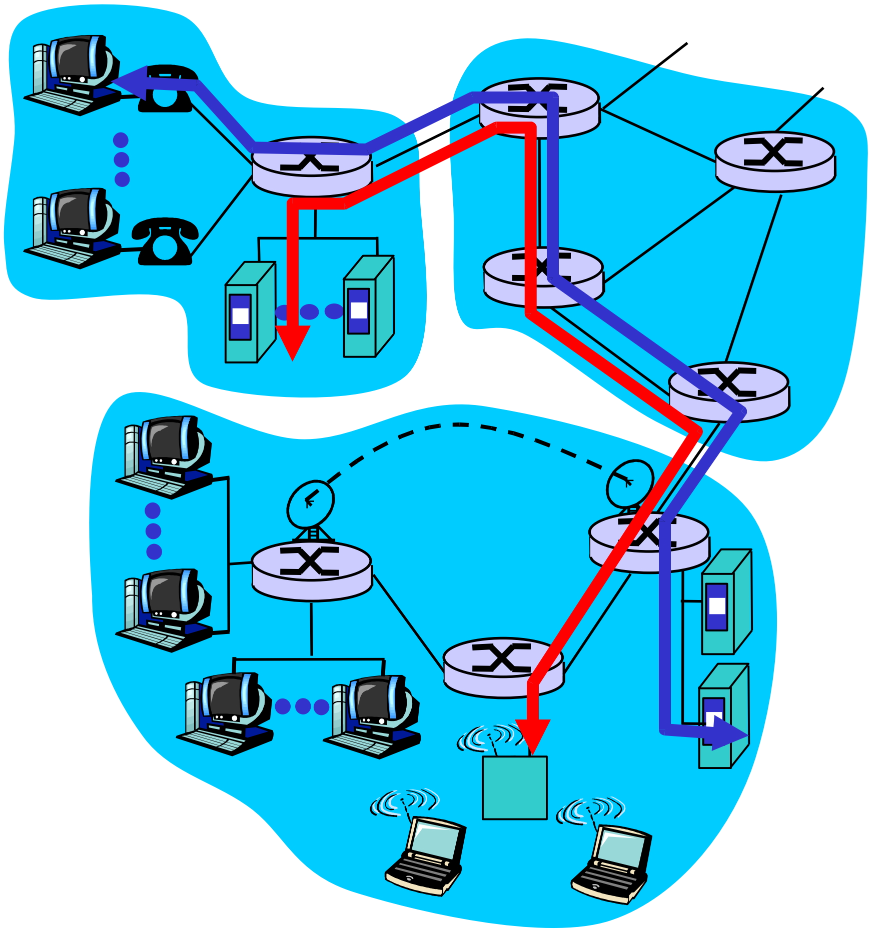
- Mesh of interconnected routers
- *The fundamental question*: how is data transferred through net?
  - circuit-switching: dedicated circuit per call (e.g. telephone network)
  - packet-switching: data sent through net in discrete "chunks"



Network Core: Circuit Switching

End-to-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)*

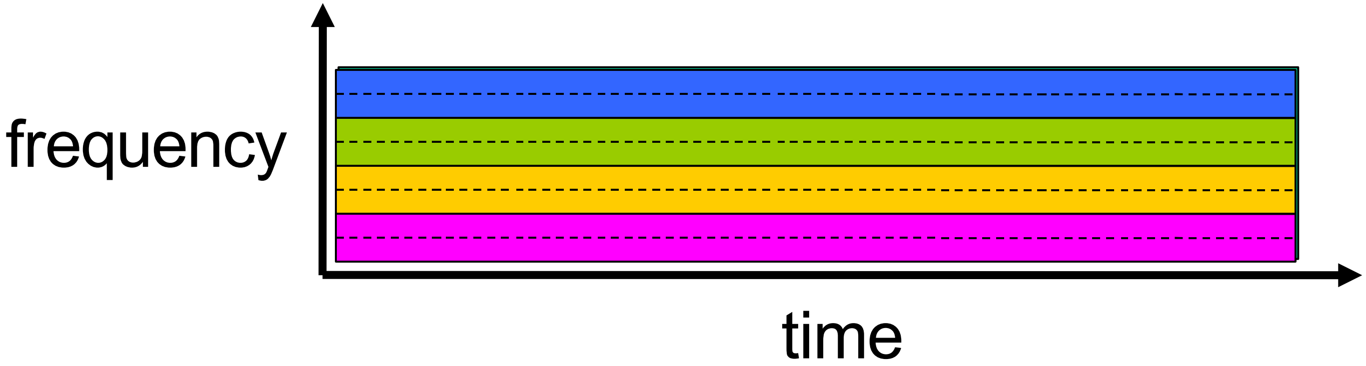
- There are two common ways of dividing link bandwidth into "pieces":
  - frequency division
  - time division

