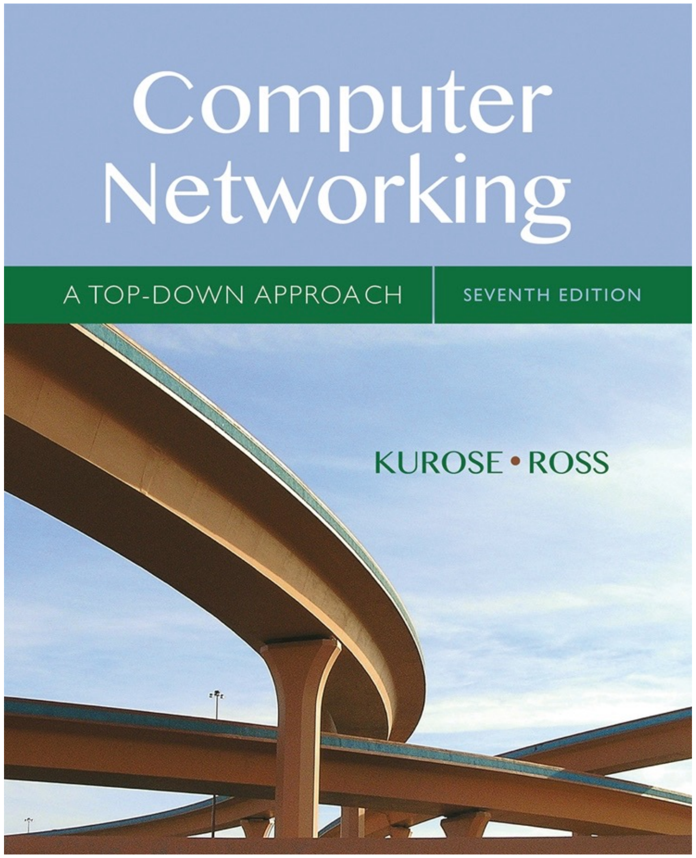
Chapter 1
Introduction



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

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.

Thanks and enjoy! JFK/KWR

Administrivia

- All homework released on Blackboard
  - Due on Sep 3, 11:59 pm
  - No late submissions are accepted
- Quiz 1, Sep 4 in class
  - Pen and paper, multiple choice questions
  - Materials to cover: Overview and Chapter 1
  - Calculator is allowed, but no smart device

Chapter 1: Introduction

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

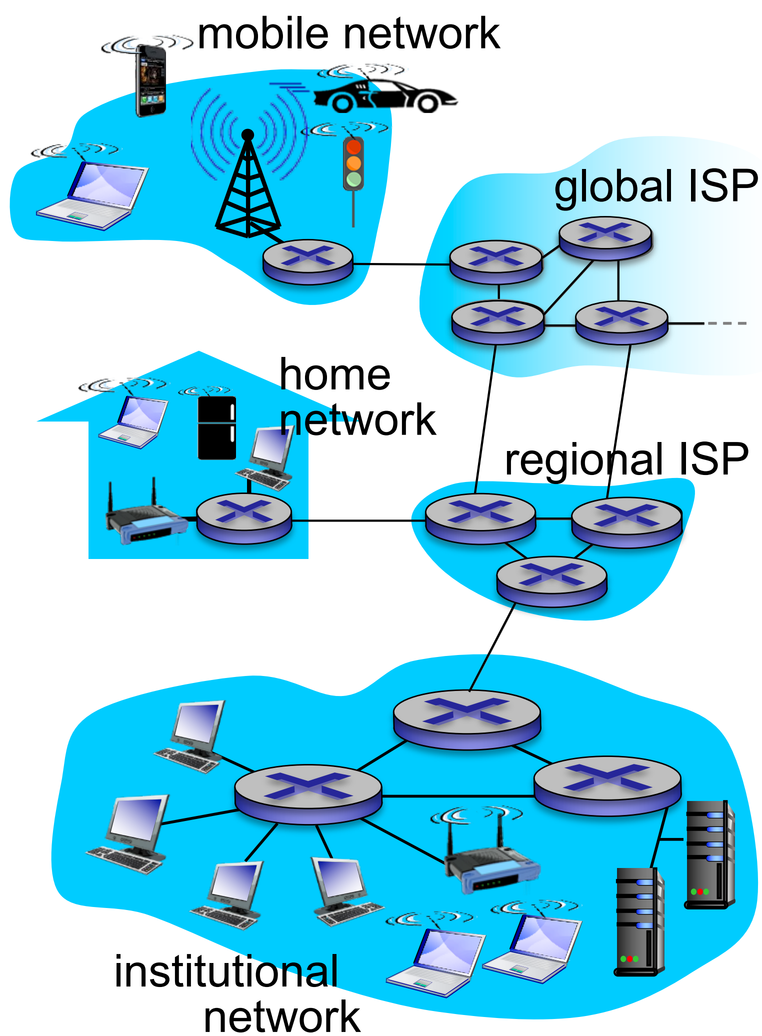
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

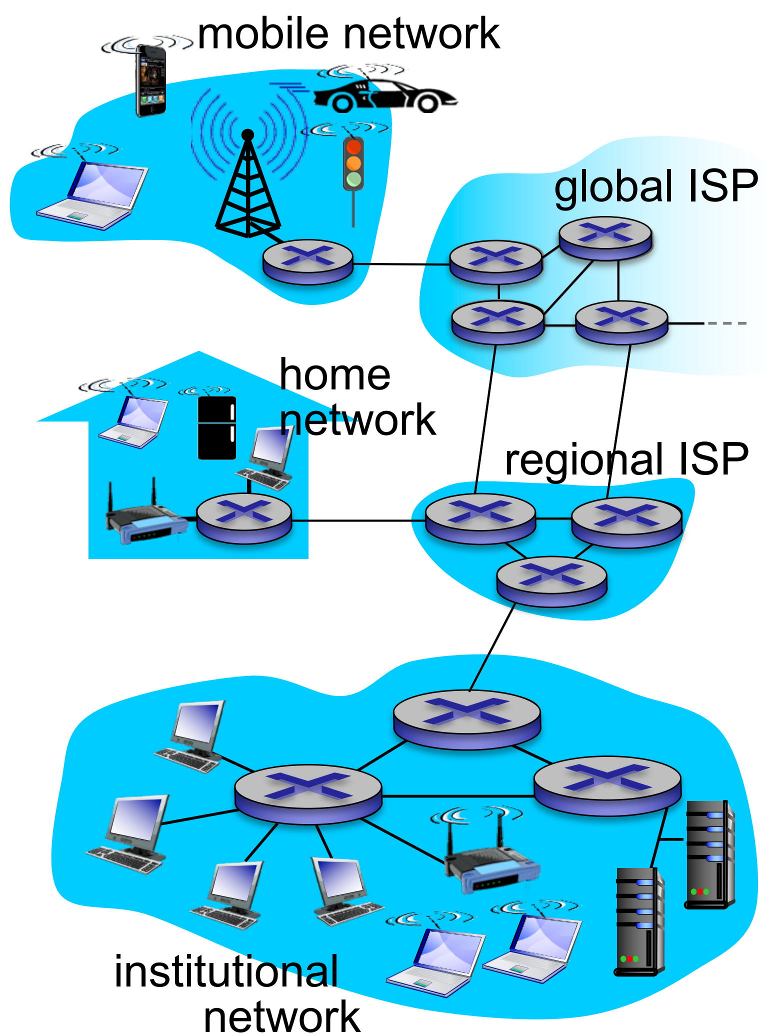
What's the Internet: "nuts and bolts" view

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



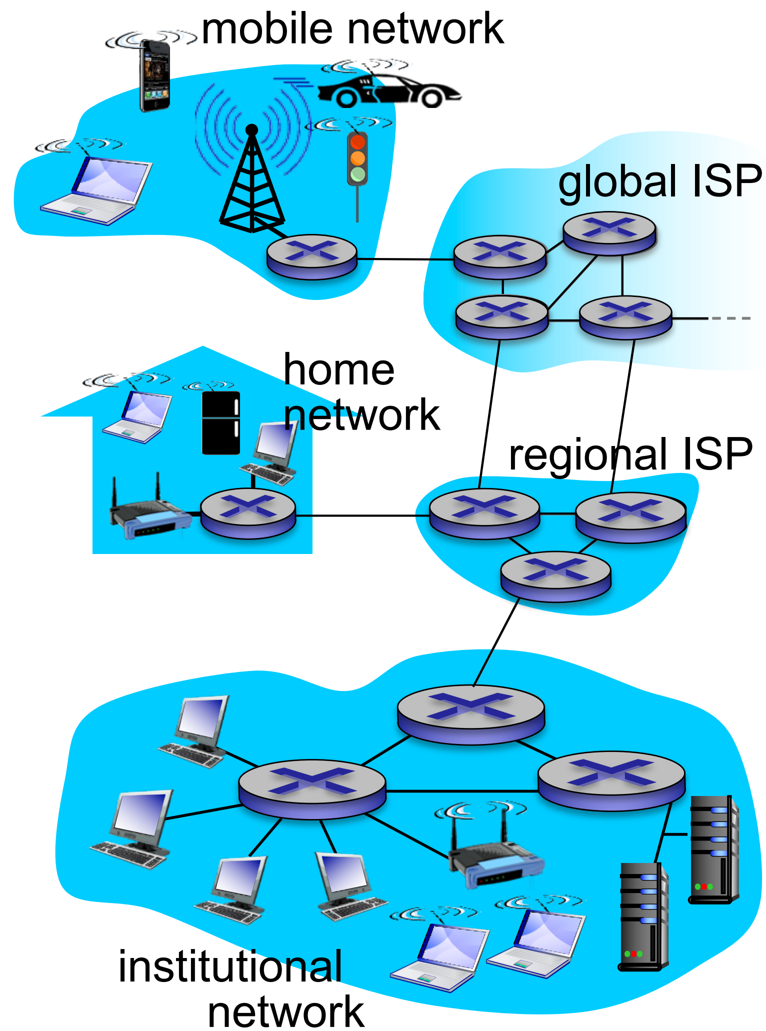
What's the Internet: "nuts and bolts" view

- *Internet:* "network of networks"
  - Interconnected ISPs
- *protocols* control sending and receiving of messages
  - e.g., TCP, IP, HTTP, Skype, 802.11
- *Internet standards*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



What's the Internet: a service view

- *infrastructure that provides services to applications:*
  - Web, VoIP, email, games, e-commerce, social nets, …
- *provides programming interfaces to applications*
  - hooks that allow sending and receiving app programs to "connect" to the Internet
  - 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 messages are received, or other events

