Chapter 1
Introduction

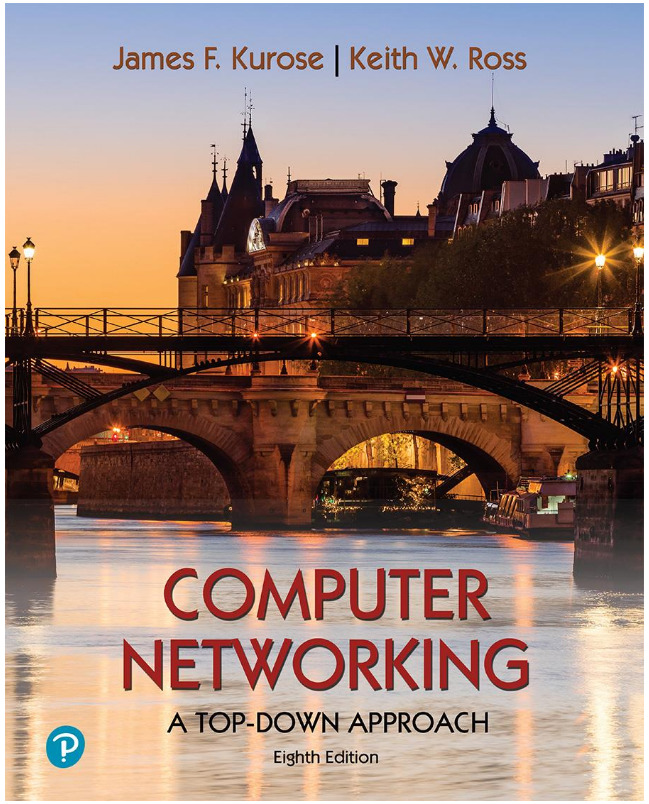
A note on the use of these PowerPoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and 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) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides 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.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020
http://gaia.cs.umass.edu/kurose_ross

Chapter 1: introduction

Chapter goal:

- Get "feel," "big picture," introduction to terminology
  - more depth, detail *later* in course
- Approach:
  - use Internet as example

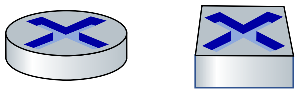
Overview/roadmap:

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

The Internet: a “nuts and bolts” view

Billions of connected computing *devices*:

- *hosts = end systems*
- running *network apps* at Internet’s “edge”



Packet switches: forward packets (chunks of data)

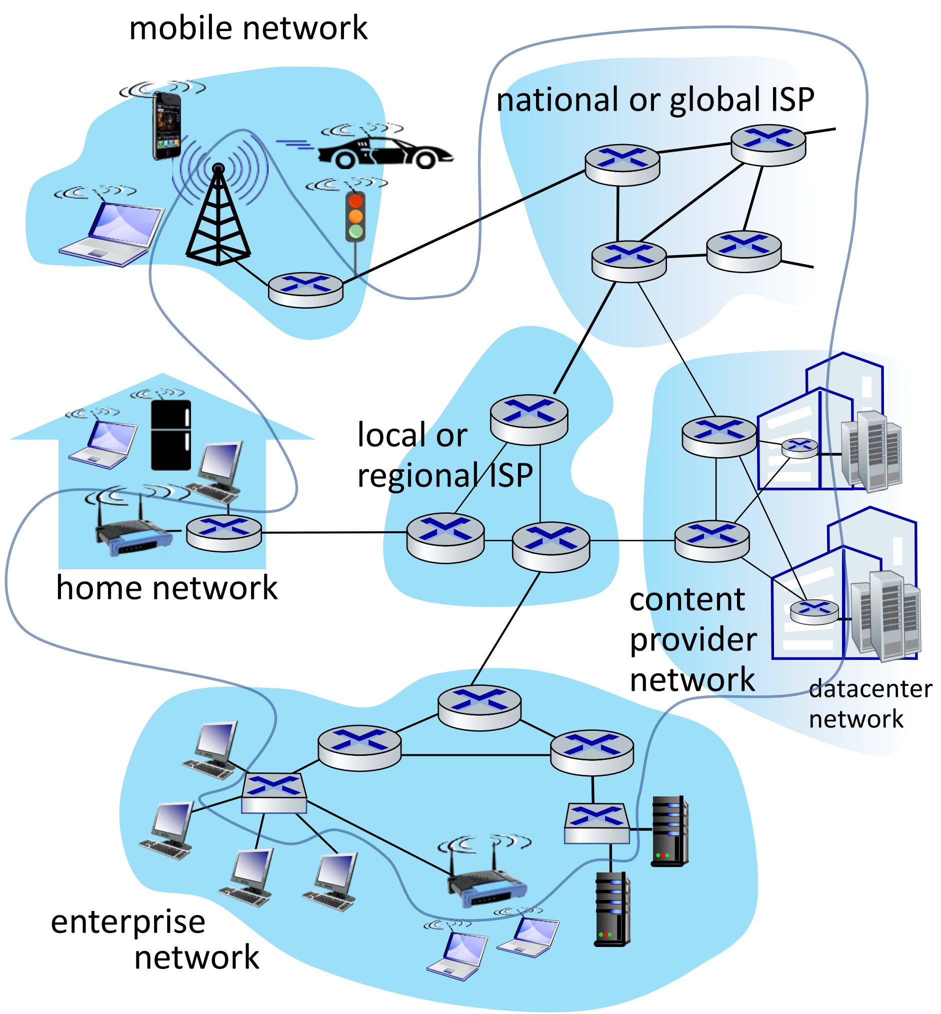
- *routers, switches*

Communication links

- fiber, copper, radio, satellite
- transmission rate: *bandwidth*

Networks

- collection of devices, routers, links: managed by an organization



“Fun” Internet-connected devices


Amazon Echo


Internet refrigerator


IP picture frame


Pacemaker & Monitor


Tweet-a-watt: monitor energy use


Web-enabled toaster + weather forecaster

Security Camera


Slingbox: remote control cable TV


AR devices

Internet phones



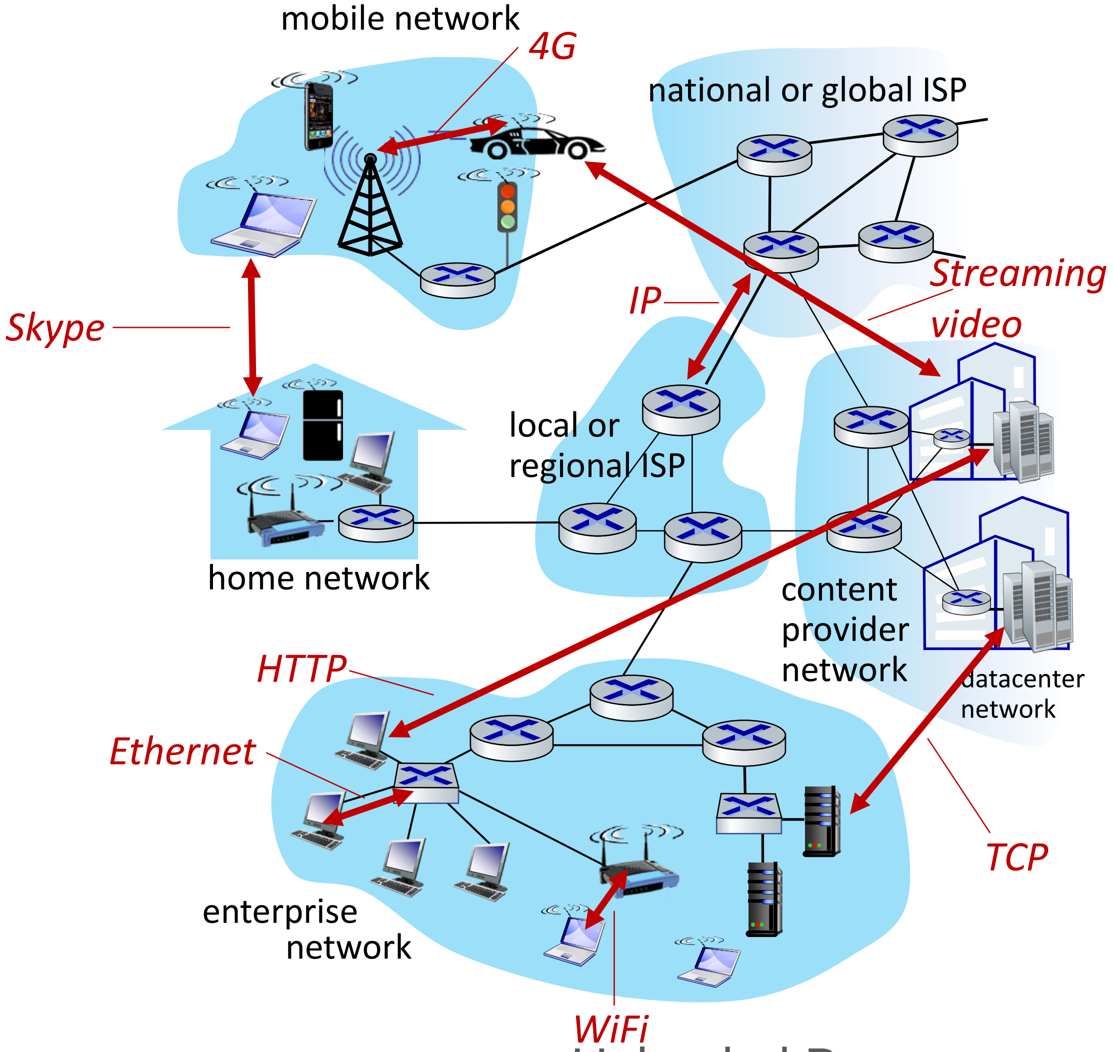
sensorized, bed mattress


Fitbit

Others?

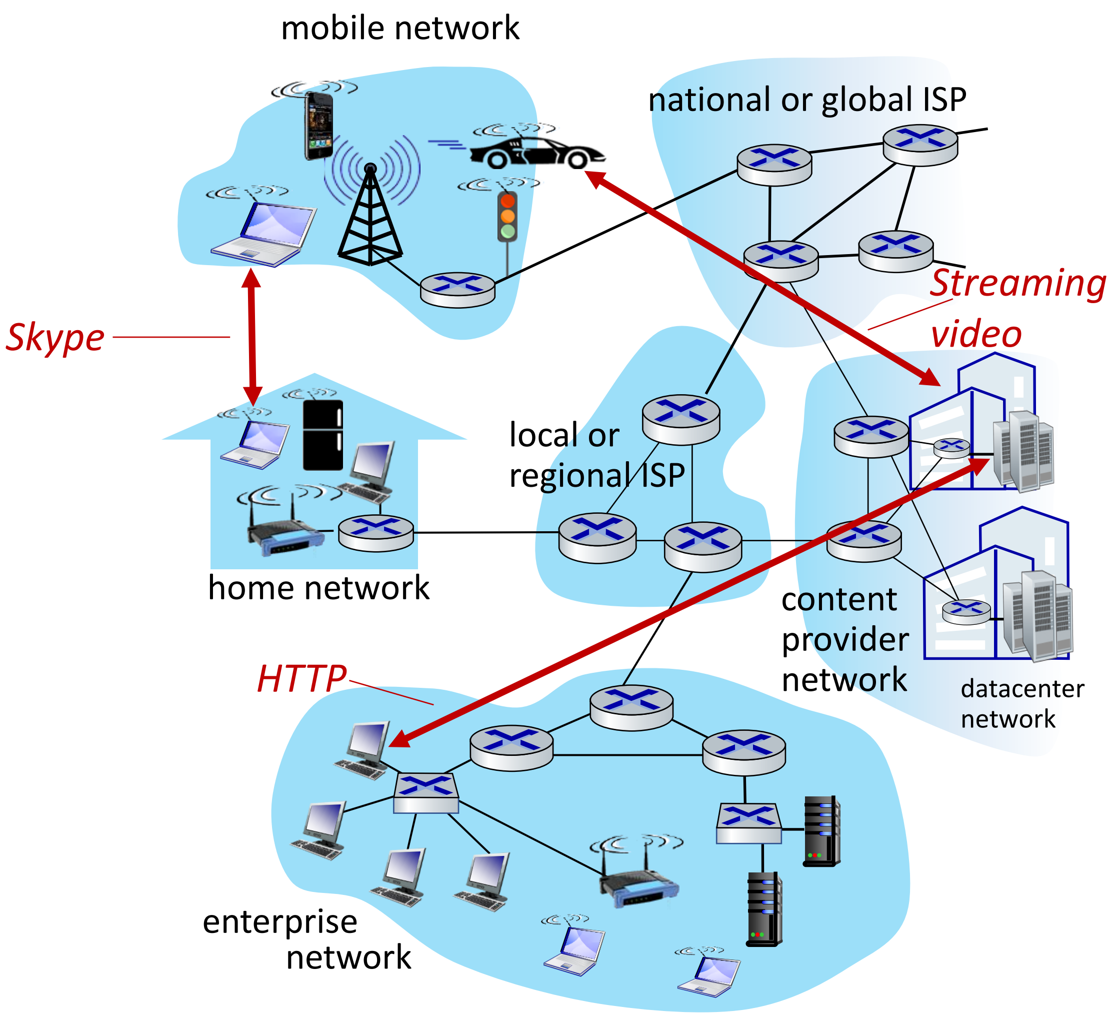
The Internet: a "nuts and bolts" view

- *Internet: "network of networks"*
  - Interconnected ISPs
- *protocols* are *everywhere*
  - control sending, receiving of messages
  - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4G, Ethernet
- *Internet standards*
  - RFC: Request for Comments
  - IETF: Internet Engineering Task Force



The Internet: a "service" view

- *Infrastructure* that provides services to applications:
  - Web, streaming video, multimedia teleconferencing, email, games, e-commerce, social media, inter-connected appliances, …
- provides *programming interface* to distributed applications:
  - "hooks" allowing sending/receiving apps to "connect" to, use Internet transport service
  - provides service options, analogous to postal service



What’s a protocol?

Human protocols:

- “what’s the time?”
- “I have a question”
- introductions

… specific messages sent

… specific actions taken when message received, or other events

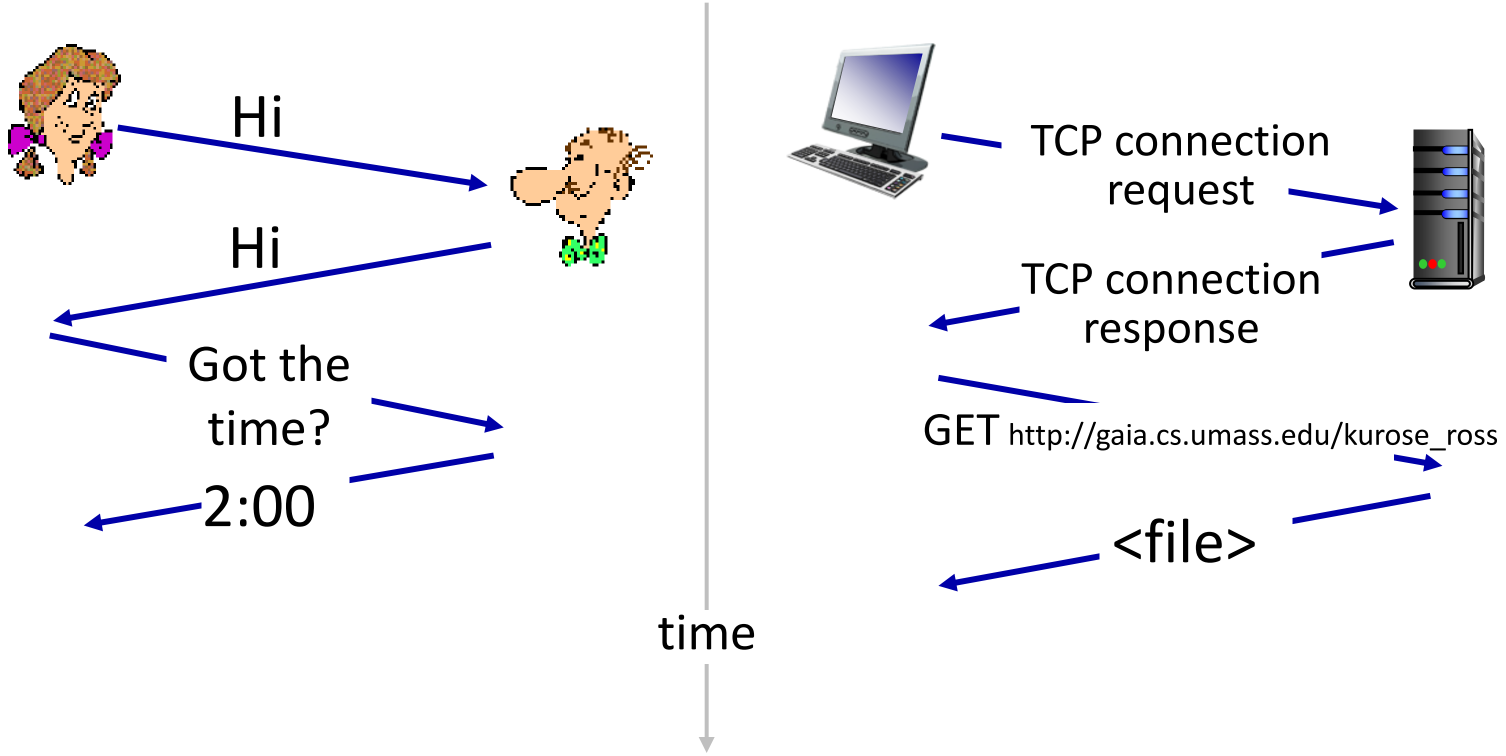
Network protocols:

- computers (devices) rather than humans
- all communication activity in Internet governed by protocols

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

What’s a protocol?

A human protocol and a computer network protocol:



Q: other human protocols?

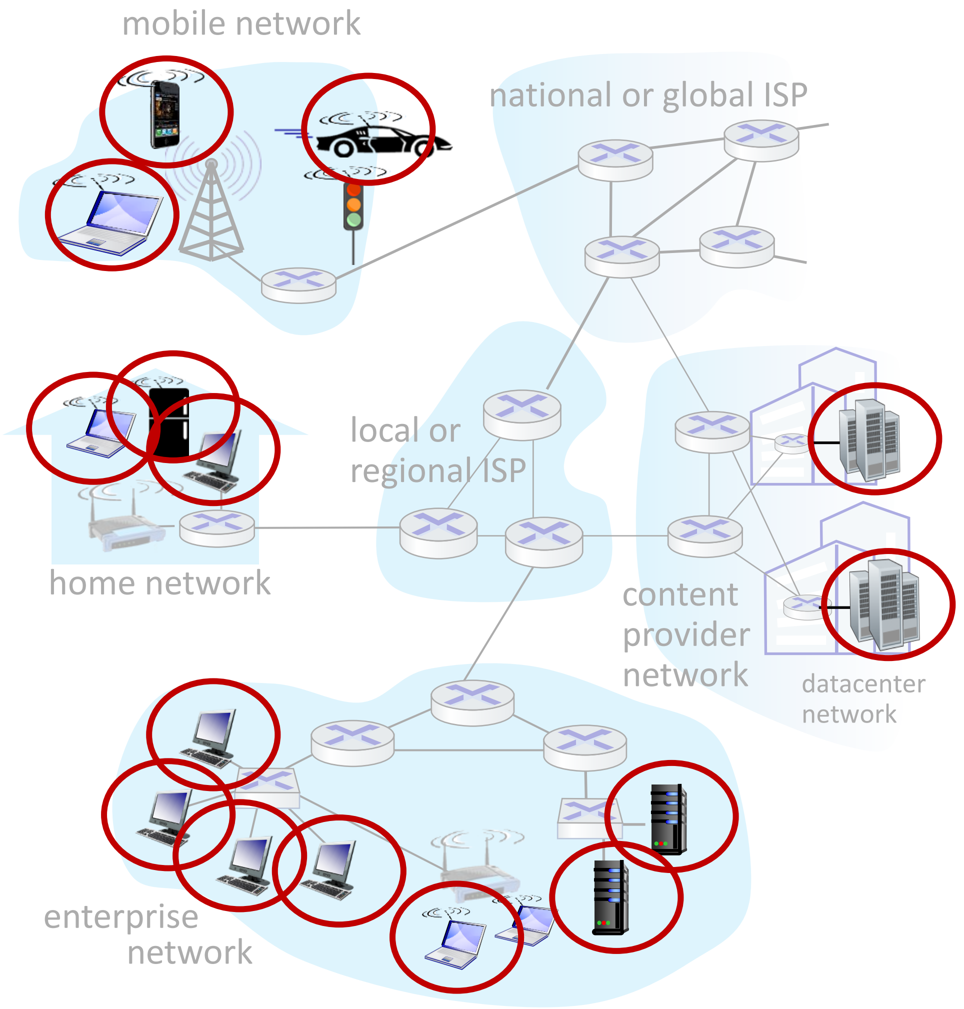
Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

A closer look at Internet structure

Network edge:

- hosts: clients and servers
- servers often in data centers



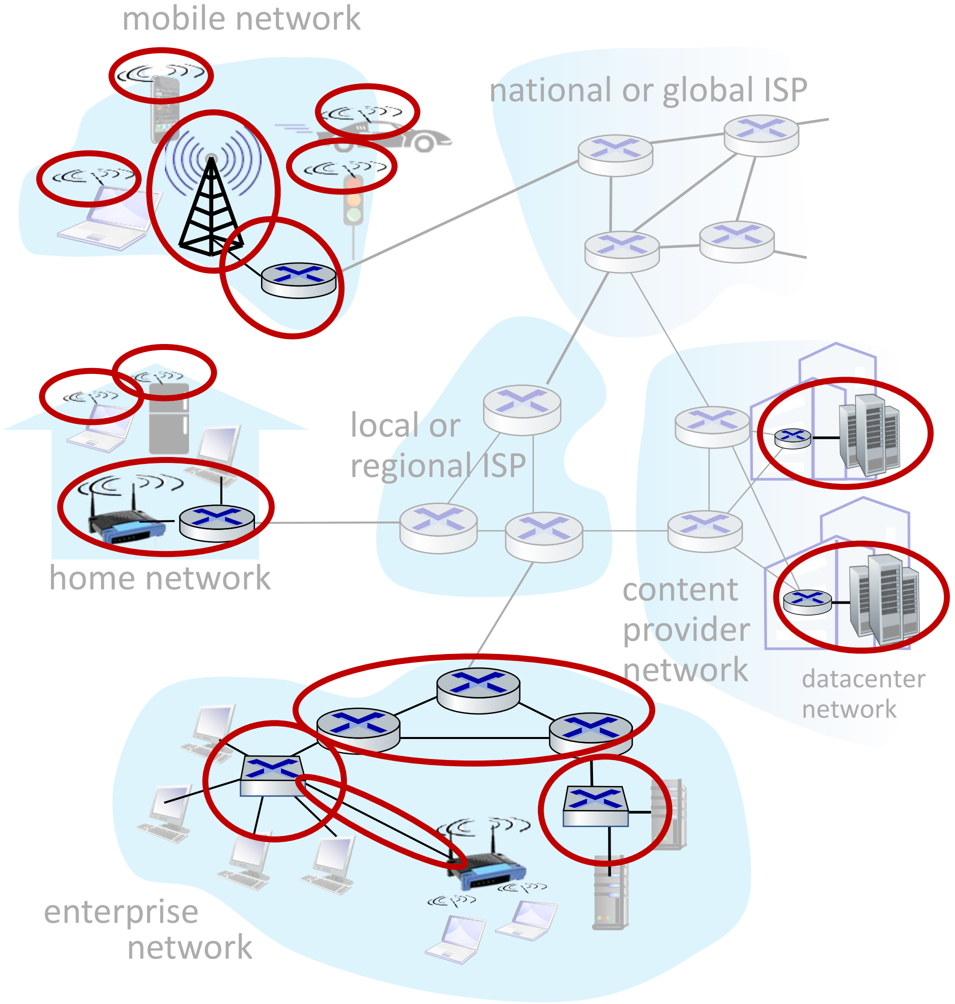
A closer look at Internet structure

Network edge:

- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links

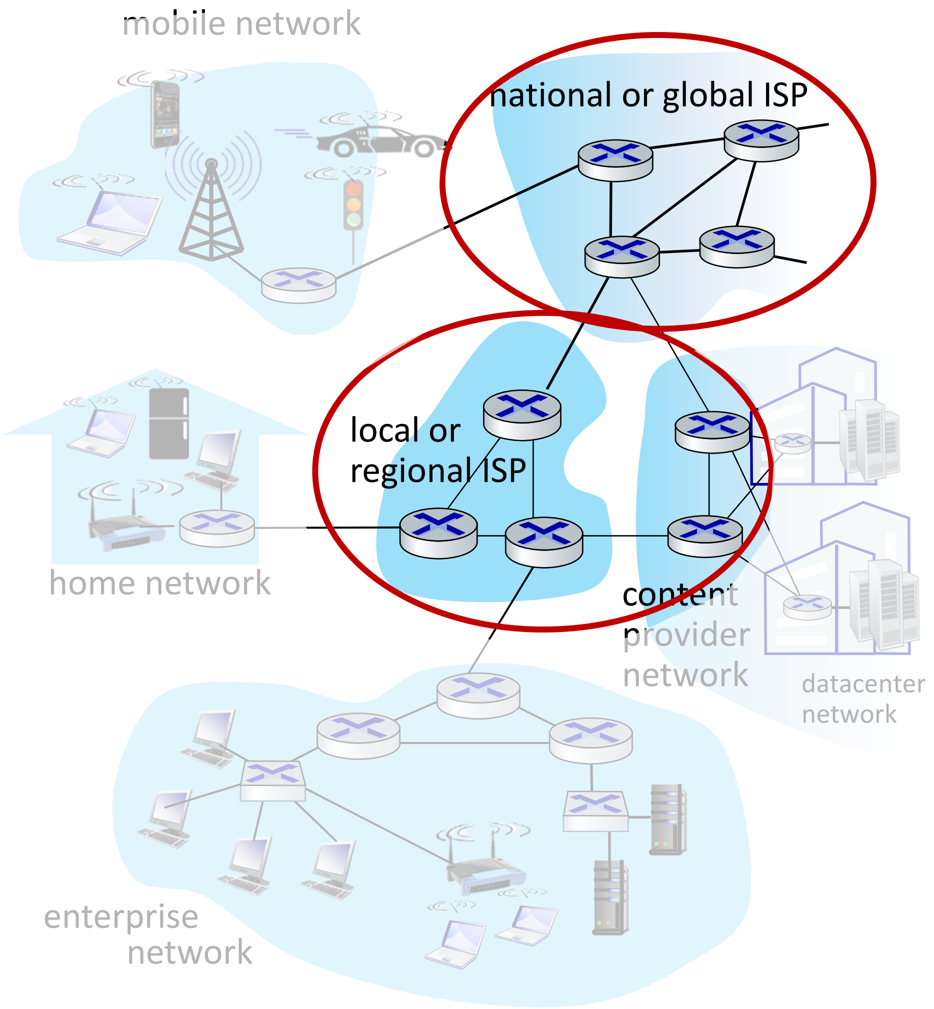


A closer look at Internet structure

Network edge:
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:
- wired, wireless communication links

Network core:
- interconnected routers
- network of networks



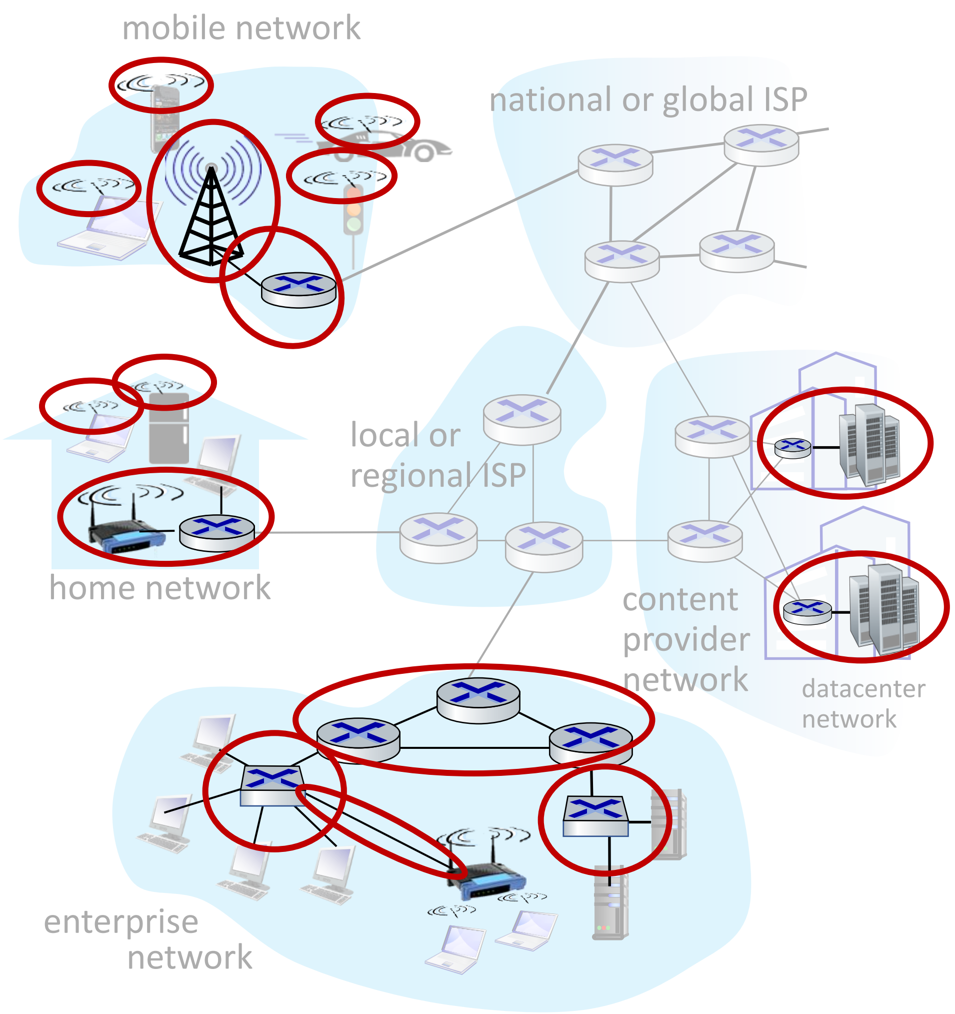
Access networks and physical media

Q: How to connect end systems to edge router?

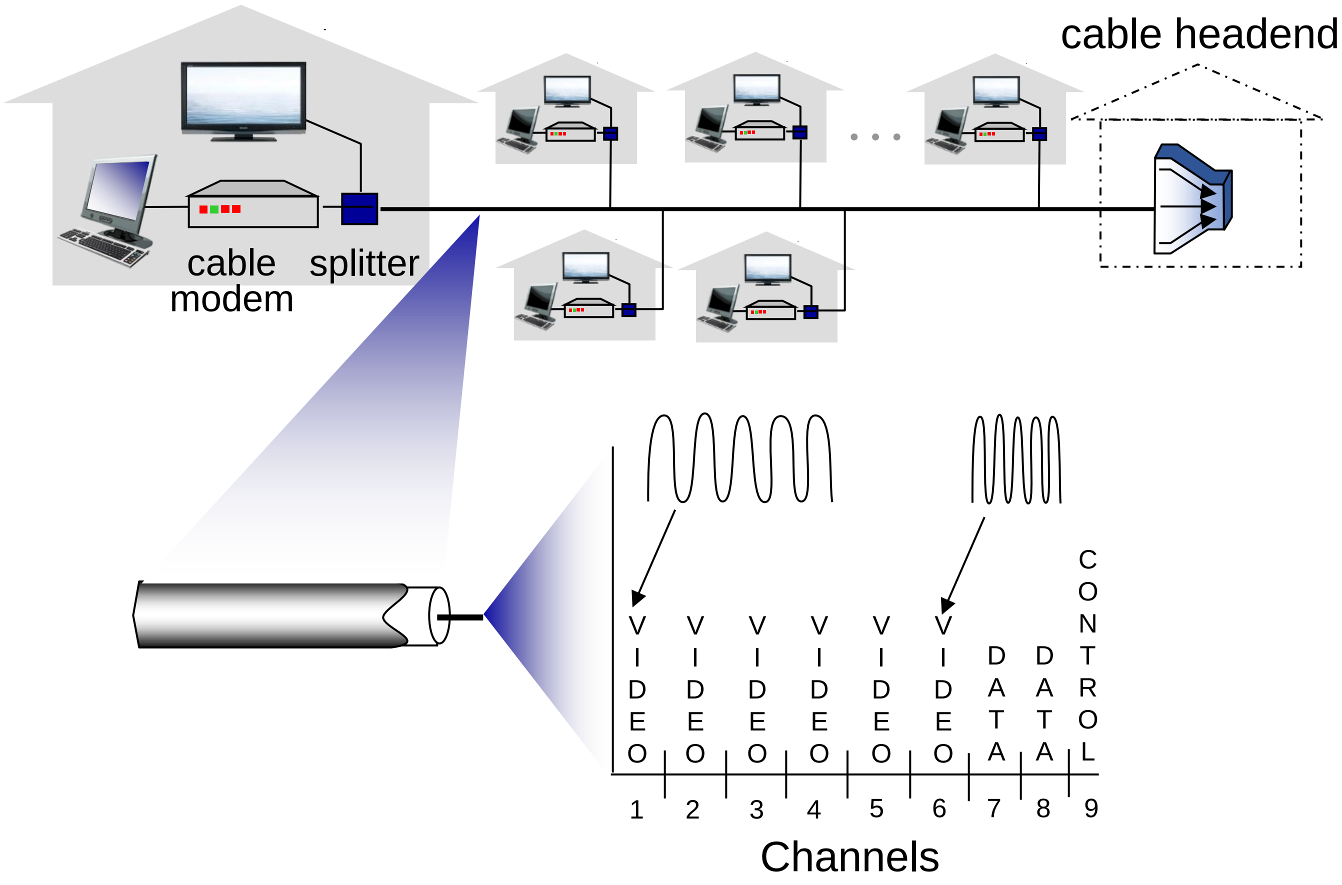
- residential access nets
- institutional access networks (school, company)
- mobile access networks (WiFi, 4G/5G)

What to look for:

- transmission rate (bits per second) of access network?
- shared or dedicated access among users?

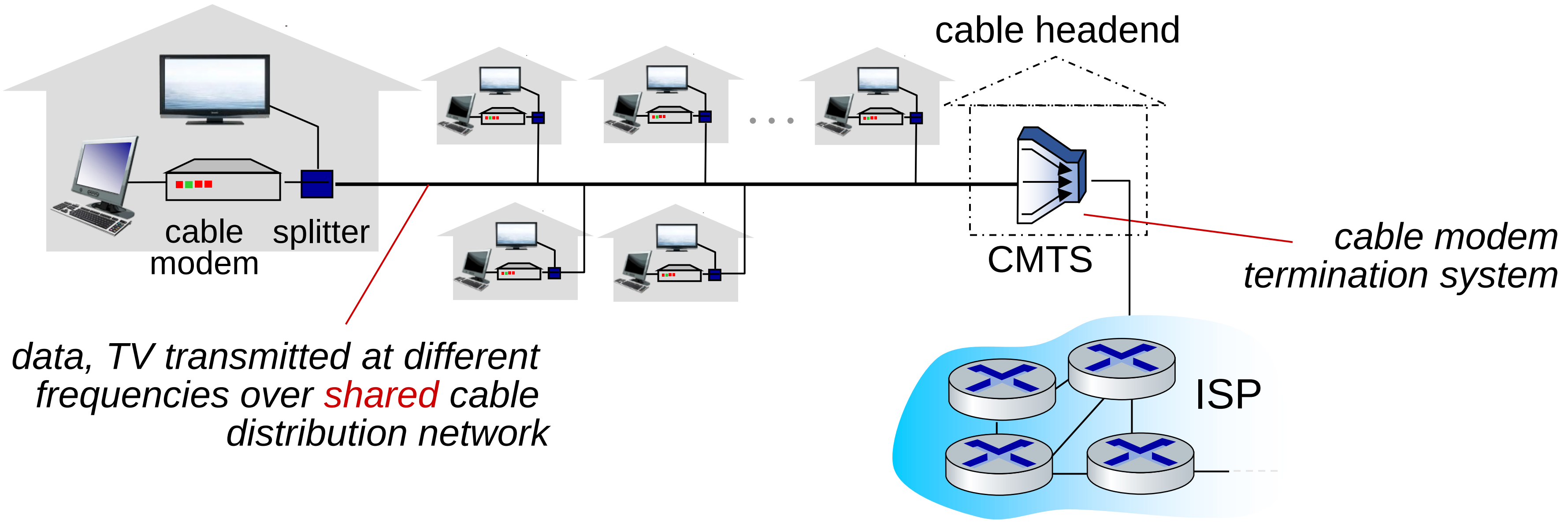


Access networks: cable-based access



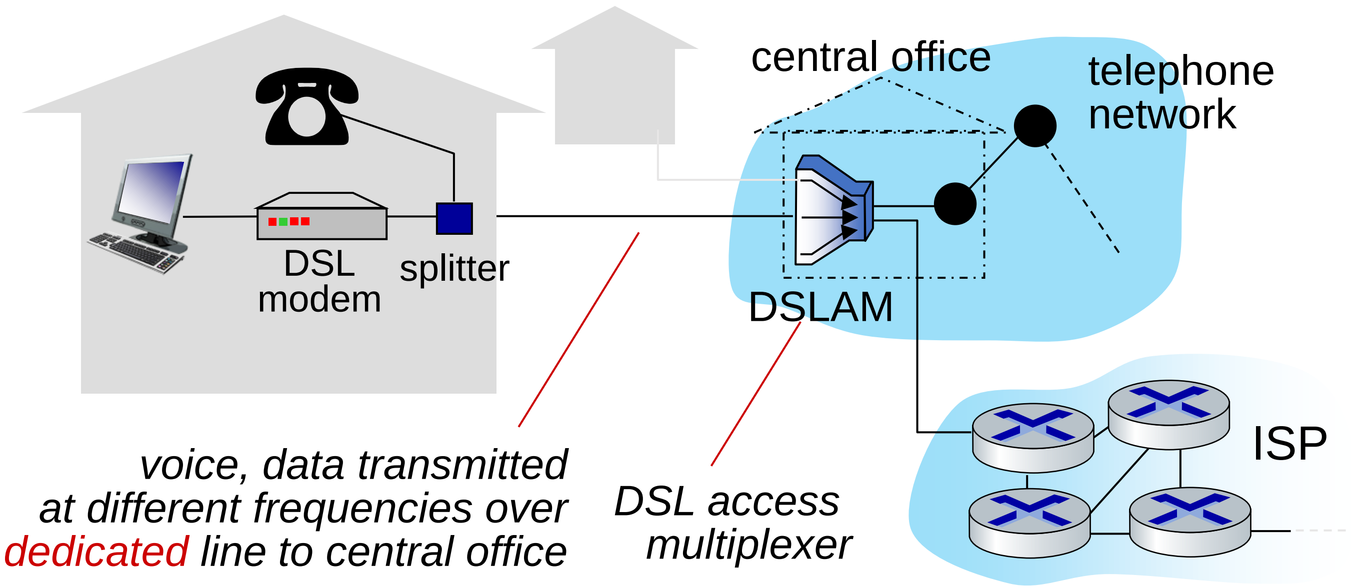
frequency division multiplexing (FDM): different channels transmitted in different frequency bands

Access networks: cable-based access



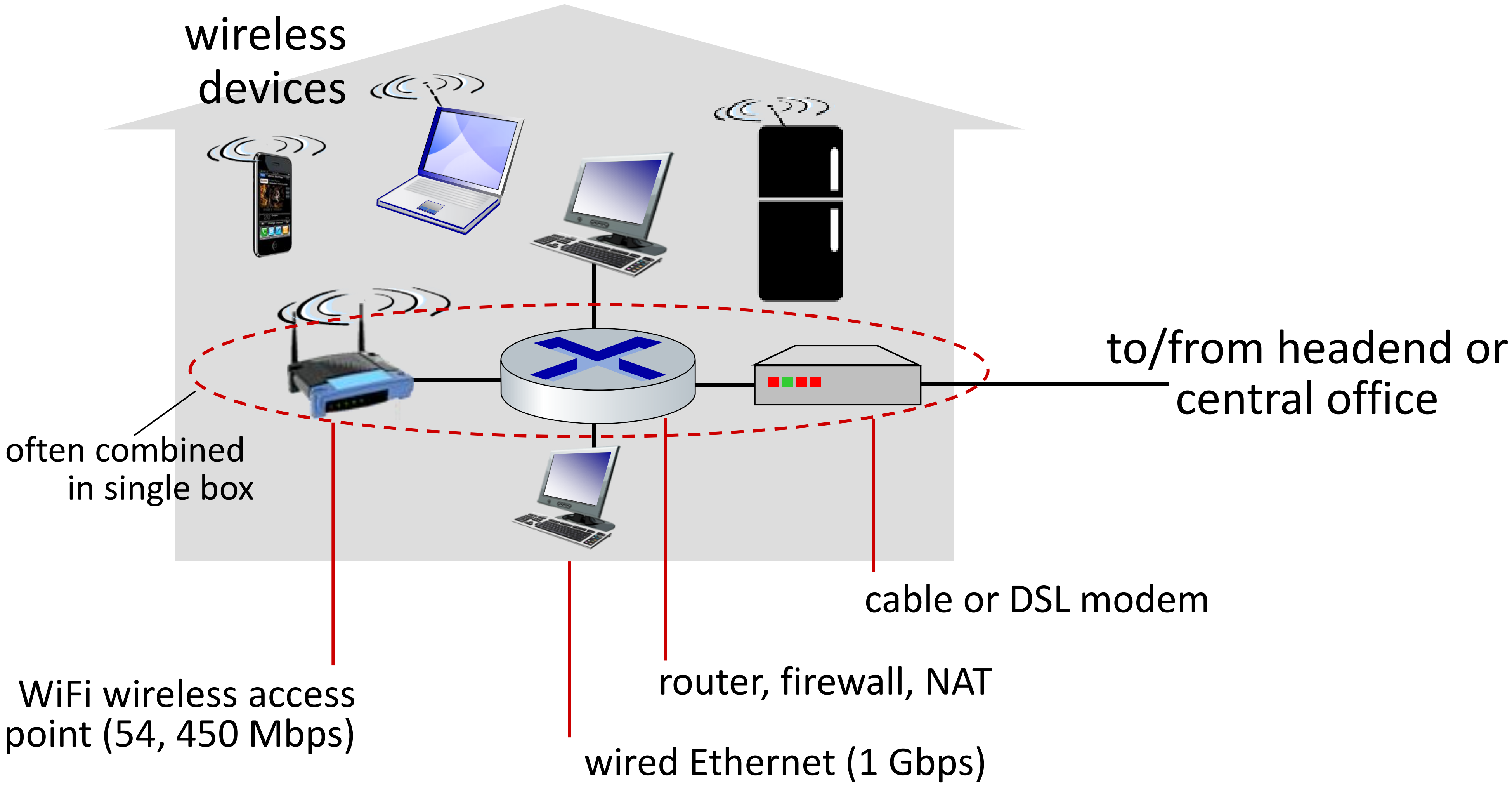
- HFC: hybrid fiber coax
  - asymmetric: up to 40 Mbps – 1.2 Gbps downstream transmission rate, 30–100 Mbps upstream transmission rate
- network of cable, fiber attaches homes to ISP router
  - homes *share access network* to cable headend

Access networks: digital subscriber line (DSL)



- use *existing* telephone line to central office DSLAM
  - data over DSL phone line goes to Internet
  - voice over DSL phone line goes to telephone net
- 24-52 Mbps dedicated downstream transmission rate
- 3.5-16 Mbps dedicated upstream transmission rate

Access networks: home networks



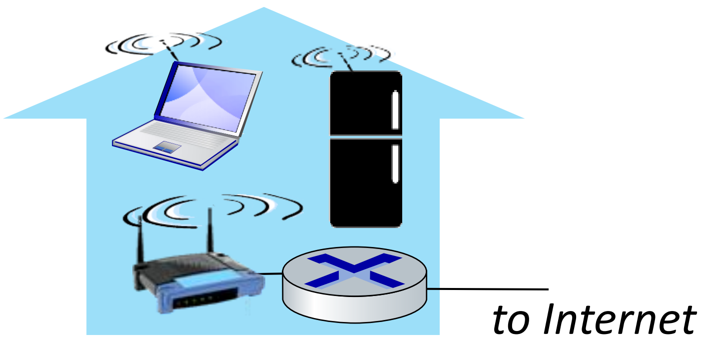
Wireless access networks

Shared *wireless* access network connects end system to router

- via base station aka "access point"

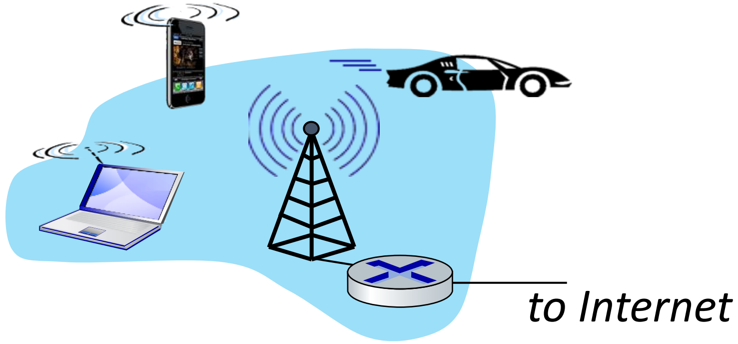
Wireless local area networks (WLANs)

- typically within or around building (~100 ft)
- 802.11b/g/n (WiFi): 11, 54, 450 Mbps transmission rate

