



Chair for Network Architectures and Services – Prof. Carle
Department of Computer Science
TU München

Master Course Computer Networks IN2097

**Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.**

**Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
<http://www.net.in.tum.de>**



Technische Universität München



Course Outline (tentative)

- Part 1: Internet protocols
 1. Overview on Computer Networks
 2. Application Layer
 3. Transport Layer
 4. Network Layer
 5. Link Layer
 6. Wireless and Mobile Networks
 7. Interactive Communication – Voice and Video Services
- Part 2: Advanced Computer Networks Principles
 8. Network Monitoring and Measurements
 9. Network design principles
 - *common themes*: signaling, indirection, virtualization, multiplexing, randomization, scalability
 - *implementation principles*: techniques
 - *network architecture*: the big picture, synthesis
 - *Future Internet* approaches
 10. Network Simulation (if time permits)



Acknowledgements

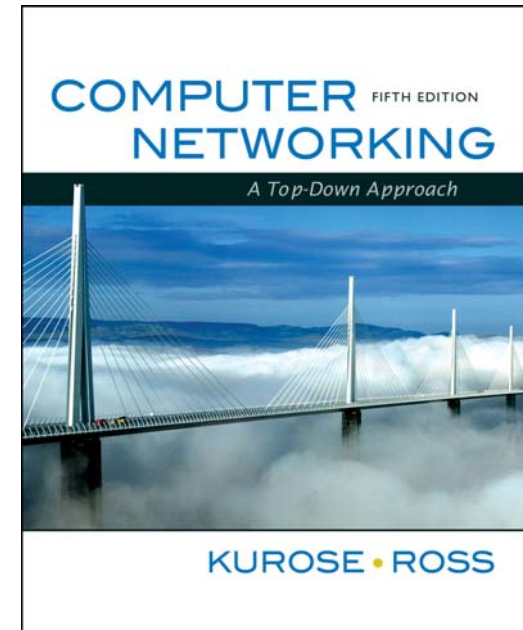
- *Part 1 of this lecture is based on the book*
Computer Networking: A Top Down Approach,
5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.
- The lecture is based to a significant extent
on slides by Jim Kurose and Keith Ross



Jim Kurose, University of
Massachusetts, Amherst



Keith Ross
Polytechnic Institute of
New York University





Course organization

- Lecture
 - Monday, 16:15-17.45, MI H2 first weekly, then typically bi-weekly
 - Friday, 10:15-11.45, MI H2 weekly
- Exercises
 - After start of exercises, typically bi-weekly Monday 16:15-17.45, MI 00.08.038
- Students are requested to subscribe to lecture and exercises at <http://www.net.in.tum.de/en/teaching/ws0910/lectures/masterkurs-rechnernetze/>
- Email list, svn access
 - for subscribers of course
- Questions and Answers / Office hours
 - Prof. Dr. Georg Carle, carle@net.in.tum.de
 - After the course and upon appointment (typically Thursday 11-12)
 - Christian Grothoff, Ph.D., grothoff@net.in.tum.de
- Course Material
 - Slides are available online. Slides may be updated during the course.



Grading

- Students are requested to subscribe to lecture and exercises at <http://www.net.in.tum.de/en/teaching/ws0910/lectures/masterkurs-rechnernetze/>
- Exercises
 - Successfully participating at exercises gives a bonus of 0,3 for overall grade
- Practical assignments
 - Two practical assignments are planned
 - You have to succeed in at least one
 - They will be graded
- Our concept for grading
 - Final examinations will be oral and give an individual grade. You must pass the oral exam for being successful in the course.
 - For overall grade, grade of one practical assignment gives 25% of final grade
 - If your grade for a second practical assignment is better than your oral grade, it is accounted for by another 25%



Questions

- Who studies what?
 - Diploma degree?
 - Master in Informatics? Master in Information Systems [Wirtschaftsinformatik]?
 - Other Master courses?
 - Bachelor in Informatics?
- Which previous relevant courses?
 - IN0010 - Grundlagen Rechnernetze und Verteilte Systeme?
 - What else?



Chapter 1: Introduction

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
- ❑ protocol layers, service models

Our goal:

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



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




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

-  PC
-  server
-  wireless laptop
-  cellular handheld

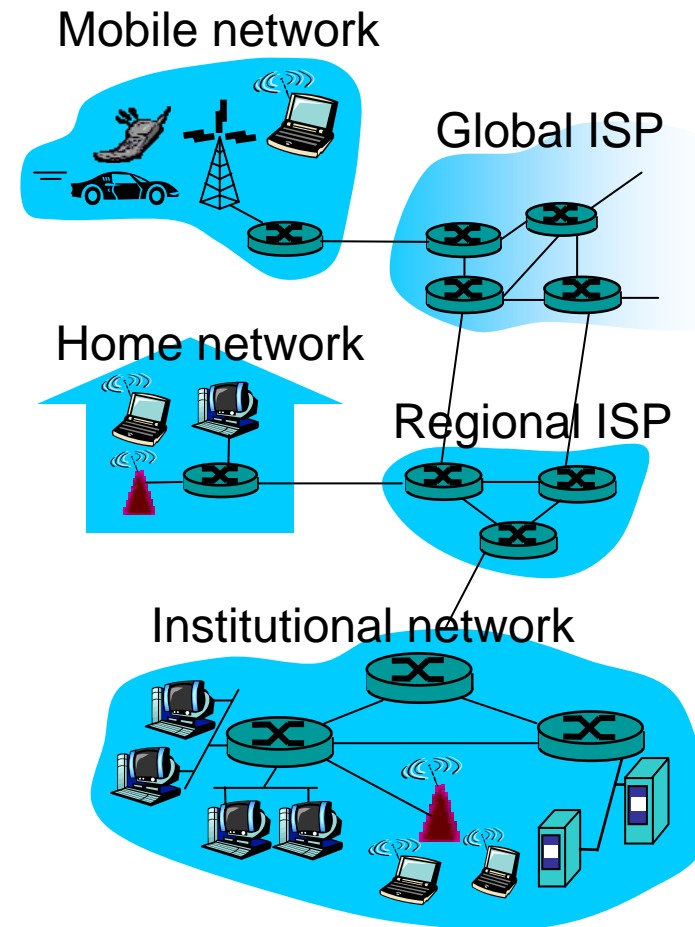
- millions of connected computing devices: *hosts = end systems*
 - running *network apps*

- *communication links*
 - fiber, copper, radio, satellite
 - transmission rate = *bandwidth*

-  access points
- wired links



- *routers*: forward packets (chunks of data)





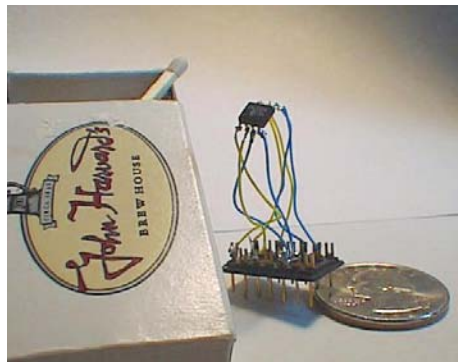
“Cool” internet appliances



IP picture frame
<http://www.ceiva.com/>
Free invitations for guests to send photos



Web-enabled toaster +
weather forecaster



World's smallest web server
in 1999



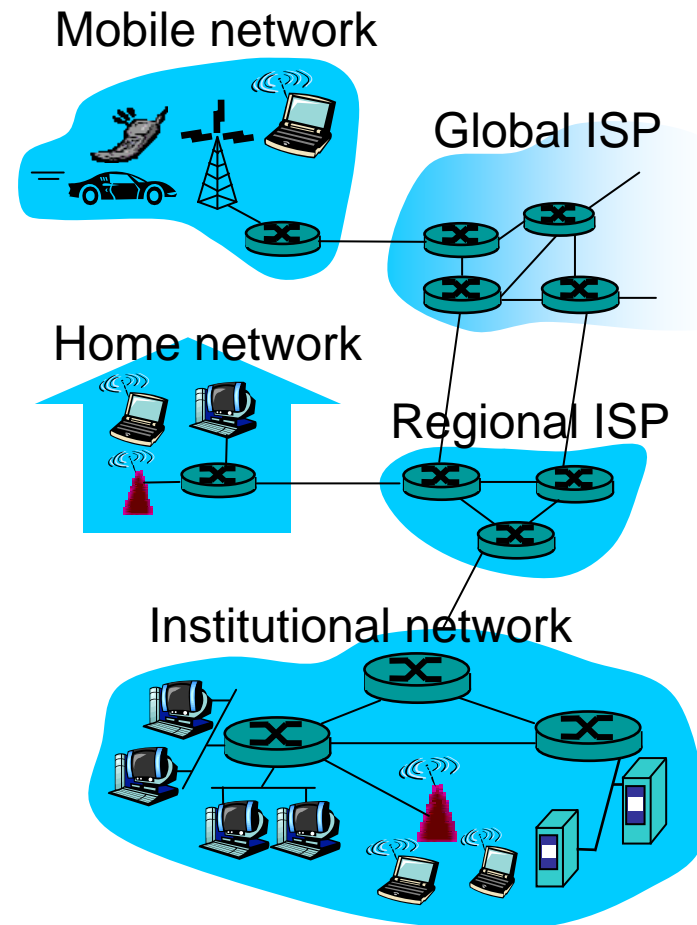
Internet phones

⇒ Who knows other cool internet appliances?