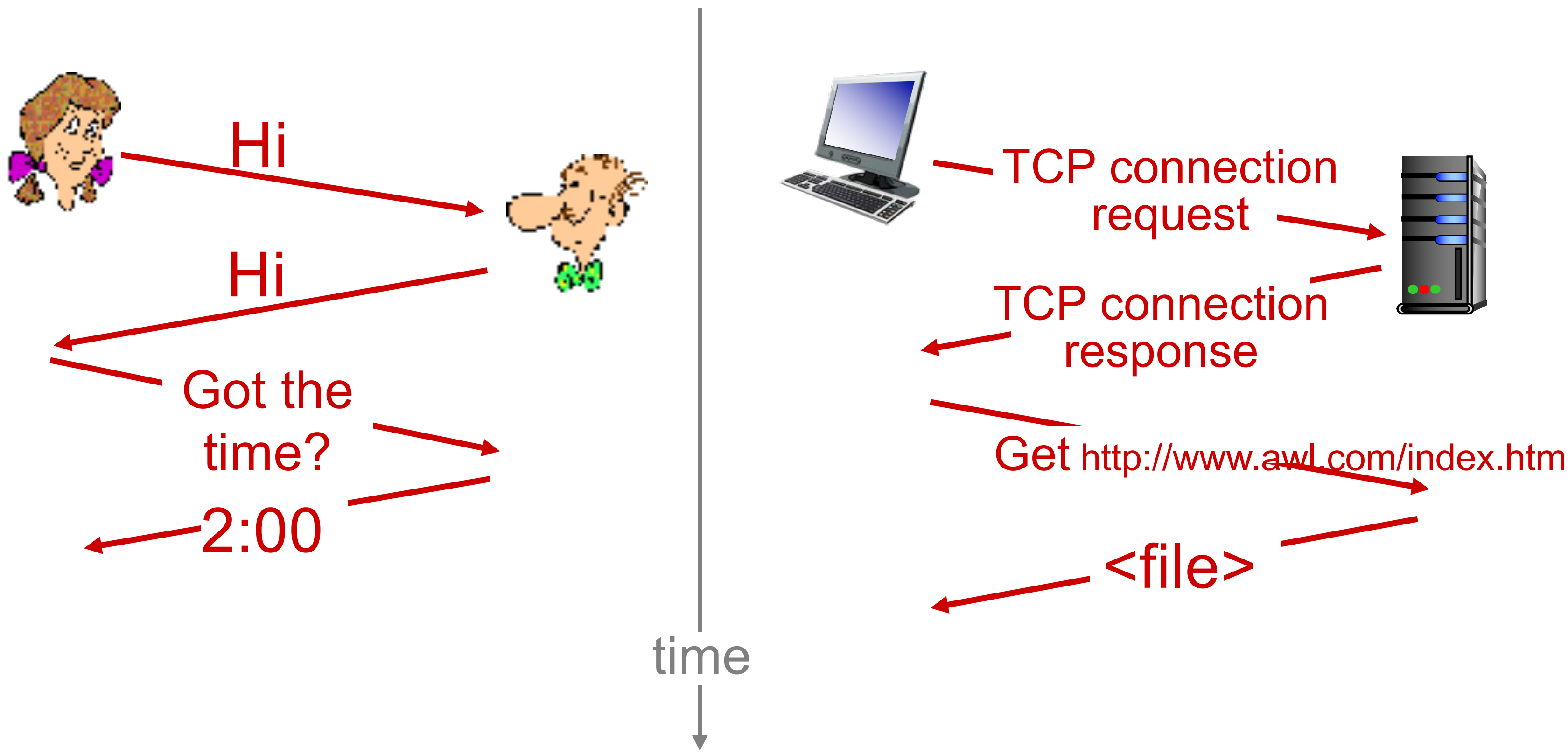
network protocols:

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

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

What's a protocol?

a human protocol and a computer network protocol:



Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

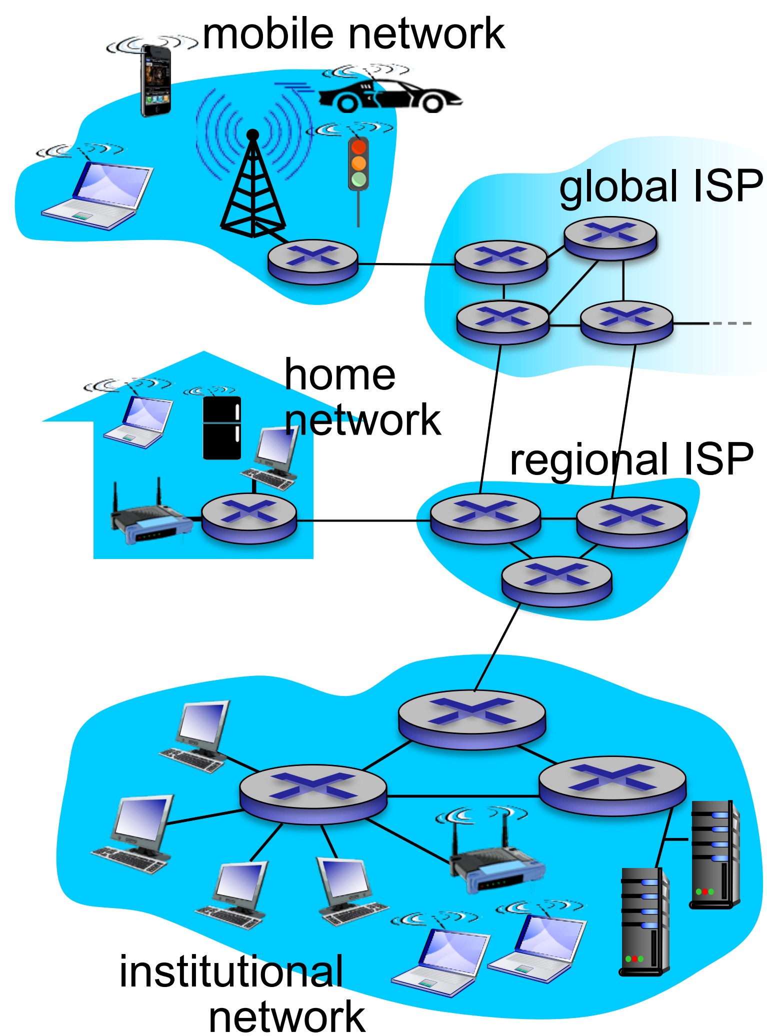
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

A closer look at network 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



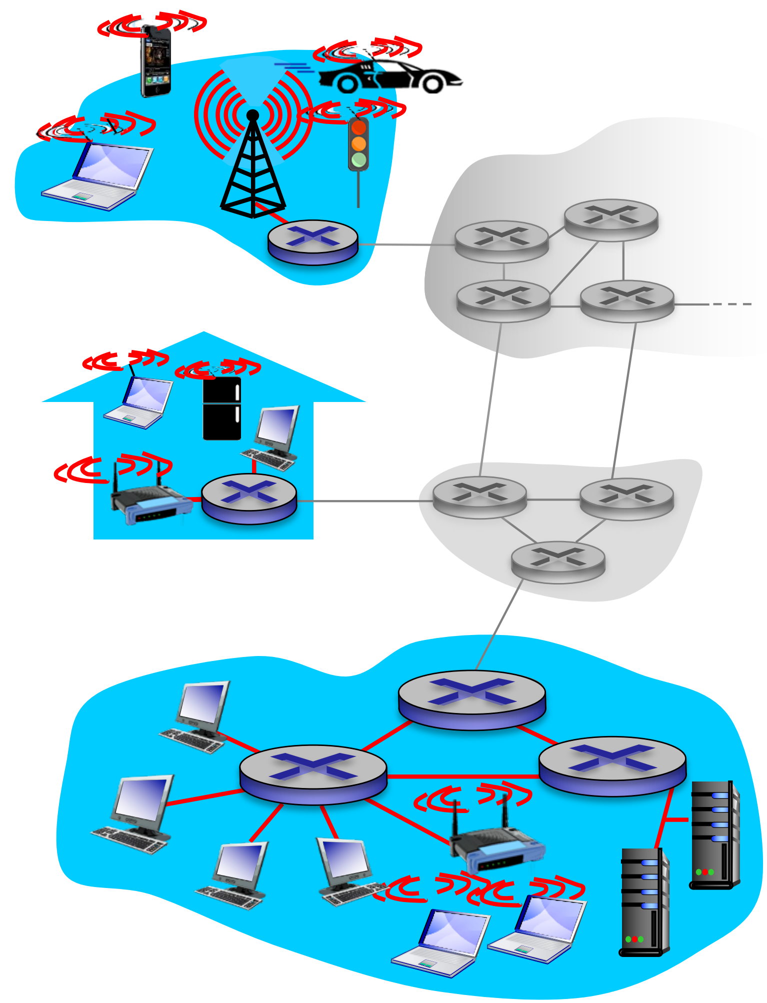
Access network: physical media

Q: How to connect end systems to edge router?

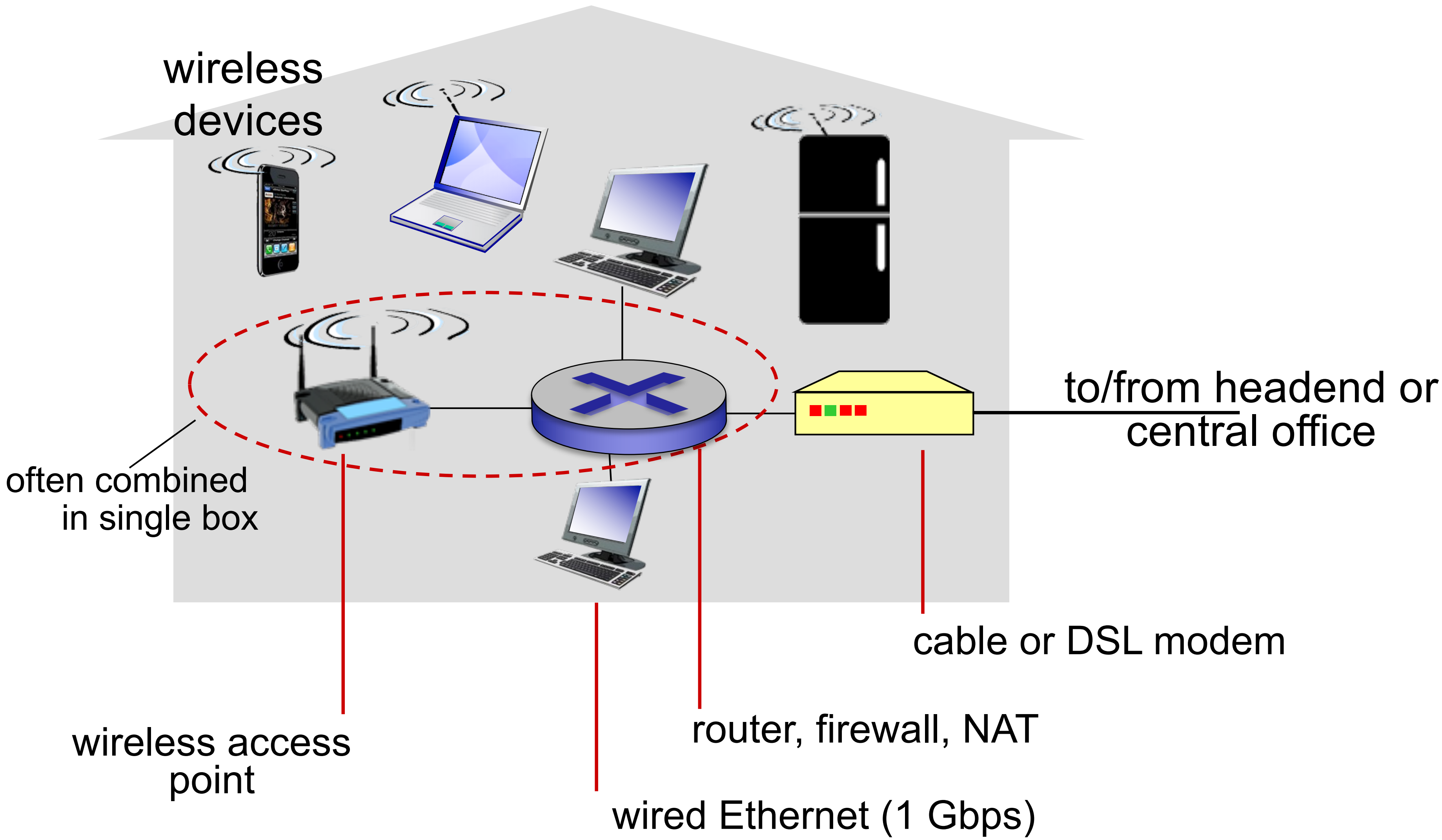
- residential access nets
- institutional access networks (school, company)
- mobile access networks

keep in mind:

