

CS 332: Computer Networks Introduction

Professor Doug Szajda

Before we do anything else:

- Names:
- The course website: <https://cmsc332-fall-2025.github.io/>
- Slides: I will post them. Not always before class.

Thanks!

- I've taught this course many times, the most recent being Fall 2021. Each time calls for a rethinking of the topics and new material.
- Much of the material I'll use in the course has been supplied by Professor Patrick Traynor of the University of Florida
 - ▶ In fact, some assignments, almost all of the slides, etc, are taken word for word.



Thanks!

- I don't feel bad about this: Patrick took this course with me at UR in the Spring of 2001.
 - ▶ So in a sense, it's payback! (And some of the material was originally mine!)
- Some of the material in these slides is also taken from the instructor material from your text
- So the authors, Jim Kurose and Keith Ross, deserve credit for them

A recurring theme

- I don't always like the images Patrick chose for his slides.
So I replace some.
- The question: Is an image from me or from Patrick?
Something to ponder...

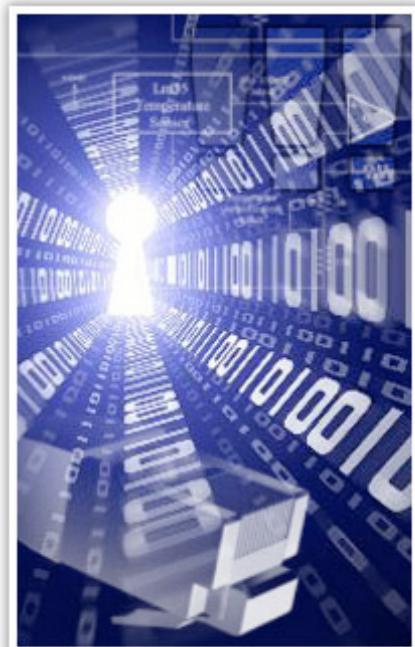
What's this all about?



The Silk Road:
Silk, spices, horses, diamonds
Religion, technology, culture and disease
Shut down around 1400 due to fall of Mongols

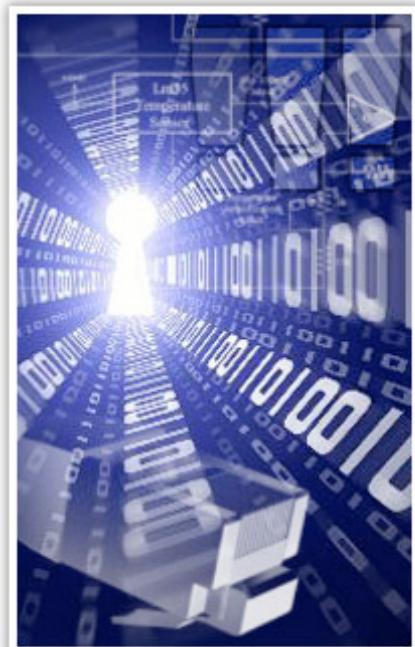
A Modern Day Silk Road

- We live with nearly constant access to the most extensive system ever built by human beings.
 - ▶ *We may never build anything bigger.*
- The Internet quickens the exchange of ideas, goods, news, and improves the quality of life for a large portion of the world's population.
 - ▶ And as we've seen, such instant exchange of information can also cause harm
- It even impacts you where you least expect it.
 - ▶ E.g. Shipping/Supply Chain Management/Appliances



A Modern Day Silk Road

- We live with nearly constant access to the most extensive system ever built by human beings.
 - ▶ *We may never build anything bigger.*
- The Internet quickens the exchange of ideas, goods, news, and improves the quality of life for a large portion of the world's population.
- And think of what may have been had we not had it during the COVID pandemic!



Why Do I Need This Course?

- As engineers and scientists, you need to understand the underpinnings of our global communications networks.
- With this information, you will be able to help design and implement the next generation of networked systems.
- As everything “comes online”, you need to understand the implications (both good and bad) and architecture of these systems.



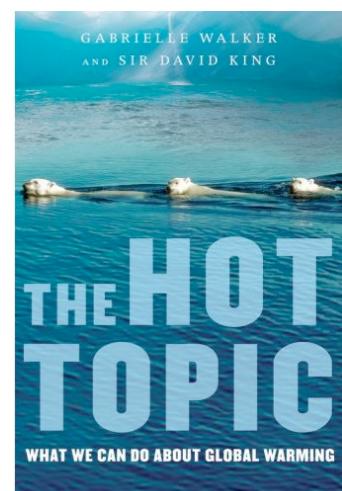
when you look at things
on the internet, the one
question you will ask
is the same one everyone
asks, and to which there
is no answer: WHY
WOULD A PERSON DO
THAT

Goals

- My Goal: *To provide students with the tools to evaluate current, and develop new, networked systems.*
 - ▶ Networking Fundamentals
 - ▶ Recognize trade-offs between different technologies.
 - ▶ Design and implement software with a communications interface.
 - ▶ Prepare you for advanced work in this area.
- I love this material and want to help you all love it too.
- But know that we go beyond just “how the Internet works” to *why* it works the way it does!

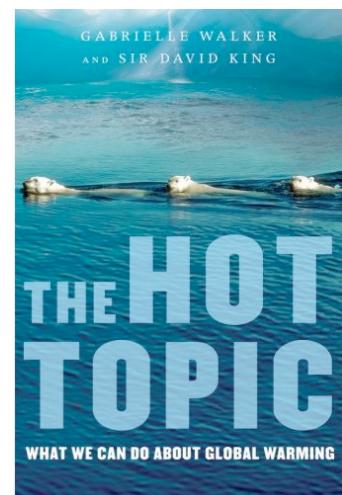
What Topics Will We Cover?

- This is an introductory course, so you will get a bit of everything:
 - ▶ TCP, IP, UDP, DNS, BGP, Email, P2P, Routing Algorithms, Congestion Control, Queuing Fundamentals, Network Management, Wireless, Cellular, Security, Ethics and lots more... (including some new protocols that are gaining widespread use)
- I will be maintaining the course at:
<https://cmsc332-fall-2025.github.io/>
- Assignments, slides and other information will be made available there.



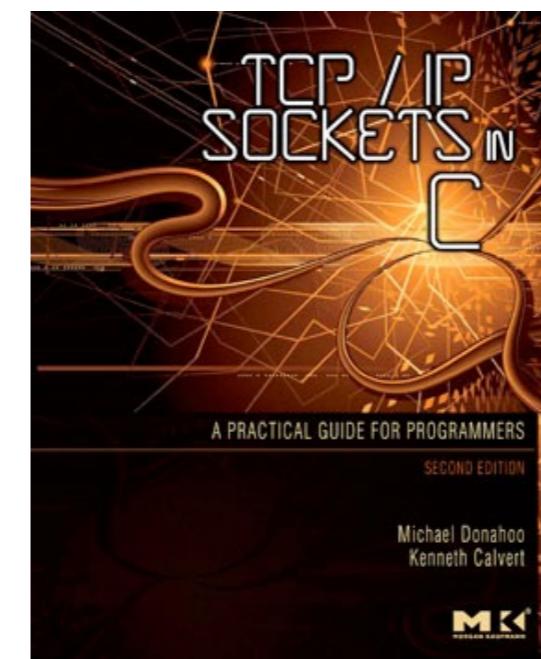
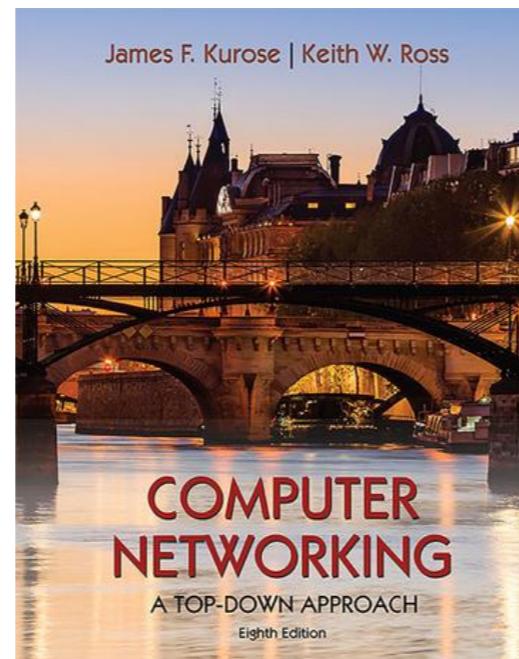
What Topics Will We Cover?

- I will also be creating a course Slack group
 - ▶ I urge you to take advantage of that to communicate with me or with each other
 - ▶ But please be aware, that I won't be responding instantaneously



Textbooks

- There are two *required* books for this class:
 - ▶ Computer Networking:A Top-Down Approach (8th edition)
 - ▶ TCP/IP Sockets in C: Practical Guide for Programmers
- Readings will come from the first; projects will be made easier using the second.



Assignments/Workload

- There will be a reading assignment for nearly every class.
 - ▶ Reading must be done *before* the class period.
- We will have a total of *four* homework assignments:
 - ▶ Problems will come from the book and the professor.
 - ▶ Check the website.
- We will also do *four* programming projects:
 - ▶ All programming must be done in C or C++



Course Calendar

- The course calendar contains a listing of readings, assignments, and deadlines (or at least will as I add to it).
- The page also contains links to additional readings.
- Check frequently! There will be changes! Students are responsible (*I will do my best to make announcements*).