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

- ❑ **protocols** control sending, receiving of messages
 - 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
- ❑ **communication infrastructure** enables distributed applications:
 - Web, VoIP, email, games, e-commerce, file sharing
- ❑ **communication services provided to applications:**
 - 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 messages received, or other events

network protocols:

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

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



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

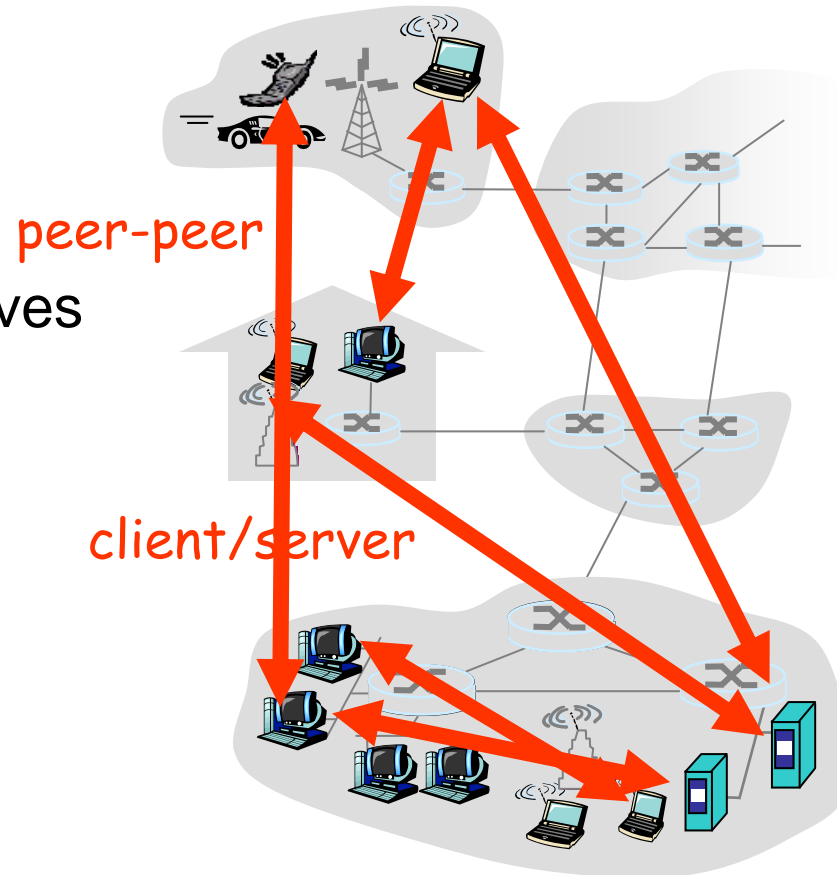
1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models



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





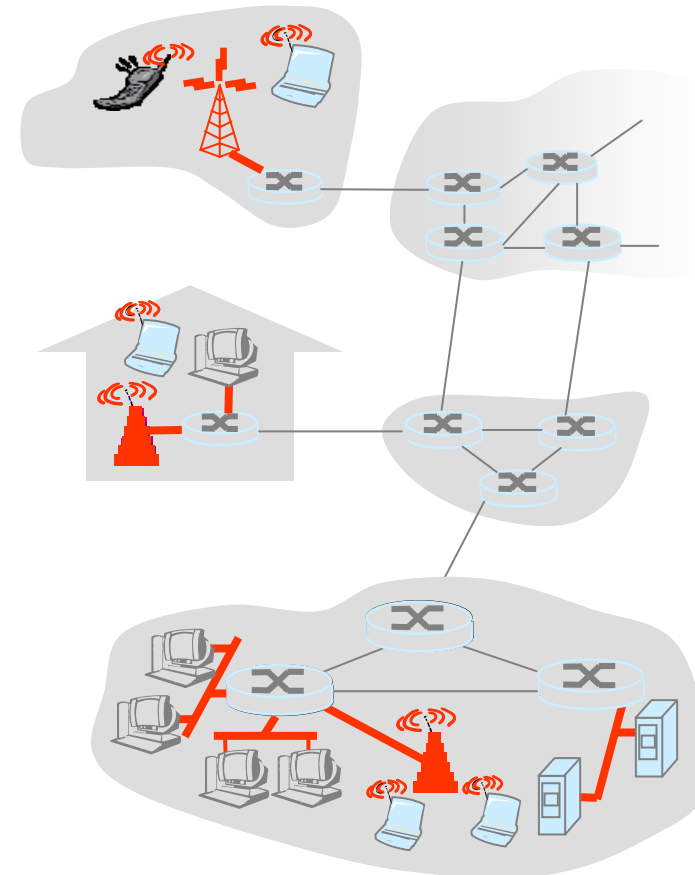
Access networks and physical media

Q: How to connect end systems to edge router?

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

Relevant:

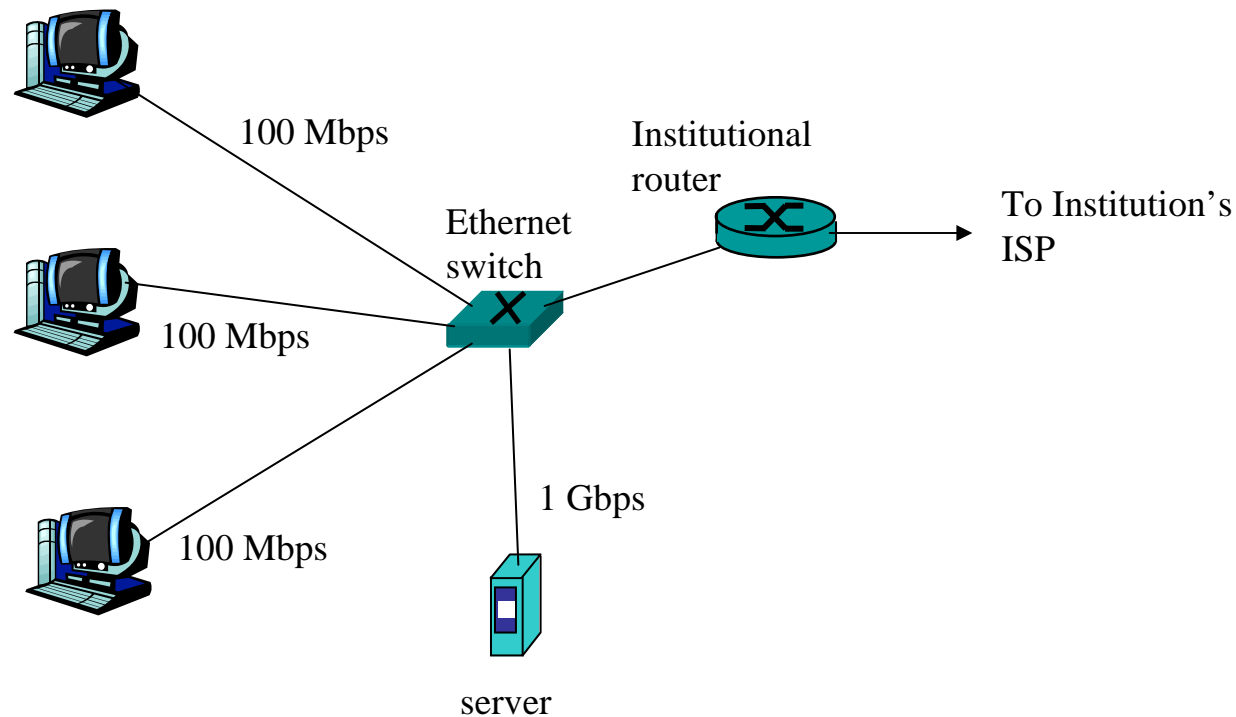
- ❑ bandwidth (bits per second) of access network?
- ❑ shared or dedicated?





Ethernet Internet access

- Typically used in companies, universities, etc
 - 10 Mbps, 100Mbps, 1Gbps, 10Gbps Ethernet
 - Today, end systems typically connect into Ethernet switch

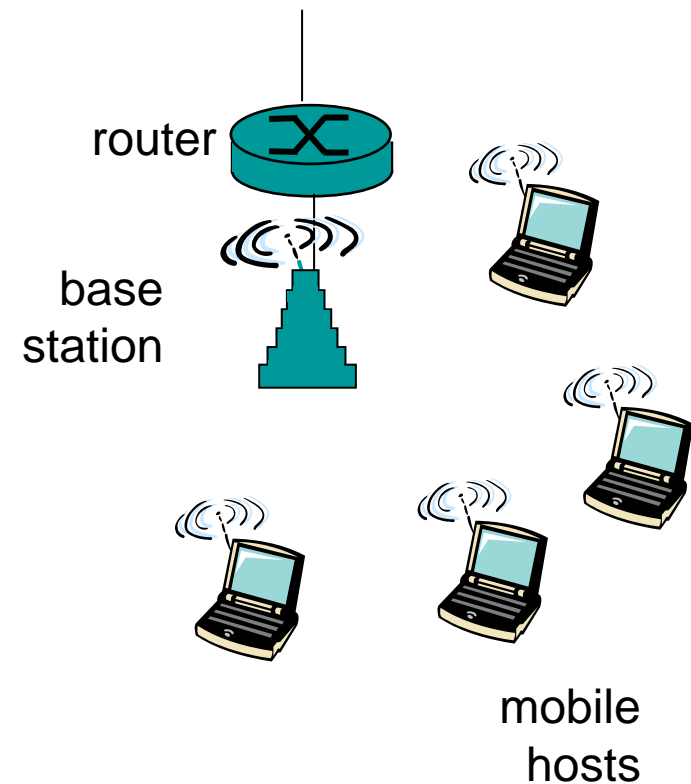


⇒ why?