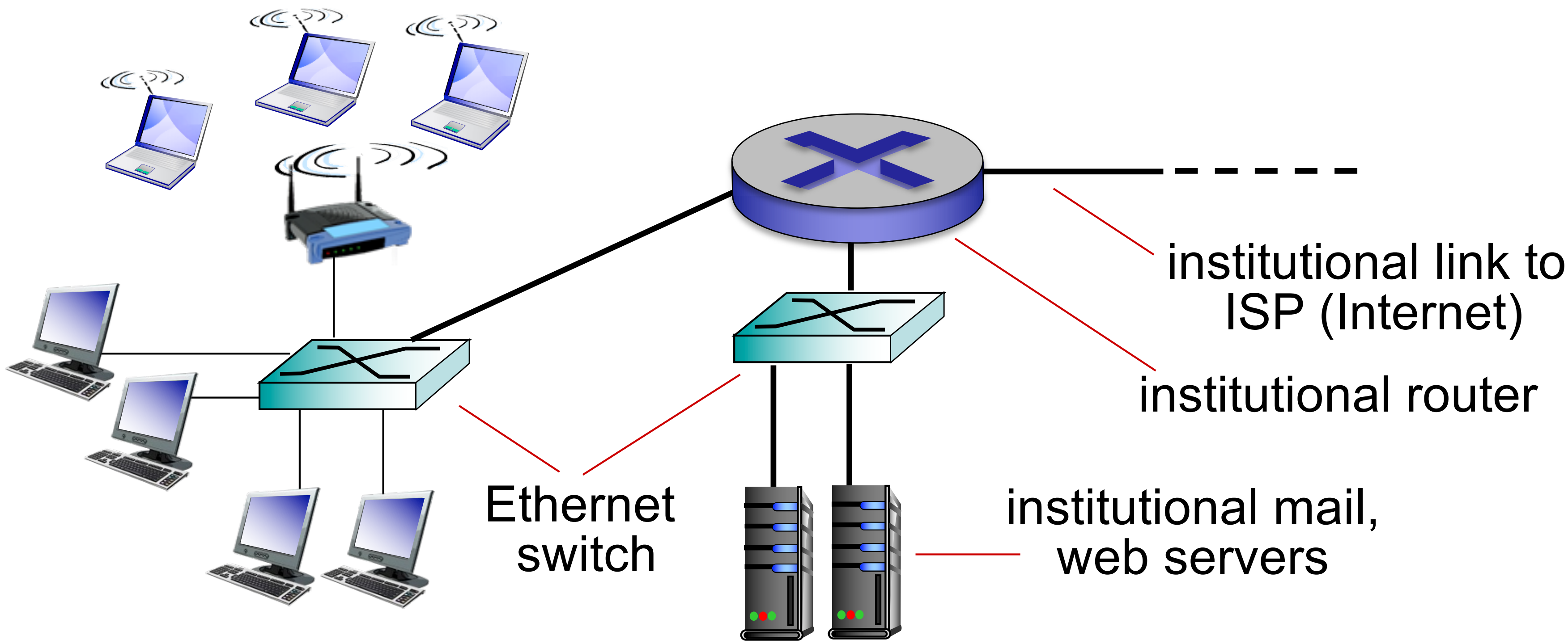
- bandwidth (bits per second) of access network?
- shared or dedicated?



Home access networks



Enterprise access networks (Ethernet)



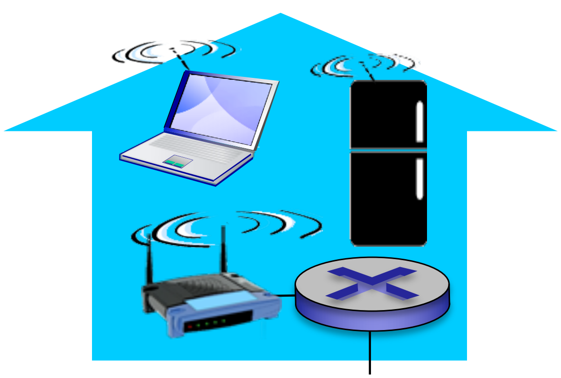
- Typically used in companies, universities, etc.
- 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- Today, end systems typically connect to Ethernet switches

Wireless access networks

- shared *wireless* access network connects end system to routers
  - via base station, aka "access point"

wireless LANs:

- within building (100 ft.)
- 802.11b/g/n/ac/ax (WiFi): 11, 54, 450 Mbps, 600 Mbps,10 Gbps transmission rate



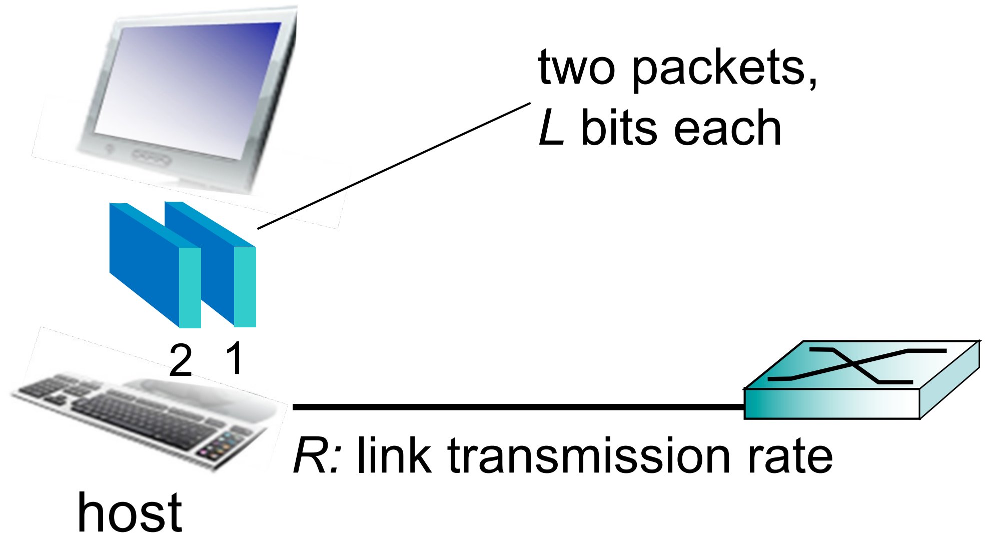
wide-area wireless access

- provided by telco (cellular) operator, 10's km
- 3G, 4G: LTE, 5G

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*



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

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

radio link types:

- terrestrial microwave
  - e.g., up to 45 Mbps channels
- local-area (e.g., WiFi)
  - 54 Mbps
- wide-area (e.g., cellular)
  - 4G cellular: ~ 10 Mbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