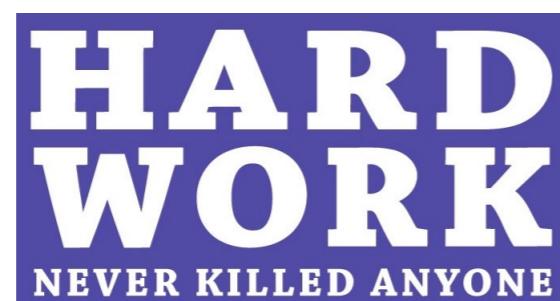
Course Schedule of Topics and Due Dates

- Course schedule is subject to change as the semester progresses.
- All work should be submitted by **11:59PM** (in Richmond, VA) on the date due.

Week	Lecture (Tue)	Lecture (Thr)	Lab (Fri)
1	Aug 26: Placeholder (for now)	Aug 28: Same here	Aug 29:
2	Sept 2:	Sept 4:	Sept 5:
3	Sept 9:	Sept 11:	Sept 12:
4	Sept 16:	Sept 18:	Sept 19:
5	Sept 23:	Sept 25:	Sept 26:
6	Sept 30:	Oct 2:	Oct 3:
7	Oct 7:	Oct 9:	Oct 10:
8	Oct 14: Fall Break	Oct 16:	Oct 17:
9	Oct 21:	Oct 23:	Oct 24:

Expectations

- This is going to be *a challenging course*. The key to success is sustained effort. Failure to keep up with readings and assignments will result in a poor understanding of the material, which is not what any of us wants.
- So what do we get for all our hard work?
 - ▶ Perhaps a step toward helping to change the world (hopefully for the better)...



Grading

- Two midterms, let's discuss
- Final Exam
- Four programming projects
 - The last one typically a more involved group project
- Perhaps a few labs
- Perhaps some ungraded homework
- Why the hand waving? Because I'm changing my grading structures and still thinking through how I'd like to do it. More on this in the coming weeks.

Lateness

- Exams are take home with a strict due date/time
- Projects will be submitted via Github Classroom.
 - ▶ You'll check out a starter repo, just as you all did in CS 240
- Late assignments are assessed a 10% per-day late penalty, with a maximum of four days.
- Students with legitimate reasons should contact the professor before the deadline to apply for an extension.
 - ▶ Failure to start on time is YOUR issue!



Academic Integrity

- Academic dishonesty, whether from *cheating, copying, fabricating results* or through *any other dishonest practice* will not be tolerated.
 - I take this personally - you should too.
- I assume by default that you are honest. Please don't disappoint me.

Course Outline

- Introduction to Networking (Chapter 1)
- Application layer (Chapter 2)
- Transport layer (Chapter 3)
- Network layer (Chapters 4 & 5)
- Link layer (some physical layer topics) (Chapter 6)
- Wireless, Mobility and Android (Chapter 7)
- Network security (Chapter 8)

So this is a networks class...

- What might that mean?
- Well, before we can answer the questions we'll consider, perhaps we should consider what questions we need to ask.
- Thought experiment. You have two devices, A and B, connected by two wires. One in each direction. These wires transmit information in binary. Messages can be transmitted from A to B or from B to A at any time. What kind of questions do you need to ask and solve in order to make this work?
 - Break into groups of 2 to discuss.

So this is a networks class...

- Thought experiment. You have two devices, A and B, connected by two wires. One in each direction. These wires transmit information in binary. Messages can be transmitted from A to B or from B to A at any time. What kind of questions do you need to ask and solve in order to make this work?
- What did we ignore here?

An Aside

- Labs periods: I neglected to mention this on Tuesday.

So this is a networks class...

- Thought experiment. You have two devices, A and B, connected by two wires. One in each direction. These wires transmit information in binary. Messages can be transmitted from A to B or from B to A at any time. What kind of questions do you need to ask and solve in order to make this work?
- What did we ignore here? What if device A has a processor that operates at 4 GHz, but device B has a processor that operates at 520 MHz?
 - How does that change things? Or perhaps put differently, what might now be required?

So this is a networks class...

- What might that mean?
- Well, before we can answer the questions we'll consider, perhaps we should consider what questions we need to ask.
- Another question: You now have many devices. Not all are directly connected to each other, but there is a path from any device to any other device. What questions do you need to ask and answer now?
 - ▶ Have at it. Hint: Networks are systems for transporting data. Thinking of other types of transportation systems might be helpful here.

Chapter I: Roadmap

I.1 What is the Internet?

I.2 Network edge

I.3 Network core

I.4 Delay & loss in packet-switched networks

I.5 Protocol layers and their service models

I.6 Networks Under Attack

I.7 History of Computer Networking and the Internet

I.8 Summary

What's the Internet: “Nuts and Bolts” View

millions of connected computing devices:

hosts = end systems

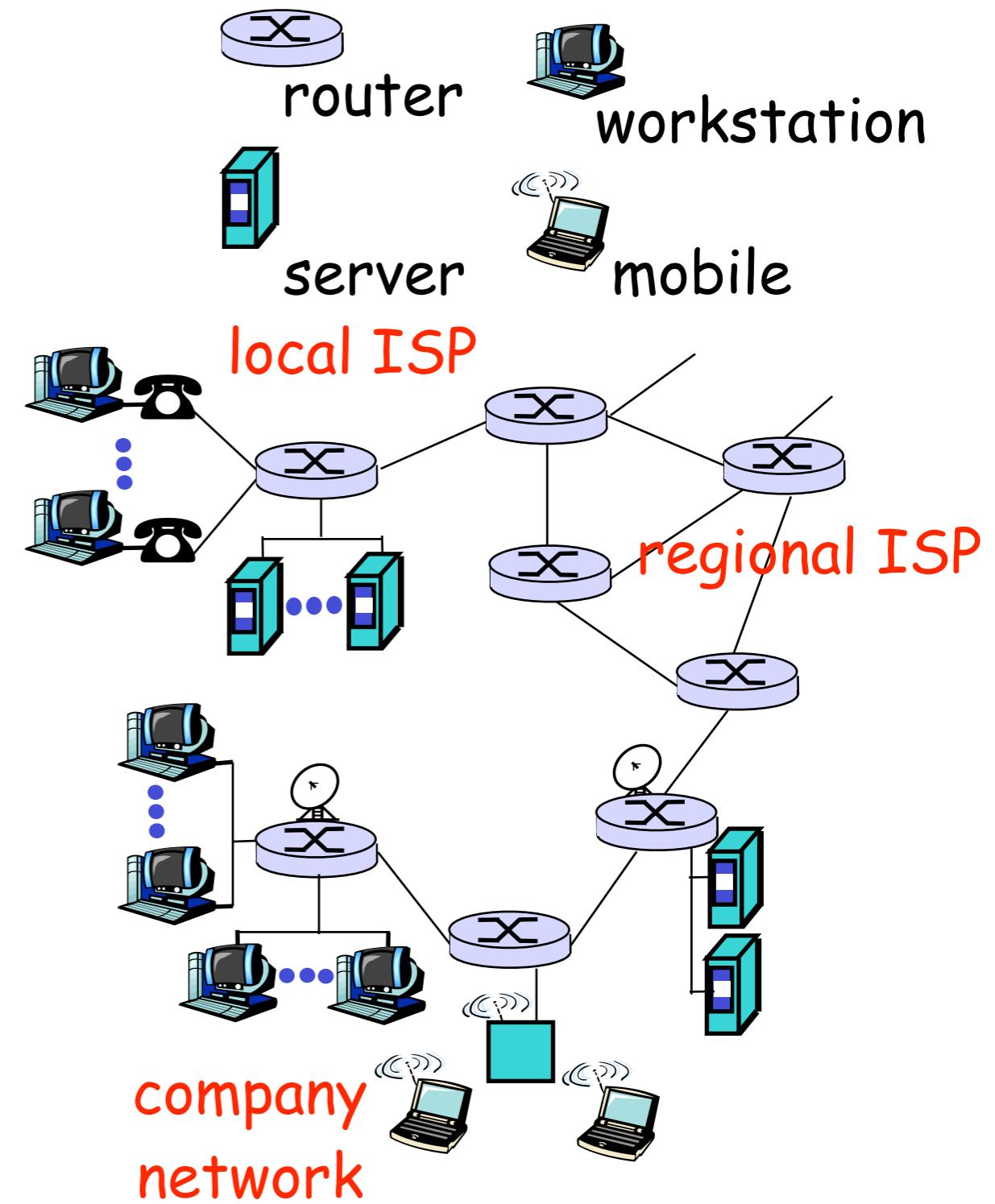
running network apps

communication links

fiber, copper, radio, satellite

transmission rate = bandwidth

routers: forward packets
(chunks of data)



Cool Internet Appliances

“Fun” Internet-connected devices



Amazon Echo



Internet refrigerator



Security Camera



Internet phones



IP picture frame



Slingbox: remote control cable TV



Gaming devices



sensorized, bed mattress



Fitbit



diapers



Tweet-a-watt:
monitor energy use



bikes



cars



scooters

Pacemaker & Monitor



Web-enabled toaster +
weather forecaster

Others?