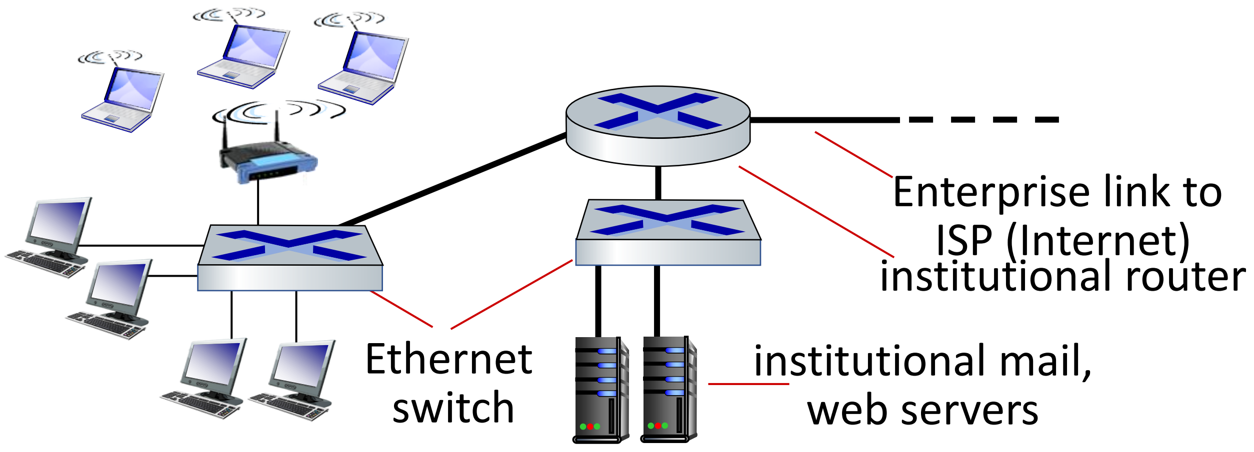


Wide-area cellular access networks

- provided by mobile, cellular network operator (10's km)
- 10's Mbps
- 4G/5G cellular networks (6G coming)



Access networks: enterprise networks

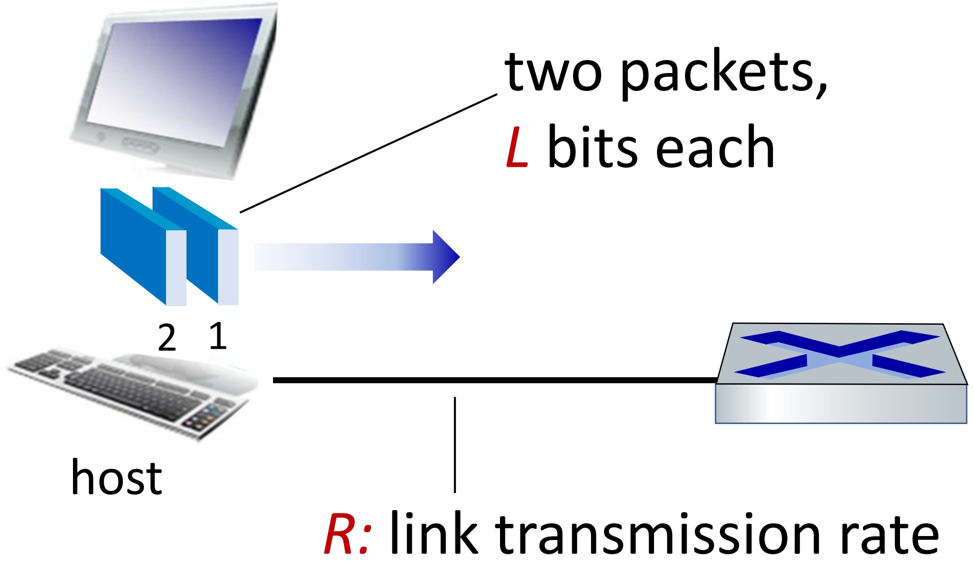


- companies, universities, etc.
- mix of wired, wireless link technologies, connecting a mix of switches and routers (we'll cover differences shortly)
  - Ethernet: wired access at 100Mbps, 1Gbps, 10Gbps
  - WiFi: wireless access points at 11, 54, 450 Mbps

Host: sends *packets* of data

host sending function:

- takes application message
- breaks into smaller chunks, known as *packets*, of length *L* bits
- transmits packet into access network at *transmission rate R*
  - link transmission rate, aka link *capacity, aka link bandwidth*



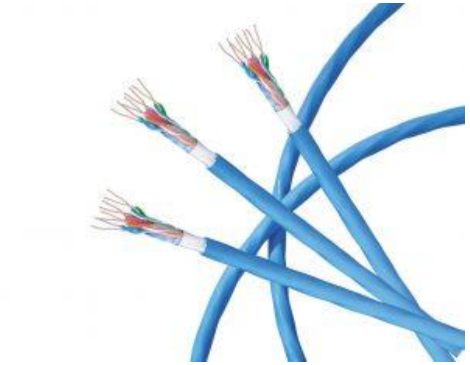
packet transmission delay = time needed to transmit *L*-bit packet into link = $\frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$

Links: physical media

- bit: propagates between transmitter/receiver pairs
- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: copper, fiber, coax
- unguided media:
  - signals propagate freely, e.g., radio

Twisted pair (TP)

- two insulated copper wires
  - Category 5: 100 Mbps, 1 Gbps Ethernet
  - Category 6: 10Gbps Ethernet



Links: physical media

Coaxial cable:

- two concentric copper conductors
- bidirectional
- broadband:
  - multiple frequency channels on cable
  - 100's Mbps per channel

Fiber optic cable:

- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (10's-100's Gbps)
- low error rate:
  - repeaters spaced far apart
  - immune to electromagnetic noise

Links: physical media

Wireless radio

- signal carried in electromagnetic spectrum
- no physical "wire"
- broadcast and "half-duplex" (sender to receiver)
- propagation environment effects:
  - reflection
  - obstruction by objects
  - Interference/noise

Transmission Modes

- **Simplex:**
  - The signal is sent in one direction (only one device can transmit the signal)
- **Half-Duplex:**
  - The signal is sent in both directions, but one at a time
- **Full-Duplex:**
  - The signal is sent in both directions at the same time

Radio link types:

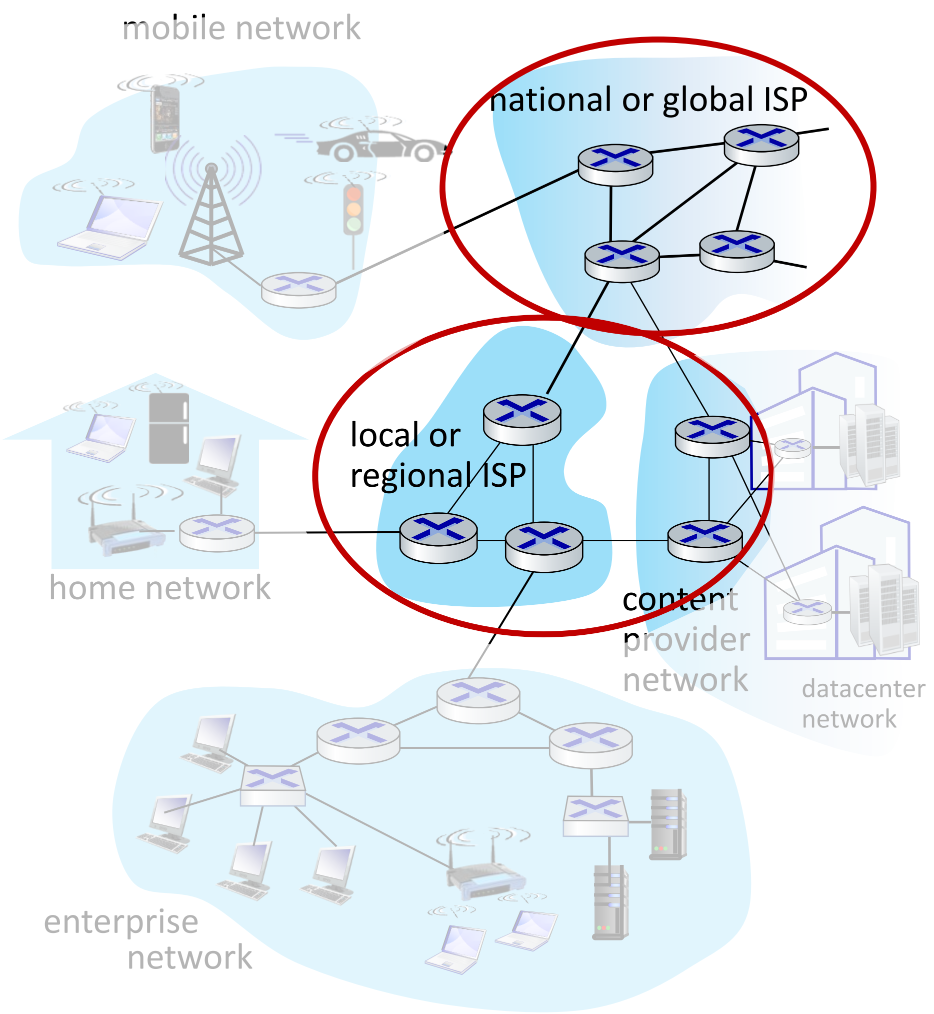
- Wireless LAN (WiFi)
  - 10-100's Mbps; 10's of meters
- wide-area (e.g., 4G cellular)
  - 10's Mbps over ~10 Km
- Bluetooth: cable replacement
  - short distances, limited rate
- terrestrial microwave
  - point-to-point; up to 45 Mbps channels
- satellite
  - up to 45 Mbps per channel
  - 270 msec end-end delay
  - geosynchronous versus low-earth-orbit

Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- **Network core:** packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

The network core

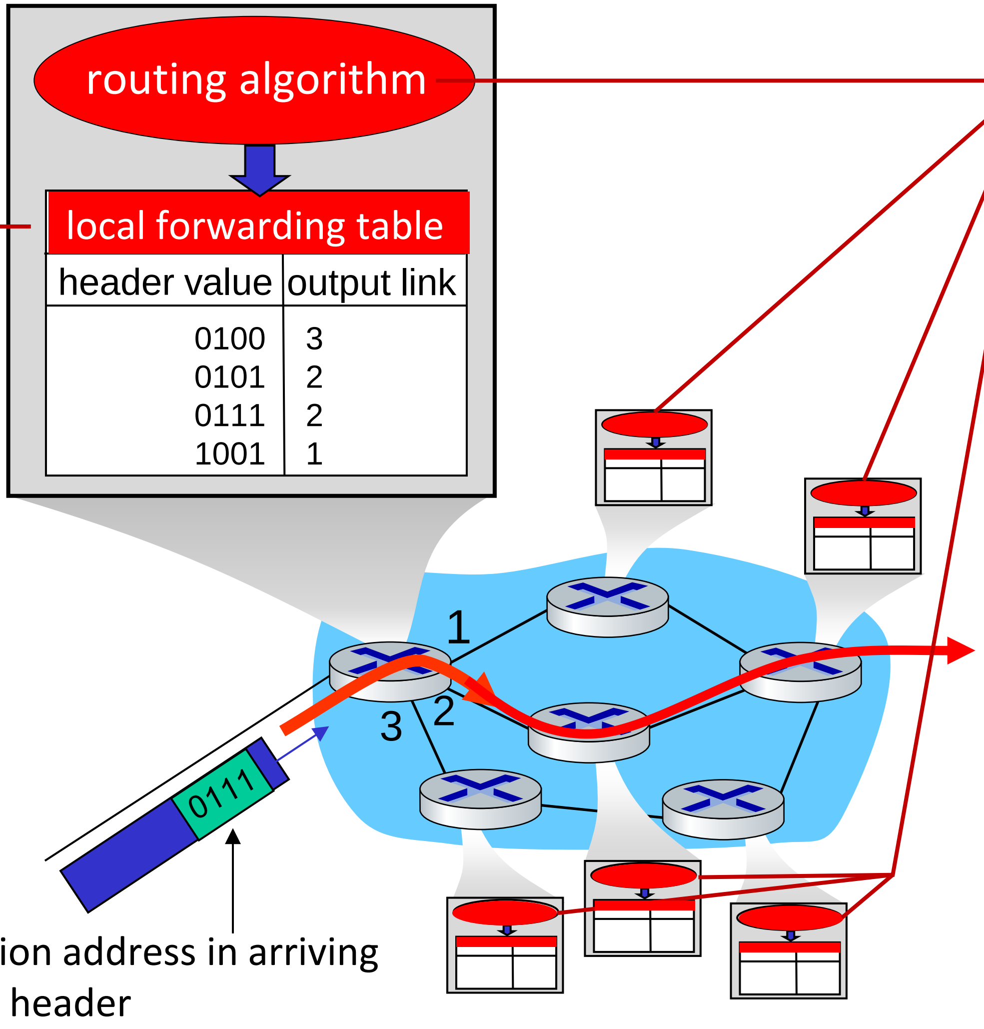
- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
  - network forward packets from one router to the next, across links on path from source to destination
  - each packet transmitted at full link capacity



Two key network-core functions

Forwarding:

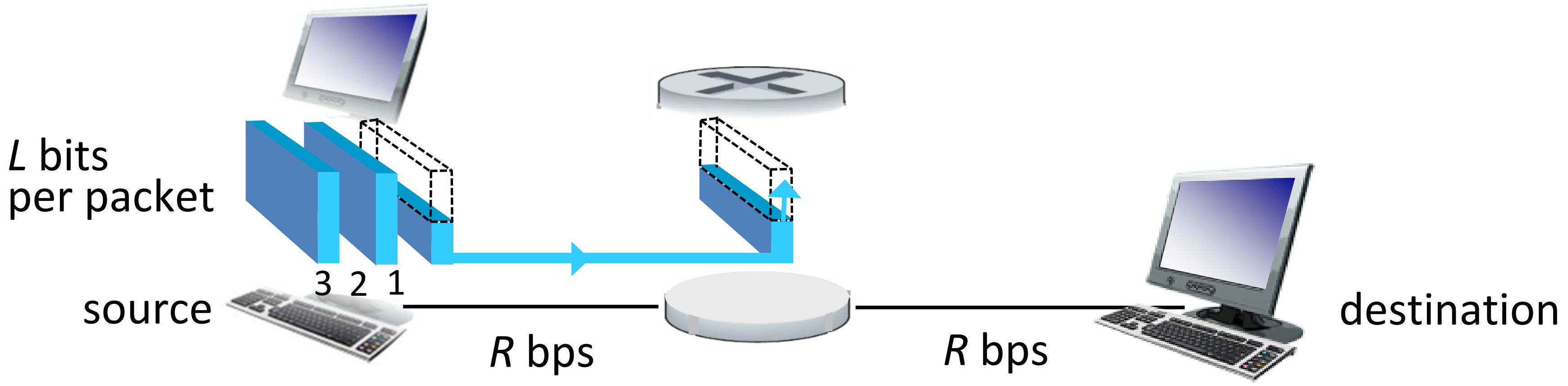
- *local* action: move arriving packets from router's input link to appropriate router output link



Routing:

- *global* action: determine source-destination paths taken by packets
- routing algorithms

Packet-switching: store-and-forward

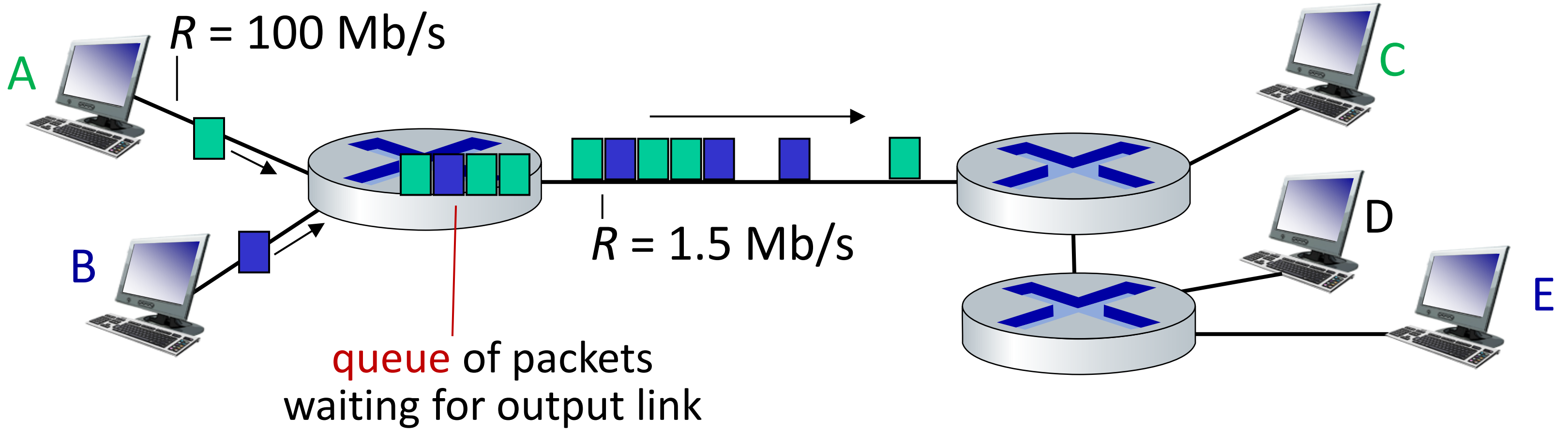


- Transmission delay: takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *Store and forward: entire* packet must arrive at router before it can be transmitted on next link
- *End-end delay:* 2*L*/*R* (above), assuming zero propagation delay (more on delay shortly)

One-hop numerical example:

- *L* = 10 Kbits
- *R* = 100 Mbps
- one-hop transmission delay = 0.1 msec

Packet-switching: queueing delay, loss



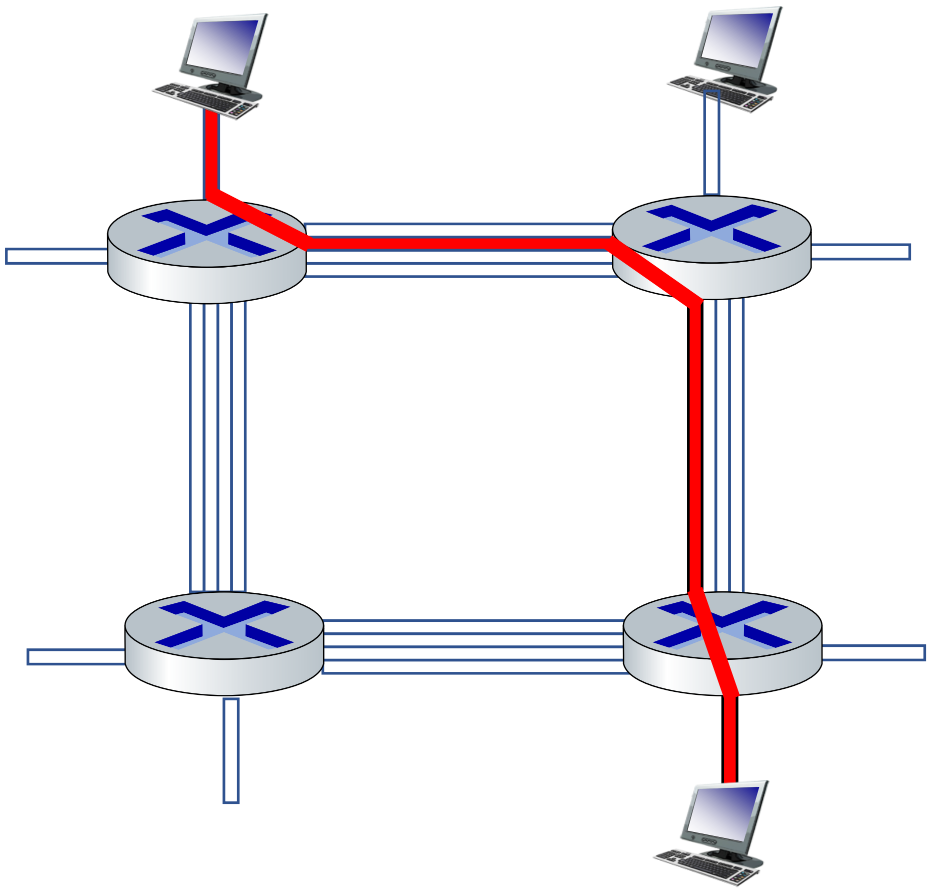
Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up

Alternative to packet switching: circuit switching

end-end resources allocated to, reserved for “call” between source and destination

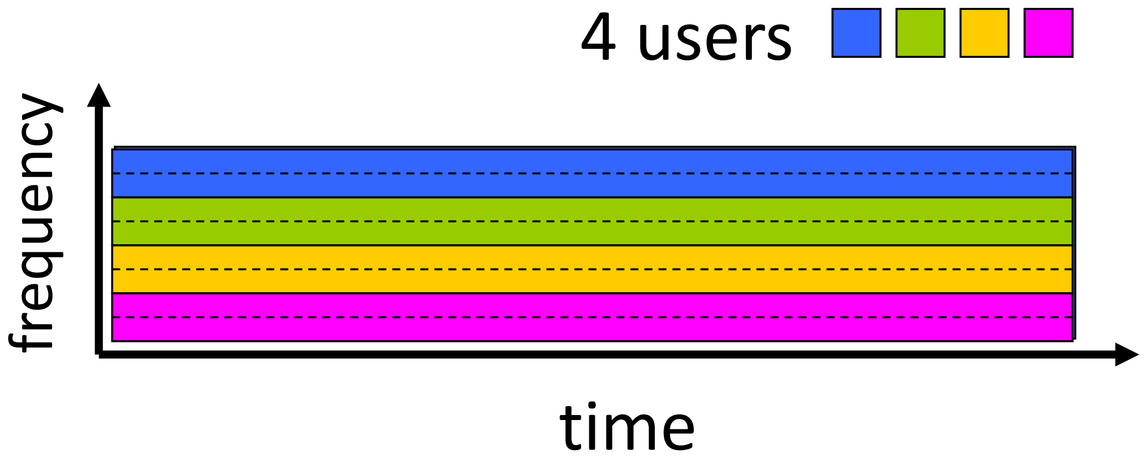
- in diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (no sharing)
- commonly used in traditional telephone networks



Circuit switching: FDM and TDM

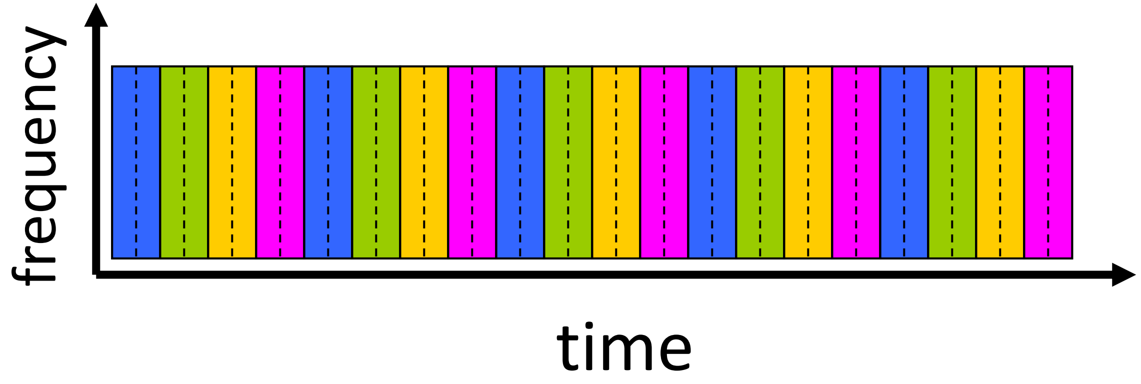
Frequency Division Multiplexing (FDM)

- optical, electromagnetic frequencies divided into (narrow) frequency bands
- each call allocated its own band, can transmit at max rate of that narrow band

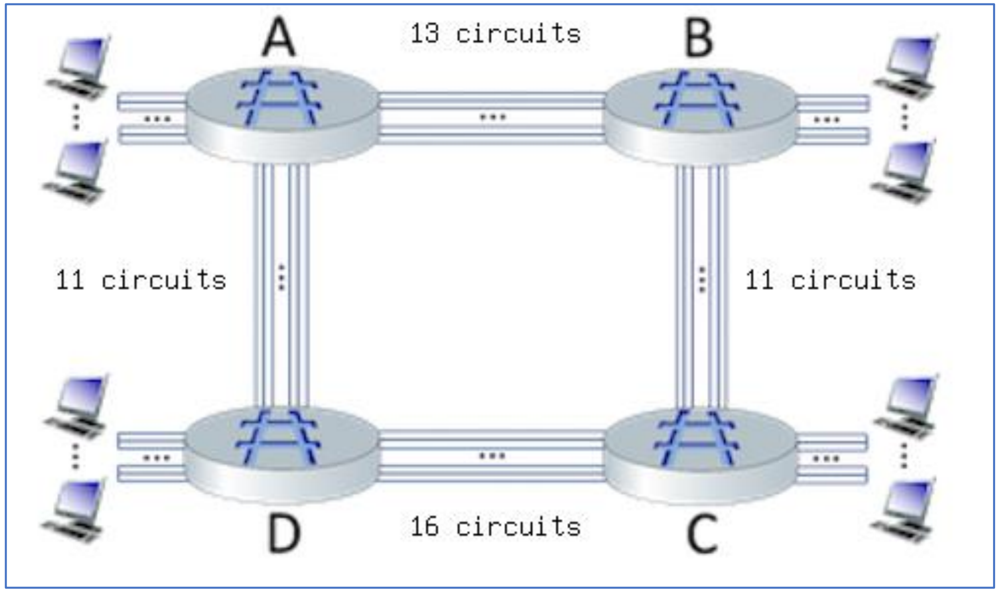


Time Division Multiplexing (TDM)

- time divided into slots
- each call allocated periodic slot(s), can transmit at maximum rate of (wider) frequency band, but only during its time slot(s)



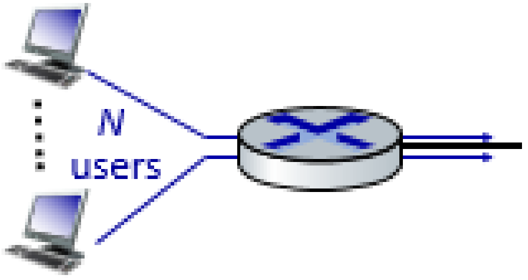
Example:



Suppose there are 13 circuits between A and B, 11 circuits between B and C, 16 circuits between C and D, and 11 circuits between D and A.