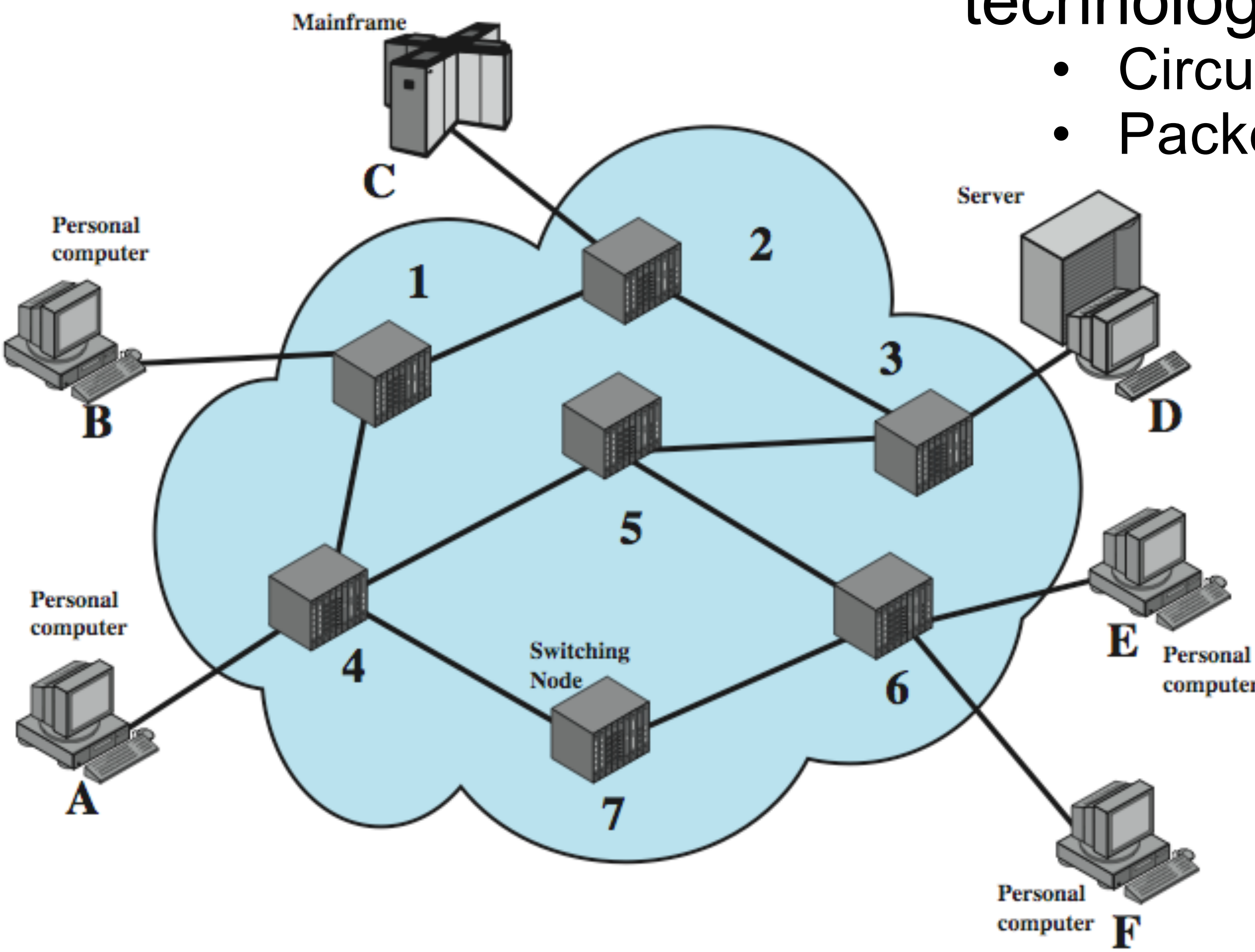
1.5 protocol layers, service models

1.6 networks under attack: security

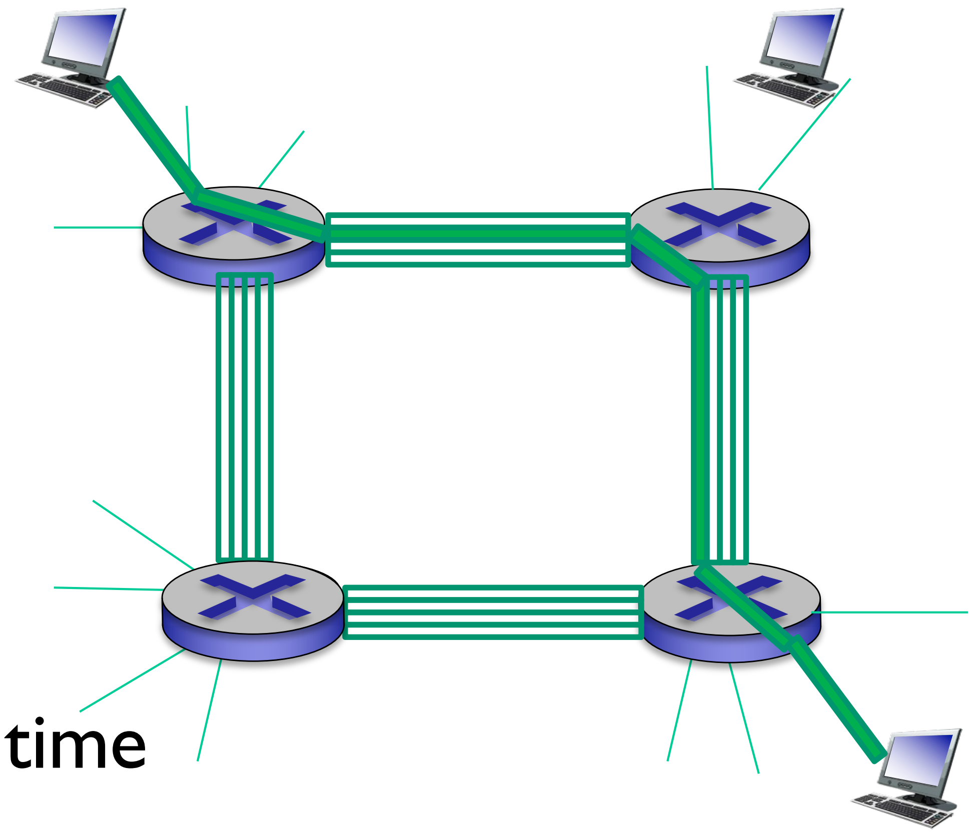
Switched Network

Two different switching technologies

- Circuit switching
- Packet switching



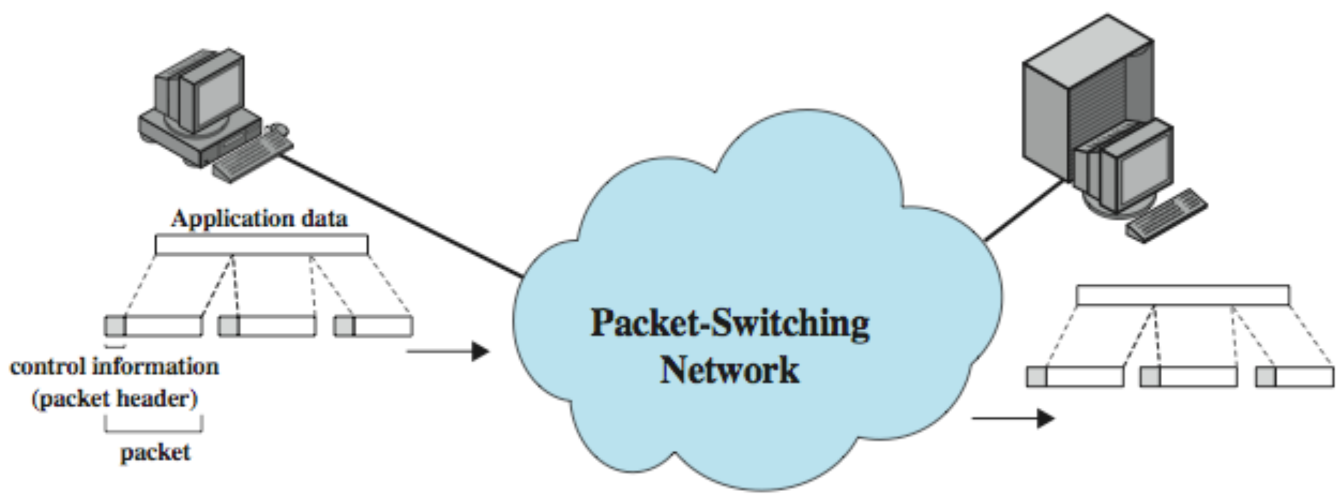
Circuit Switching



- uses a dedicated path between two hosts
- has three phases
  - establish
  - transfer
  - disconnect
- set up (connection) takes time
- inefficient
  - channel capacity dedicated for duration of connection
  - if no data, capacity wasted *(no sharing)*
- once connected, transfer is transparent
- commonly used in traditional telephone networks

Packet Switching

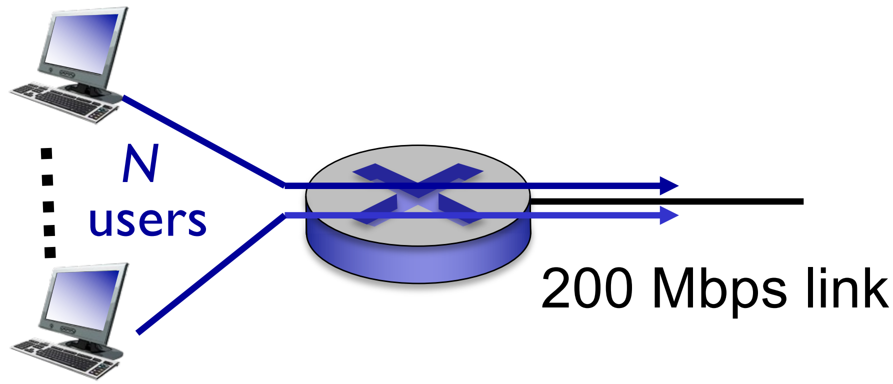
- circuit switching was designed for voice
- packet switching was designed for data
- hosts break application-layer messages into packets
- packets contain user data and control info
  - user data may be part of a larger message
  - control info includes routing (addressing) info
- packets are received, stored briefly (buffered), and transmitted to the next node



Packet switching versus circuit switching

packet switching allows more users to use network!

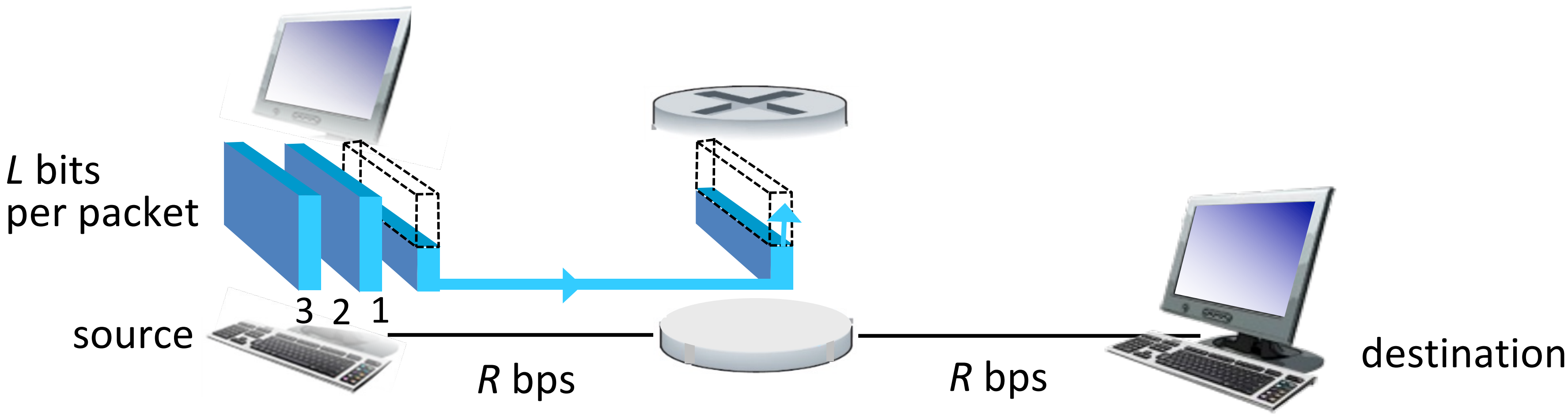
Example:
- 200 Mb/s link
- each user:
  - 20 Mb/s when "active"
  - active 30% of time



- *Circuit switching:*
  - 10 users
- *Packet switching:*
  - with 19 users, probability > 10 active at the same time is less than 0.011

- Great for bursty data
  - resource sharing
  - simpler, no call setup
- Excessive congestion possible: packet delay and loss
  - protocols needed for reliable data transfer, congestion control

Packet-switching: store-and-forward



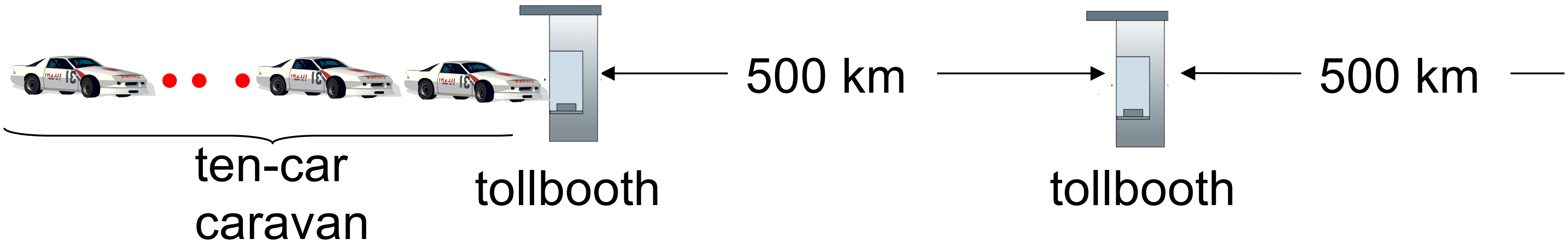
- takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- end-end delay = 2*L*/*R* (assuming zero propagation delay)

more on delay shortly …

one-hop numerical example:

- *L* = 7.5 Mbits
- *R* = 1.5 Mbps
- one-hop transmission delay = 5 sec

Example: Store-and-forward, Car – Caravan Analogy

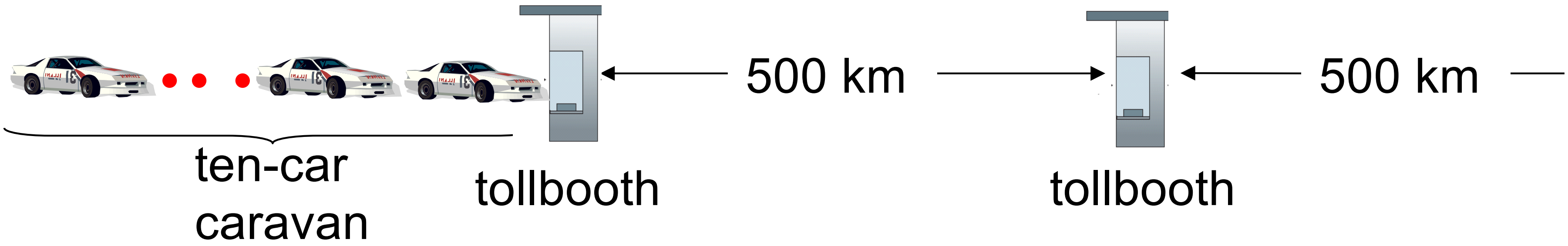


The caravan has 10 cars. The tollbooth services (i.e., transmits) a car at a rate of one car per 2 seconds. After the service, a car proceeds to the next tollbooth, which is 500 kilometers away at a rate of 20 kilometers per second. When the first car of the caravan arrives at a tollbooth, it must wait at the entrance to the toll booth until all of the other cars in its caravan have arrived.

Q1. Once a car enters service at the tollbooth, how long does it take until it leaves service?