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 (HSDPA)
 - next cellular network technology: LTE (10’s Mbps) over wide area

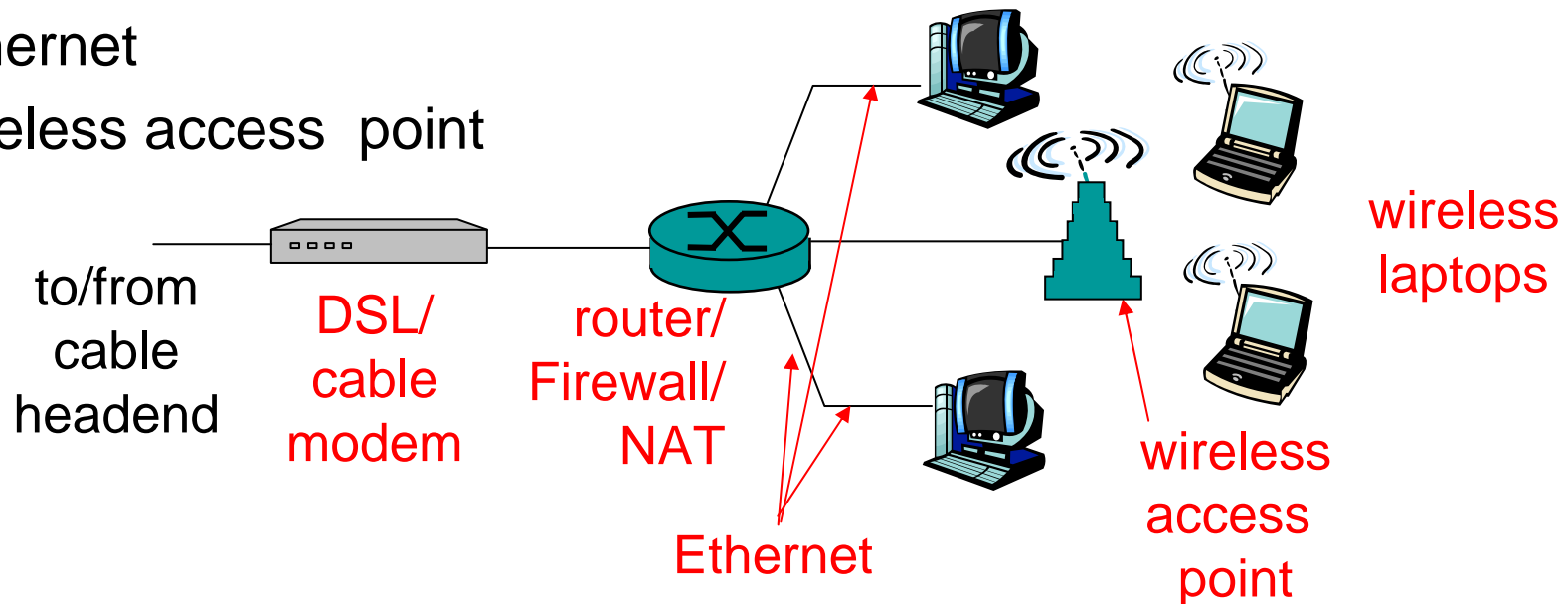




Home networks

Typical home network components:

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



⇒ Our research project AuthoNe: targetting many innovations



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

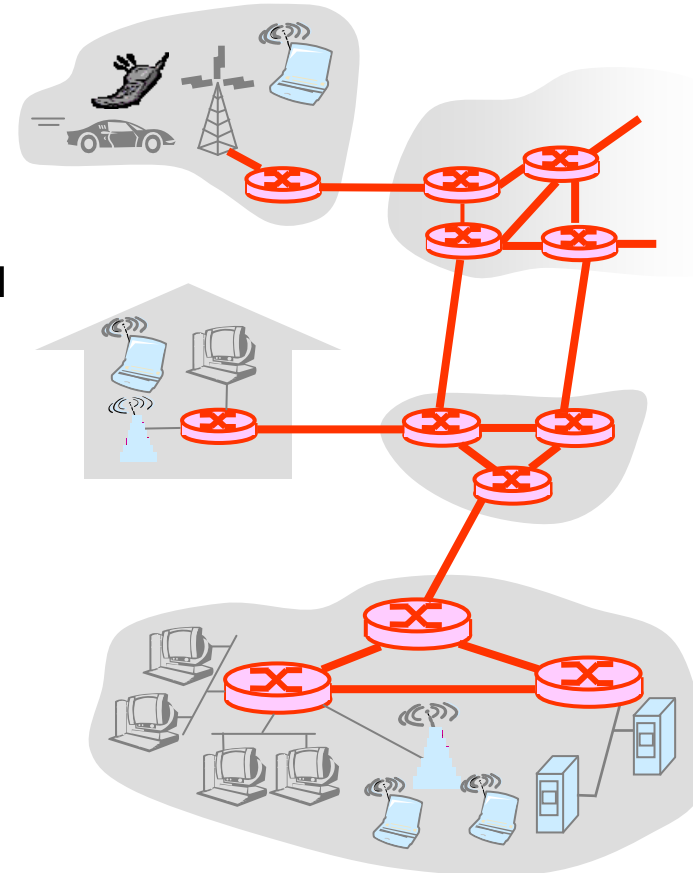
1.4 Delay, loss and throughput in packet-switched networks

1.5 Protocol layers, service models



The Network Core

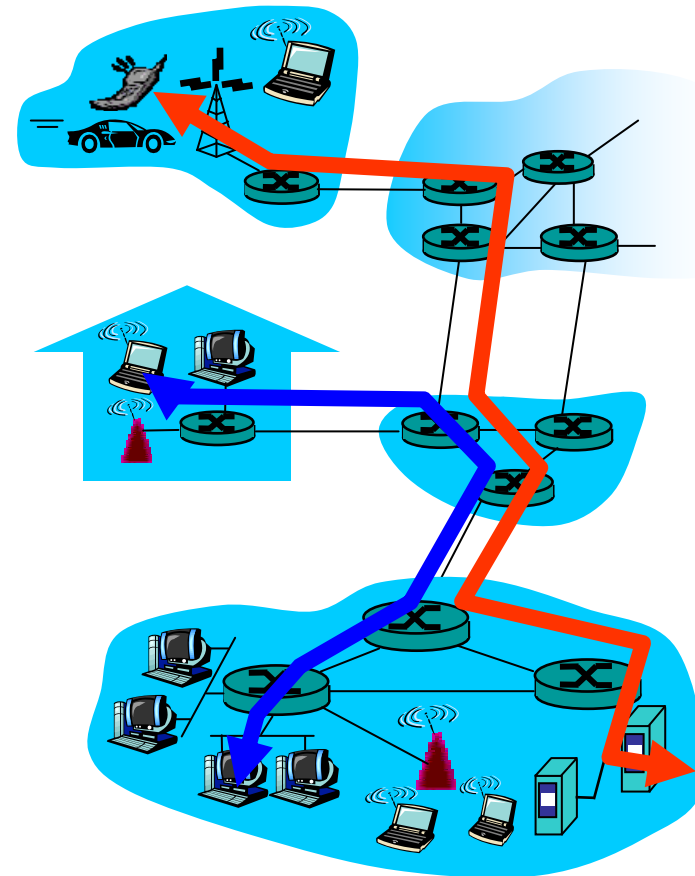
- mesh of interconnected routers
- **the fundamental question**: how is data transferred through net?
 - **circuit switching**: dedicated circuit per call: telephone net
 - **packet-switching**: data sent thru net in discrete “chunks”





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 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
- Inefficient for bursty sources (⇒why?)
- Quality guarantee, but call blocking





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:

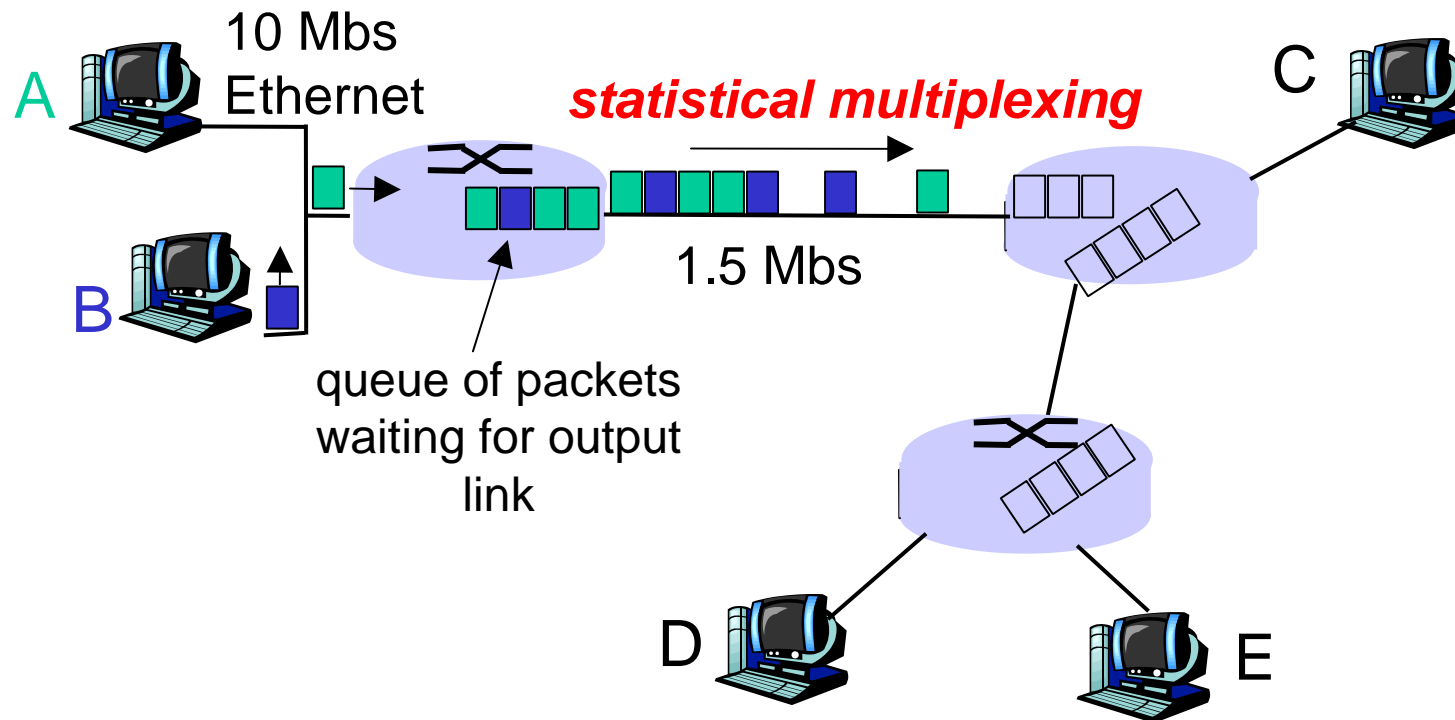
- ❑ aggregate resource demand can exceed amount available
- ❑ congestion: packets queue, wait for link use
- ❑ store and forward: packets move one hop at a time
 - Node receives complete packet before forwarding

Bandwidth division into “pieces”

Dedicated allocation
Resource reservation



Packet Switching: Statistical Multiplexing

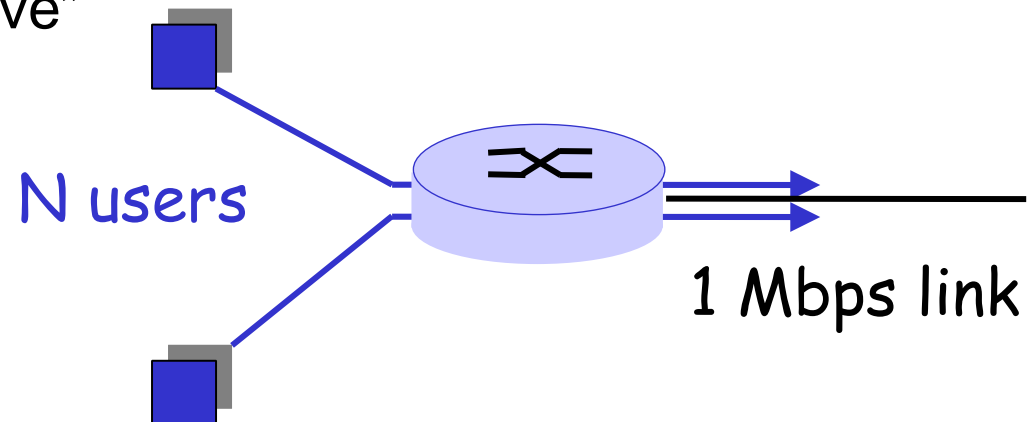


- ❑ Sequence of A & B packets does not have fixed pattern → **statistical multiplexing.**
- ❑ In TDM each host gets same slot in revolving TDM frame.



Packet switching versus circuit switching

- For bursty sources, Packet switching allows more users to use network! Example:
 - 1 Mbit link
 - each user:
 - 100 kbps when “active”
 - active 10% of time
 - circuit-switching:
 - 10 users
 - packet switching:
 - with 35 users, probability > 10 active less than .0004





Packet switching versus circuit switching

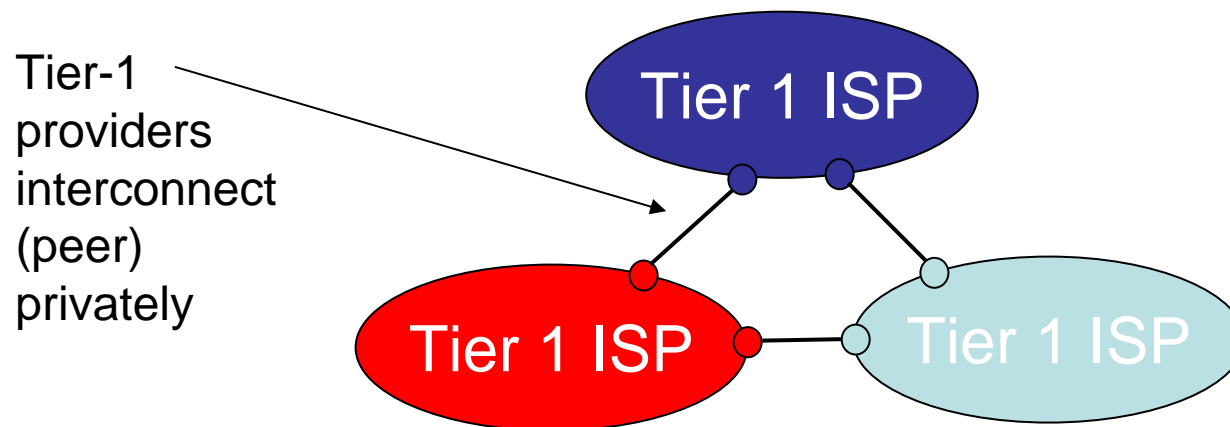
Is packet switching obviously better than circuit switching?

- packet switching is great for bursty data
 - resource sharing
 - simpler, no call setup
- possibility of **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
 - Internet-wide still an unsolved problem (⇒later)