- What is the maximum number of connections that can be ongoing in the network at any one time? **51 connections.**
- Suppose that these maximum number of connections are all ongoing. What happens when another call connection request arrives to the network, will it be accepted? **No.**
- Suppose that every connection requires 2 consecutive hops, and calls are connected clockwise. For example, a connection can go from A to C, from B to D, from C to A, and from D to B. With these constraints, what is the is the maximum number of connections that can be ongoing in the network at any one time? **22 connections.**

Packet switching Vs. circuit switching



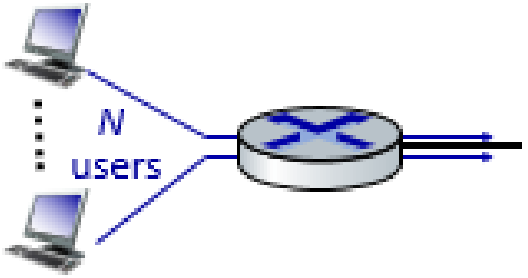
- **Problem:** users share a 1.5 Mbps link. Also suppose each user requires 250 Kbps when transmitting, but each user transmits only 10% of the time.
  - When circuit switching is used, how many users can be supported?

    1.5 Mbps / 250 Kbps per user = 6 users.
  - Packet switching is used. Find the probability that a given user is transmitting.

    P(a given user is transmitting) = p = 0.10
  - Suppose there are 20 users. Find the probability that at any given time, exactly ***n*** users are transmitting simultaneously.

P(exactly *n* users are transmitting simultaneously) = $\binom{20}{n} p^n (1-p)^{20-n}$

Packet switching Vs. circuit switching



- Suppose there are 20 users. Find the probability when 7 or more users transmitting simultaneously.

P(7 or more users are transmitting simultaneously)
= 1 – P(6 or less users are transmitting simultaneously)
= 1 – [P(6 users are transmitting simultaneously) + P(5 users are transmitting simultaneously) + … + P(1 user is transmitting simultaneously) + P(no user is transmitting)]
= $1 - \sum_{n=0}^{6} \binom{20}{n} p^n (1-p)^{20-n}$

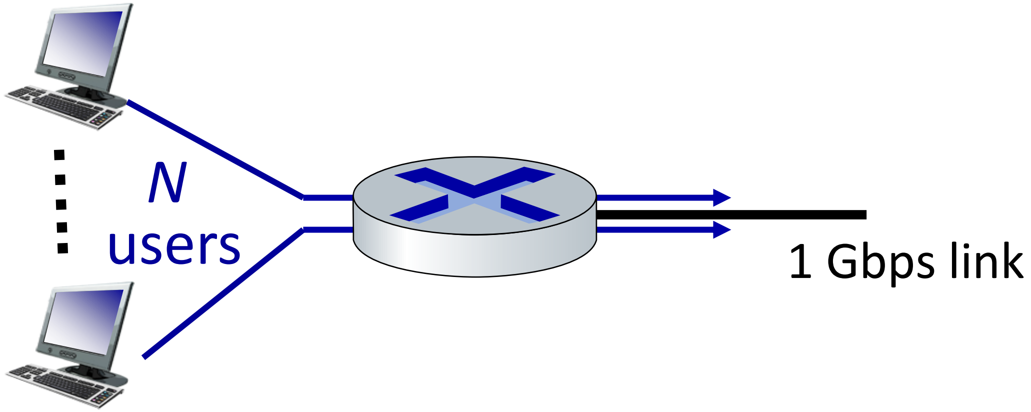
→ P(7 or more users are transmitting simultaneously) = 0.0023861

- Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive

Packet switching versus circuit switching

Example:

- 1 Gb/s link
- each user:
  - 100 Mb/s when "active"
  - active 10% of time



Question: How many users can use this network under circuit-switching and packet-switching?

- *circuit-switching:* 10 users
- *packet-switching:* with 35 users, probability > 10 active at same time is less than 0.0004 *

Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

packet switching allows more users to use network!

- Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive

Packet switching versus circuit switching

Is packet switching a “slam dunk winner”?

- great for “bursty” data – sometimes has data to send, but at other times not
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss due to buffer overflow
  - protocols needed for reliable data transfer, congestion control
- *Q:* How to provide circuit-like behavior?
  - bandwidth guarantees traditionally used for audio/video applications

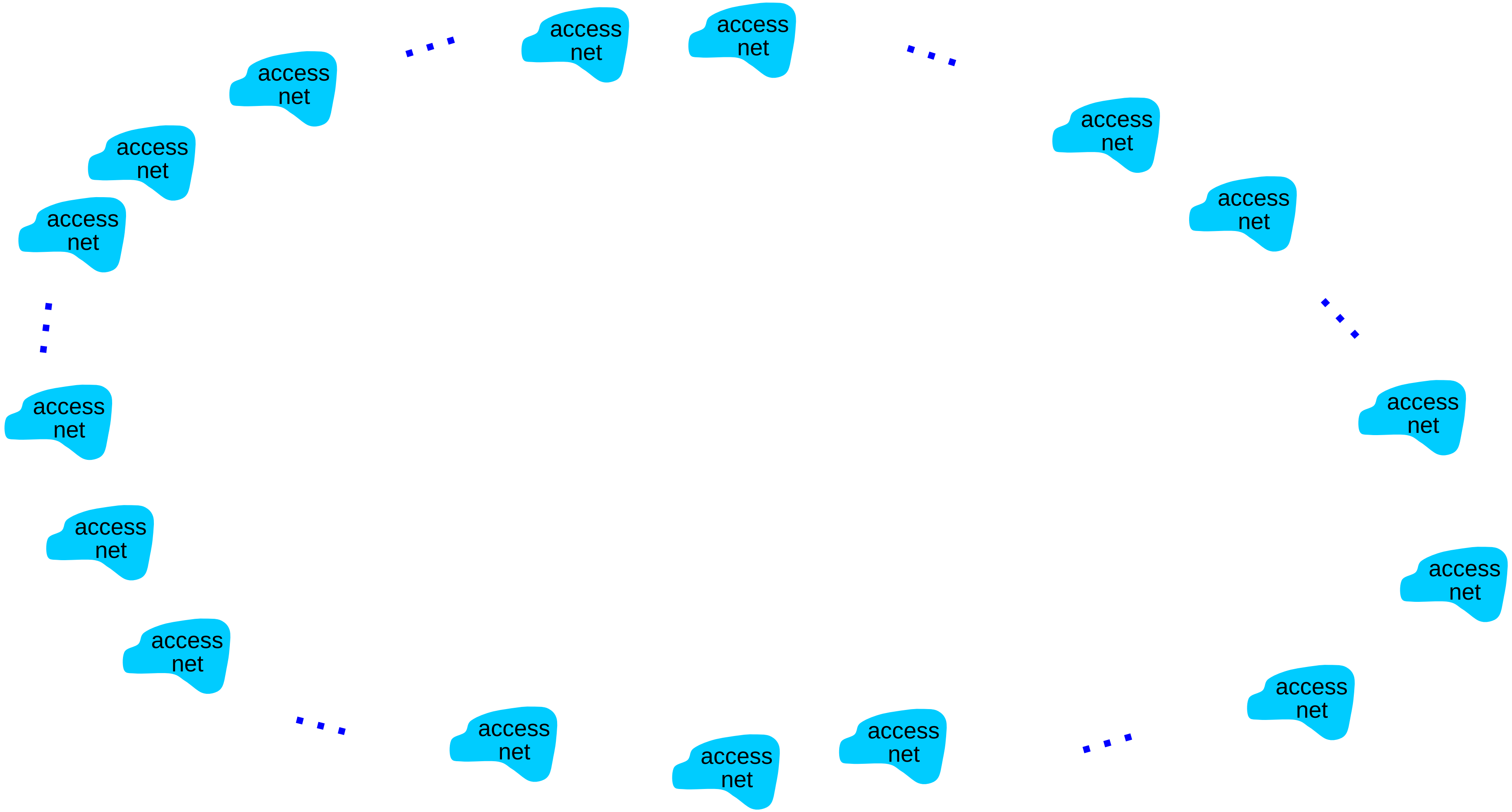
Q: human analogies of reserved resources (circuit switching) versus on-demand allocation (packet switching)?

Internet structure: a "network of networks"

- Hosts connect to Internet via access Internet Service Providers (ISPs)
  - residential, enterprise (company, university, commercial) ISPs
- Access ISPs in turn must be interconnected
  - so that any two hosts can send packets to each other
- Resulting network of networks is very complex
  - evolution was driven by economics and national policies
- Let's take a stepwise approach to describe current Internet structure

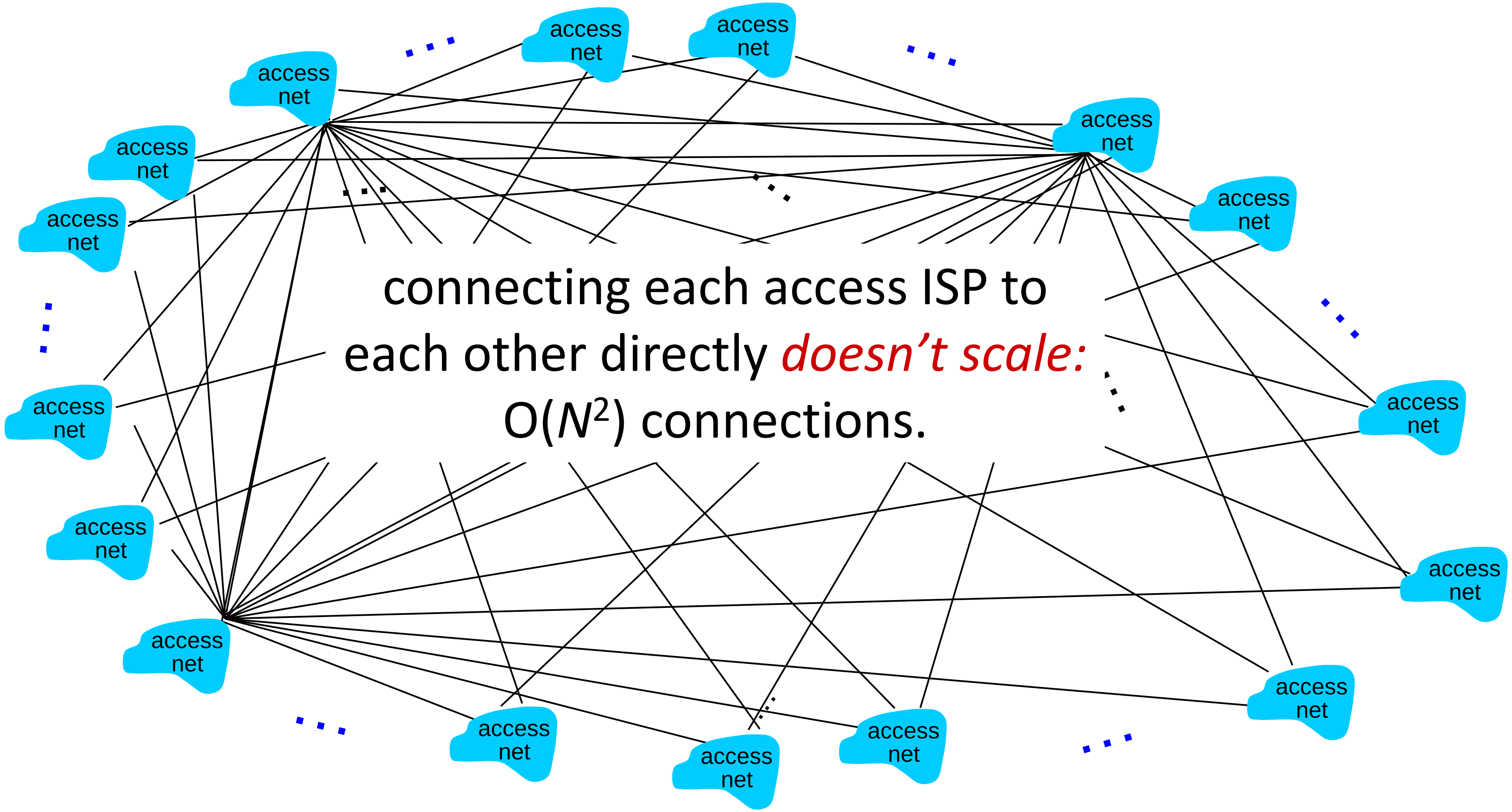
Internet structure: a "network of networks"

Question: given *millions* of access ISPs, how to connect them together?



Internet structure: a "network of networks"

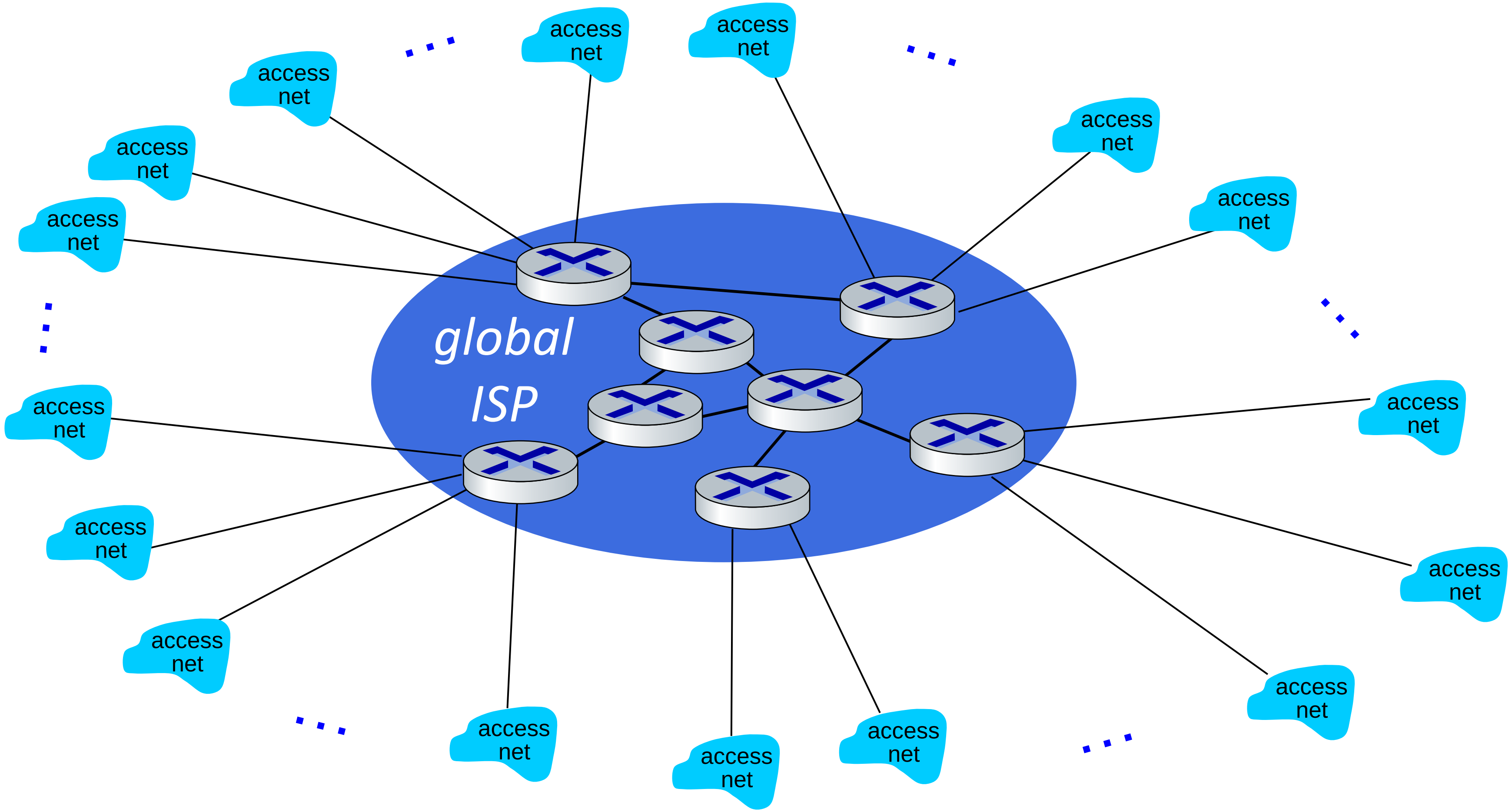
Question: given *millions* of access ISPs, how to connect them together?



Internet structure: a "network of networks"

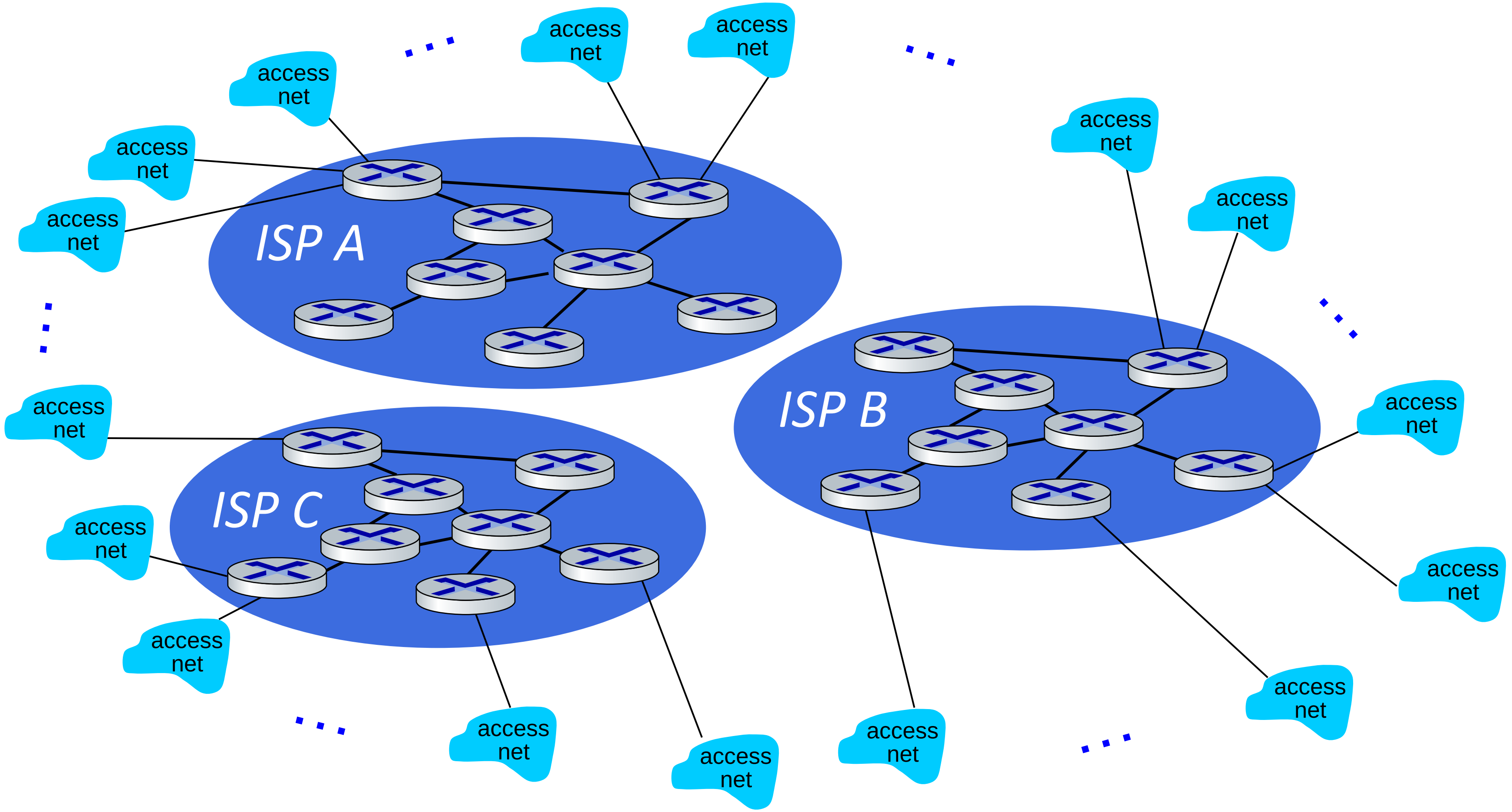
Option: connect each access ISP to one global transit ISP?