Ans: Service time is 2 seconds

Example: Store-and-forward, Car – Caravan Analogy

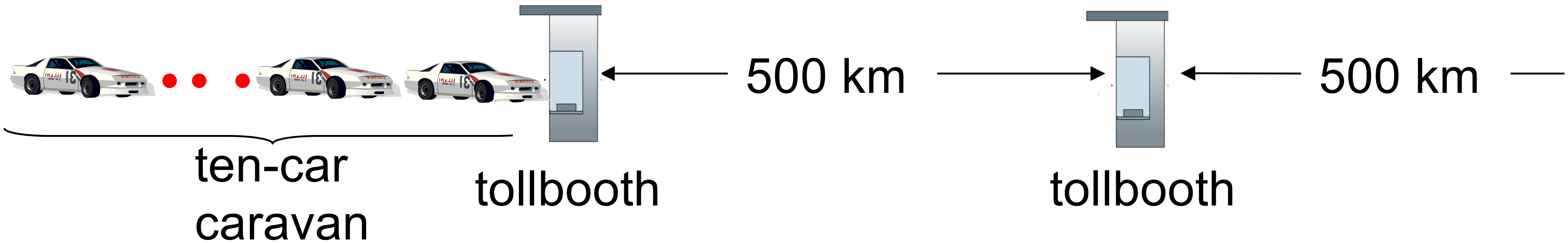


The caravan has 10 cars. The tollbooth services (i.e., transmits) a car at a rate of one car per 2 seconds. After the service, a car proceeds to the next tollbooth, which is 500 kilometers away at a rate of 20 kilometers per second. When the first car of the caravan arrives at a tollbooth, it must wait at the entrance to the toll booth until all of the other cars in its caravan have arrived.

Q2. 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)?

Ans: It takes 20 seconds to service every car.
(10 cars * 2 seconds per car)

Example: Store-and-forward, Car – Caravan Analogy

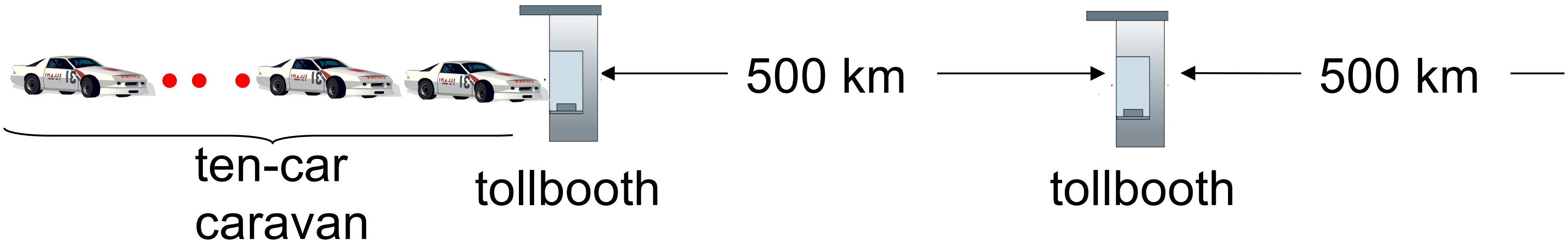


The caravan has 10 cars. The tollbooth services (i.e., transmits) a car at a rate of one car per 2 seconds. After the service, a car proceeds to the next tollbooth, which is 500 kilometers away at a rate of 20 kilometers per second. When the first car of the caravan arrives at a tollbooth, it must wait at the entrance to the toll booth until all of the other cars in its caravan have arrived.

Q3. Once the first car leaves the tollbooth, how long does It take until it arrives at the next tollbooth?

Ans: It takes 25 seconds to travel to the next toll booth (500 km / 20 km/s)

Example: Store-and-forward, Car – Caravan Analogy

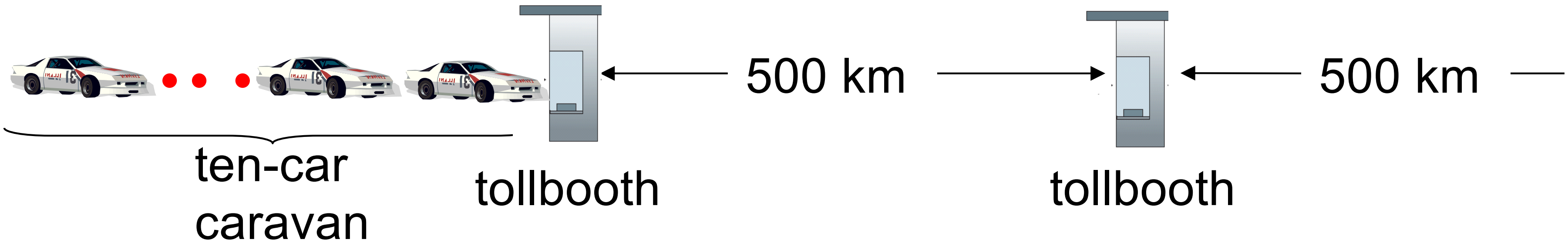


The caravan has 10 cars. The tollbooth services (i.e., transmits) a car at a rate of one car per 2 seconds. After the service, a car proceeds to the next tollbooth, which is 500 kilometers away at a rate of 20 kilometers per second. When the first car of the caravan arrives at a tollbooth, it must wait at the entrance to the toll booth until all of the other cars in its caravan have arrived.

Q4. Once the last car leaves the tollbooth, how long does it take until it arrives at the next tollbooth?

Ans: Just like in the previous question, it takes 25 seconds, regardless of the car.

Example: Store-and-forward, Car – Caravan Analogy

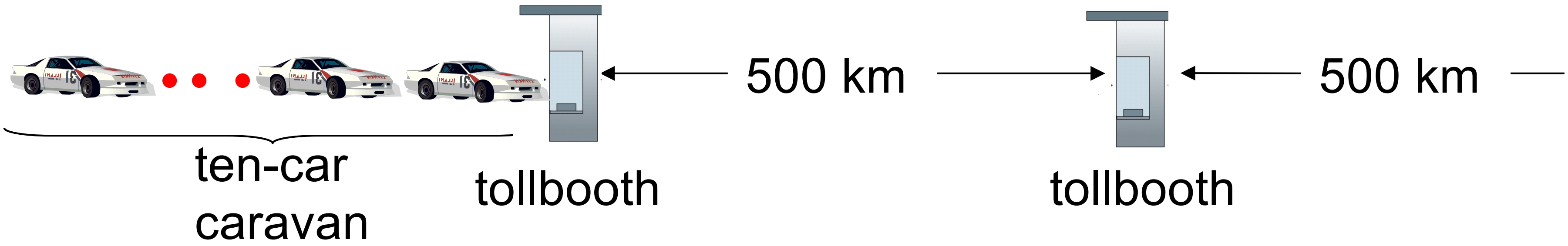


The caravan has 10 cars. The tollbooth services (i.e., transmits) a car at a rate of one car per 2 seconds. After the service, a car proceeds to the next tollbooth, which is 500 kilometers away at a rate of 20 kilometers per second. When the first car of the caravan arrives at a tollbooth, it must wait at the entrance to the toll booth until all of the other cars in its caravan have arrived.

Q5. Once the first car leaves the tollbooth, how long does it take until it enters service at the next tollbooth?

Ans: It takes 43 seconds until the first car gets serviced at the next toll booth.
(10-1 cars * 2 seconds per car + 500 km / 20 km/s)

Example: Store-and-forward, Car – Caravan Analogy

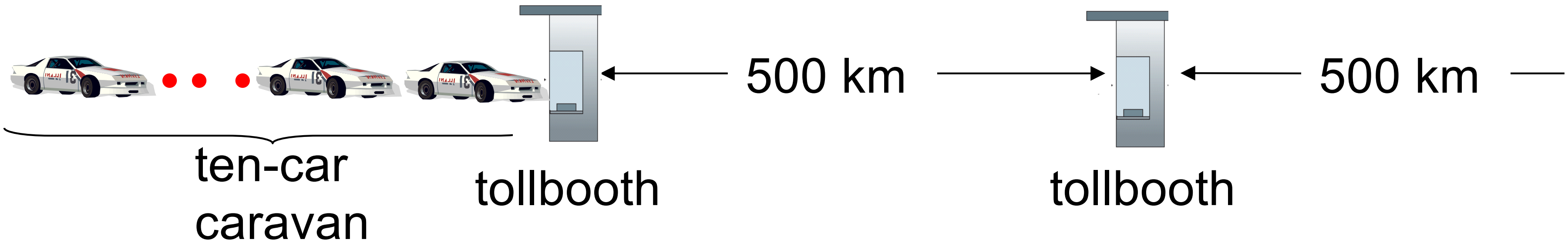


The caravan has 10 cars. The tollbooth services (i.e., transmits) a car at a rate of one car per 2 seconds. After the service, a car proceeds to the next tollbooth, which is 500 kilometers away at a rate of 20 kilometers per second. When the first car of the caravan arrives at a tollbooth, it must wait at the entrance to the toll booth until all of the other cars in its caravan have arrived.

Q6. Are there ever two cars in service at the same time, one at the first toll booth and one at the second toll booth? (Yes or No)

Ans: No, because cars can't get service at the next tollbooth until all cars have arrived.

Example: Store-and-forward, Car – Caravan Analogy



The caravan has 10 cars. The tollbooth services (i.e., transmits) a car at a rate of one car per 2 seconds. After the service, a car proceeds to the next tollbooth, which is 500 kilometers away at a rate of 20 kilometers per second. When the first car of the caravan arrives at a tollbooth, it must wait at the entrance to the toll booth until all of the other cars in its caravan have arrived.

Q7. Are there ever zero cars in service at the same time, i.e., the caravan of cars has finished at the first tollbooth but not yet arrived at the second tollbooth? Yes or No

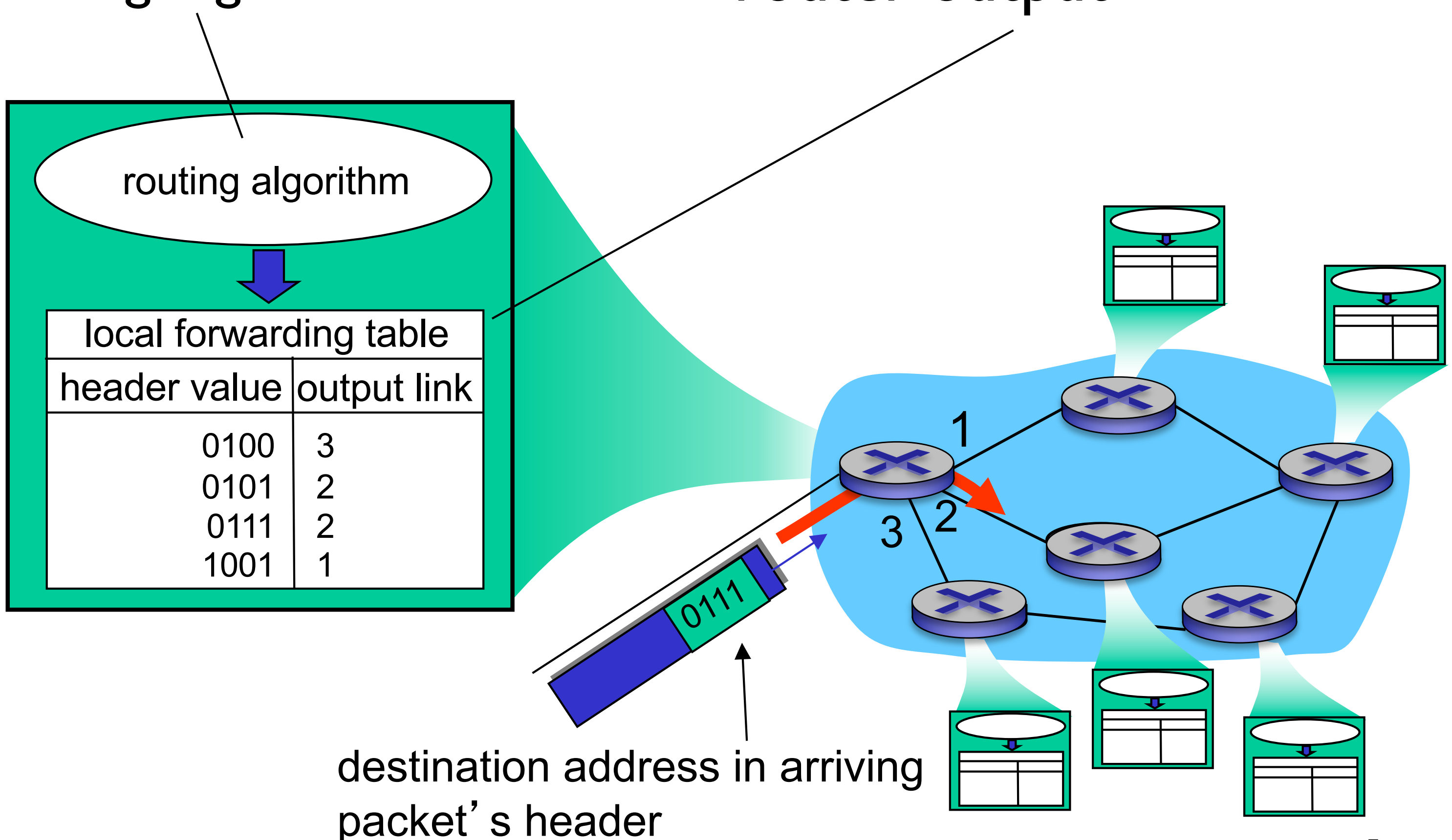
Ans: Yes. one notable example is when the last car in the caravan is serviced but is still travelling to the next tollbooth; all other cars have to walt until it arrives. Thus, no cars are being serviced.