Introduction: 1-4

What's the Internet: “Nuts and Bolts” View

protocols control sending,
receiving of msgs

e.g., TCP, IP, HTTP, FTP, PPP

Internet:
“network of networks”

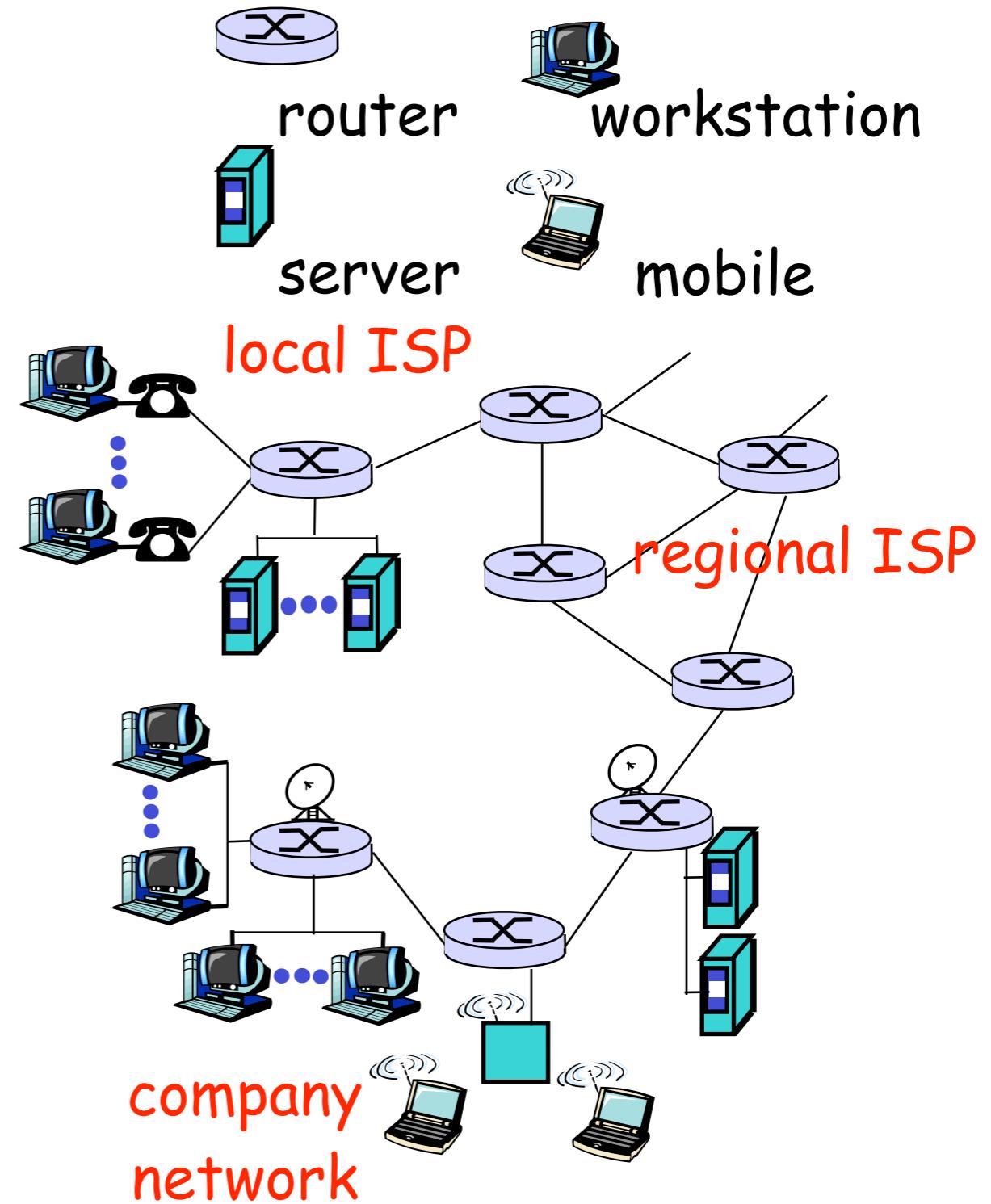
loosely hierarchical

public Internet versus private intranet

Internet standards

RFC: Request for comments

IETF: Internet Engineering Task Force
(Check out IETF website)



What's the Internet:A Service View

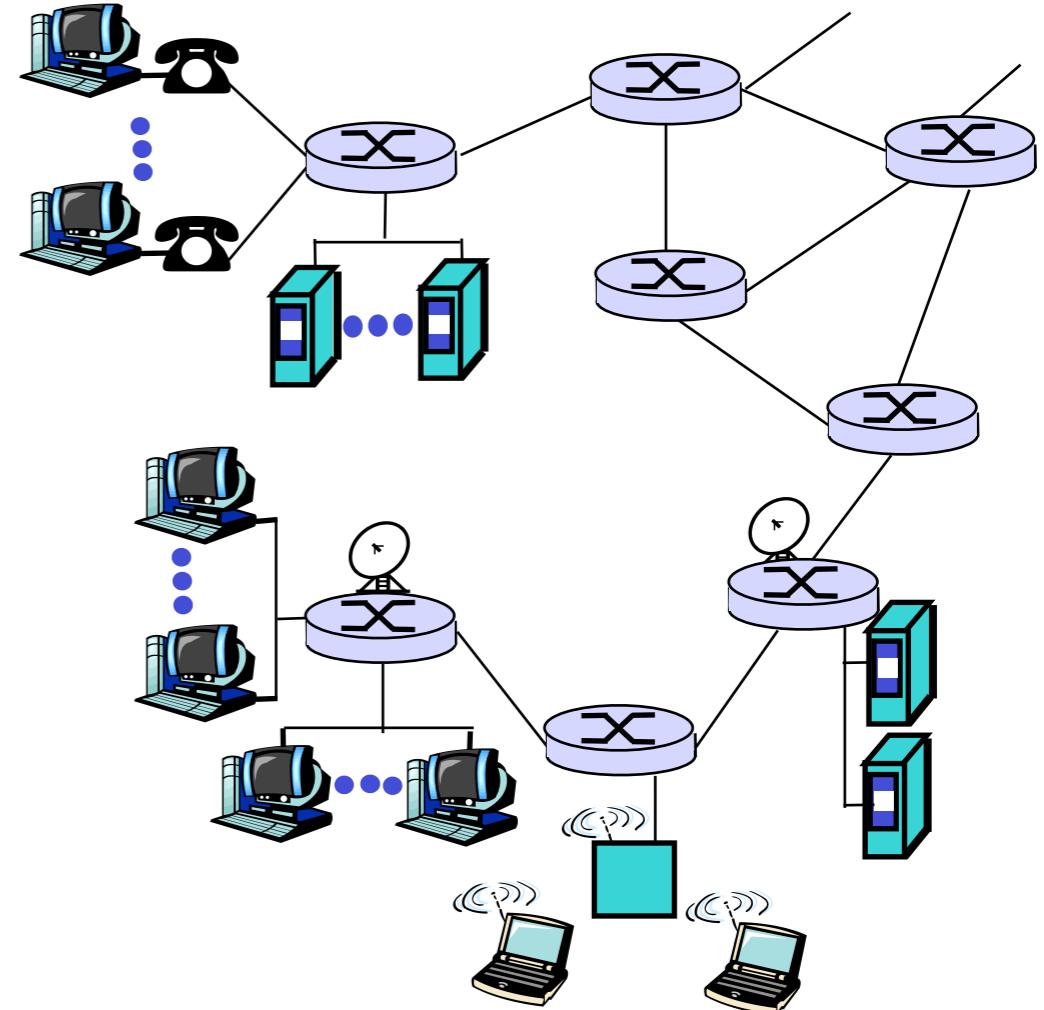
**communications infrastructure
enables distributed applications:**

Web, email, games, e-commerce, file sharing, conferencing (Zoom)

communication services provided to apps:

Connectionless **unreliable**

connection-oriented **reliable**



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

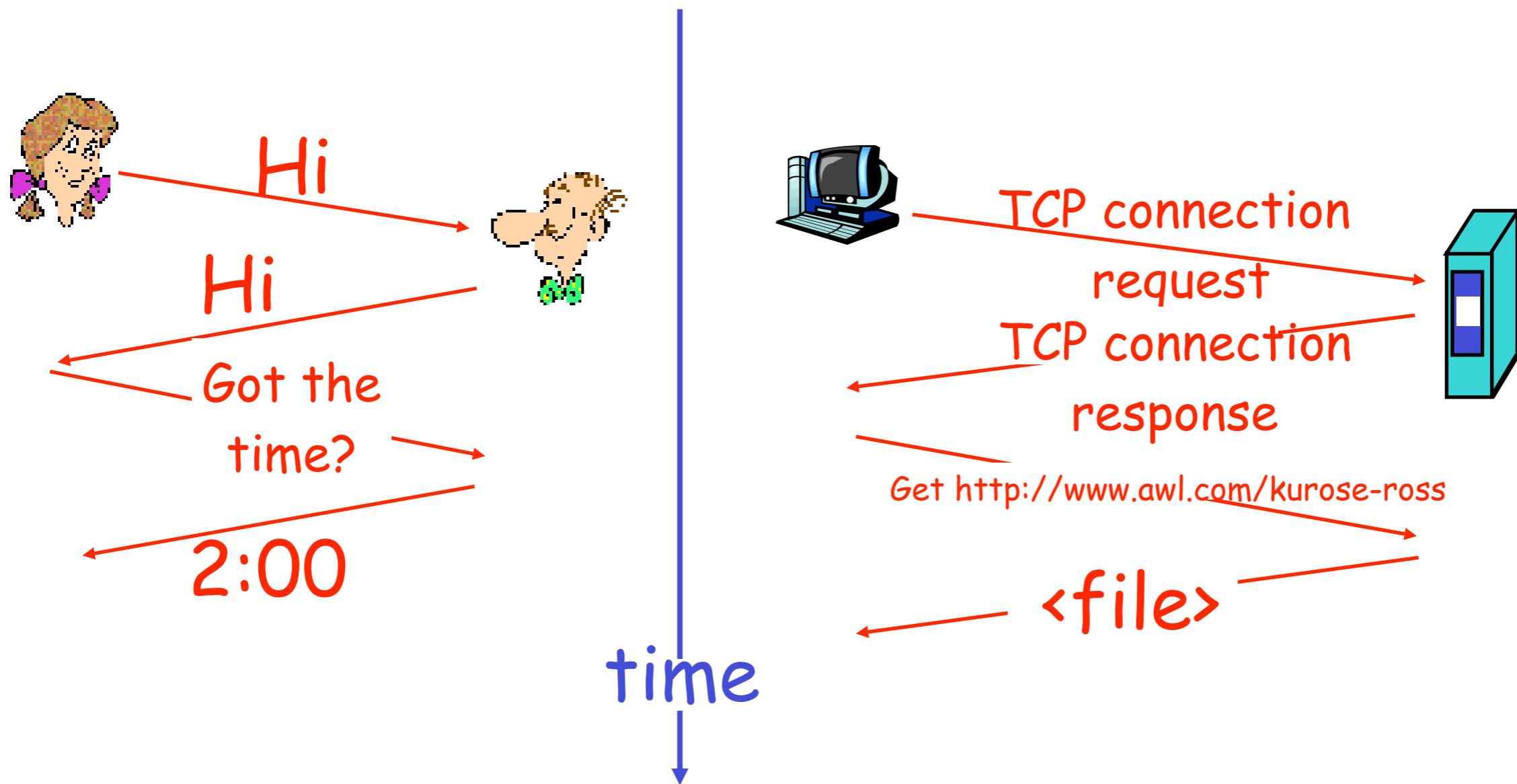
network protocols:

- 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?

- Example: A human protocol and a computer protocol:



- Question: What are some other human protocols?

Chapter I: Roadmap

I.1 What is the Internet?

I.2 Network edge

I.3 Network core

I.4 Delay & loss in packet-switched networks

I.5 Protocol layers and their service models

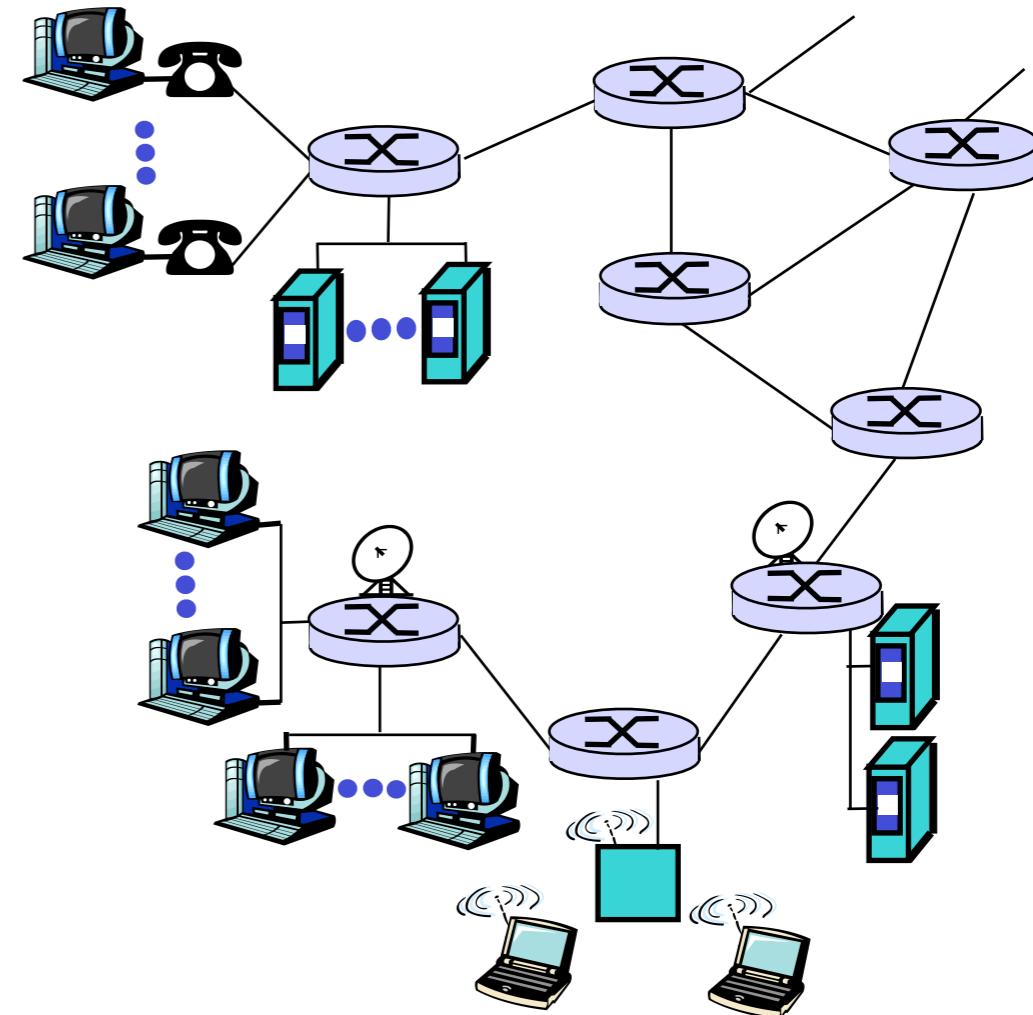
I.6 Networks Under Attack

I.7 History of Computer Networking and the Internet

I.8 Summary

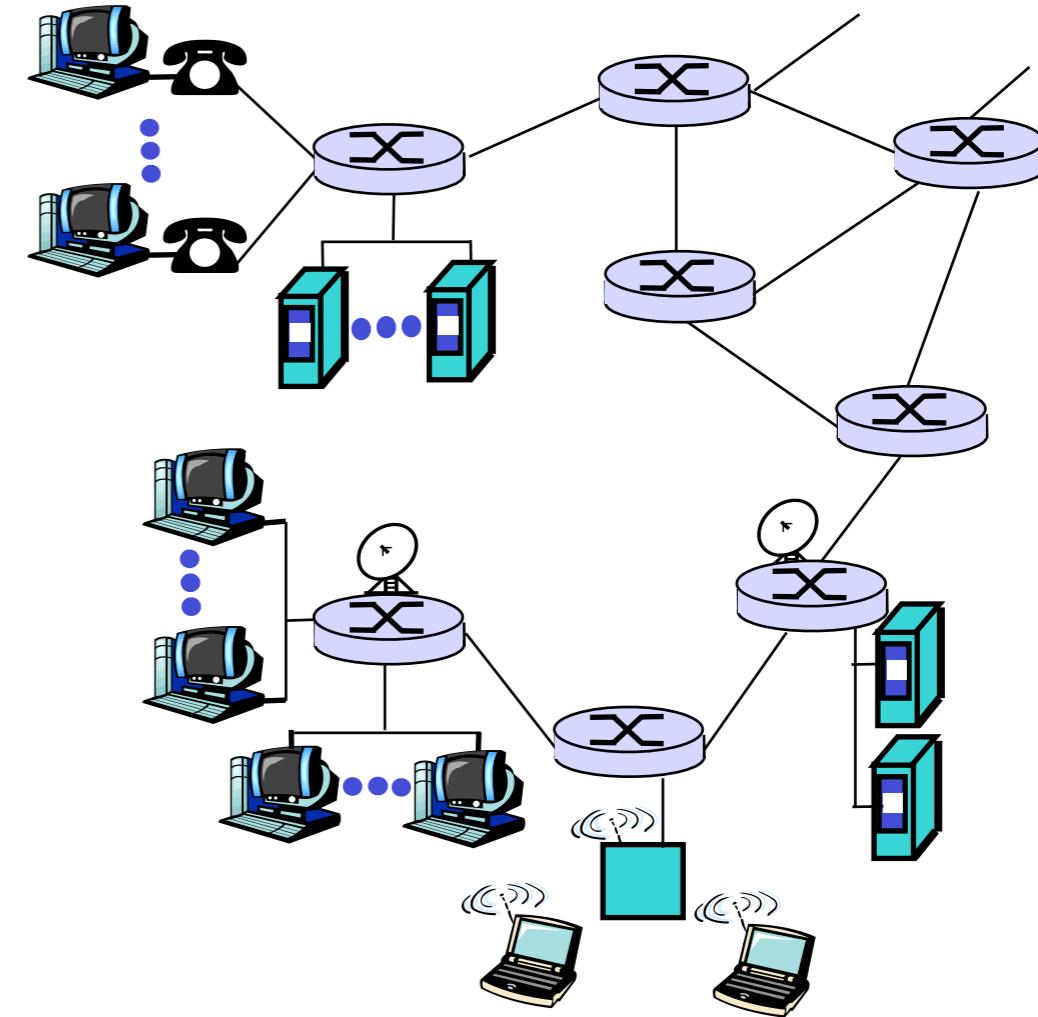
A Closer Look at Network Structure:

- **network edge:**
applications and hosts
 - **network core:**
 - ▶ routers
 - ▶ network of networks
 - **access networks,
physical media:**
communication links
-
- Guided vs unguided



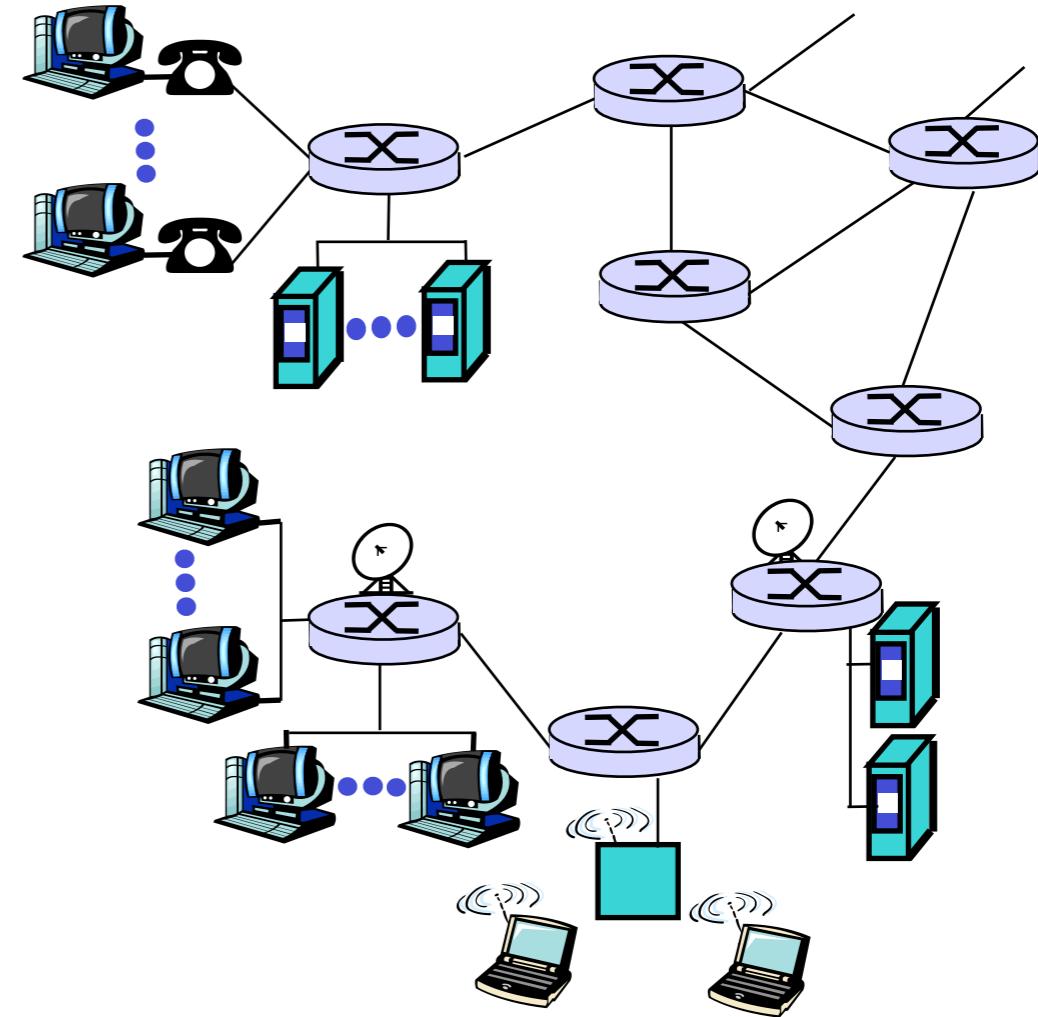
A Closer Look at Network Structure:

- **network edge:**
applications and hosts
- **network core:**
 - ▶ routers
 - ▶ network of networks
- **access networks:** the network the physically connects end system to the first router



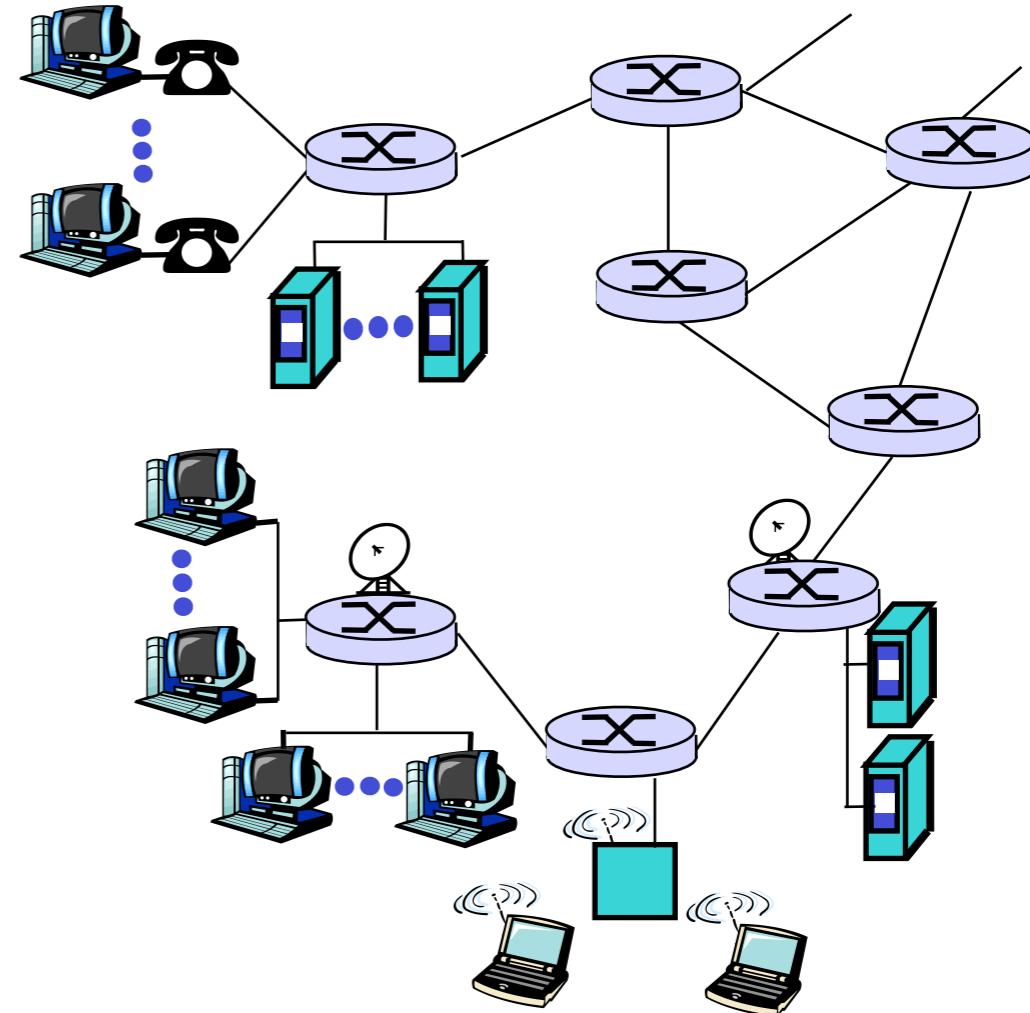
A Closer Look at Network Structure:

- **network edge:** applications and hosts
- **network core:**
 - ▶ routers
 - ▶ network of networks
- **physical media:** twisted-pair copper wire, coaxial cable, multimode fiber-optic cable, terrestrial radio spectrum, satellite radio spectrum communication links



A Closer Look at Network Structure:

- **network edge:**
applications and hosts
 - **network core:**
 - ▶ routers
 - ▶ network of networks
 - **access networks,
physical media:**
communication links
-
- Guided vs unguided



The Network Edge:

- **end systems (hosts):**

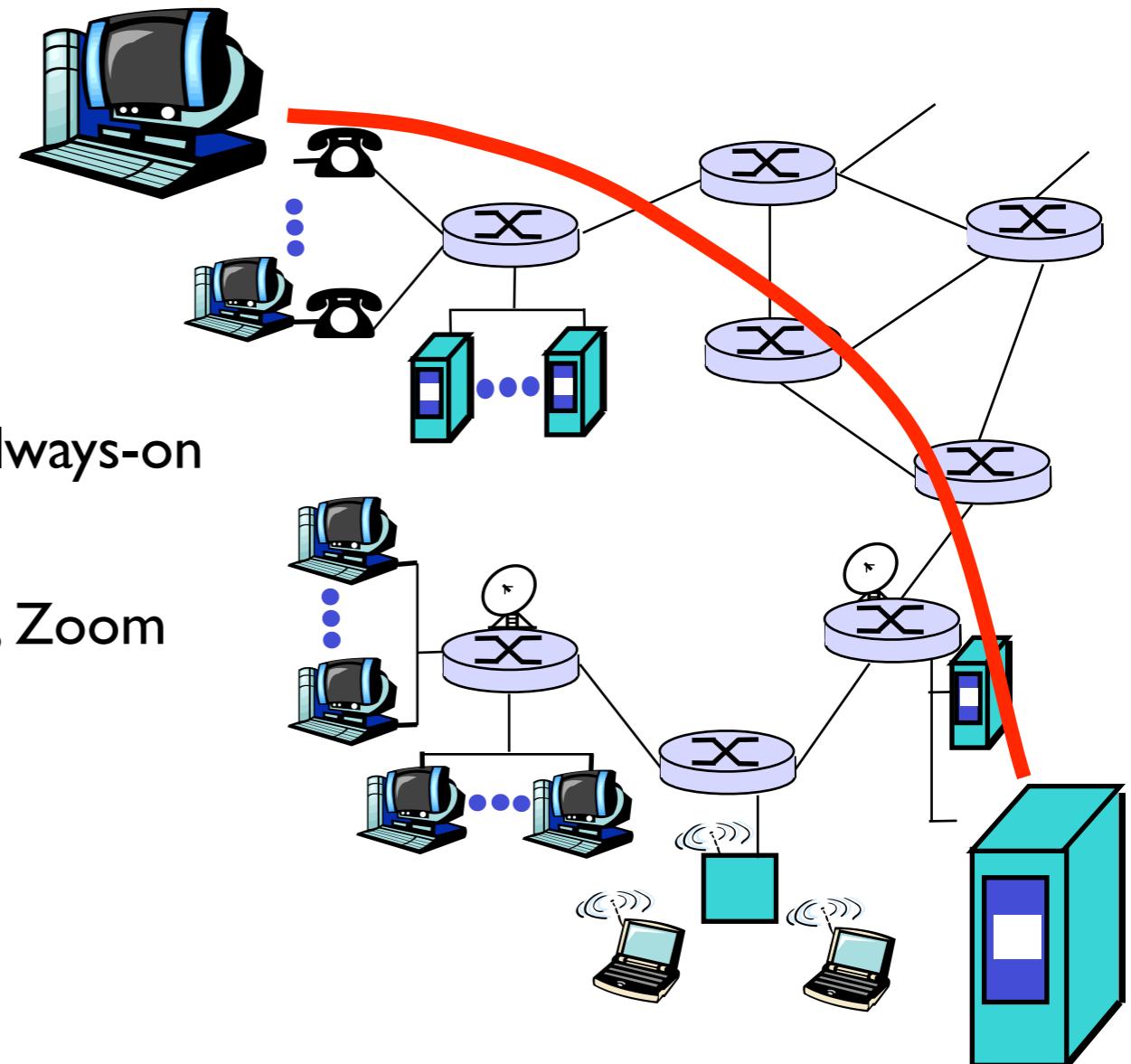
- ▶ run application programs
- ▶ e.g. Web, email, at “edge of network”

- **client/server model**

- ▶ client host requests, receives service from always-on server
- ▶ e.g. Web browser/server; email client/server, Zoom

- **peer-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

App's using TCP:

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

App's using UDP:

- streaming media, teleconferencing, DNS, Internet telephony

Chapter I: Roadmap

I.1 What is the Internet?

I.2 Network edge

I.3 *Network core*

I.4 Delay & loss in packet-switched networks

I.5 Protocol layers and their service models

I.6 Networks Under Attack

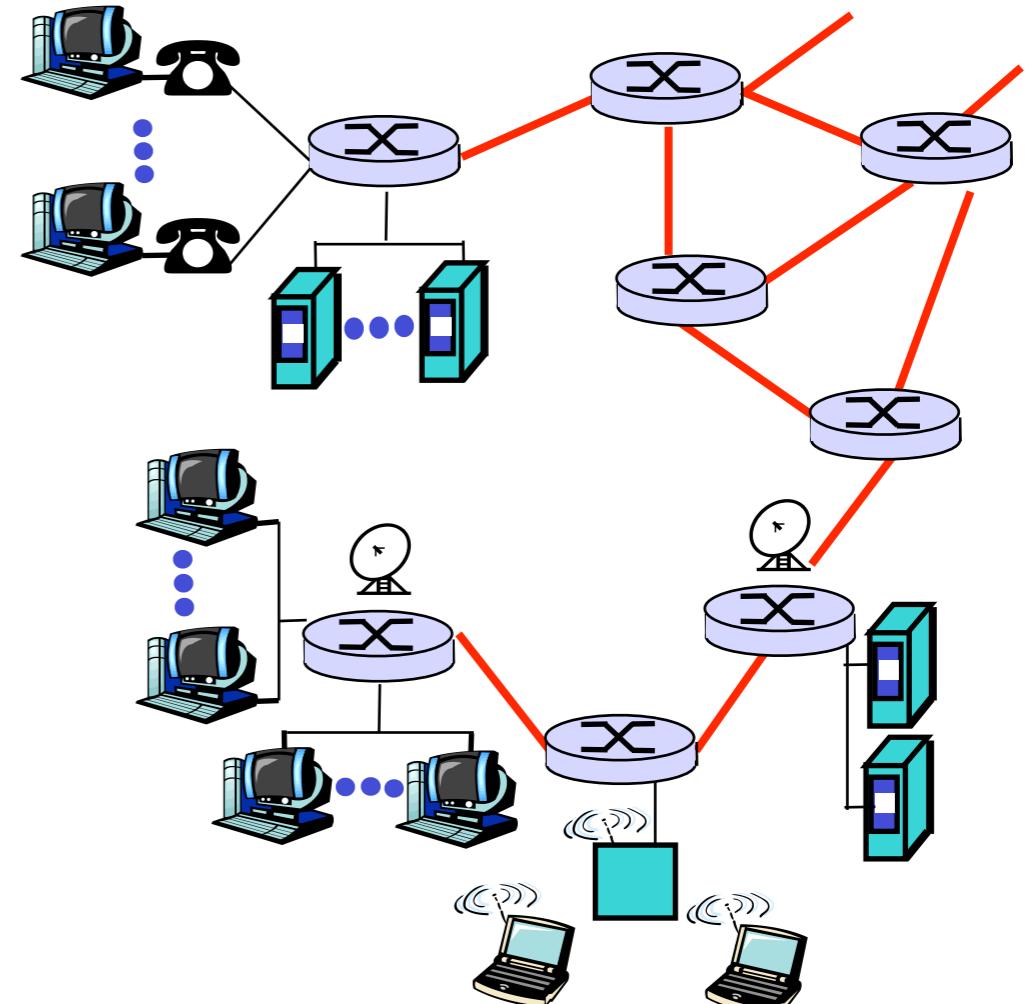
I.7 History of Computer Networking and the Internet

I.8 Summary

The Network Core

- mesh of interconnected routers
- **the fundamental question:**
how is data transferred through net?

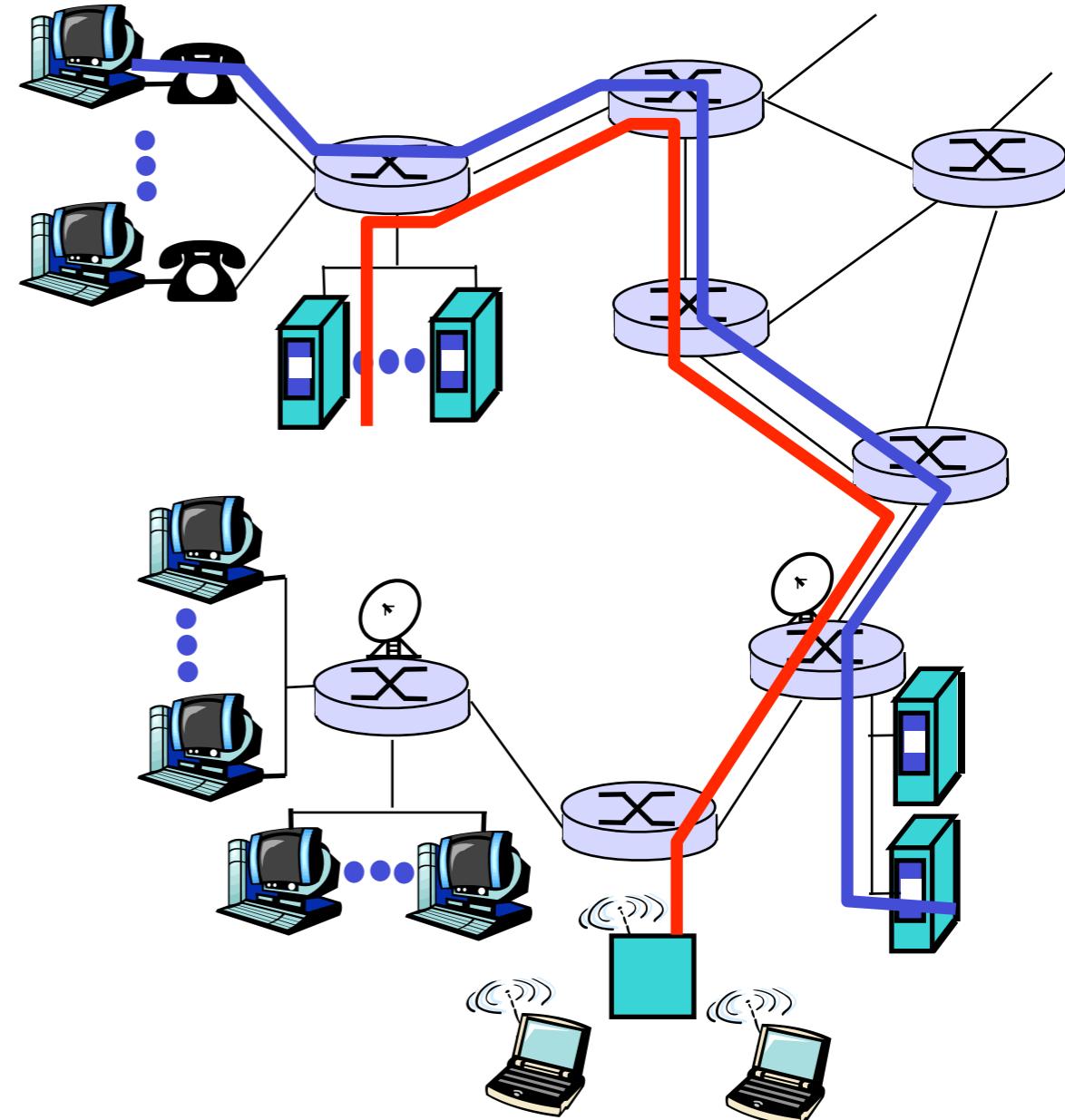
- ▶ **circuit switching:**
dedicated circuit per call:
telephone net
- ▶ **packet-switching:** data
sent thru net in discrete
“chunks”