Customer and provider ISPs have economic agreement.



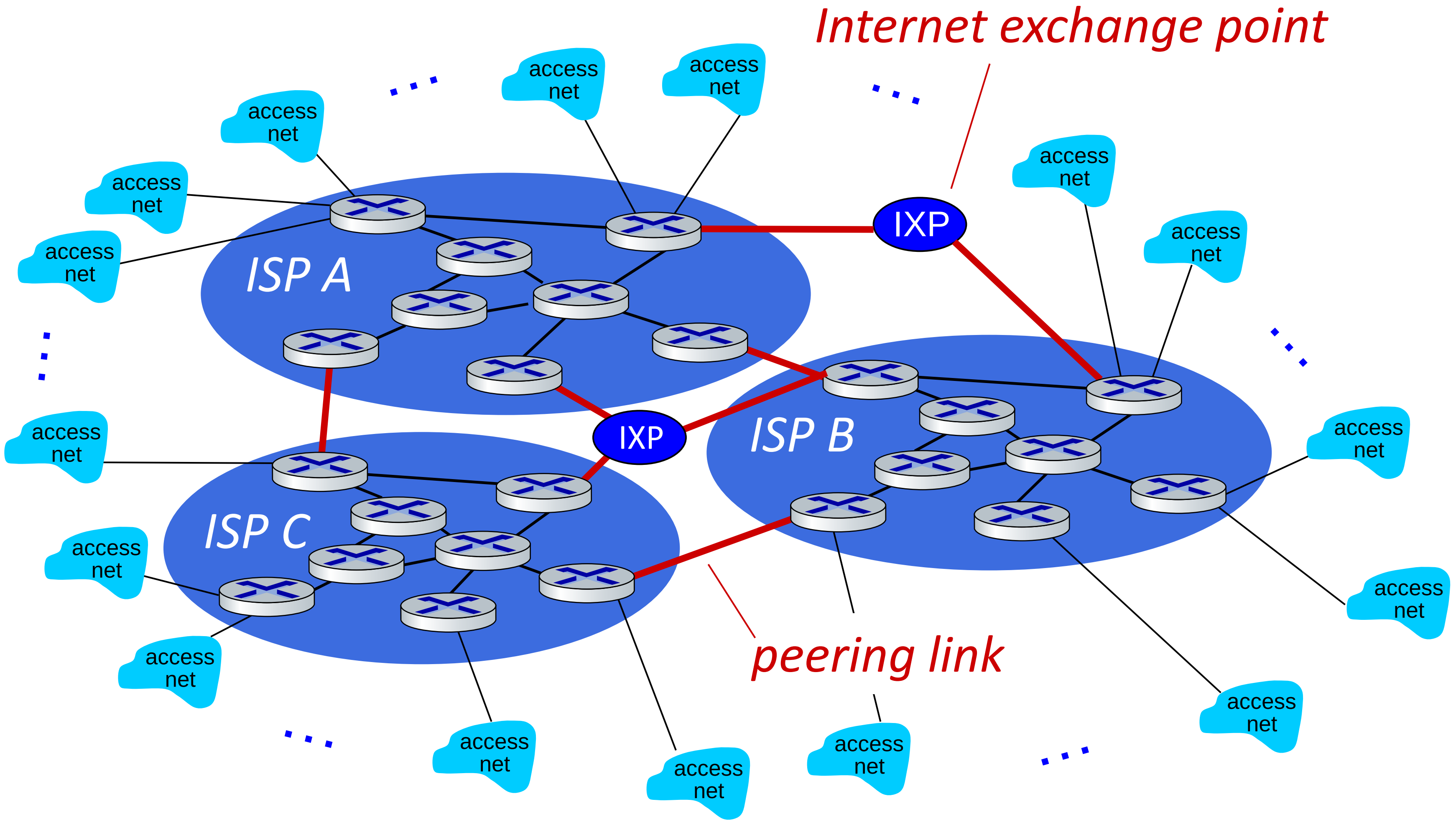
Internet structure: a "network of networks"

But if one global ISP is viable business, there will be competitors ….



Internet structure: a "network of networks"

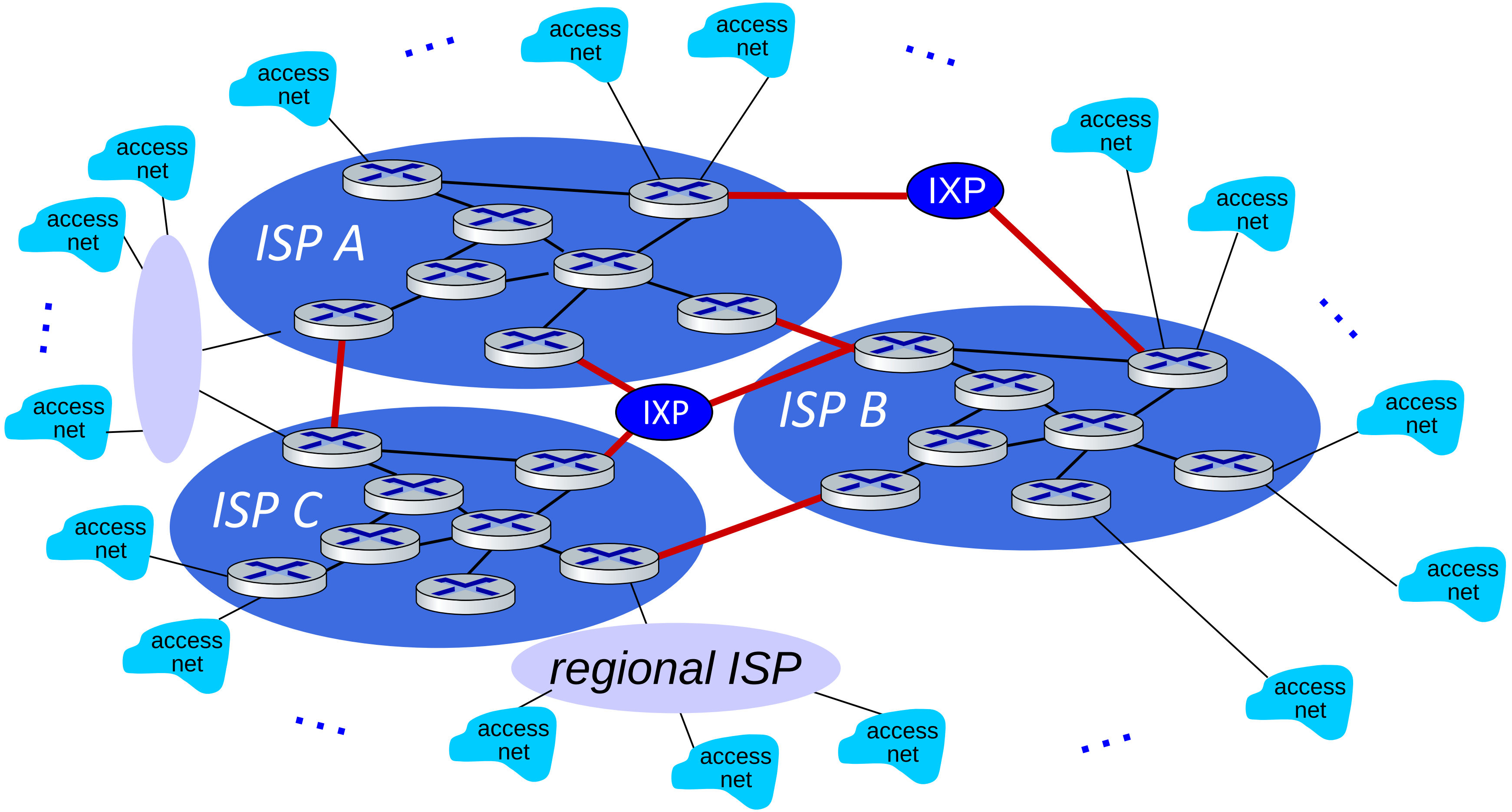
But if one global ISP is viable business, there will be competitors …. who will want to be connected



The Internet Exchange Map: https://www.internetexchangemap.com/

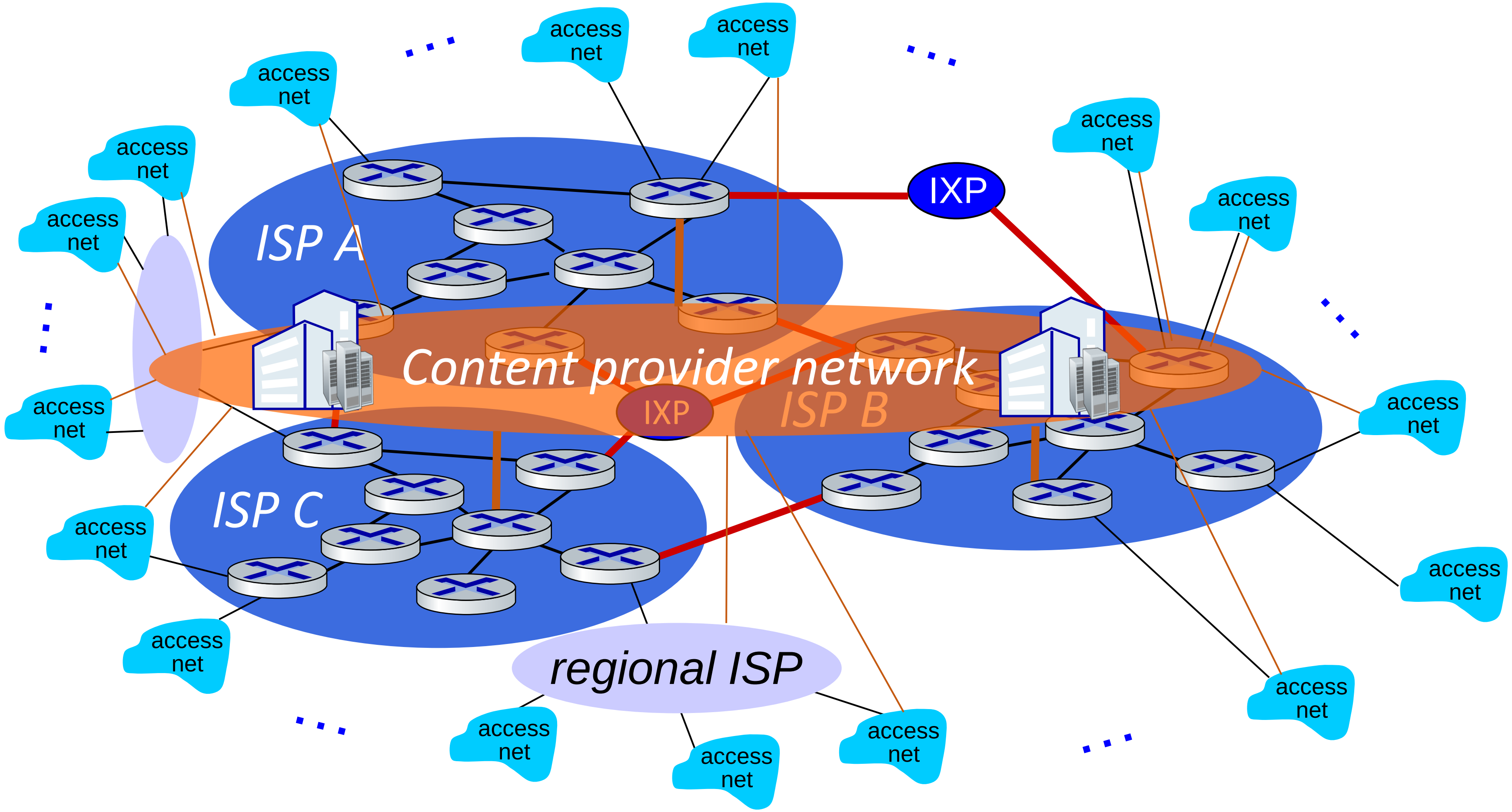
Internet structure: a “network of networks”

… and regional networks may arise to connect access nets to ISPs

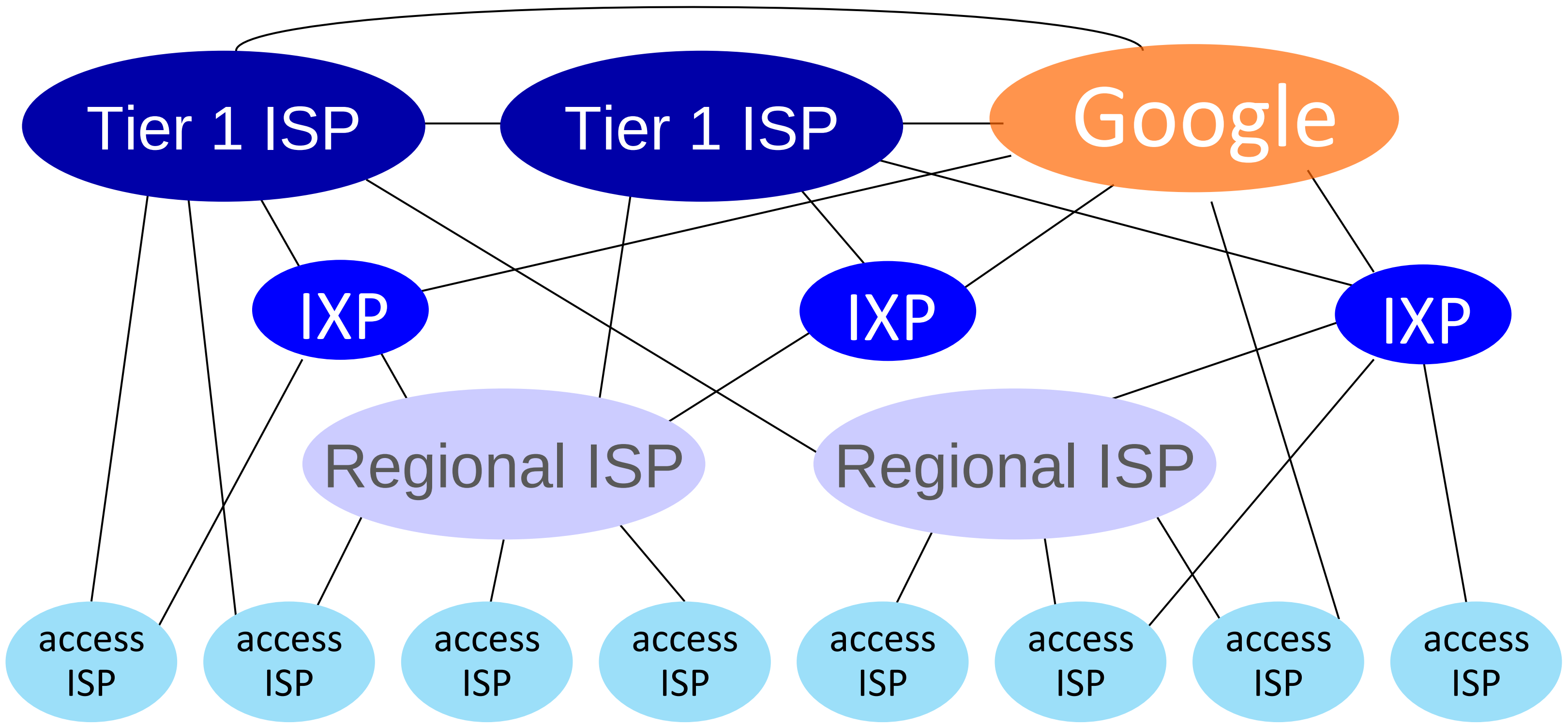


Internet structure: a "network of networks"

… and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users



Internet structure: a "network of networks"



- An Internet Exchange Point (IXP) is a physical facility where Internet infrastructure companies, such as Internet Service Providers (ISPs) and Content Delivery Networks (CDNs), interconnect.
- ISPs typically pay the IXP based on the amount of traffic they route through it.
- To reduce costs, two ISPs may establish a direct peering agreement, enabling them to exchange traffic directly and bypass the IXP.
- An ISP can connect to multiple ISPs at higher levels, a practice known as multi-homing.
- Multi-homing is highly effective for:
  - Ensuring continuous connectivity by eliminating a single point of failure.
  - Providing a cost-effective solution for maintaining network reliability.

At "center": small number of well-connected large networks

- "tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
- content provider networks (e.g., Google, Facebook): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs

Tier-1 ISP Network map: Sprint (2019)

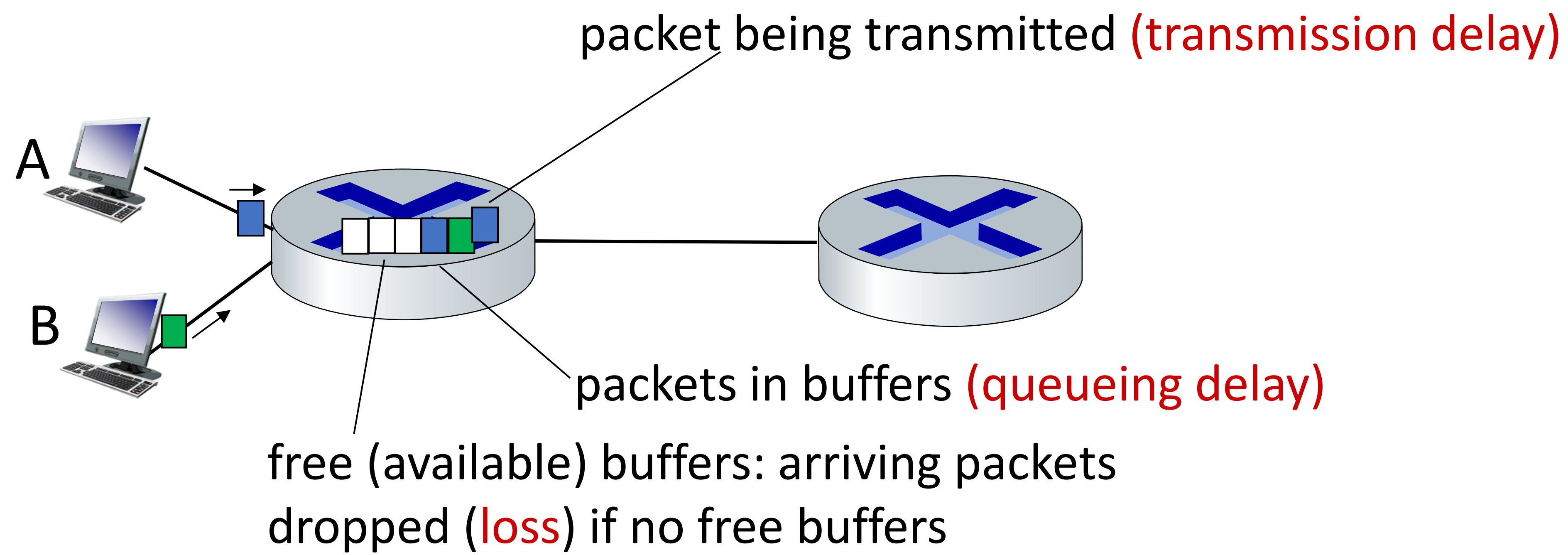
Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

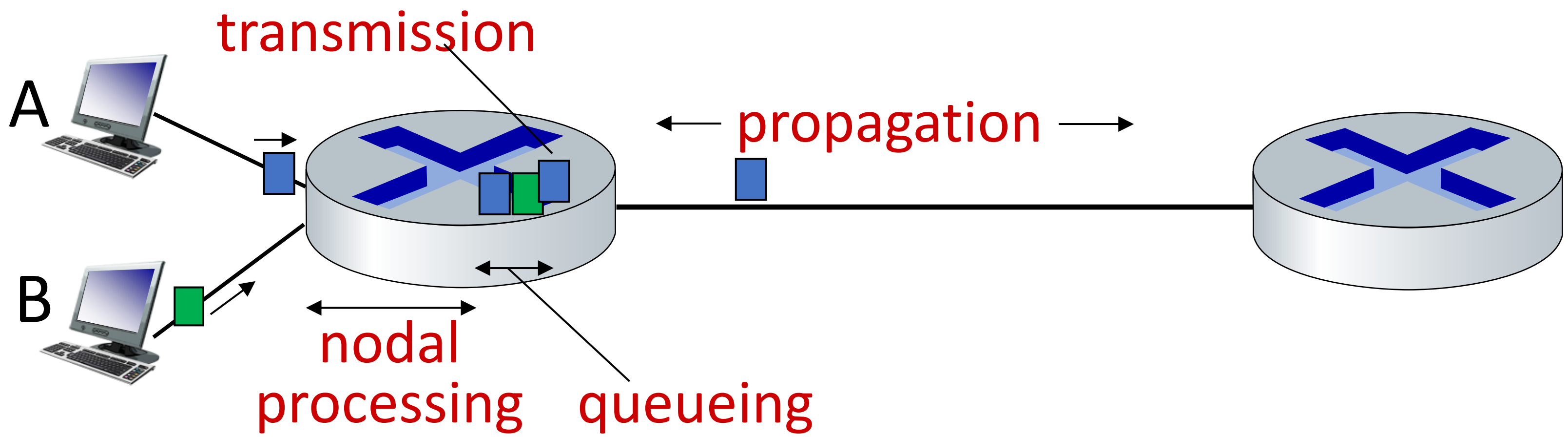
How do packet loss and delay occur?

packets *queue* in router buffers

- packets queue, wait for turn
- arrival rate to link (temporarily) exceeds output link capacity: packet loss



Packet delay: four sources

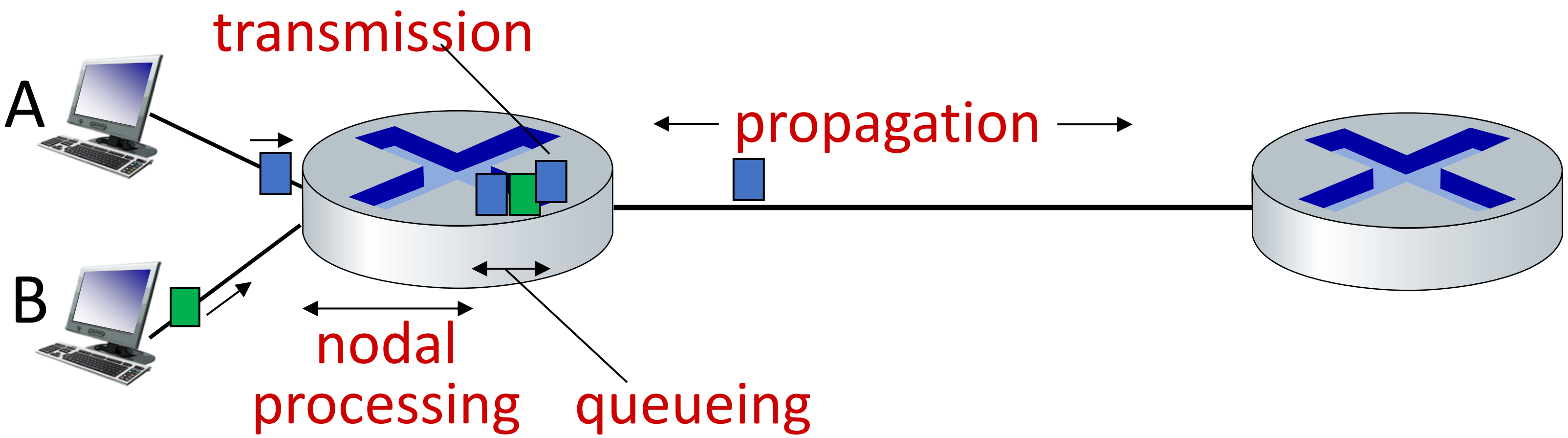


$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay
- time waiting at output link for transmission
- depends on congestion level of router