Two key network-core functions

routing: determines source-destination route taken by packets

- *routing algorithms*

forwarding: move packets from router's input to appropriate router output

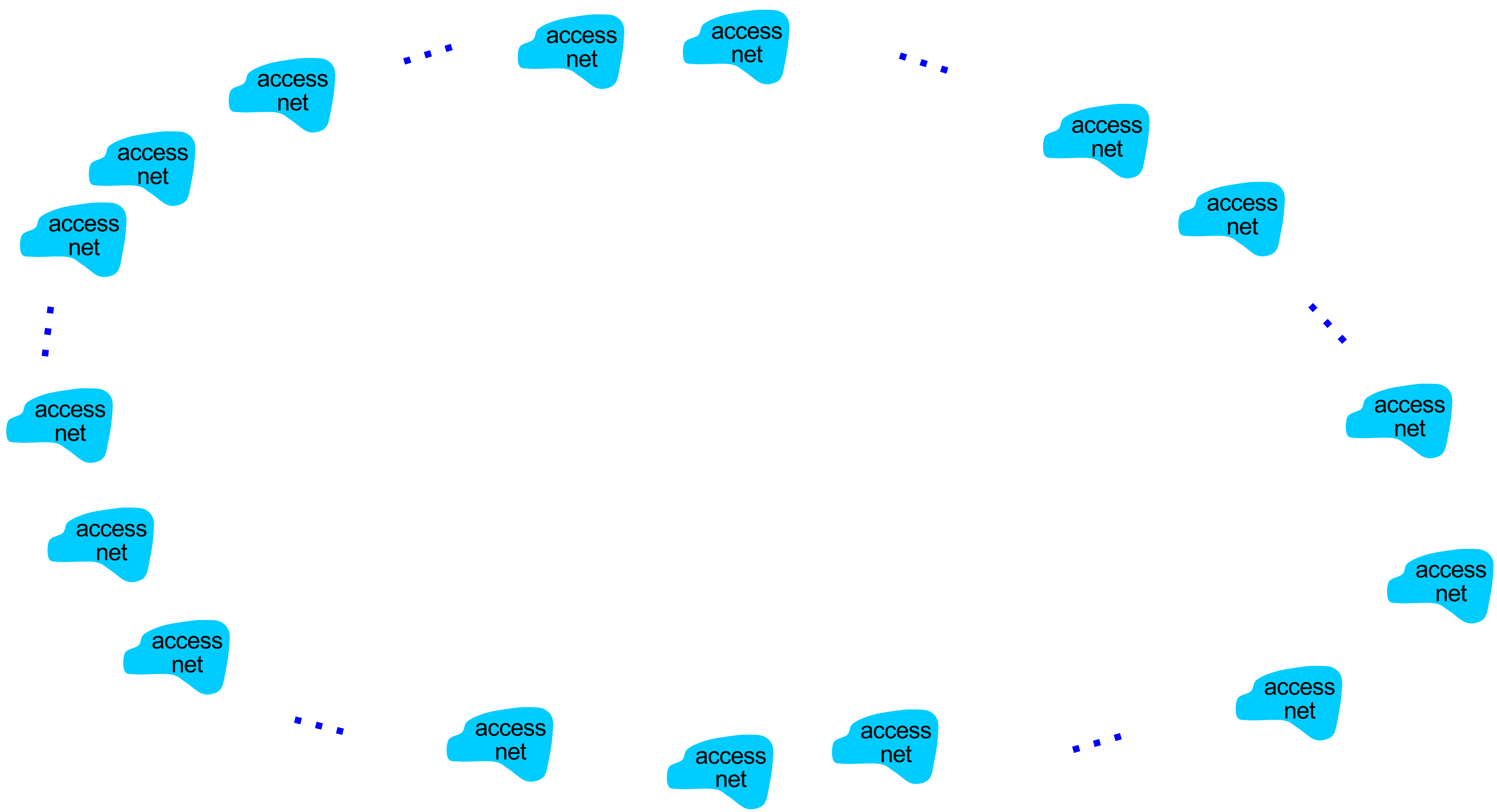


Internet structure: network of networks

- End systems connect to the Internet via access ISPs (Internet Service Providers)
  - residential, company, and university 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