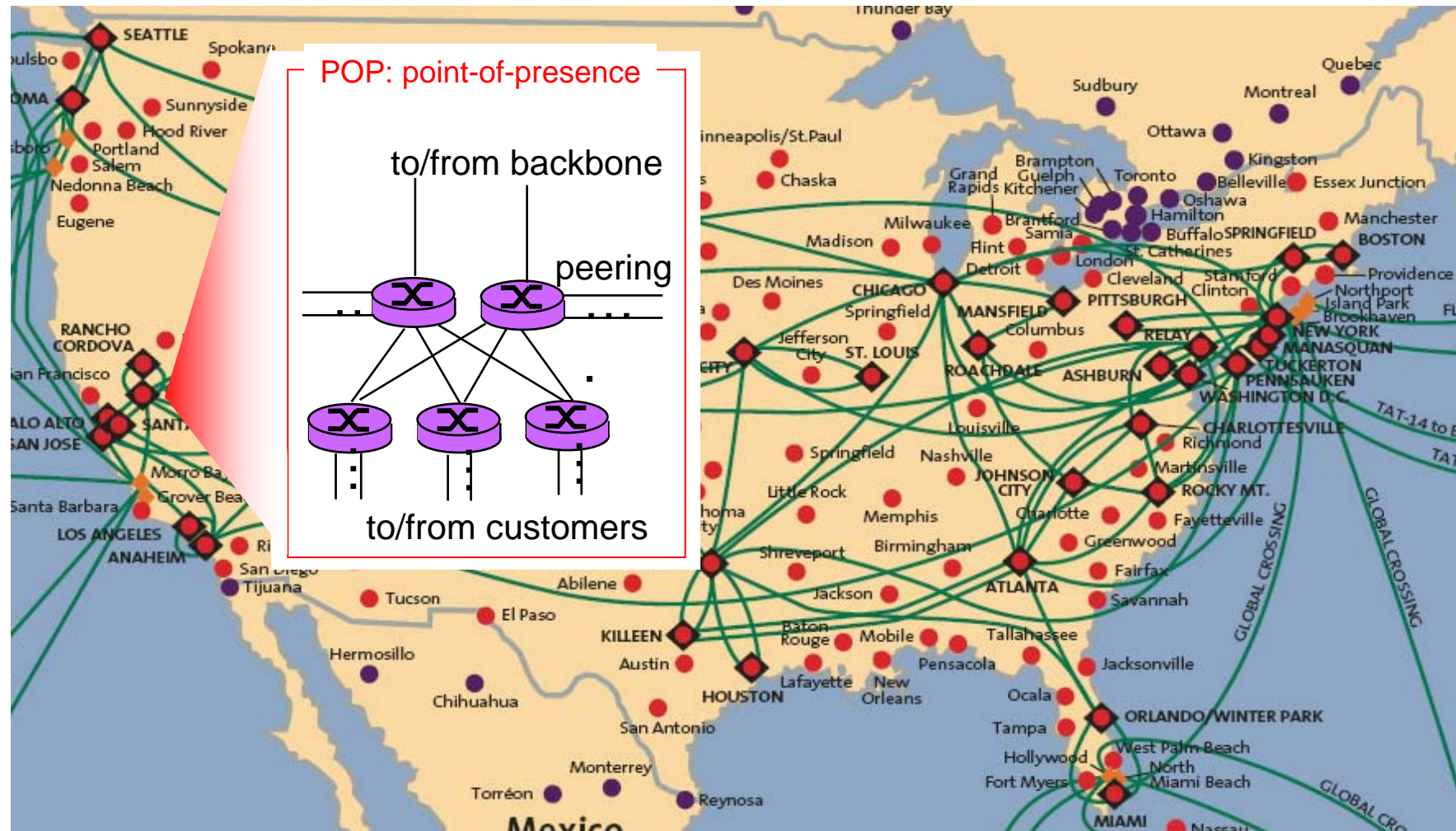
Internet structure: network of networks

- roughly hierarchical
- **at center: “tier-1” ISPs** (AT&T, Global Crossing, Level 3, NTT, Qwest, Sprint, Tata, Verizon (UUNET), Savvis, TeliaSonera), national/international coverage
 - treat each other as equals
 - can reach every other network on the Internet without purchasing IP transit or paying settlements.





Tier-1 ISP: e.g., Sprint

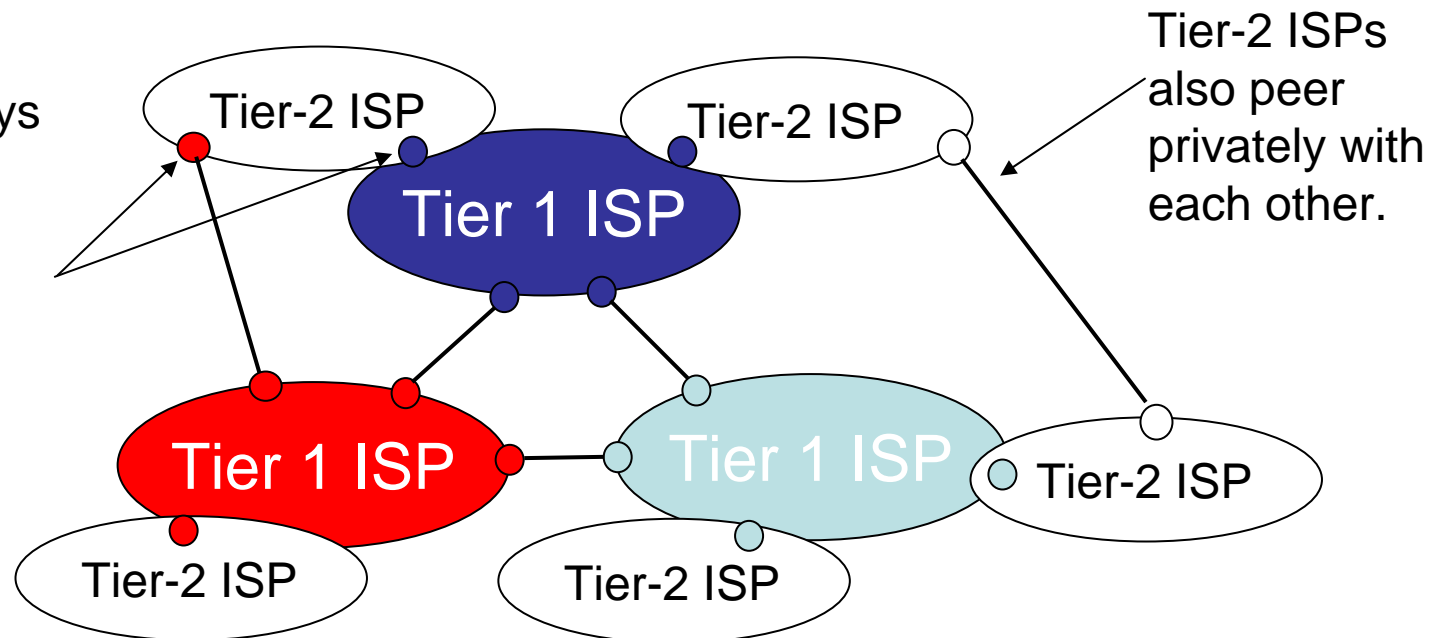




Internet structure: network of networks

- “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
- tier-2 ISP is *customer* of tier-1 provider