Packet delay: four sources



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

d_{trans}: transmission delay:

- L: packet length (bits)
- R: link *transmission rate (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed (~2x10^8 m/sec)
- $d_{prop} = d/s$

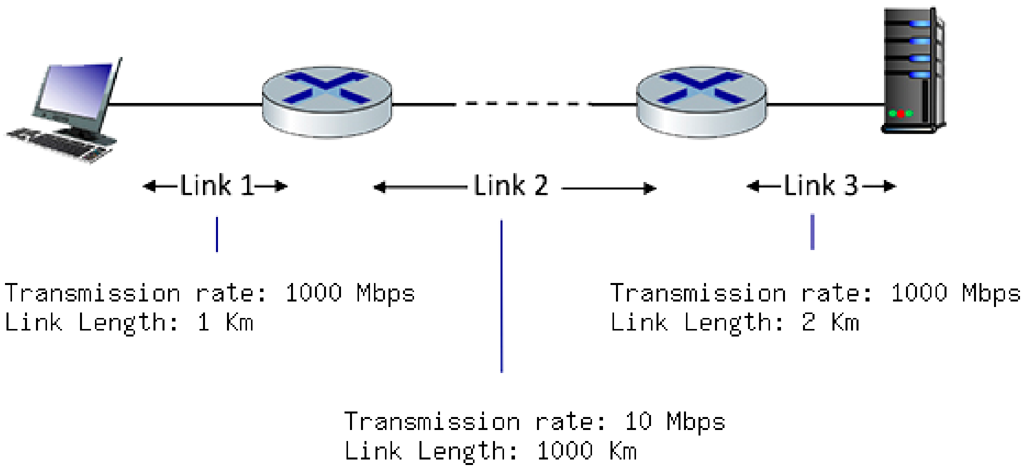
d_{trans} and d_{prop} *very* different

- Check out the online interactive exercises: http://gaia.cs.umass.edu/kurose_ross/interactive

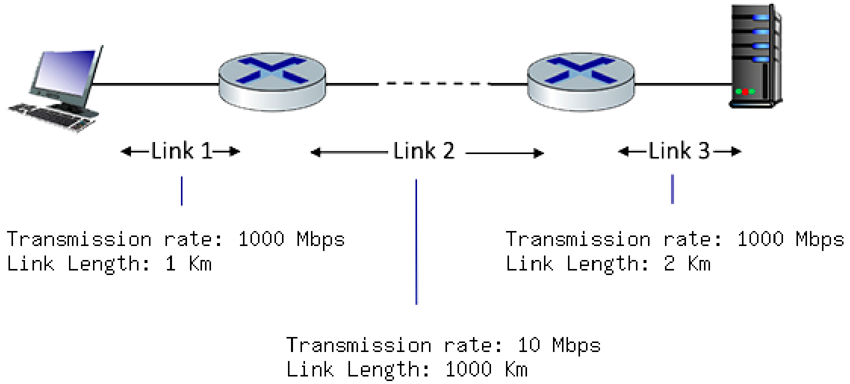
Example:

Assume the length of a packet is 12000 bits. The speed of light propagation delay on each link is $3x10^8$ m/sec.

- Q1: What is the transmission delay of link 1?
- Q2: What is the propagation delay of link 1?
- Q3: What is the total delay of link 1?
- Q4: What is the transmission delay of link 2?
- Q5: What is the propagation delay of link 2?
- Q6: What is the total delay of link 2?
- Q7: What is the total delay?



Example:



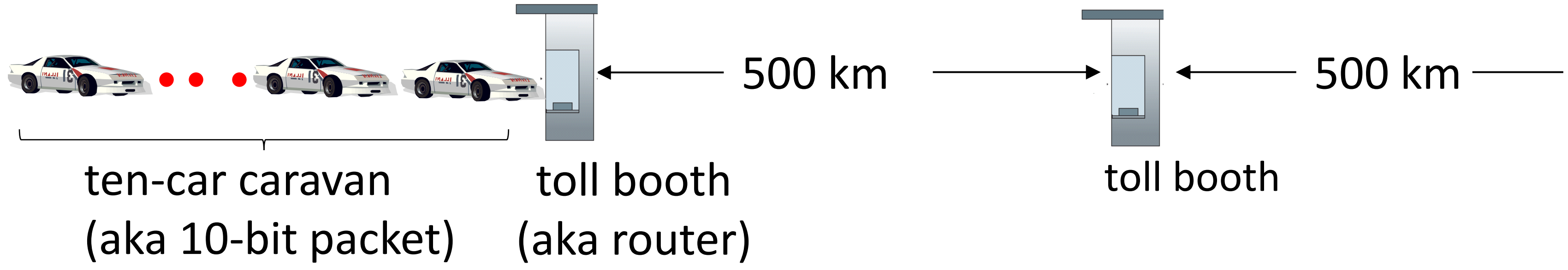
Assume the length of a packet is 12000 bits. The speed of light propagation delay on each link is 3×10^8 m/sec.

- Q1: What is the transmission delay of link 1? **1.2×10^{-5}**
- Q2: What is the propagation delay of link 1? **3.33×10^{-6}**
- Q3: What is the total delay of link 1? **15.33×10^{-6}**
- Q4: What is the transmission delay of link 2? **0.0012**
- Q5: What is the propagation delay of link 2? **0.0033**
- Q6: What is the total delay of link 2? **0.0045**
- Q7: What is the total delay? **$15.33 \times 10^{-6} + 0.0045 + 18.67 \times 10^{-6} = 0.004534$**

Example:

- Assume we want to send an image from mars to earth
- 1920X1080 pixels, assume true color (each pixel is 3 bytes), using 100Mbps
- Distance to mars 213.85 million km
- D_prop = Propagation delay = distance / speed of light
- D_prop = 213.85 * 10^9/ (3 * 10^8) = 712 sec = 11.8 min
- File size = 1920 * 1080 * 3 * 8
- Assume packet size is equal file size
- Transmission delay = 1920 * 1080 * 3 * 8 / (100 * 10^6) = 0.5 sec
- Total Time = 712 + 0.5 = 712.5 sec

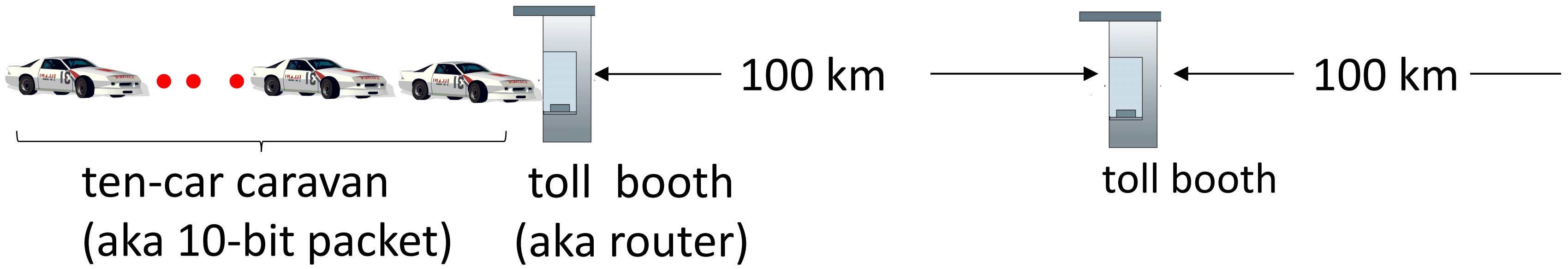
Example: caravan analogy



Suppose the caravan has 10 cars, and that the tollbooth services (that is, transmits) a car at a rate of one car per 1 seconds. Once receiving serving a car proceeds to the next tool both, which is 500 kilometers away at a rate of 20 kilometers per second. Also assume that whenever the first car of the caravan arrives at a tollbooth, it must wait at the entrance to the tollbooth until all of the other cars in its caravan have arrived, and lined up behind it before being serviced at the toll booth.

- Once a car enters service at the tollbooth, how long does it take until it leaves service? **1 sec**.
- How long does it take for the entire caravan to receive service at the tollbooth (that is the time from when the first car enters service until the last car leaves the tollbooth)? **10 sec.**
- Once the first car leaves the tollbooth, how long does it take until it arrives at the next tollbooth? **25 sec.**
- Once the first car leaves the tollbooth, how long does it take until it enters service at the next tollbooth? **34 sec.**

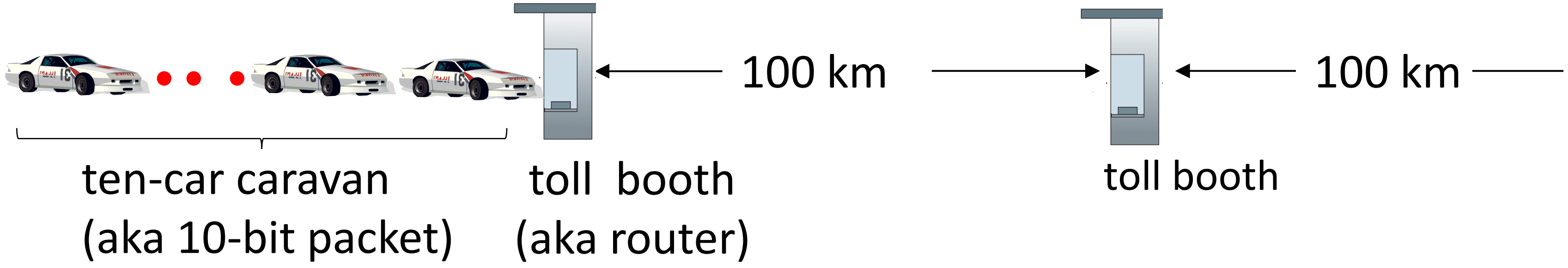
Caravan analogy



- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car ~ bit; caravan ~ packet
- *Q:* How long until caravan is lined up before 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
- *A:* 62 minutes

Caravan analogy



- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q: Will cars arrive to 2nd booth before all cars serviced at first booth?*

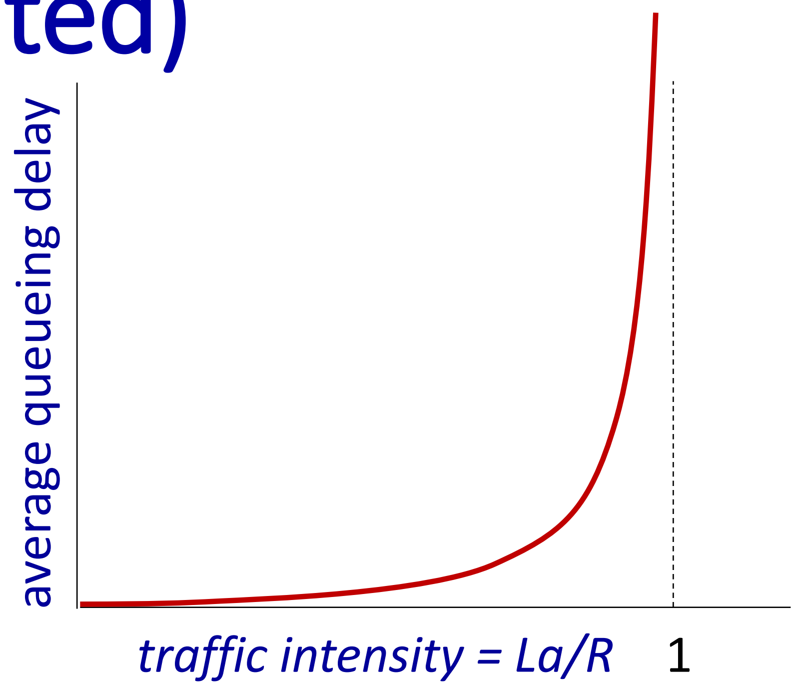
  A: Yes! after 7 min, first car arrives at second booth; three cars still at first booth

Packet queueing delay (revisited)

- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- α*:* average packet arrival rate

$$\frac{L \cdot \alpha}{R} : \frac{\text{arrival rate of bits}}{\text{service rate of bits}}$$

"traffic intensity"



- $L\alpha/R$ ~ 0: avg. queueing delay small
- $L\alpha/R$ → 1: avg. queueing delay large
- $L\alpha/R$ > 1: more "work" arriving is more than can be serviced - average delay infinite!

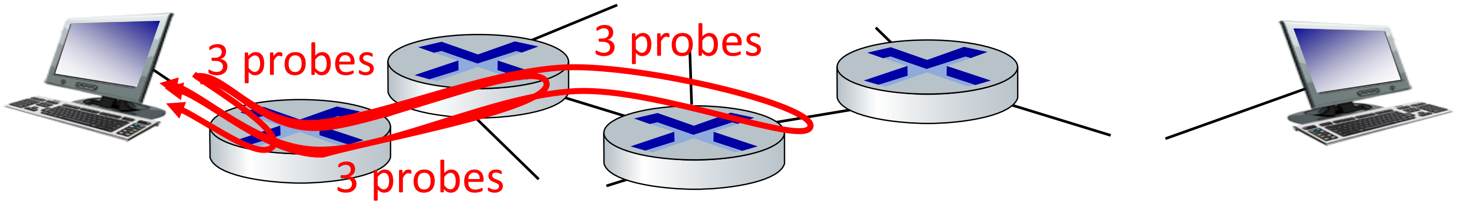
Packet queueing delay

Assume a constant transmission rate of $R = 700000$ bps, a constant packet-length $L = 4600$ bits, and a is the average rate of packets/second. Traffic intensity $I = La/R$, and the queuing delay is calculated as $I(L/R)(1 - I)$ for $I < 1$.

- In practice, does the queuing delay tend to vary a lot? **Yes**.
- Assuming that $a = 34$, what is the queuing delay? **1.1401 ms**
- Assuming that $a = 90$, what is the queuing delay? **1.588 ms**
- Assuming the router's buffer is infinite, the queuing delay is 1.588 ms, and 860 packets arrive. How many packets will be in the buffer 1 second later? **231 packets**
- If the buffer has a maximum size of 792 packets, how many of the 860 packets would be dropped upon arrival from the previous question? **68**

“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 (with time-to-live field value of *i*)
  - router *i* will return packets to sender
  - sender measures time interval between transmission and reply



Real Internet delays and routes

traceroute: gaia.cs.umass.edu to www.eurecom.fr

3 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

3 delay measurements to border1-rt-fa5-1-0.gw.umass.edu

trans-oceanic link

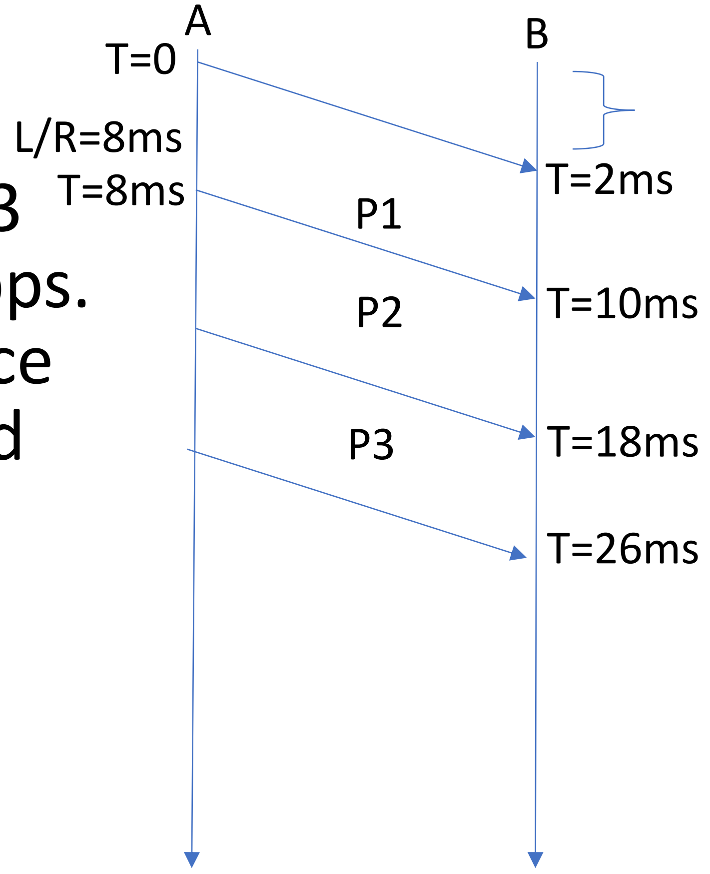
looks like delays *decrease*! Why?

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

* Do some traceroutes from exotic countries at www.traceroute.org

Example

Two directly connected devices need to transmit a 3 Kbytes file over a link with a bandwidth of R = 1 Mbps. Each packet has a size of L = 1 Kbyte and the distance between the devices is 500 km. Assuming the speed of light is 2.5×10^8 m/s, calculate the total time to deliver the file.

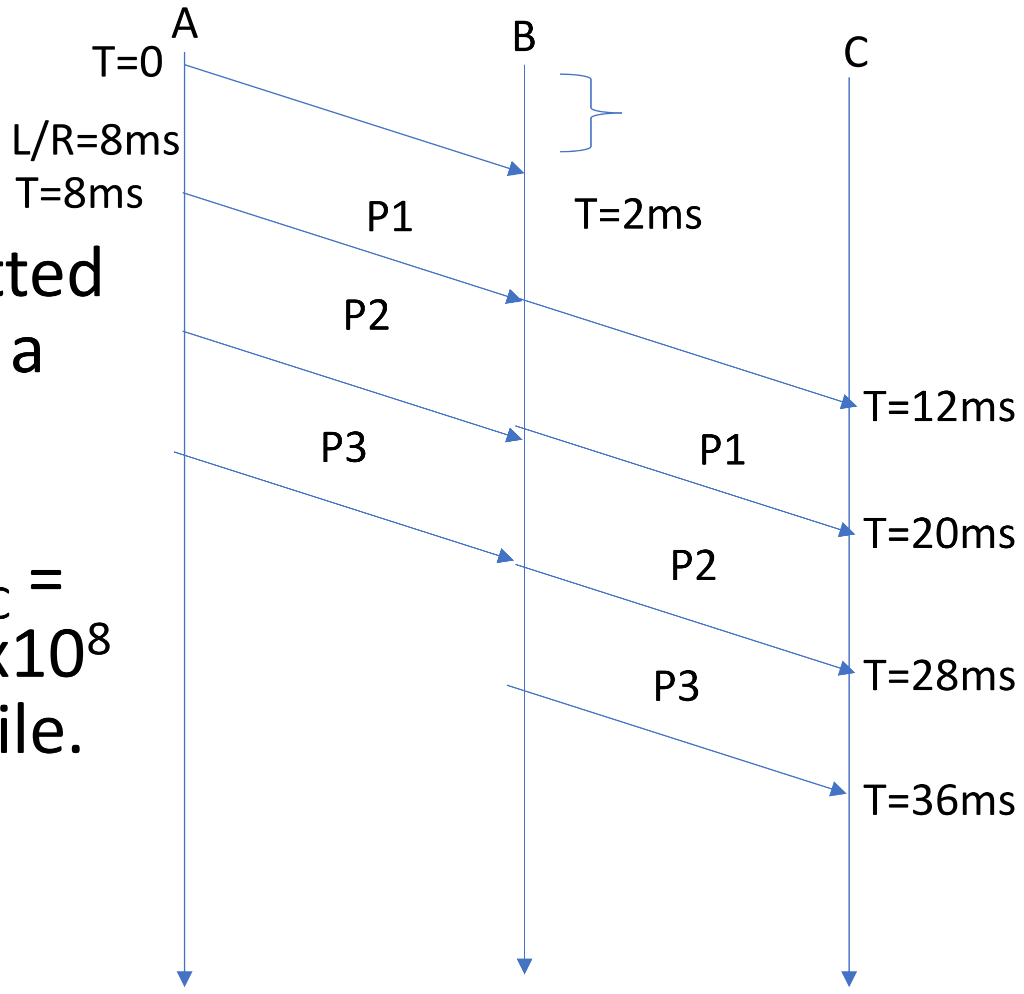


- Number of packets?
  - $N = F/L = 3 * 10^3 * 8 / (1 * 10^3 * 8) = 3$ packets
- Propagation Delay (d_{prop})?
  - $d_{prop} = d/s = 500 * 10^3 / (2.5 * 10^8) = 2$ ms
- Transmission Delay (d_{trans})?
  - $d_{trans} = L/R = 1 * 10^3 * 8 / (1 * 10^6) = 8$ ms
- Total time to deliver the file?
  - End-to-End delay $= d_{prop} + N * d_{trans} = 2 + 3 * 8 = 26$ ms

Example

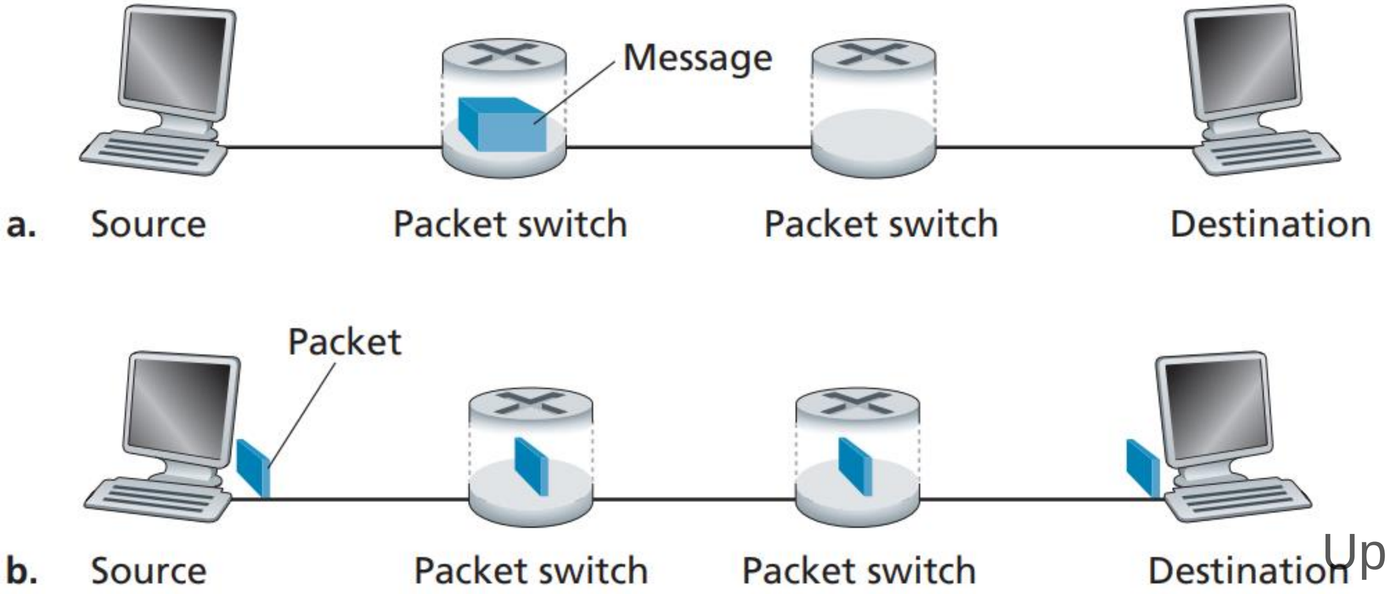
A packet of size L = 1 Kbyte is to be transmitted over a path of 2 links. Each link transmits at a rate of R = 1 Mbps. The total file size to be transmitted is 3 Kbytes file. The distance between the devices is 500 km ($d_{A\rightarrow B} = d_{B\rightarrow C}$ = 500 km). Assuming the speed of light is 2.5×10^8 m/s, calculate the total time to deliver the file.