Circuit Switching: FDM and TDM

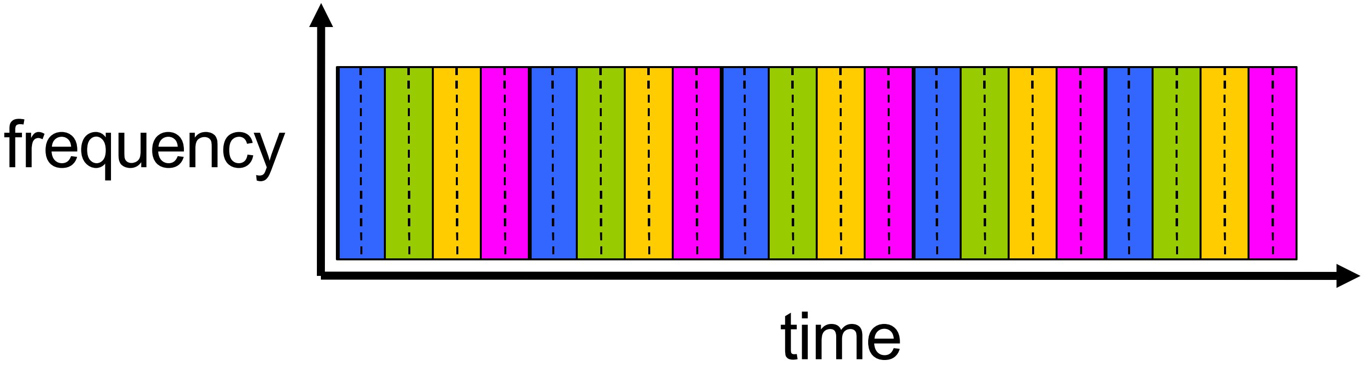
Example:

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!

Network Core: Packet Switching

Each end-to-end data stream is divided into *packets*

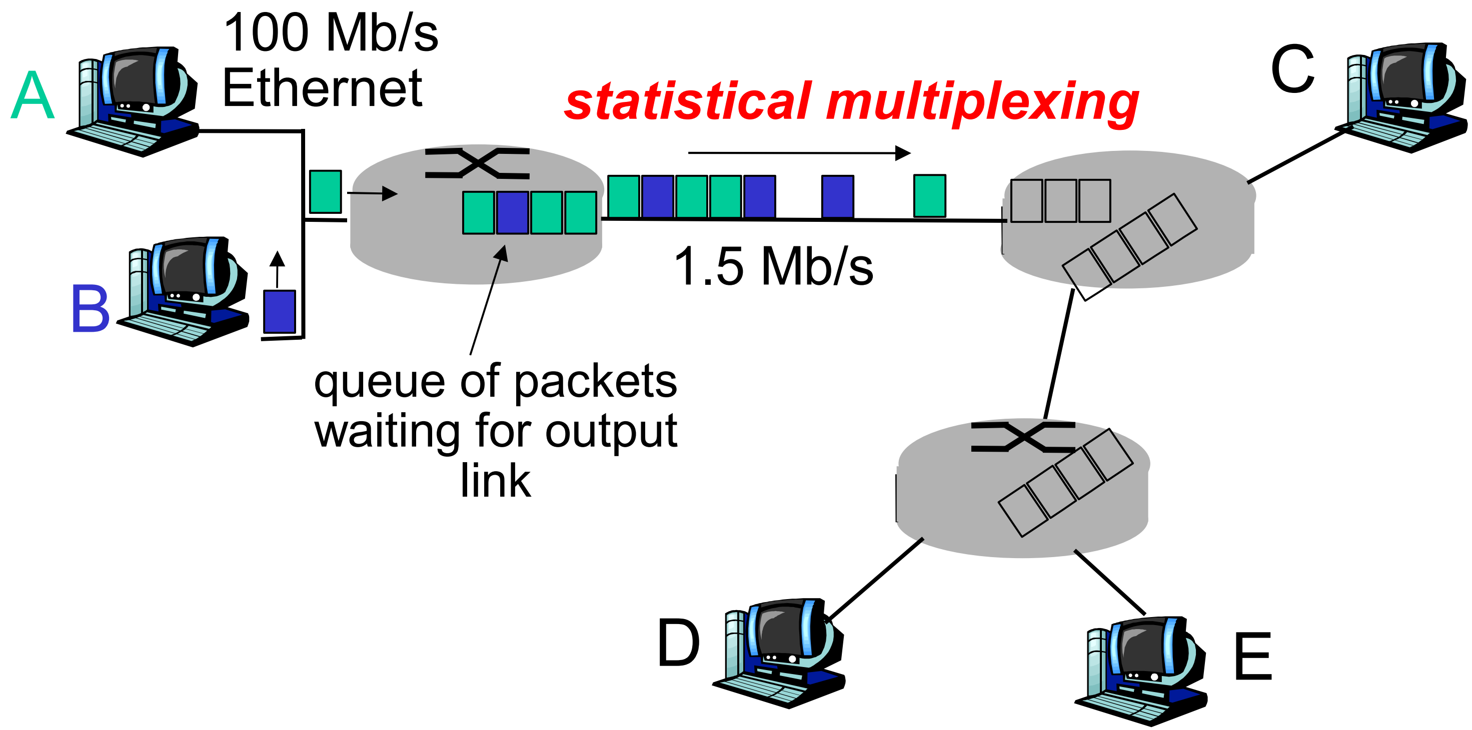
- user A, B packets *share* network resources
- each packet uses full link bandwidth
- resources used *as needed*