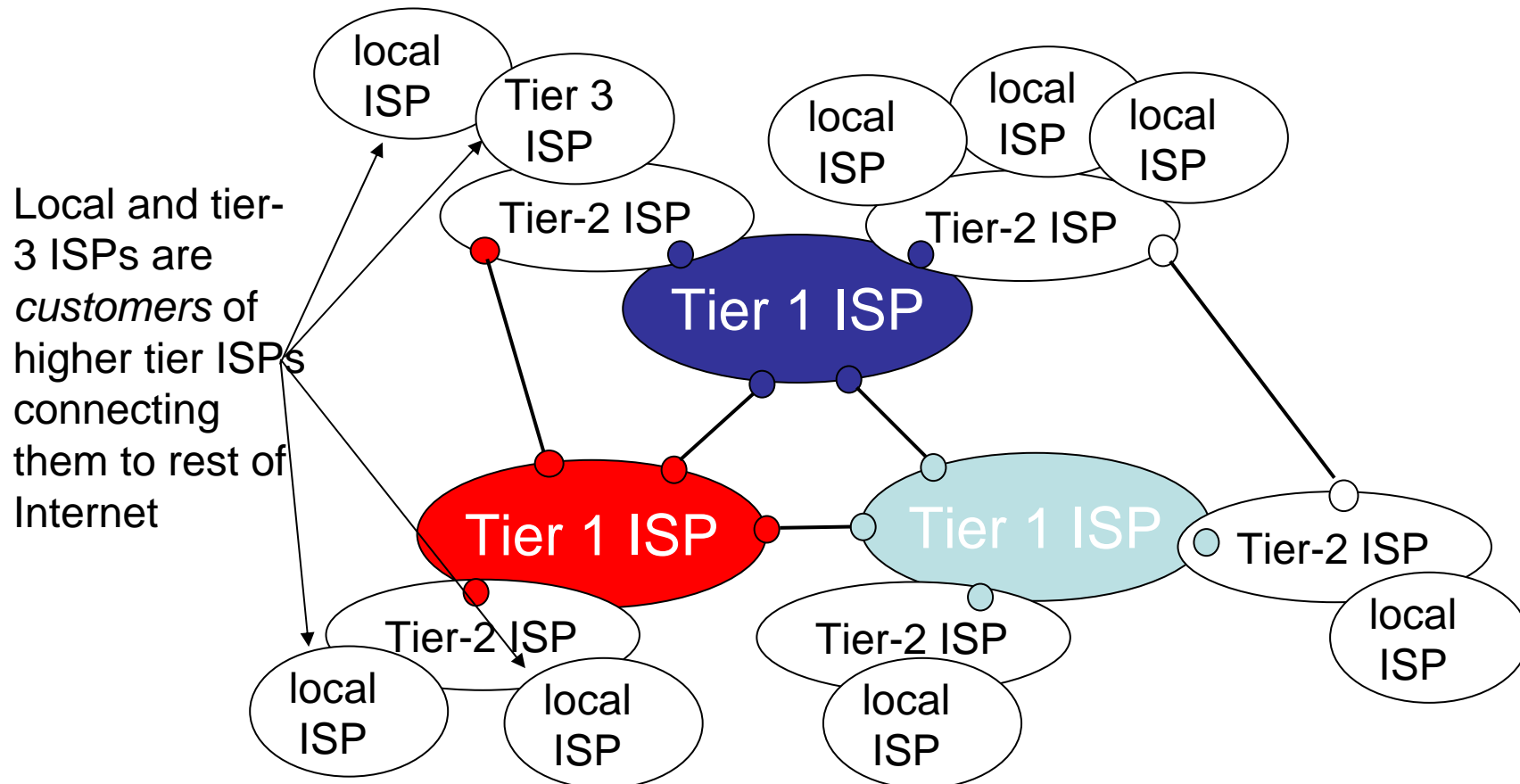




Internet structure: network of networks

□ “Tier-3” ISPs and local ISPs

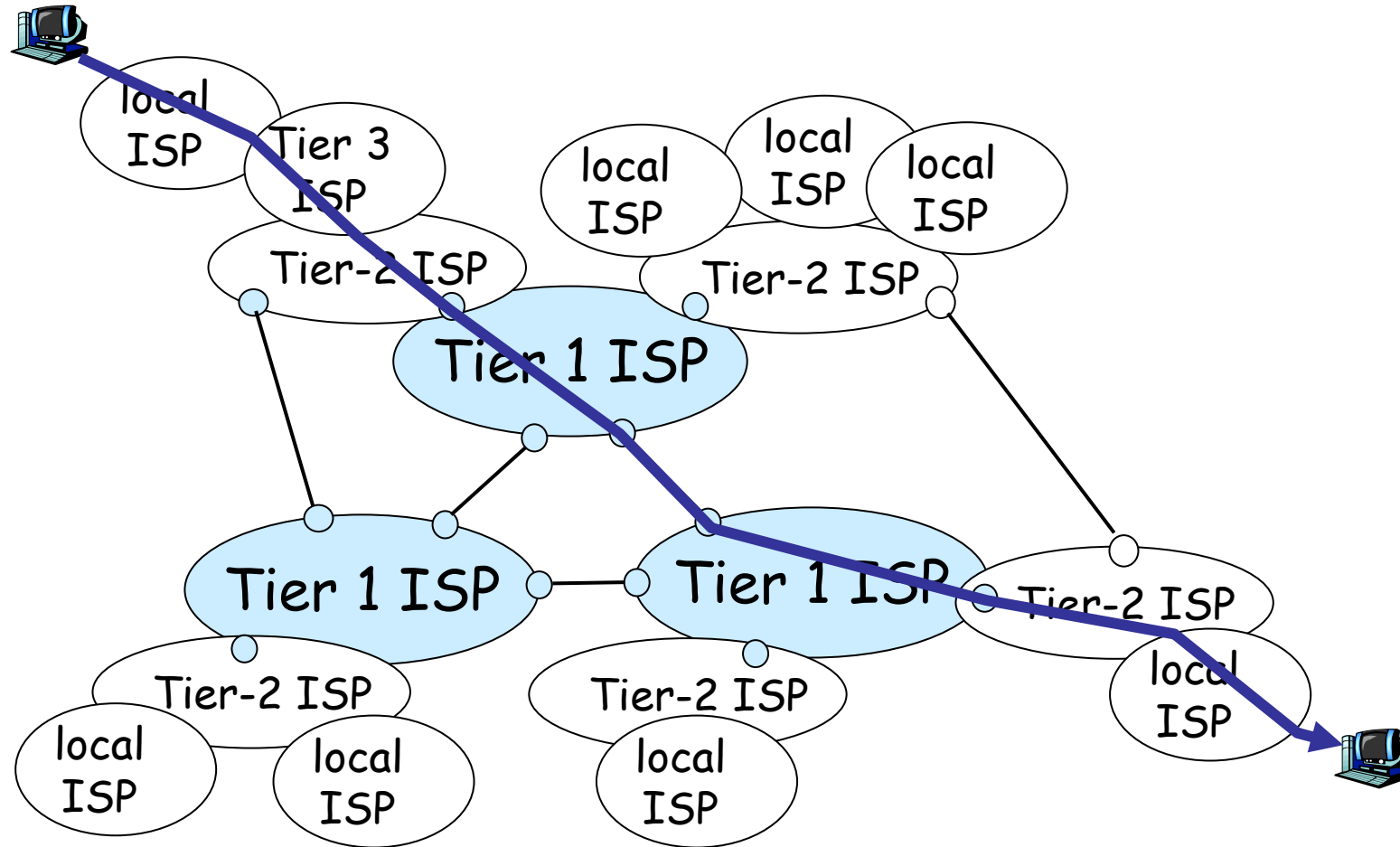
- last hop (“access”) network (closest to end systems)





Internet structure: network of networks

- a packet passes through many networks!





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

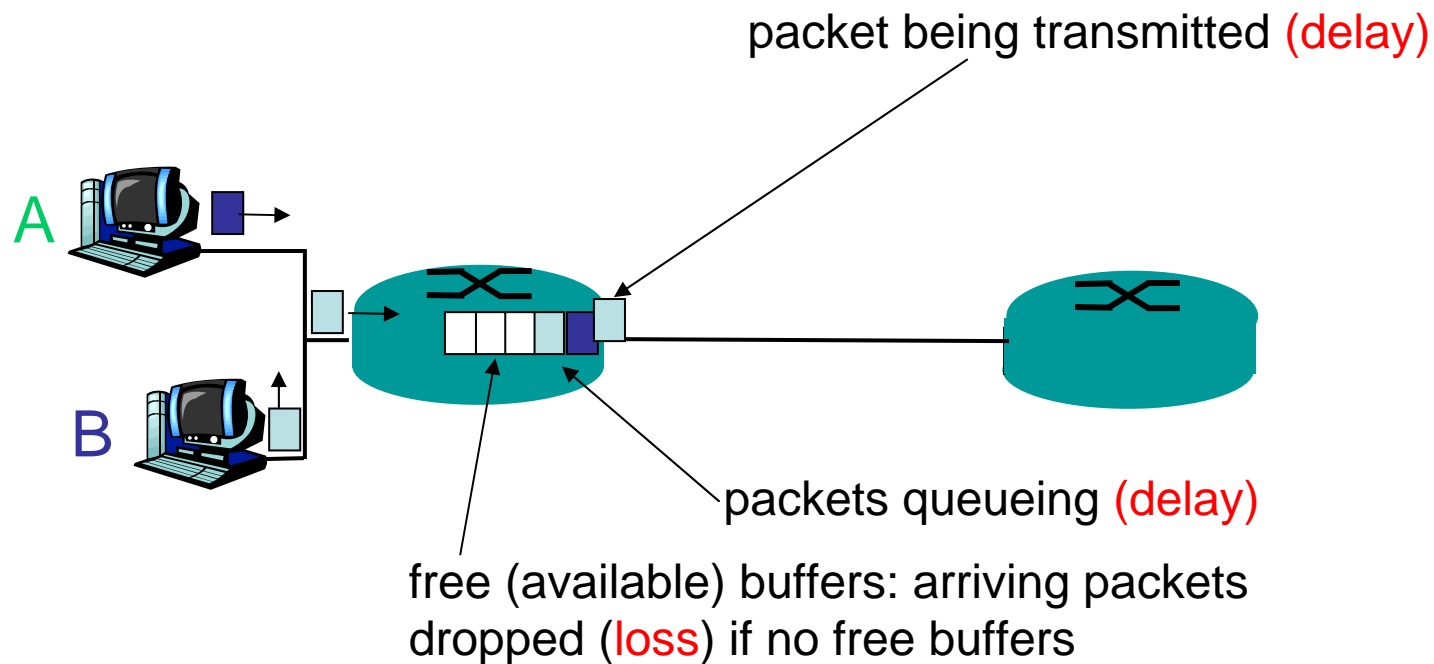
1.5 Protocol layers, service models



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





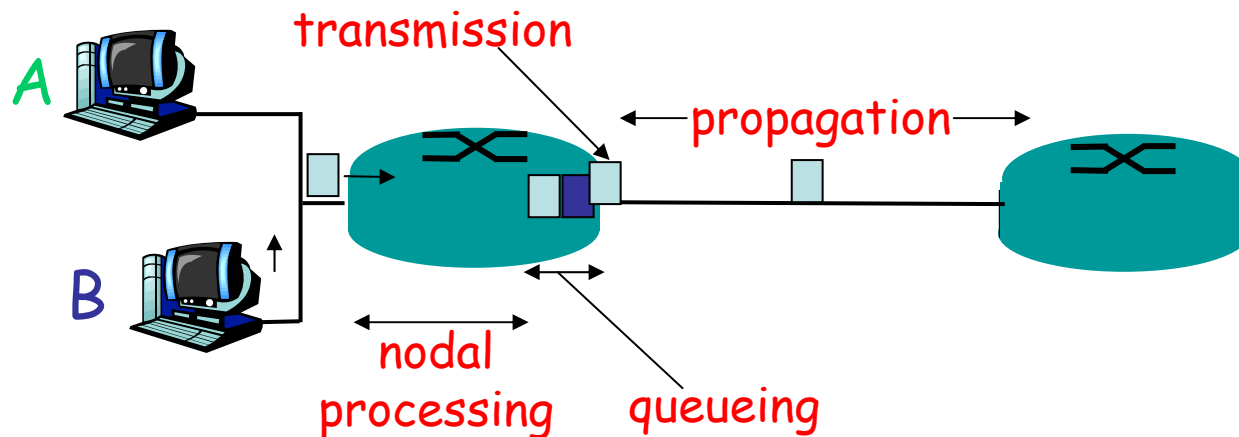
Four sources of packet delay

□ 1. nodal processing:

- check bit errors
- determine output link

□ 2. queueing

- time waiting at output link for transmission
- depends on congestion level of router