- Number of packets?
  - $N = F/L = 3 * 10^3 * 8 / (1 * 10^3 * 8) = 3$ packets
- Propagation Delay (d_{prop})?
  - $d_{prop} = d/s = 500 * 10^3 / (2.5 * 10^8) = 2$ ms
- Transmission Delay (d_{trans})?
  - $d_{trans} = L/R = 1 * 10^3 * 8 / (1 * 10^6) = 8$ ms
- Total time to deliver the file?
  - End-to-End delay = $2*(d_{prop}+d_{trans})+(N-1)*d_{trans}$ = 36 ms



Example

P31. In modern packet-switched networks, including the Internet, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as *message segmentation*. Figure 1.27 illustrates the end-to-end transport of a message with and without message segmentation. Consider a message that is 10^6 bits long that is to be sent from source to destination in Figure 1.27. Suppose each link in the figure is 5 Mbps. Ignore propagation, queuing, and processing delays.



Example (Cont.)

a. Consider sending the message from source to destination *without* message segmentation. How long does it take to move the message from the source host to the first packet switch? Keeping in mind that each switch uses store-and-forward packet switching, what is the total time to move the message from source host to destination host?

- Time to send message from source host to first packet switch = d_{trans} = L/R = 10^6 / (5 * 10^6) = 0.2 sec
- With store-and-forward switching, the total time to move message from source host to destination host = d_{trans} * (# of links) = 0.2 * 3 = 0.6 sec

Example (Cont.)

b. Now suppose that the message is segmented into 100 packets, with each packet being 10,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?

- Time to send 1st packet from source host to first packet switch = d_{trans} = L/R = $10^4 / (5 * 10^6)$ = 2 msec
- Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = 2* d_{trans} = 2 * 2 = 4 ms

Example (Cont.)

c. How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part (a) and comment.

- Time at which 1st packet is received at the destination host = d_{trans} * (# of links) = 2 * 3 = 6 ms
- After this, every 2 ms one packet will be received; thus, time at which last (100th) packet is received = 6 + 99 * 2 = 204 ms = 0.204 sec.
- It can be seen that delay in using message segmentation is significantly less (almost 1/3rd).

Example (Cont.)

d. In addition to reducing delay, what are reasons to use message segmentation?

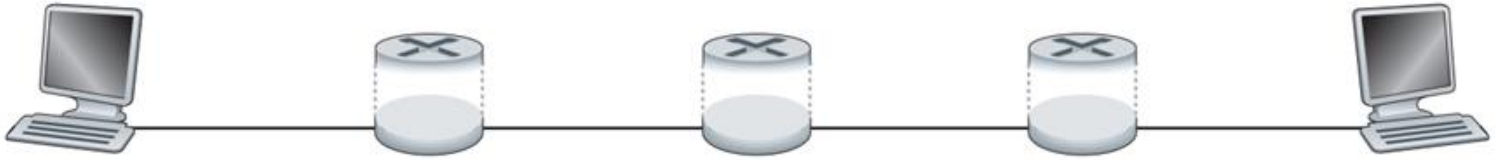
- Without message segmentation, if bit errors are not tolerated, if there is a single bit error, the whole message has to be retransmitted (rather than a single packet).
- Without message segmentation, huge packets (containing HD videos, for example) are sent into the network. Routers have to accommodate these huge packets. Smaller packets have to queue behind enormous packets and suffer unfair delays.

e. Discuss the drawbacks of message segmentation.

- Packets have to be put in sequence at the destination.
- Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

Example

A packet of size ***1*** MB (megabyte), including an overhead of ***200*** KB (kilobytes), is to be transmitted over a path of ***4*** links. Each link transmit at a rate of ***R = 8*** Gbps (gigabits per second). The total file size to be transmitted is ***1*** GB (gigabytes). Assume that there are no queueing delays, the processing delay is negligible, and the propagation delay for each link is 5 milliseconds. Calculate the total time to deliver the file.



- Number of links (X) = 4
- Data per packet = L - overhead = $1 * 10^6 - 200 * 10^3$ = 800000 bytes
- Number of packets (N) = (file size)/(data per packet) = 10^9 / (800000) = 1250 packets
- Propagation Delay (d_{prop}) = 5 ms
- Transmission Delay (d_{trans}) = L/R = $1 * 10^6 * 8 / (8 * 10^9)$ = 1 ms
- End-to-End delay = X * ($d_{prop} + d_{trans}$) + (N - 1) * d_{trans} = 4 * (5 + 1) + (1250 - 1) * 1 = 4 * 6 + 1249 = 24 + 1249 = 1273 ms = 1.273 seconds

Example

Consider sending a file of 6 Mbits over a path of X links. Each link transmits at a rate of 4 Mbps. Suppose that a packet-switched datagram network is used, and it segments the file into 450 packets with each packet having a 500 bits header. Assume that there are no queuing delays, the processing delay is negligible, and each of the X links has a propagation delay of 7 milliseconds. Calculate the maximum number of links, X, that is along the path such that the total time to deliver the file is less than or equal to 2.107 seconds.

$t_{trans.} = (6 \text{ Mbits}/450 + 500 \text{ bits})/4 \text{ Mbps} = 0.00346 \text{ seconds}$

$$\text{Total time} = 2.107 \text{ s} \geq X \times (t_{trans.} + t_{prop.}) \text{ (1st packet)} + 449 \text{ packets} \times t_{trans.}$$
$$\geq (X + 449) \times 0.00346 + X \times 0.007$$

→ $X \leq (2.107 - 1.55354)/0.01046 = 52.9$

→ Maximum number of links X = 52 links

Example

Consider sending a large file of ***F*** bits from Host A to Host B. There are ***two*** links (and one switch) between A and B, and the links are uncongested (that is, no queuing delays). Host A segments the file into segments of ***S*** bits each and adds ***40*** bits of header to each segment, forming packets of ***L = 40 + S*** bits. Each link has a transmission rate of ***R*** bps. Find the value of ***S*** that minimizes the delay of moving the file from Host A to Host B. Disregard propagation delay.

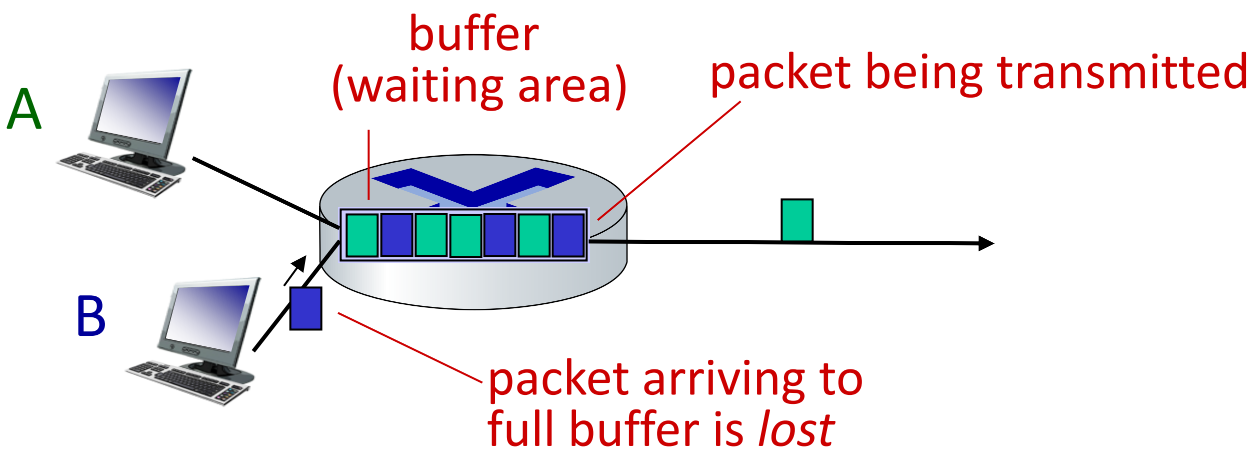
Time at which the 1^{st} packet is received at the destination $= \frac{S+40}{R} \times 2$ seconds. After this, one packet is received at destination every $\frac{S+40}{R}$ sec.

$\Rightarrow$ Delay in sending the whole file $= delay = \frac{S+40}{R} \times 2 + (\frac{F}{S} - 1) \times (\frac{S+40}{R}) = \frac{S+40}{R} \times (\frac{F}{S} + 1)$

$\Rightarrow$ To calculate the value of S which leads to the minimum delay, $\frac{d}{dS} delay = 0 \Rightarrow S = \sqrt{40F}$

Packet loss

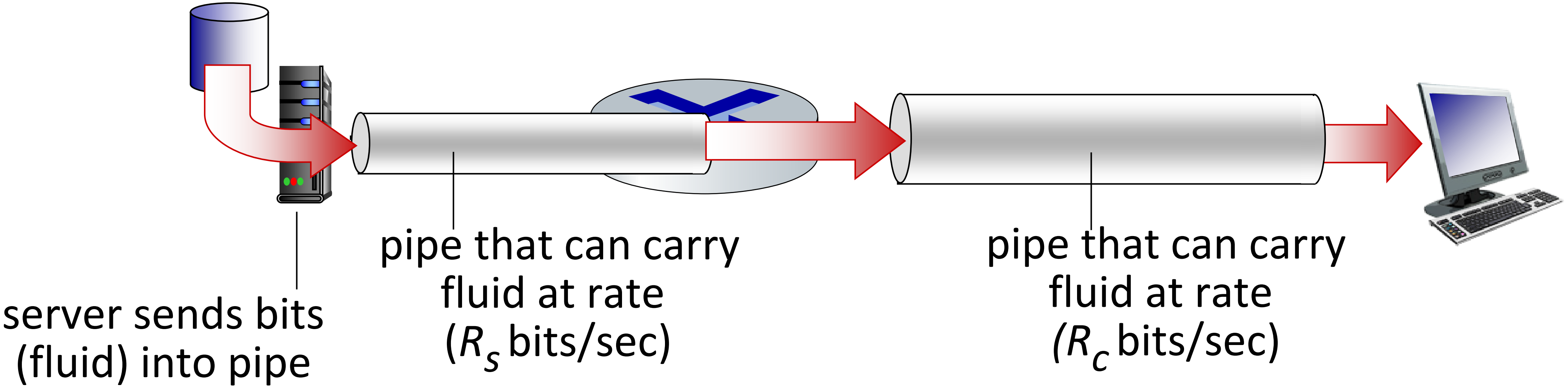
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



* Check out the Java applet for an interactive animation on queuing and loss

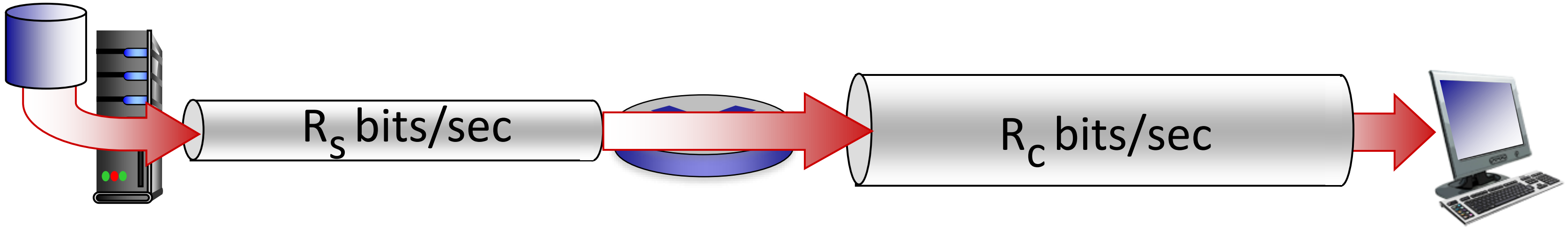
Throughput

- *throughput:* rate (bits/time unit) at which bits are being sent from sender to receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time

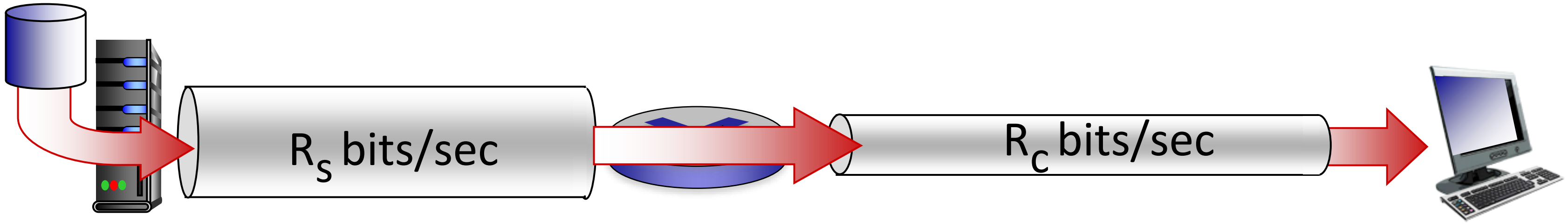


Throughput

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



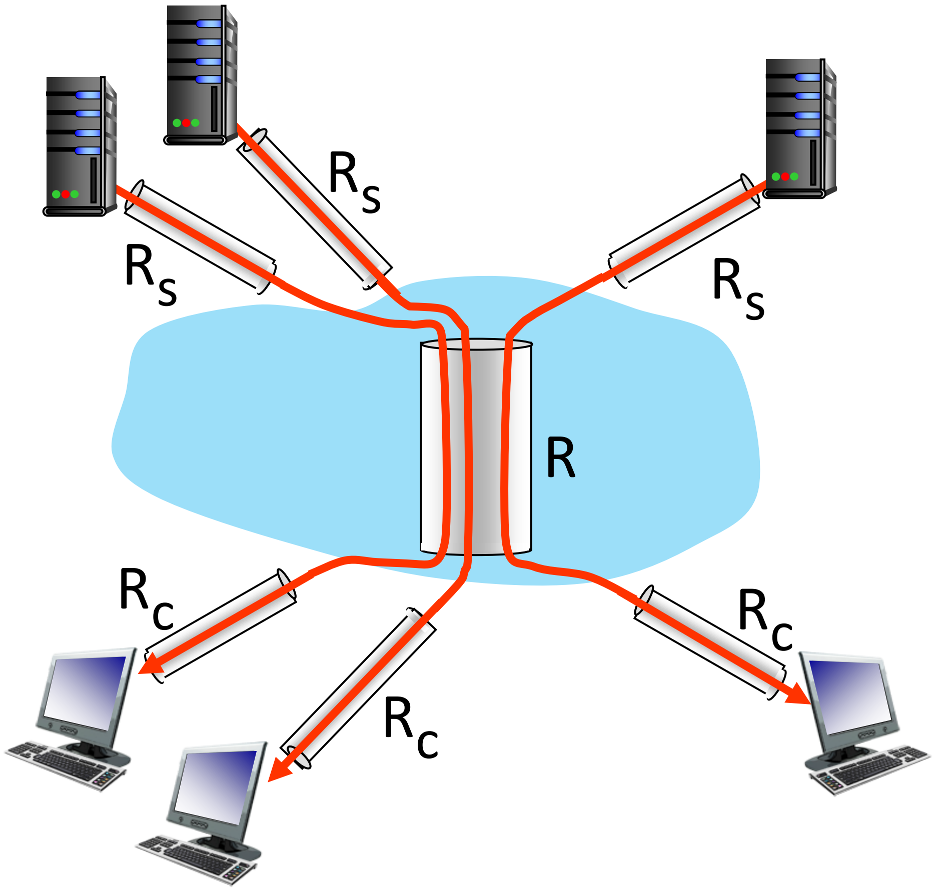
$R_s > R_c$ What is average end-end throughput?



bottleneck link

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

Throughput: network scenario

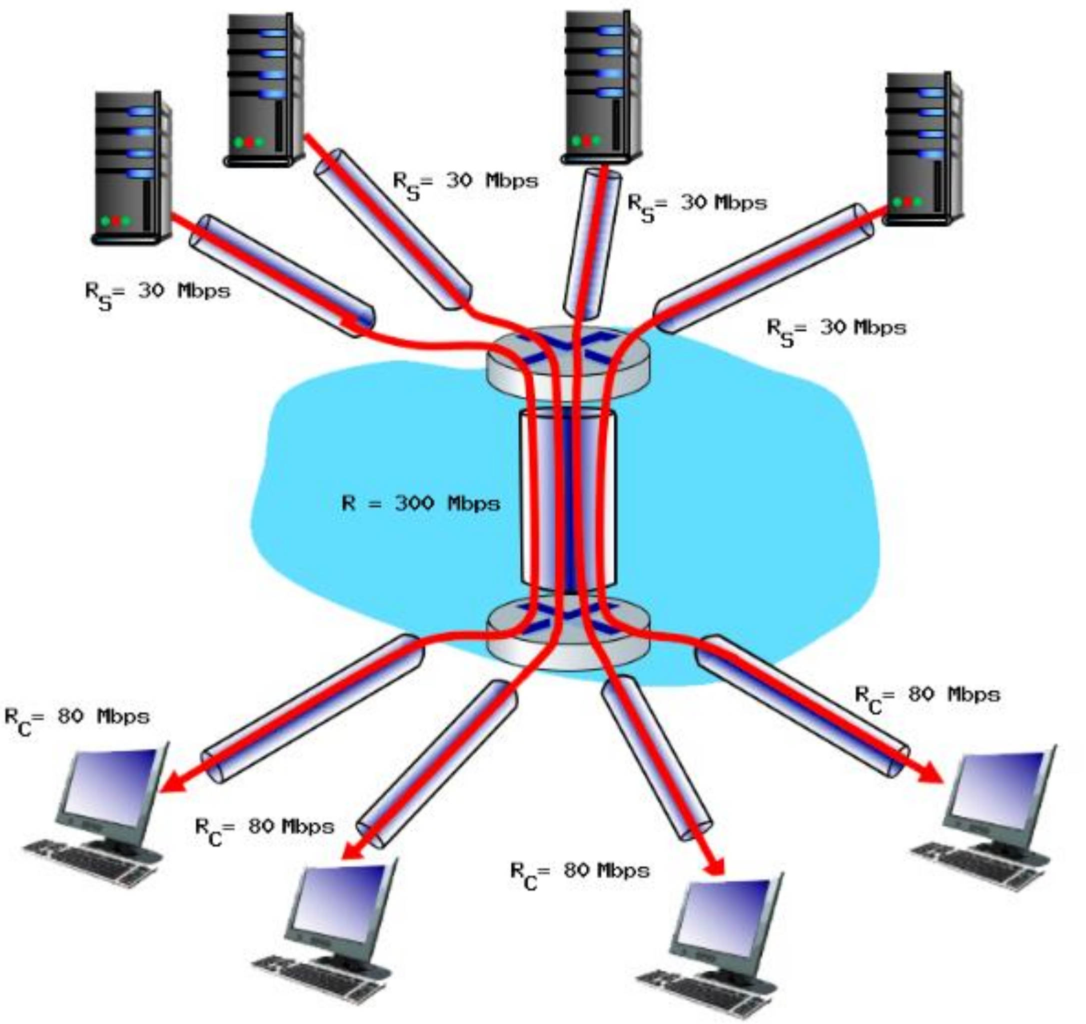


10 connections (fairly) share backbone bottleneck link *R* bits/sec

- per-connection end-end throughput: $min(R_c, R_s, R/10)$
- in practice: R_c or R_s is often bottleneck

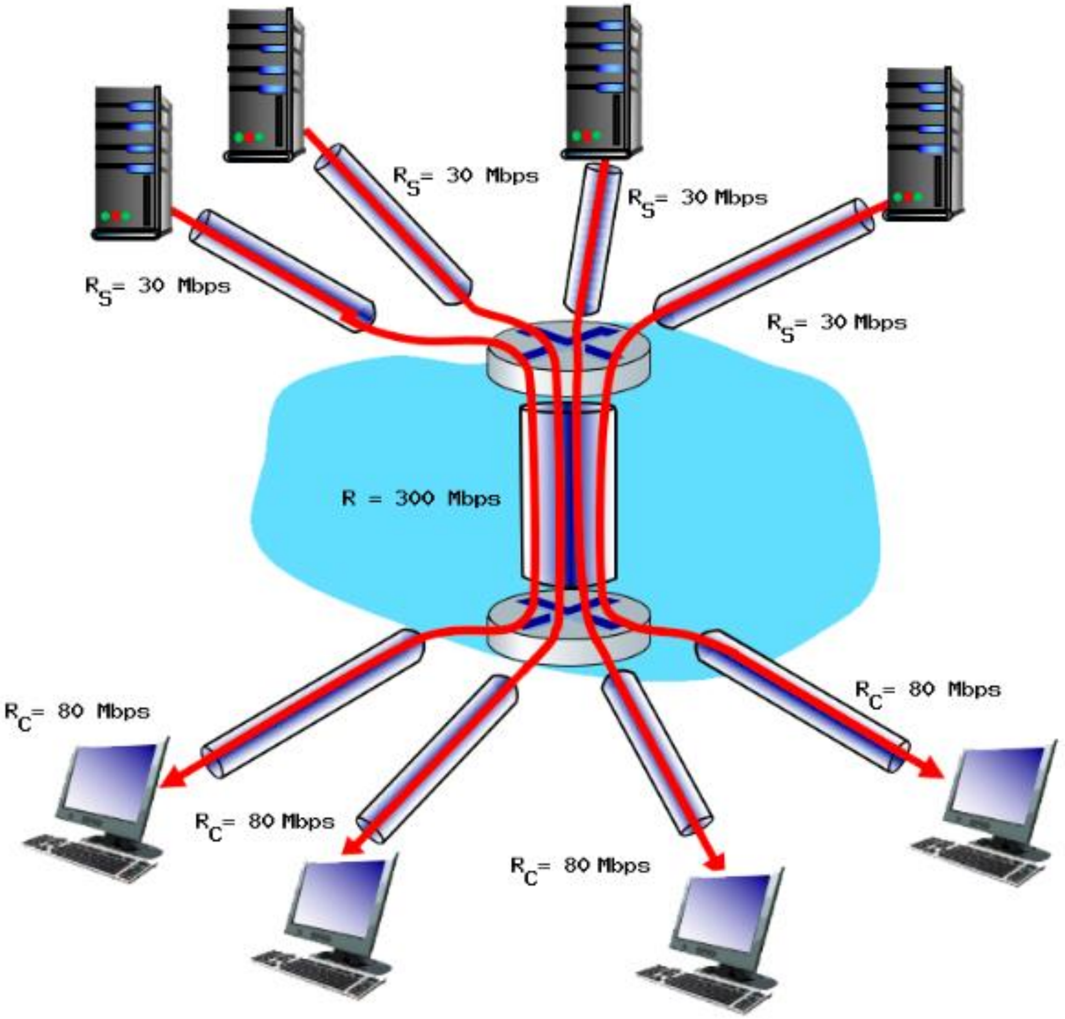
- Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/

Example:



- Consider this scenario, with four different servers connected to four different clients over four three-hop paths.
- The four pairs share a common middle hop with a transmission capacity of R = 300 Mbps.
- The four links from the servers to the shared link have a transmission capacity of R_S = 30 Mbps.
- Each of the four links from the shared middle link to a client has a transmission capacity of R_C = 80 Mbps.