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

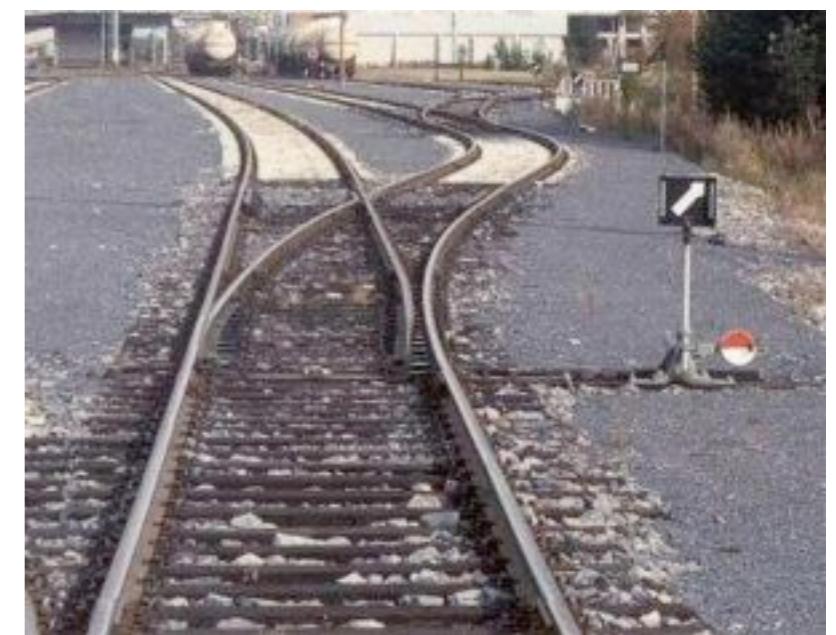
network resources
(e.g., bandwidth)

divided into “pieces”

- pieces allocated to calls

- resource piece *idle* if not used by owning call
(no sharing)

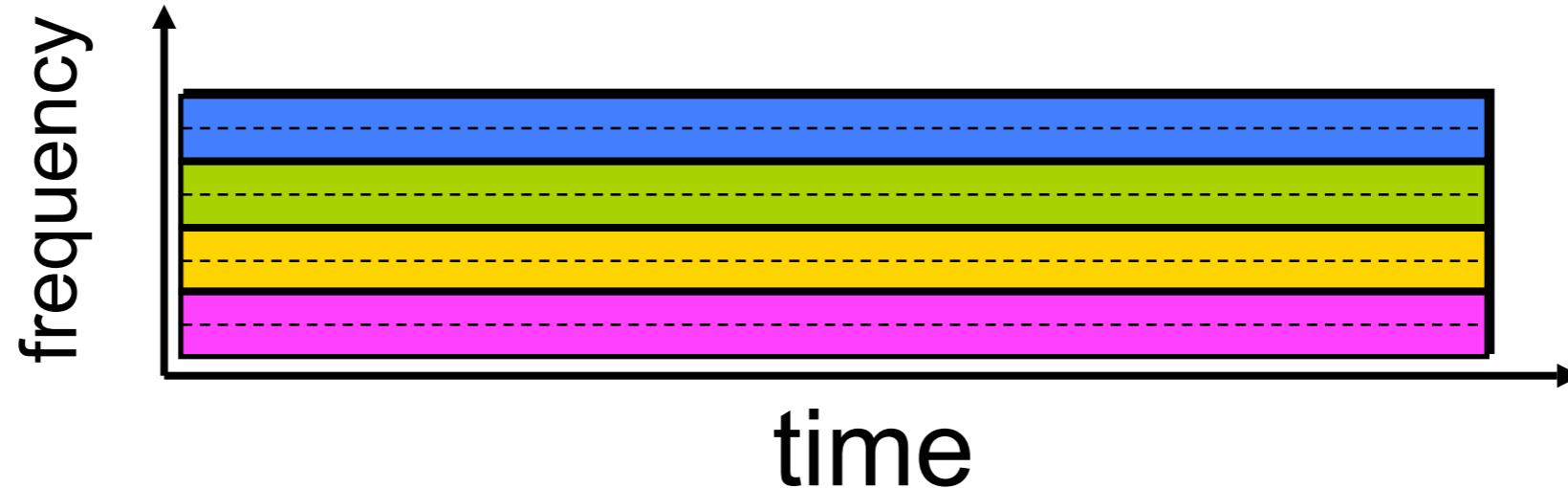
- dividing link bandwidth into “pieces”
 - frequency division
 - time division



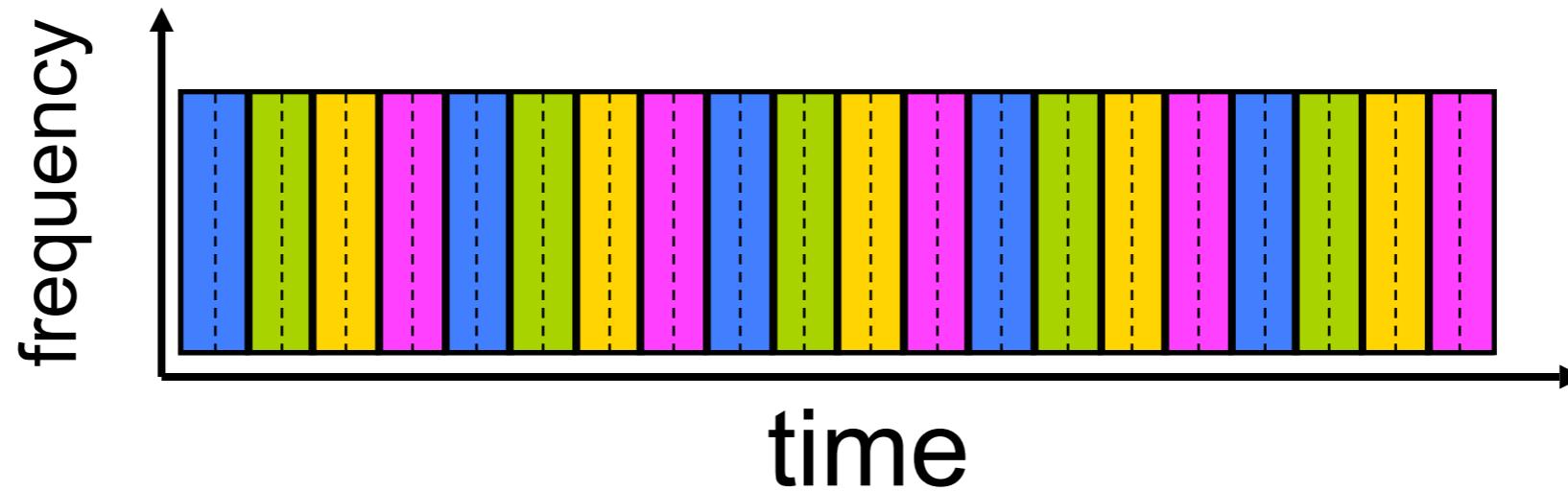
Circuit Switching: FDM and TDM

Example:

F
D
M



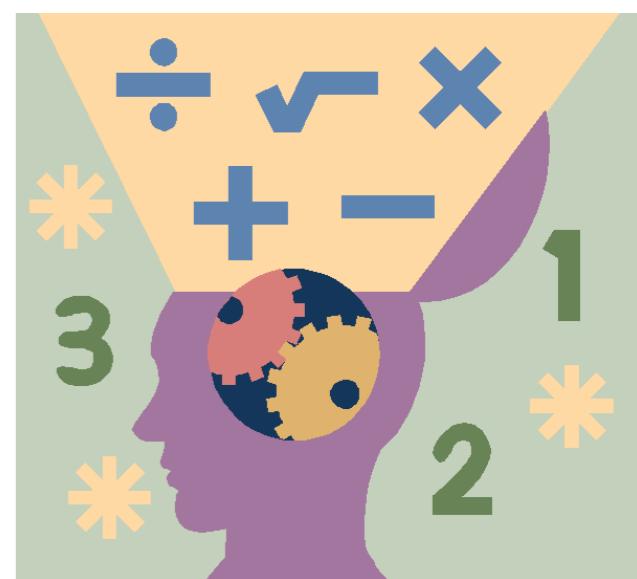
T
D
M



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-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