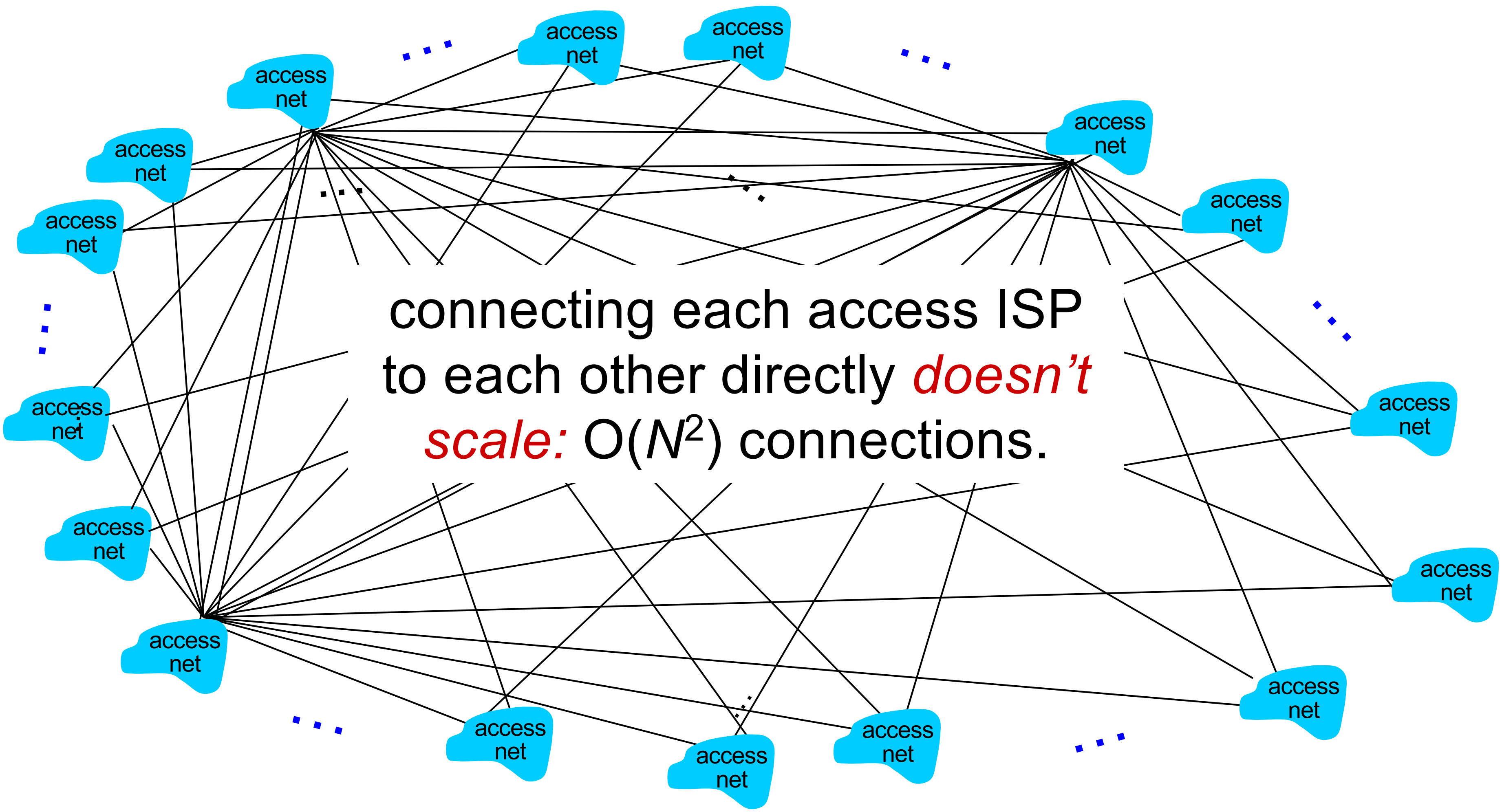
Internet structure: network of networks

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



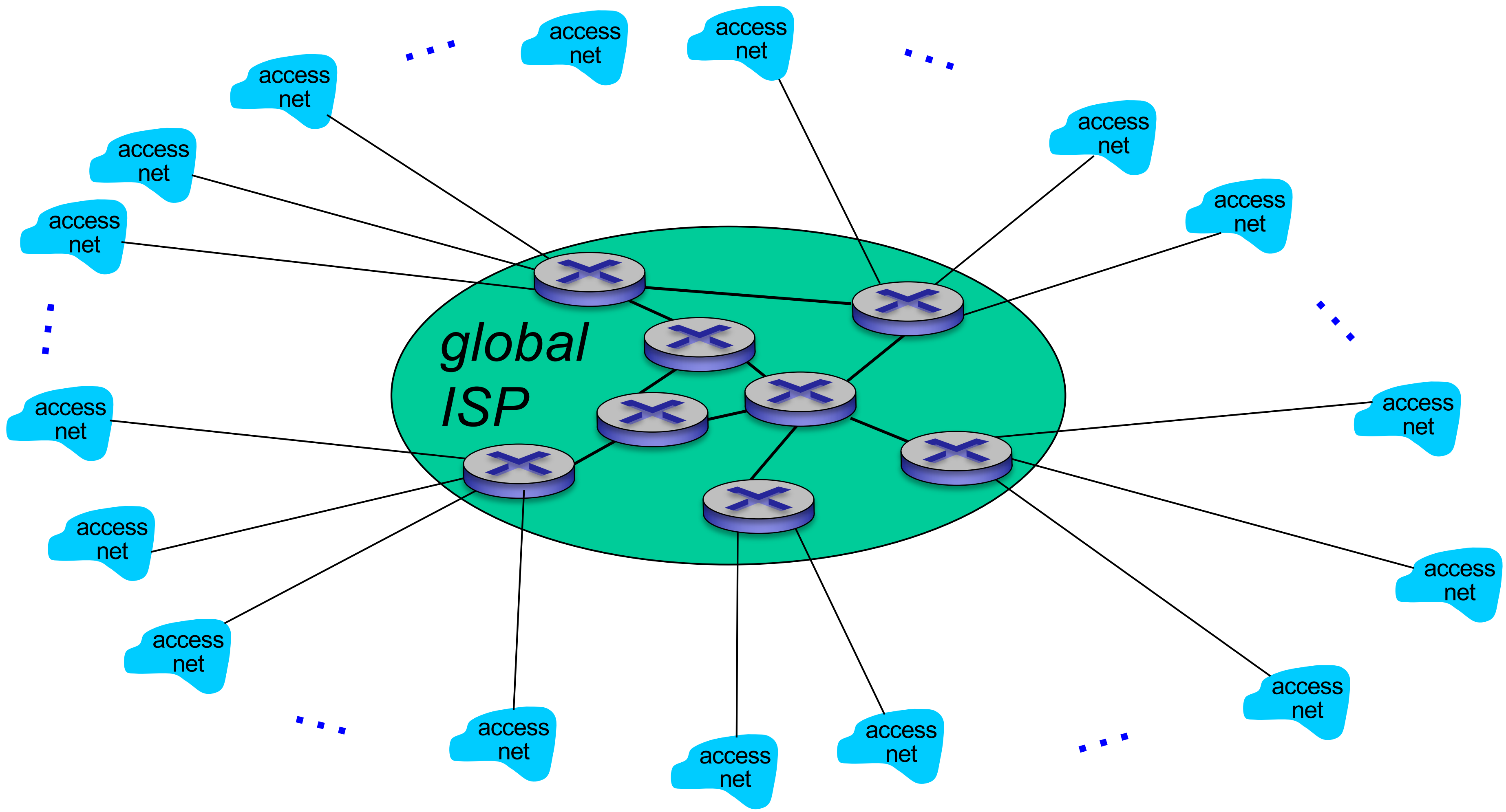
Internet structure: network of networks

Option: connect each access ISP to every other access ISP?



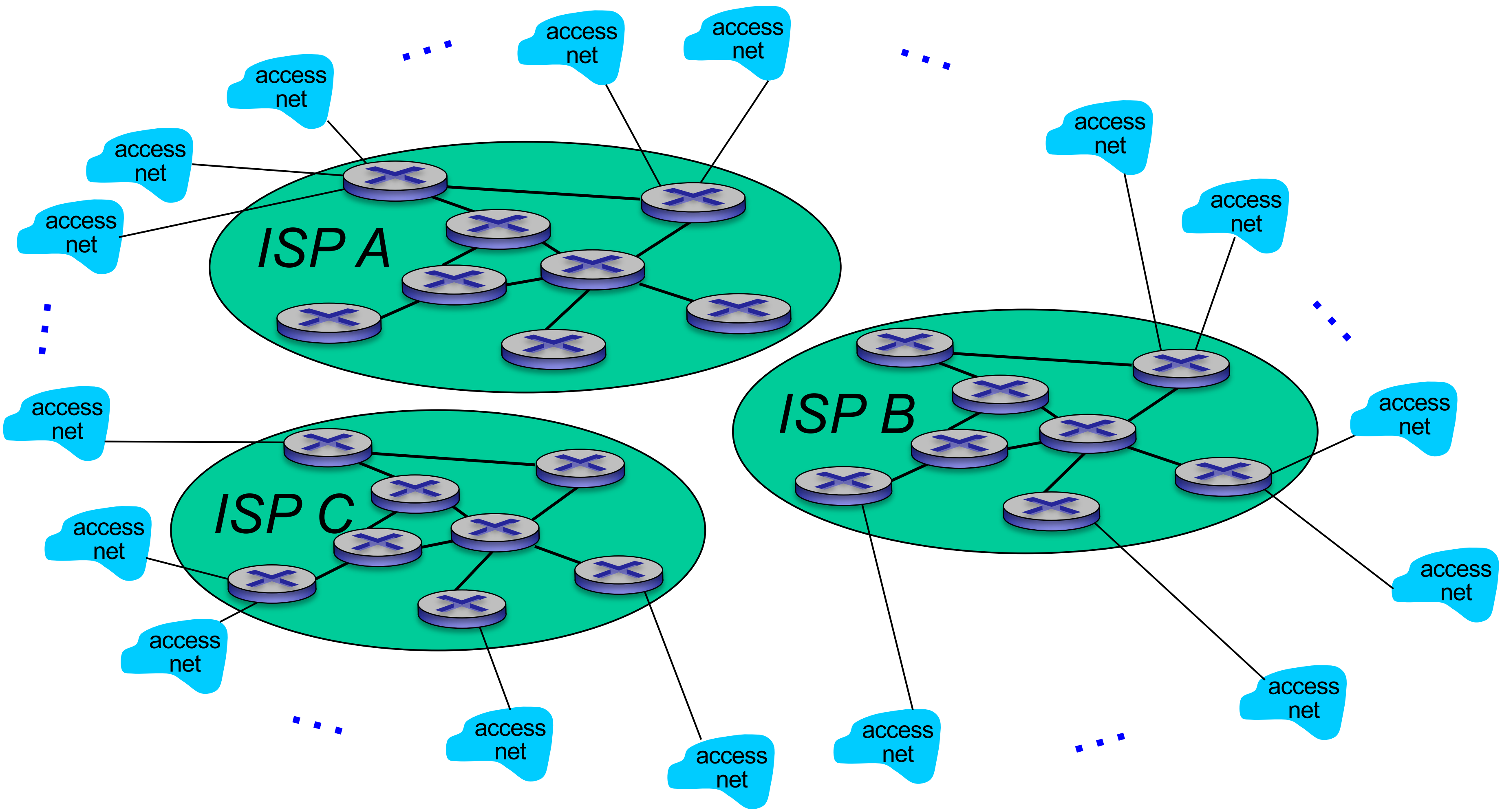
Internet structure: network of networks

Option: connect each access ISP to one global transit ISP?
Customer and provider ISPs have economic agreement.