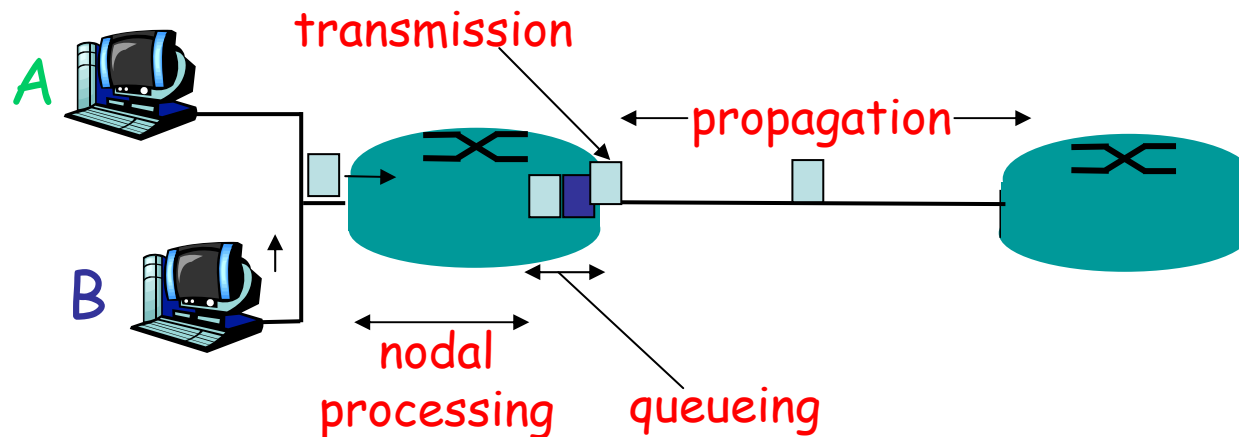
Delay in packet-switched networks

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





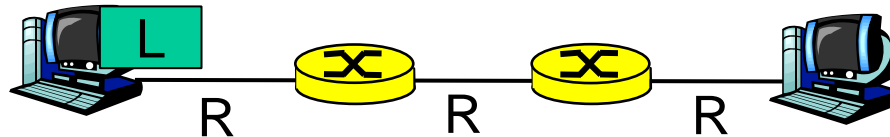
Nodal delay

- d_{proc} = processing delay
 - typically a few microseconds 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 microseconds to hundreds of msecs

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



Packet-switching: store-and-forward



- Takes L/R seconds to transmit (push out) packet of L bits on to link or R bps
- Entire packet must arrive at router before it can be transmitted on next link: store and forward
- delay = $3L/R$

Example:

Circuit Switching:

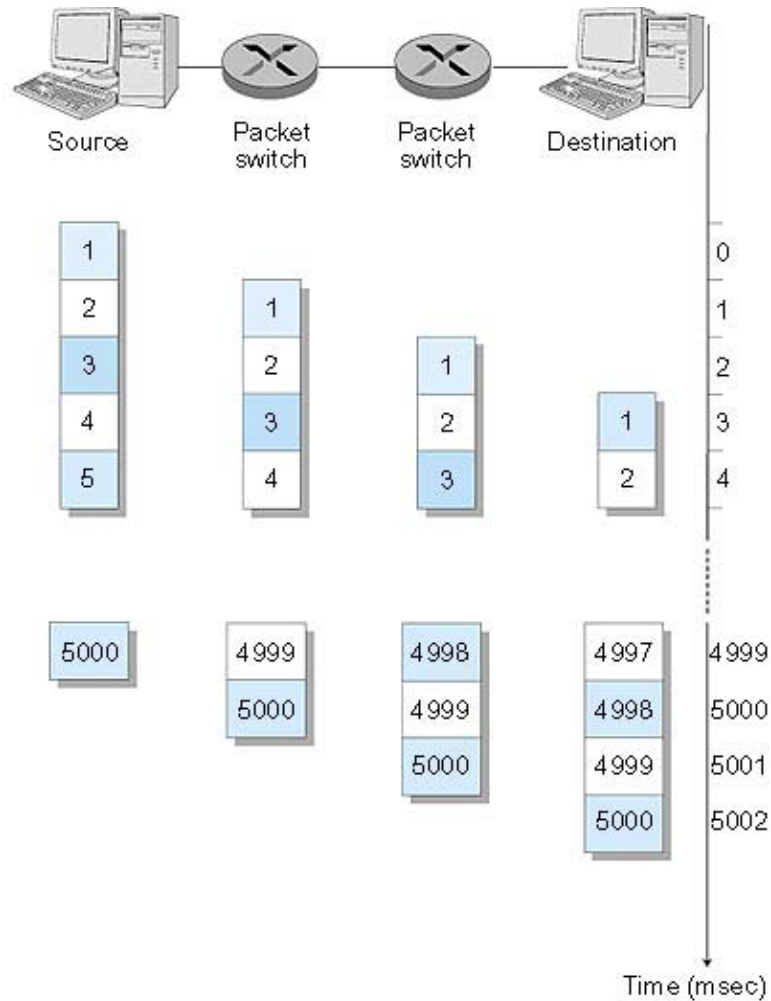
- $L = 7.5$ Mbits
- $R = 1.5$ Mbps
- Transmission delay = 5 sec

Packet Switching:

- $L = 7.5$ Mbits
- $R = 1.5$ Mbps
- Transmission delay = 15 sec



Packet Switching: Message Segmenting



Now break up the message into 5000 packets

- ❑ Each packet 1,500 bits
- ❑ 1 msec to transmit packet on one link
- ❑ *pipelining*: each link works in parallel
- ❑ Delay reduced from 15 sec to 5.002 sec (as good as circuit switched)
- ❑ What did we achieve over circuit switching?
- ❑ Drawbacks (of packet vs. Message)

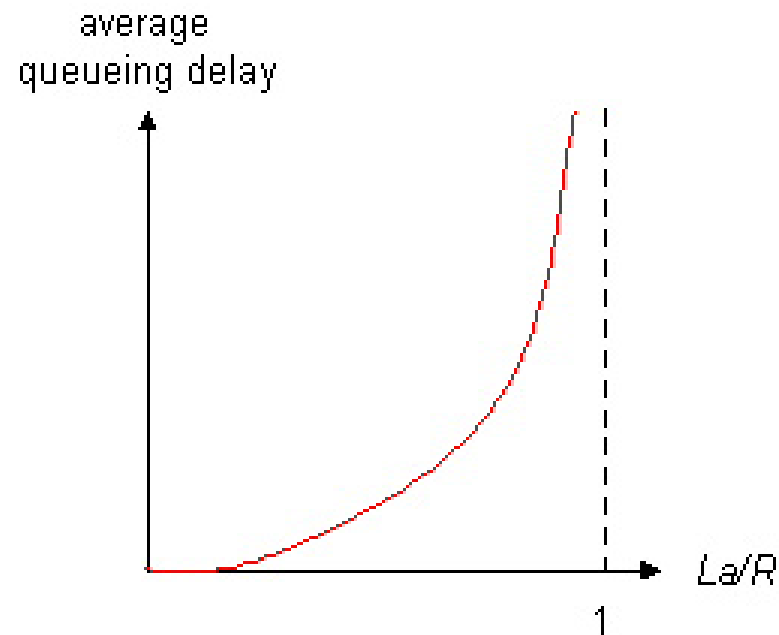


Queueing delay (revisited)

- R =link bandwidth (bit/s)
- L =packet length (bit)
- a =average packet arrival rate

traffic intensity = $L \cdot a/R$

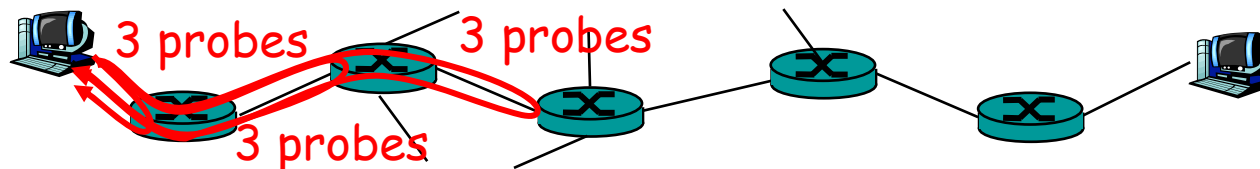
- $L \cdot a/R \sim 0$: average queueing delay small
- $L \cdot a/R \rightarrow 1$: delays become large
- $L \cdot a/R > 1$: more “work” arriving than can be serviced, average delay infinite!