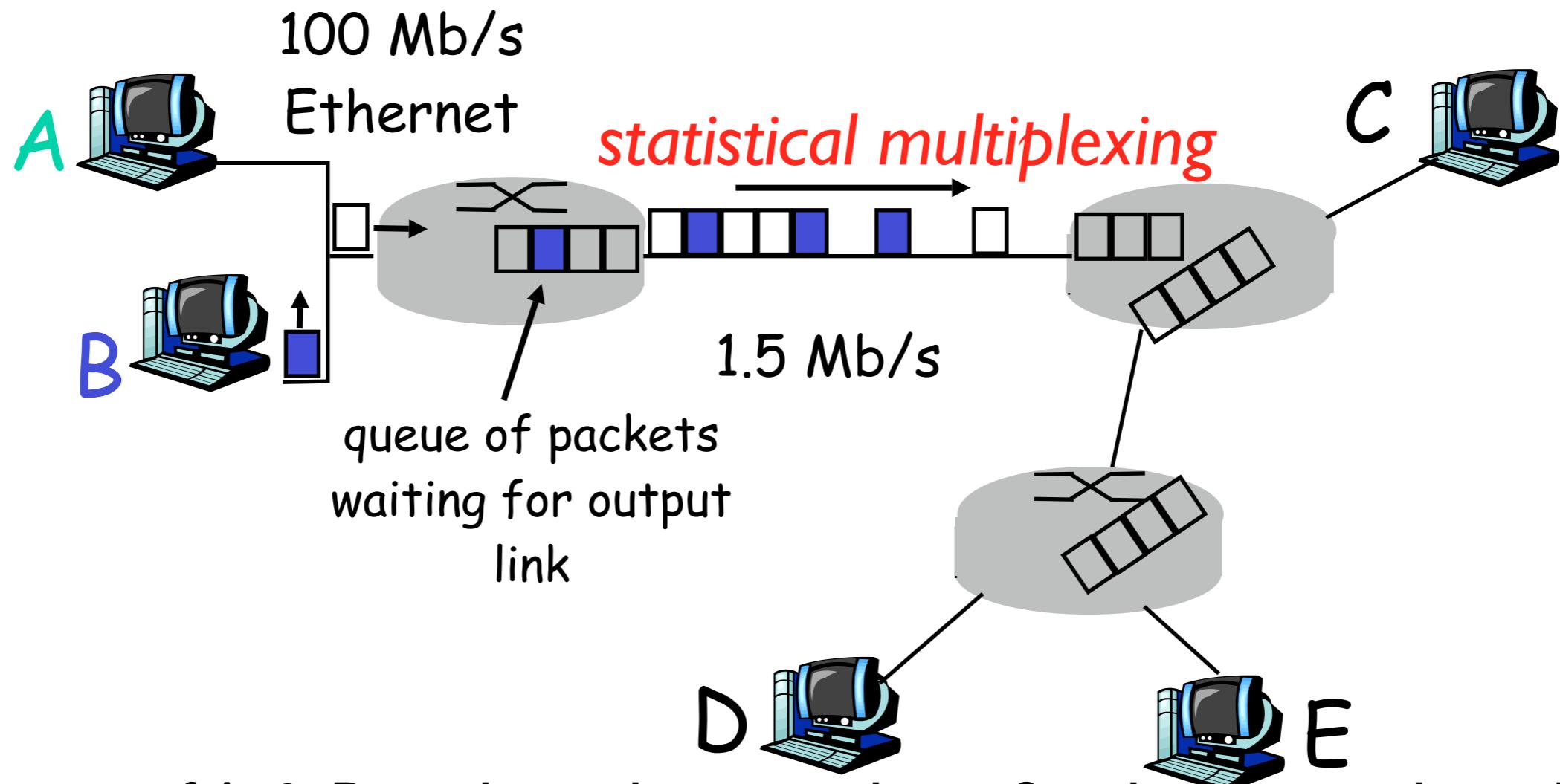
resource contention:

- 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

Question:

- Which do you think would be better for the Internet and why?
 - Take 5 minutes to think about this. I want justification here!

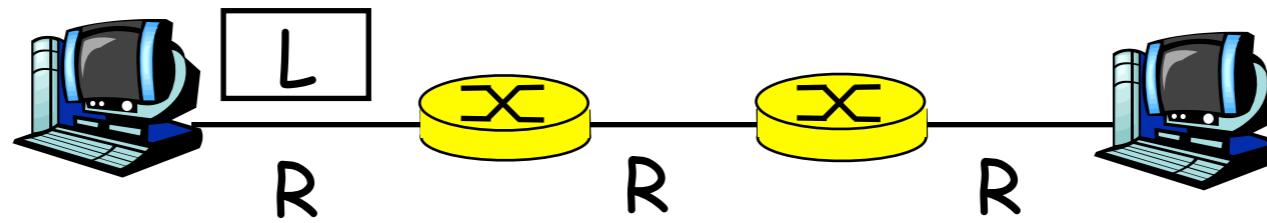
Packet Switching: Statistical Multiplexing



Sequence of A & B packets does not have fixed pattern, shared on demand : ***statistical multiplexing*** (slots allocated dynamically according to demand)

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

Packet-Switching: Store and Forward

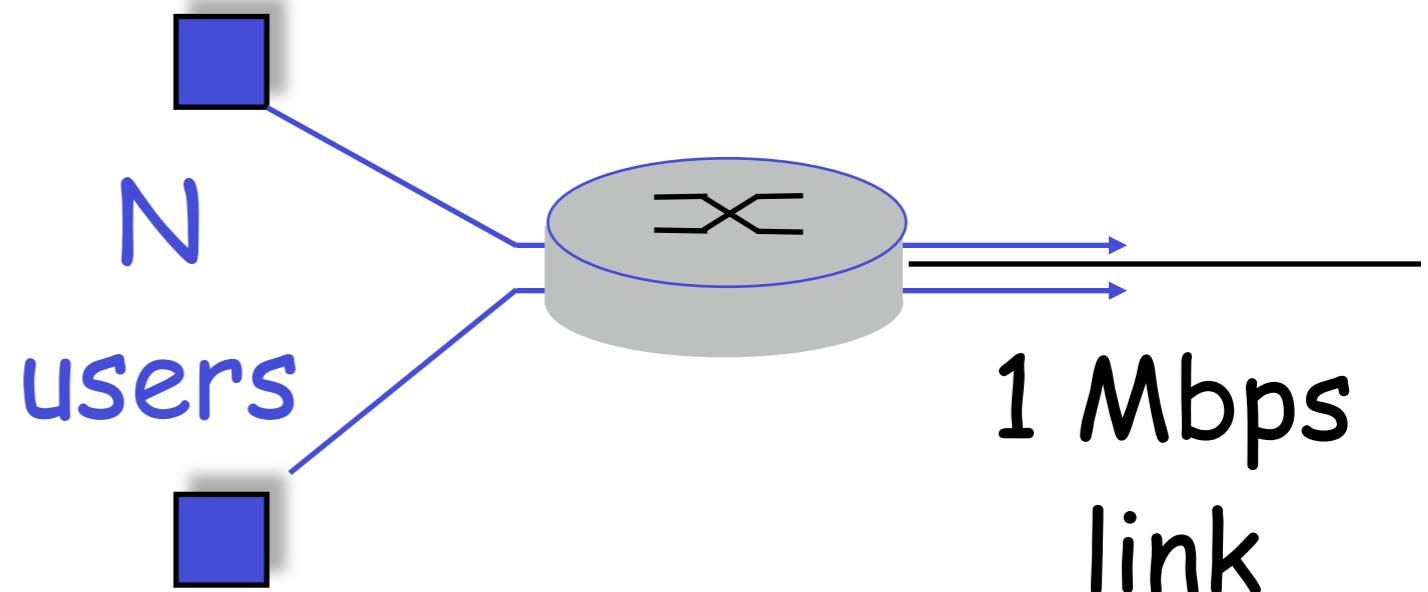


- Takes L/R seconds to transmit (push out) packet of L bits on to link of R bps
 - Entire packet must arrive at router before it can be transmitted on next link:
store and forward
 - $\text{delay} = 3L/R$ (assuming zero propagation delay)
- Example:*
- $L = 7.5 \text{ Mbits}$
 - $R = 1.5 \text{ Mbps}$
 - $\text{delay} = 15 \text{ sec}$
- more on delay shortly ...

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
- packet switching:
 - ▶ with 35 users, probability > 10 active less than .0004



Q: how did we get value 0.0004?

Conclusion

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

Conclusion

- Welcome to CMSC 332 - this will be a great class.
- Go get the books and start doing the reading (if you haven't already)!
- Regularly visit the webpage to figure out when homeworks and projects will be due.
- Questions?



"Okay, folks, that's a wrap!"