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

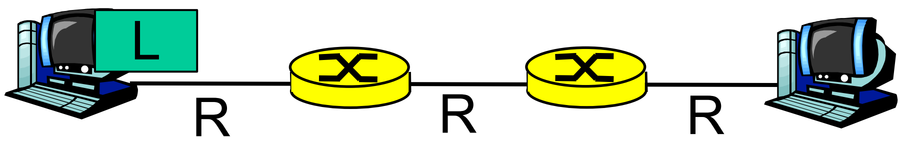
Packet Switching: Statistical Multiplexing



Sequence of A & B packets does not have fixed pattern, 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 of R bps
- Entire packet must arrive at router before it can be transmitted on next link: *store and forward*
- delay = 3L/R (assuming zero propagation delay)

Example:

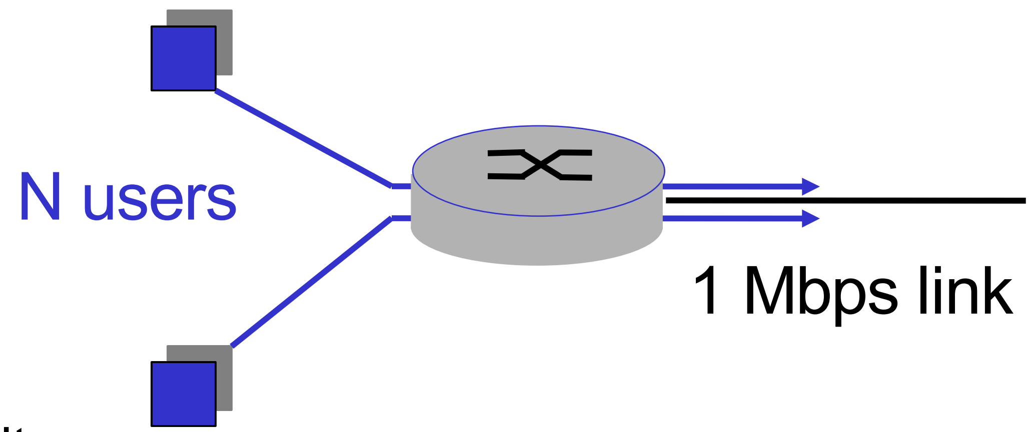
- L = 7.5 Mbits
- R = 1.5 Mbps
- delay = 15 sec
- 3 hops in the route, so packet must be pushed out 3 times

more on delay next time …

Packet switching versus circuit switching

Packet switching allows more users to use the 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 is only .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)

Recap

- Course mechanics
- What is the Internet?
  - hosts, routers, communication links
  - communications services, protocols
- Network Edge
  - client-server, peer-to-peer
  - TCP, UDP
- Network Core
  - Circuit-switched networks
    - FDM
    - TDM
  - Packet-switched networks

Next time

- Finish introduction and overview:
  - Network access and physical media
  - Internet structure and ISPs
  - Delay & loss in packet-switched networks
  - Protocol layers, service models