“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
```

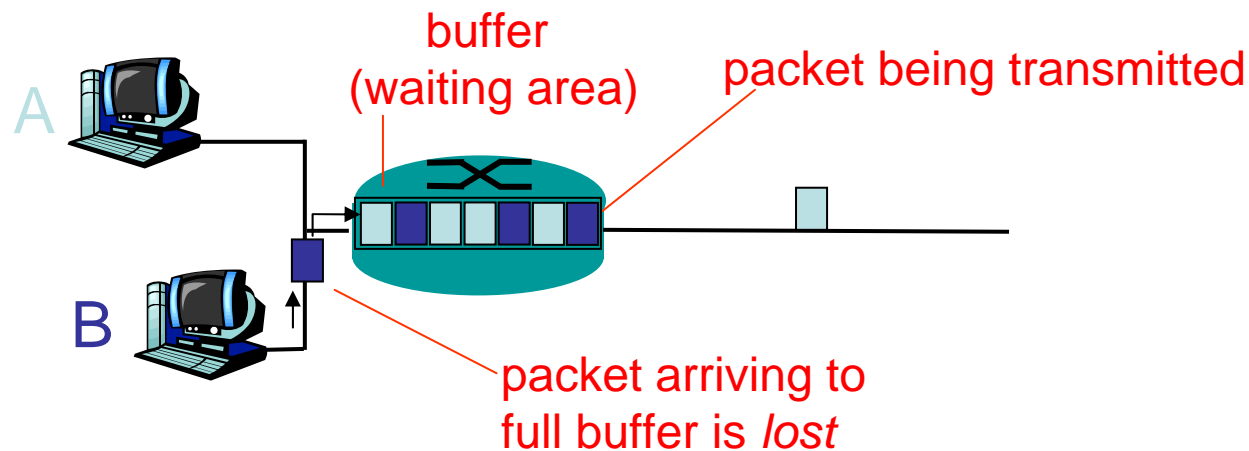
trans-oceanic link

* means no response (probe lost, router not replying)



Packet loss

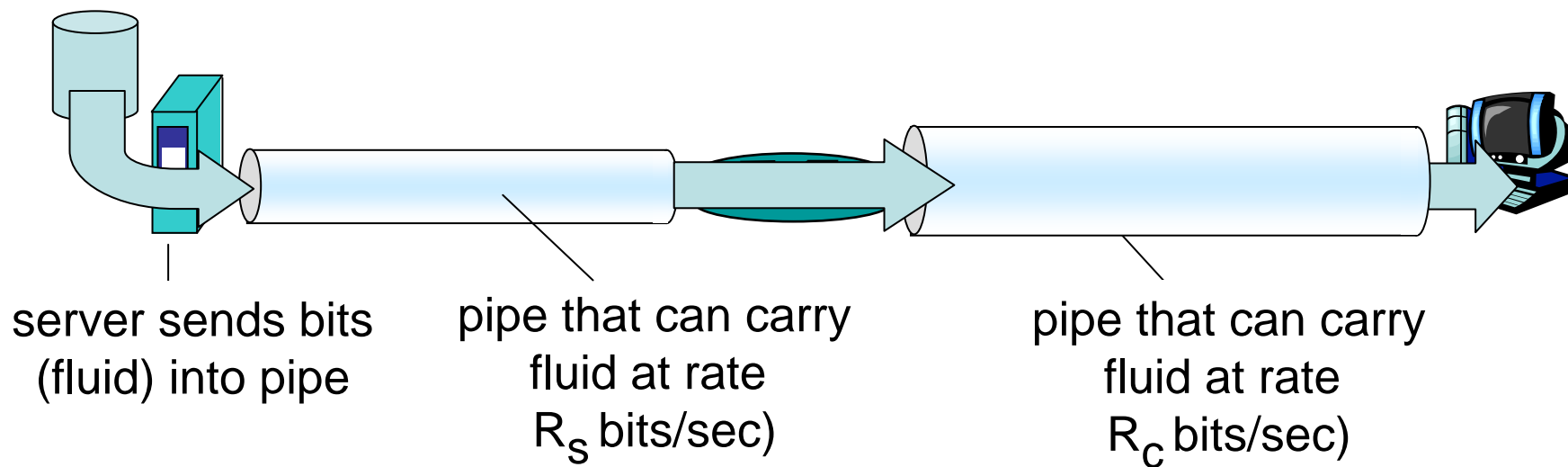
- ❑ queue (aka buffer) preceding link in buffer has finite capacity
- ❑ packet arriving to full queue dropped (aka lost)
- ❑ lost packet may be retransmitted by previous node, by source end system, or not at all





Throughput

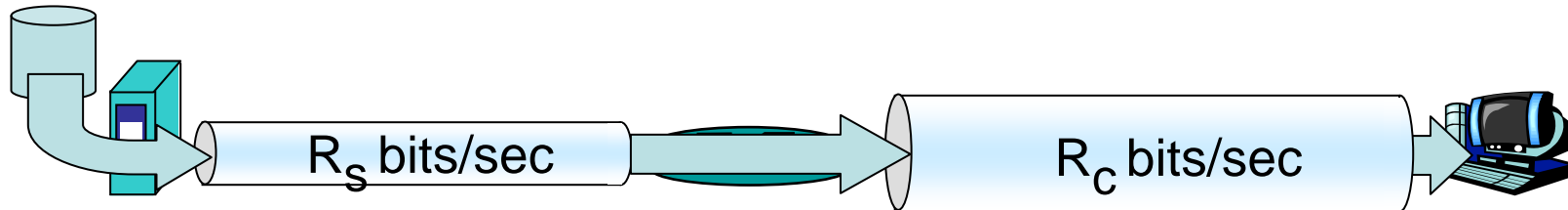
- *throughput*: rate (bits/time unit) at which bits transferred between sender/receiver
 - *instantaneous*: rate at given point in time
 - *average*: rate over longer period of time



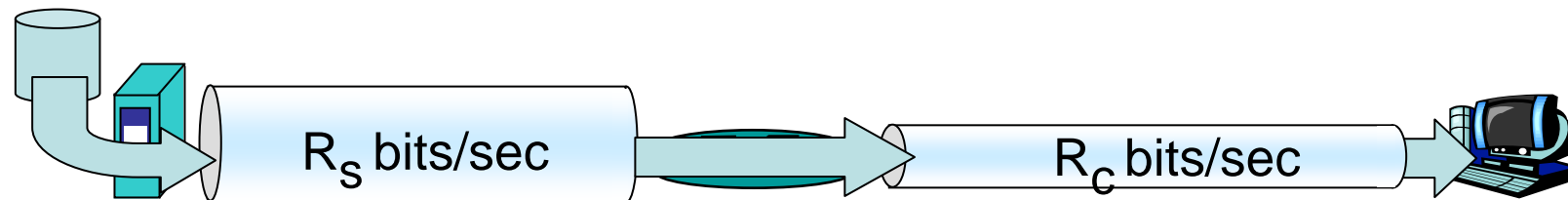


Throughput (more)

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



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



bottleneck link

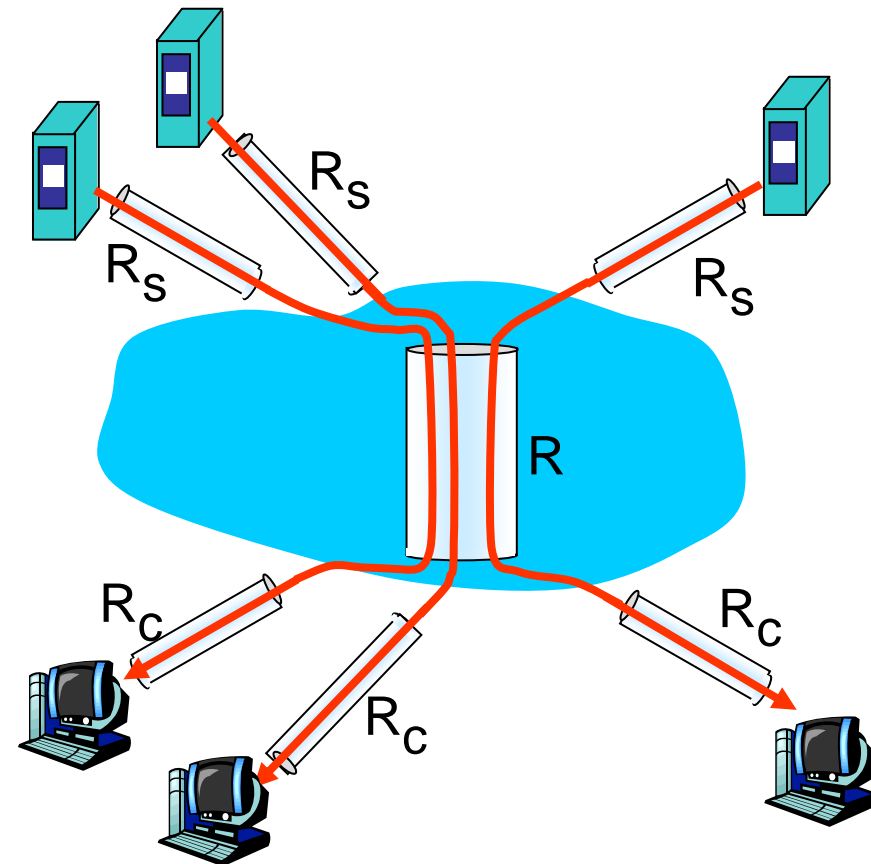
link on end-end path that constrains end-end throughput

- ⇒ measurement challenge for networks with many nodes:
identify bottleneck interfaces, e.g. with packet-pair measurements



Throughput: Internet scenario

- Example: 10 clients / servers share a bottleneck link
 - 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?



Why layering?

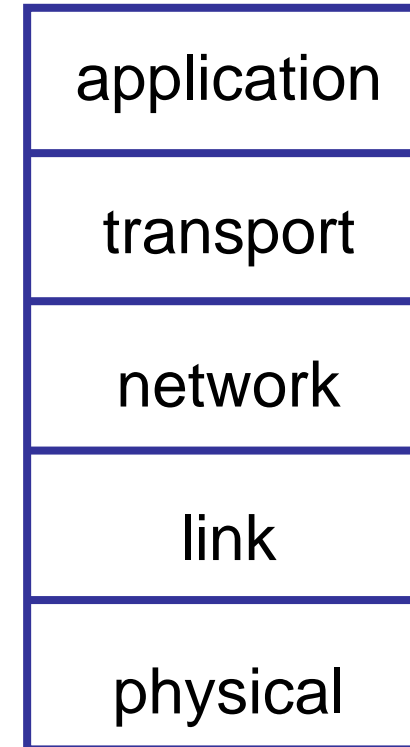
Dealing with complex systems:

- ❑ 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”





Introduction: Summary

Covered a lot 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

You now have:

- ❑ context, overview, “feel” of networking
- ❑ more depth, detail *to follow!*