Example:



- What is the maximum achievable end-end throughput (in Mbps) for each of four client-to-server pairs, assuming that the middle link is fairly shared (divides its transmission rate equally)? **30Mbps**
- Which link is the bottleneck link (R_c, R_S, or R)? **R_s**
- Assuming that the servers are sending at the maximum rate possible, what are the link utilizations for the server links (R_S)? **1**
- Assuming that the servers are sending at the maximum rate possible, what are the link utilizations for the client links (R_c)? **0.38**
- Assuming that the servers are sending at the maximum rate possible, what is the link utilizations for the shared link (R)? **0.4**

Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

Network security

- Internet not originally designed with (much) 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!
- We now need to think about:
  - How bad guys can attack computer networks
  - How we can defend networks against attacks
  - How to design architectures that are immune to attacks

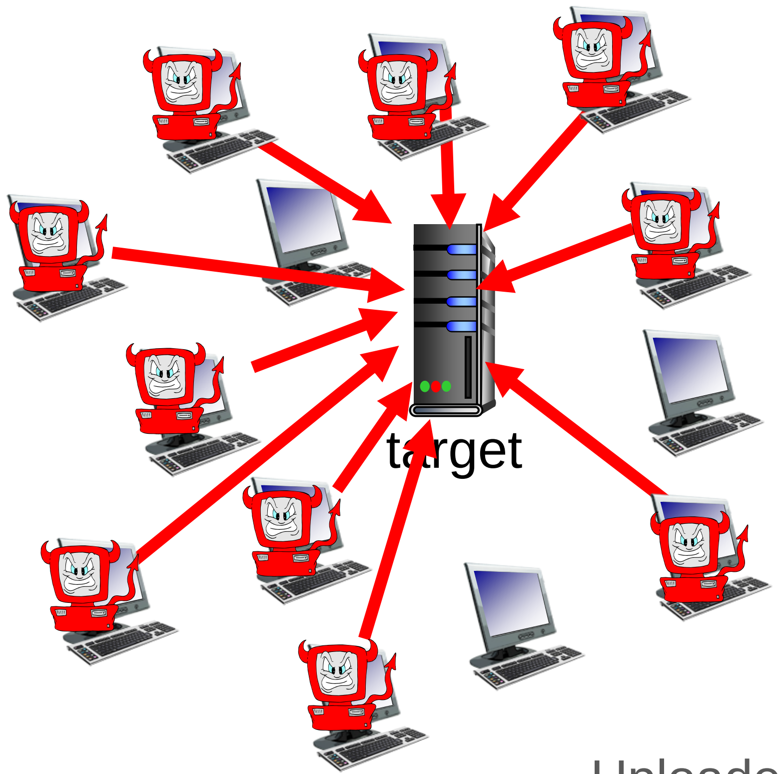
Bad guys: malware

- malware can get in host from:
  - *virus:* self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  - *worm:* self-replicating infection by passively receiving object that gets itself executed
- spyware malware can record keystrokes, web sites visited, upload info to collection site
- infected host can be enrolled in botnet, used for spam or distributed denial of service (DDoS) attacks

Bad guys: denial of service

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

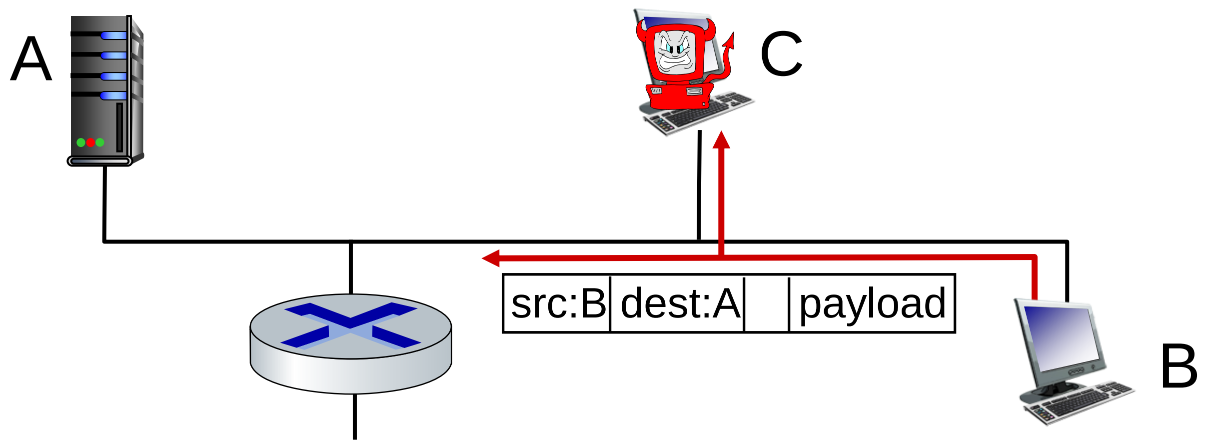
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Bad guys: packet interception

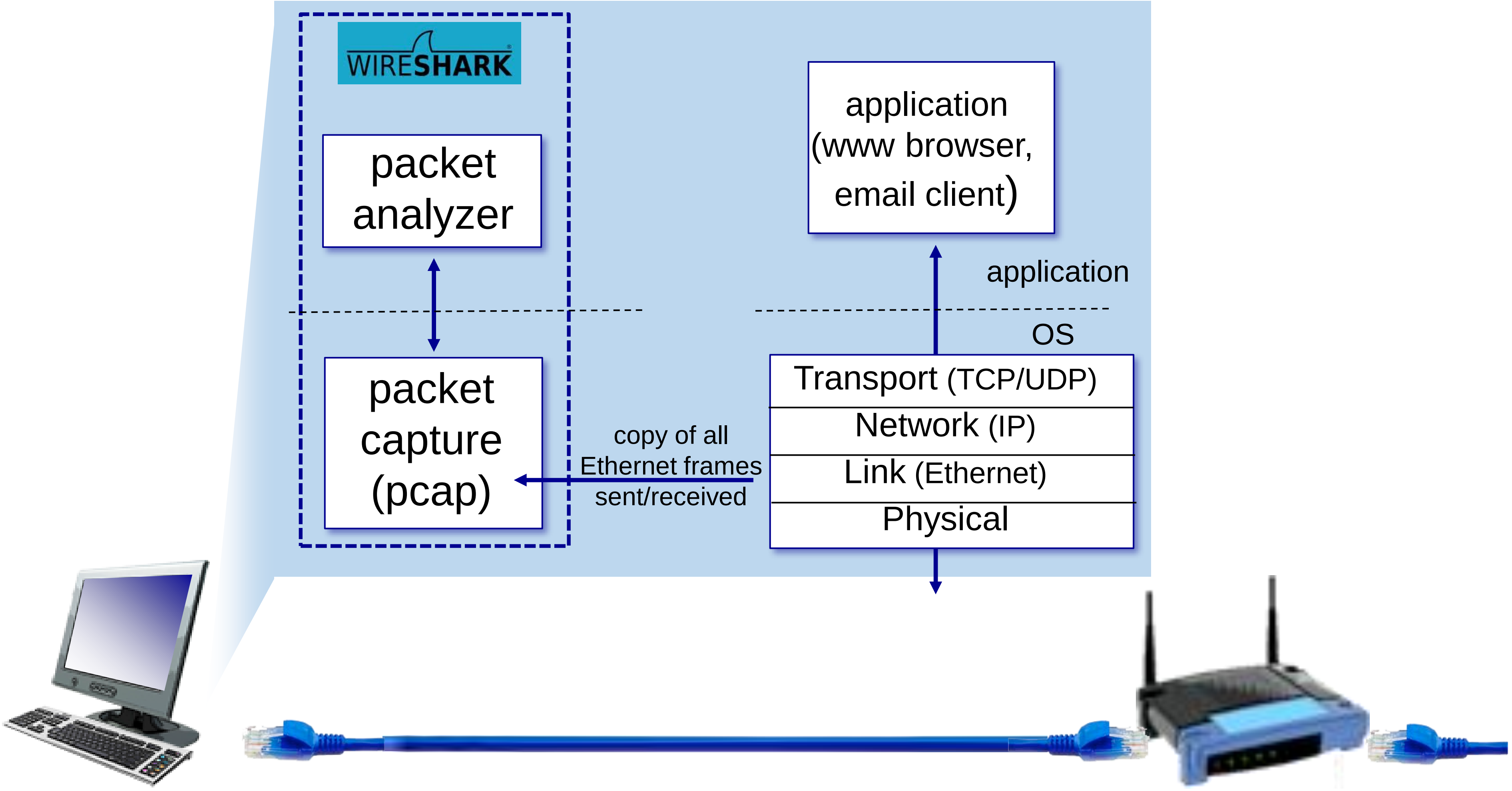
packet "sniffing":

- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



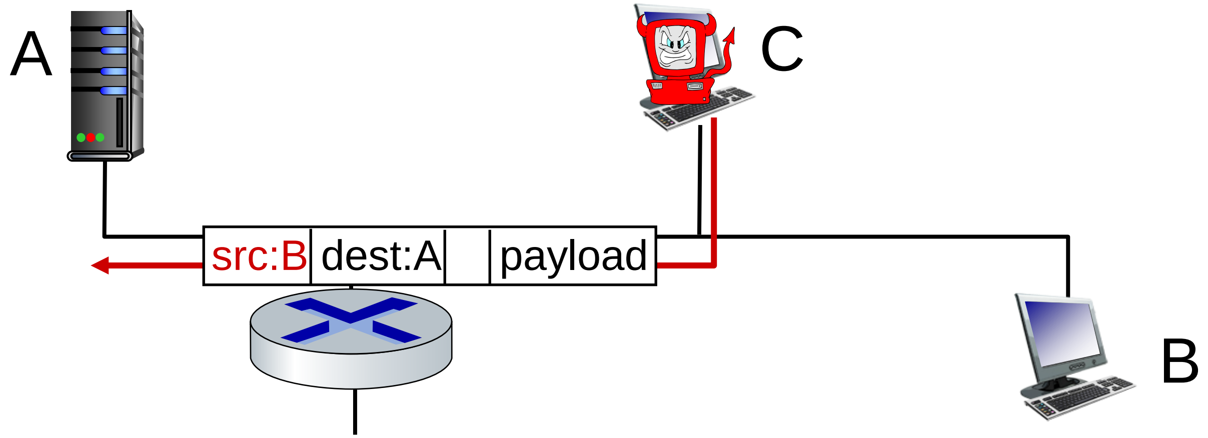
Wireshark software used for our end-of-chapter labs is a (free) packet-sniffer

Wireshark


WIRESHARK
packet analyzer
packet capture (pcap)
copy of all Ethernet frames sent/received
application (www browser, email client)
application
OS
Transport (TCP/UDP)
Network (IP)
Link (Ethernet)
Physical

Bad guys: fake identity

IP spoofing: send packet with false source address



… lots more on security (throughout, Chapter 8)

Chapter 1: roadmap

- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- **Protocol layers, service models**
- History

Protocol “layers” and reference models

Networks are complex, with many “pieces”:

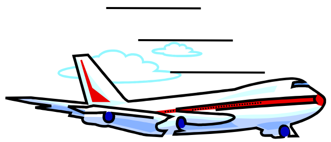
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

is there any hope of *organizing* structure of network?

.... or at least our *discussion* of networks?

Example: organization of air travel



ticket (purchase)	ticket (complain)
baggage (check)	baggage (claim)
gates (load)	gates (unload)
runway takeoff	runway landing
airplane routing	airplane routing

airplane routing

airline travel: a series of steps, involving many services

Example: organization of air travel

layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

Q: describe in words the service provided in each layer above

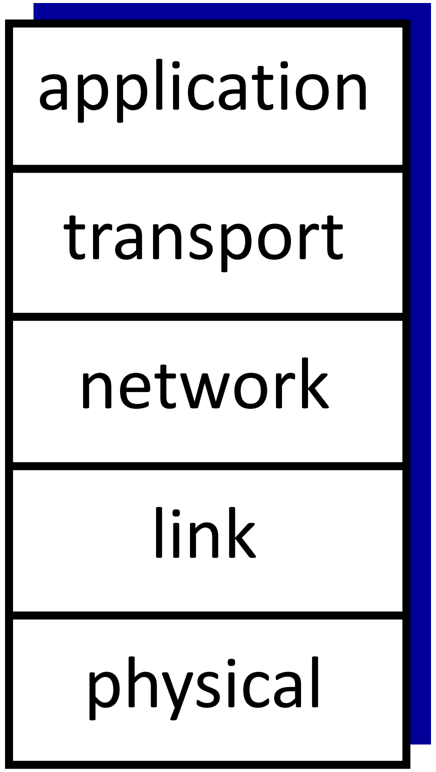
Why layering?

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 in layer's service *implementation*: transparent to rest of system
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?
- layering in other complex systems?

Internet protocol stack

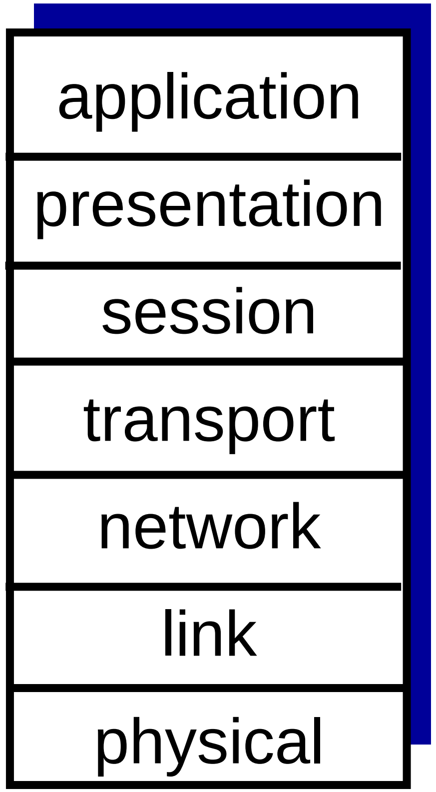
- *application:* supporting network applications
  - HTTP, DNS, SMTP, IMAP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP
- *physical:* bits "on the wire"



ISO/OSI reference model

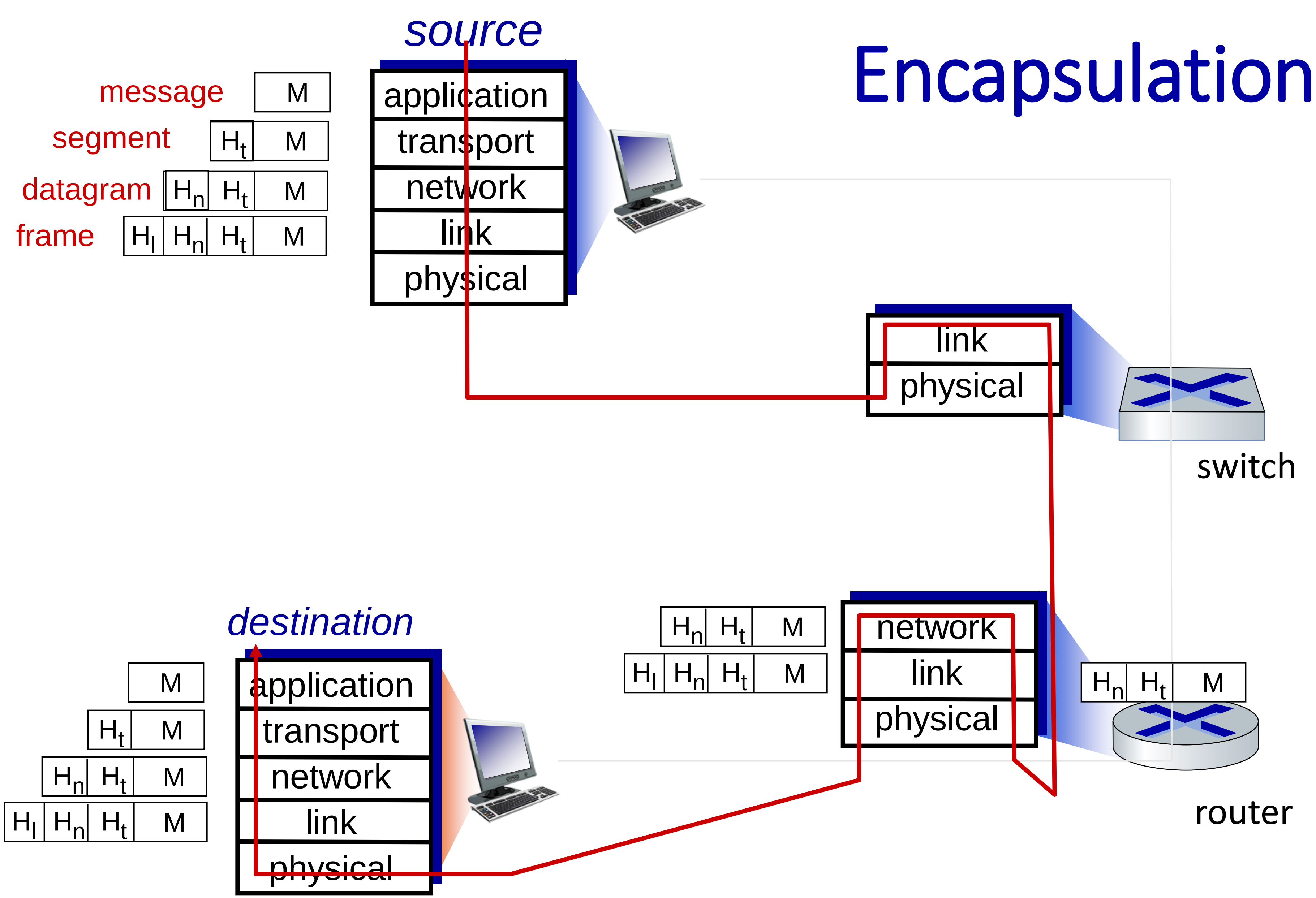
Two layers not found in Internet protocol stack!

- *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?



The seven layer OSI/ISO reference model

Encapsulation



Chapter 1: roadmap

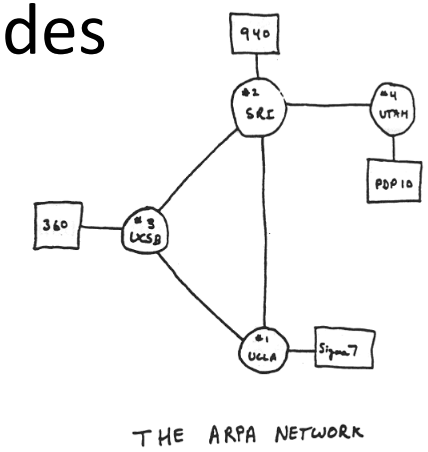
- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

Internet history

1961-1972: Early packet-switching principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational

- 1972:
  - ARPAnet public demo
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



Internet history

1972-1980: Internetworking, new and proprietary nets

- 1970: ALOHAnet satellite network in Hawaii
- 1974: Cerf and Kahn - architecture for interconnecting networks
- 1976: Ethernet at Xerox PARC
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best-effort service model
- stateless routing
- decentralized control

define today's Internet architecture

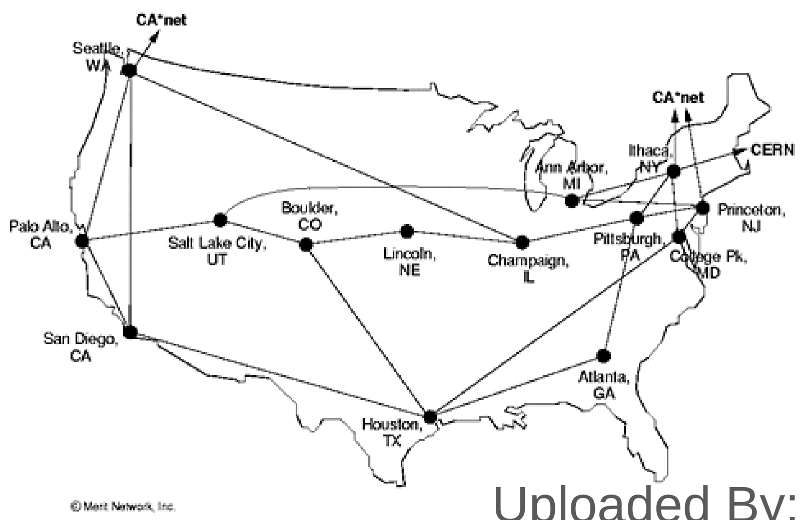
Internet history

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: smtp e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: ftp protocol defined
- 1988: TCP congestion control

- new national networks: CSnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

NSFNET T1 Network 1991



Internet history

1990, 2000s: commercialization, the Web, new applications

- early 1990s: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990s: commercialization of the Web

late 1990s – 2000s:

- more killer apps: instant messaging, P2P file sharing
- network security to forefront
- est. 50 million host, 100 million+ users
- backbone links running at Gbps

Internet history

2005-present: more new applications, Internet is "everywhere"

- ~18B devices attached to Internet (2017)
  - rise of smartphones (iPhone: 2007)
- aggressive deployment of broadband access
- increasing ubiquity of high-speed wireless access: 4G/5G, WiFi
- emergence of online social networks:
- Facebook: ~ 2.5 billion users
- service providers (Google, FB, Microsoft) create their own networks
  - bypass commercial Internet to connect "close" to end user, providing "instantaneous" access to search, video content, …
- enterprises run their services in "cloud" (e.g., Amazon Web Services, Microsoft Azure)

Chapter 1: Summary

We've covered a "ton" of material!

- Internet overview
- What's a protocol?
- network edge, access network, core
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models
- security
- history

You now have:

- context, overview, vocabulary, "feel" of networking
- more depth, detail, *and fun* to follow*!*