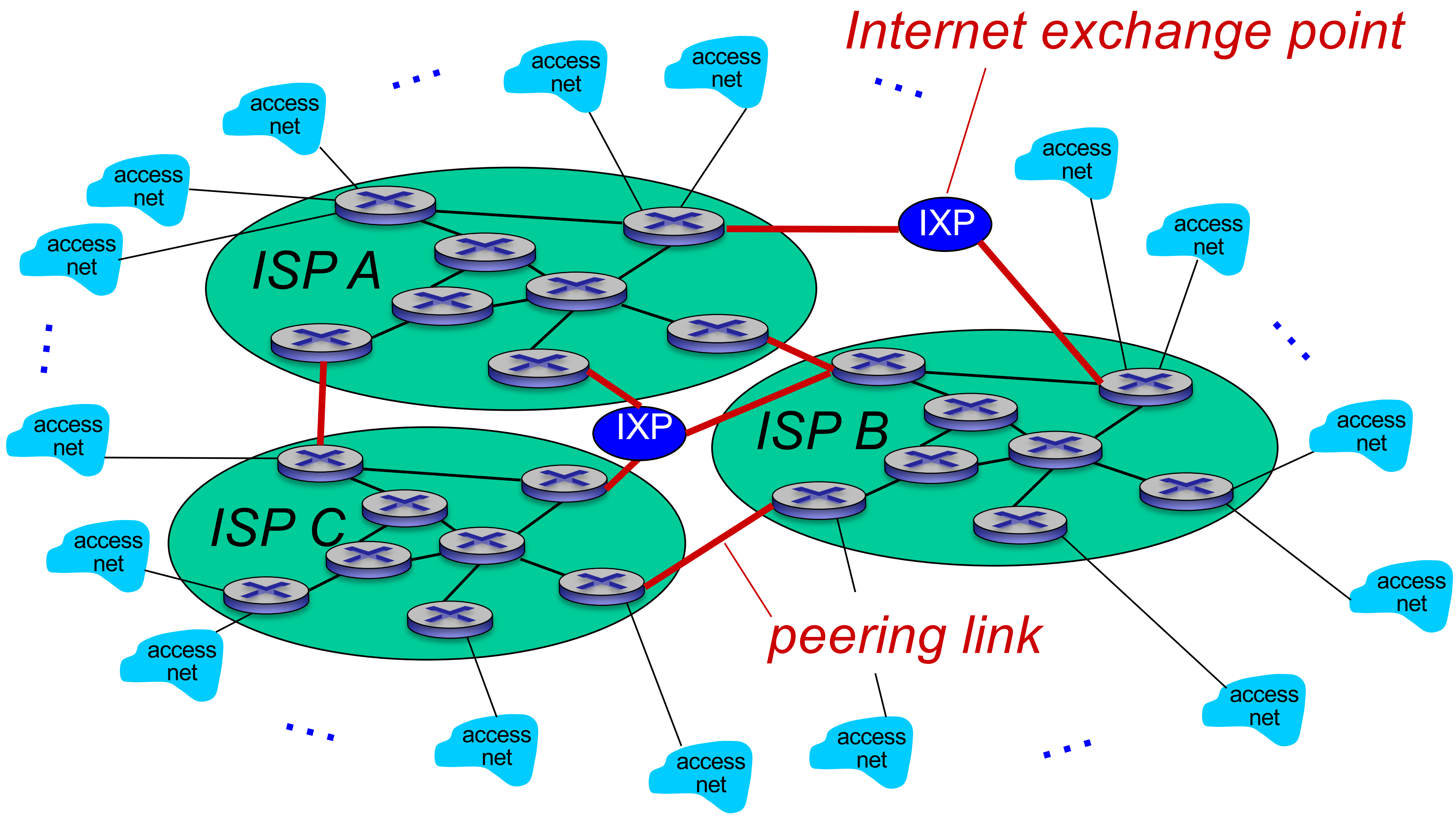
Internet structure: network of networks

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



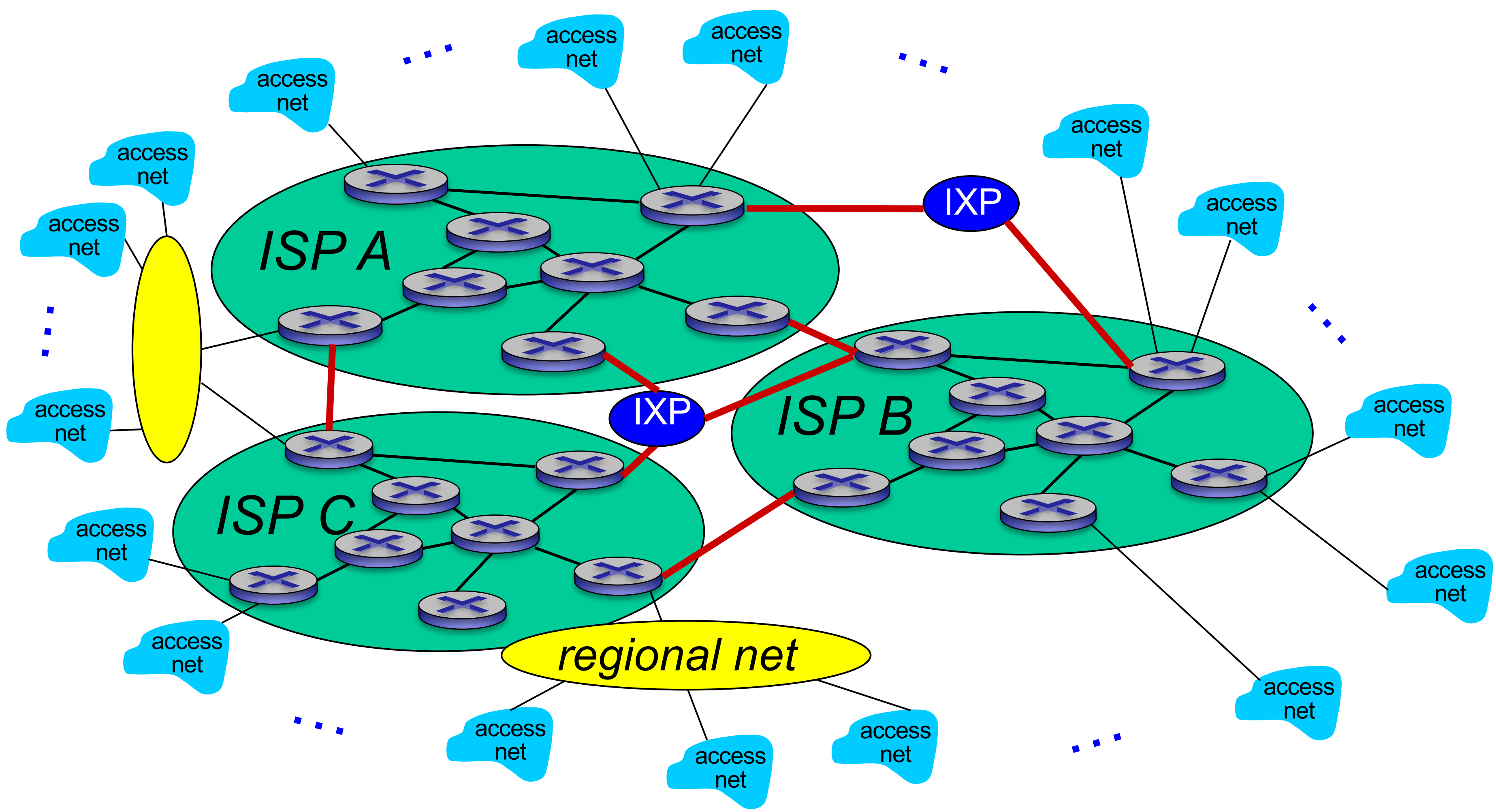
Internet structure: network of networks

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



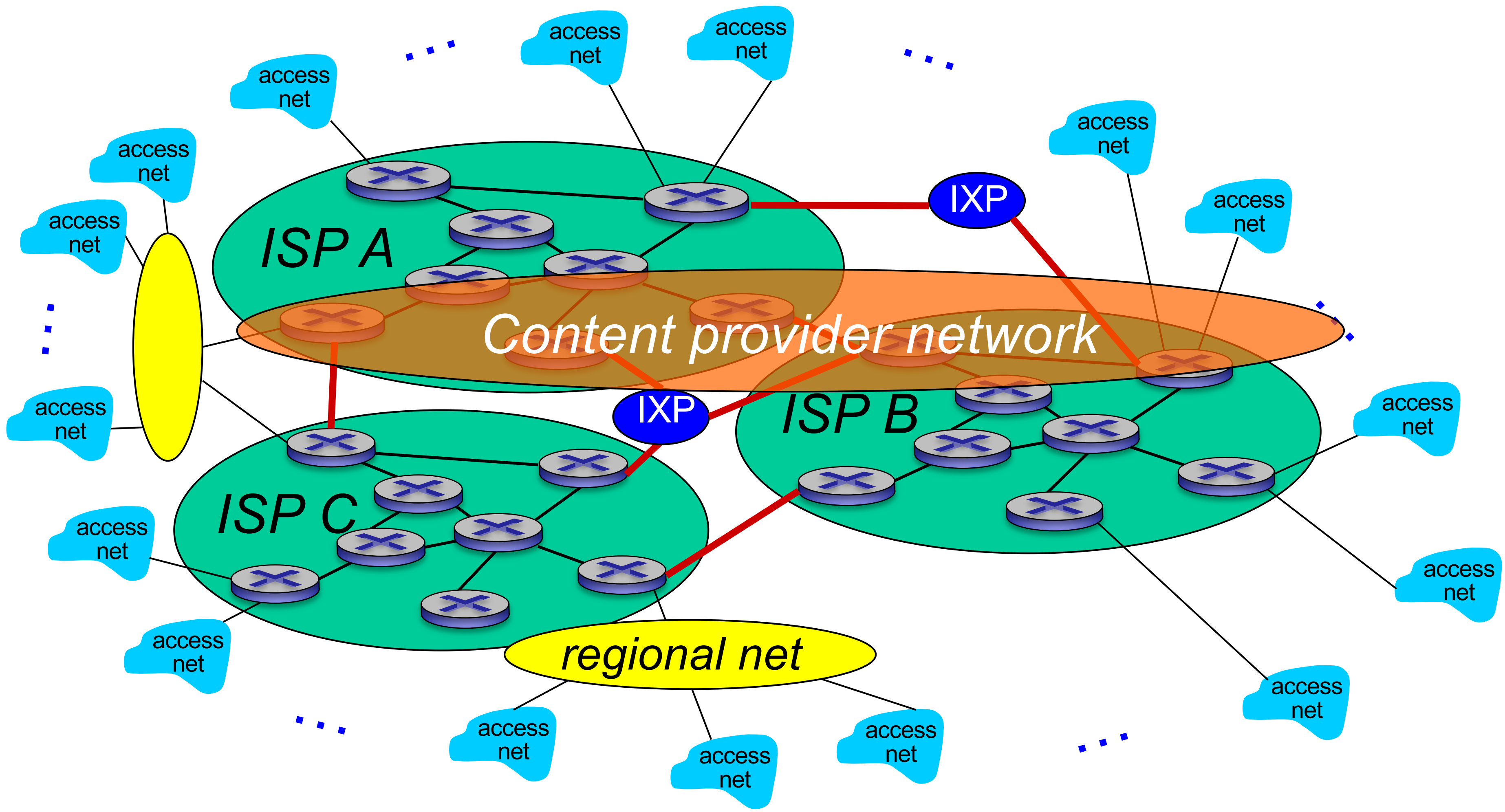
Internet structure: network of networks

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



Internet structure: 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



Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

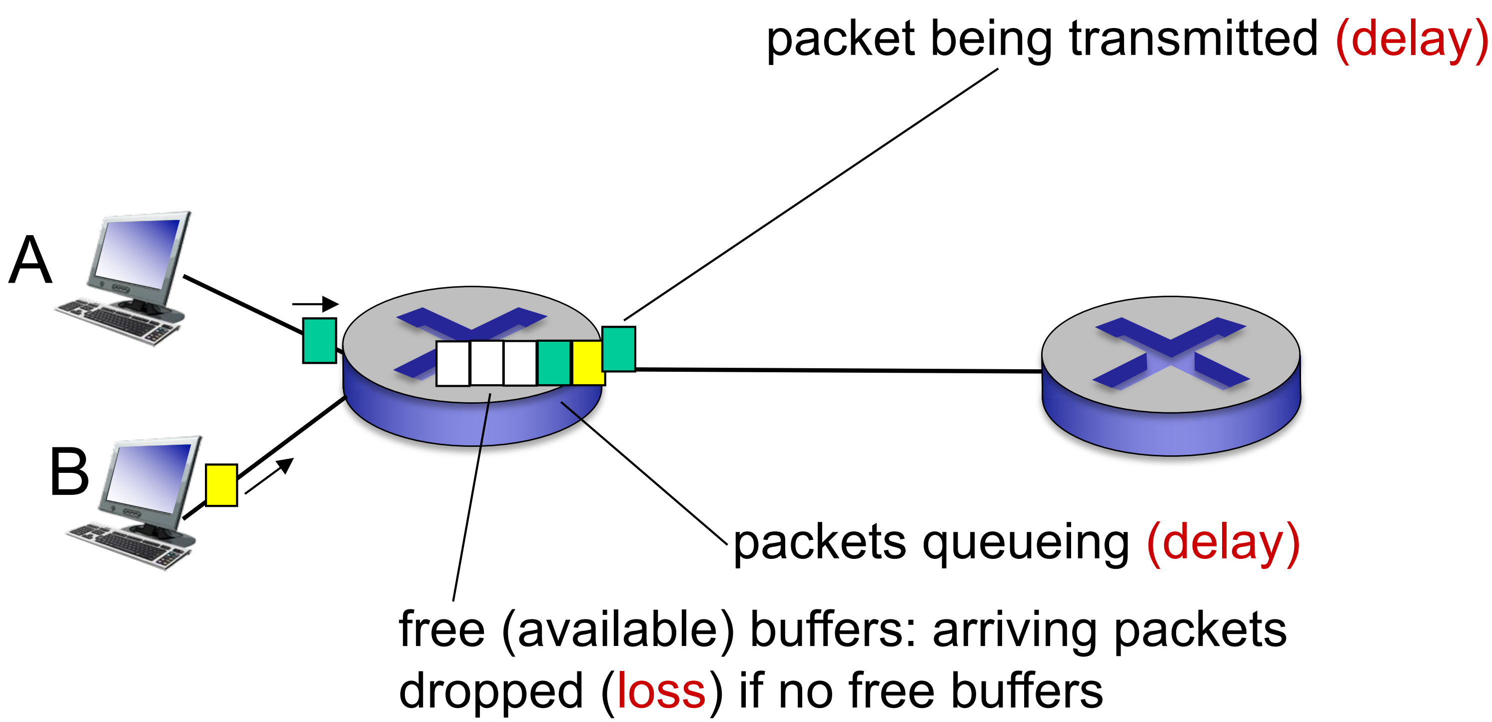
1.5 protocol layers, service models

1.6 networks under attack: security

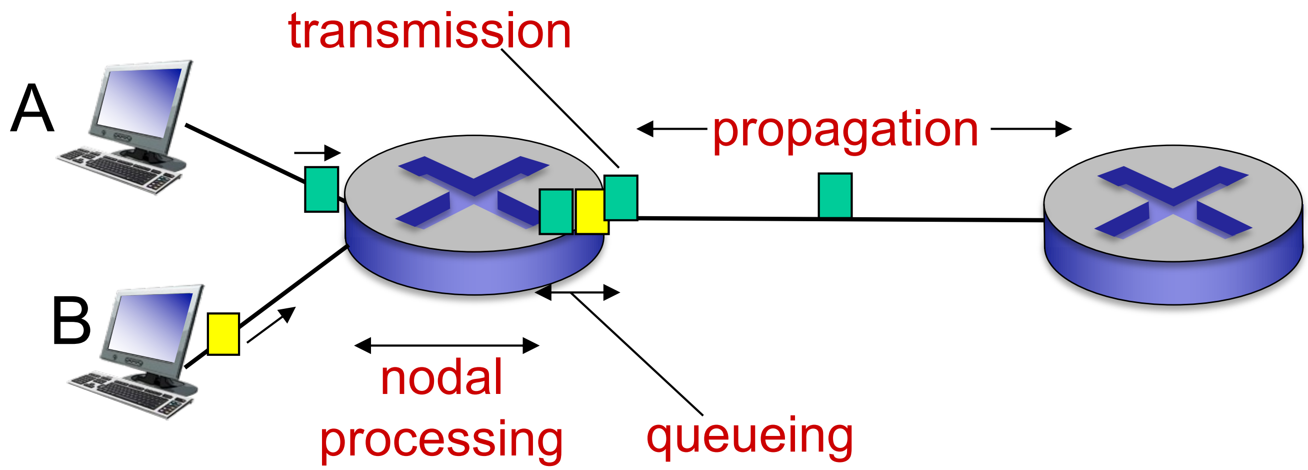
How do loss and delay occur?

packets *queue* in router buffers

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



Four sources of packet delay



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

d_{proc}: nodal processing

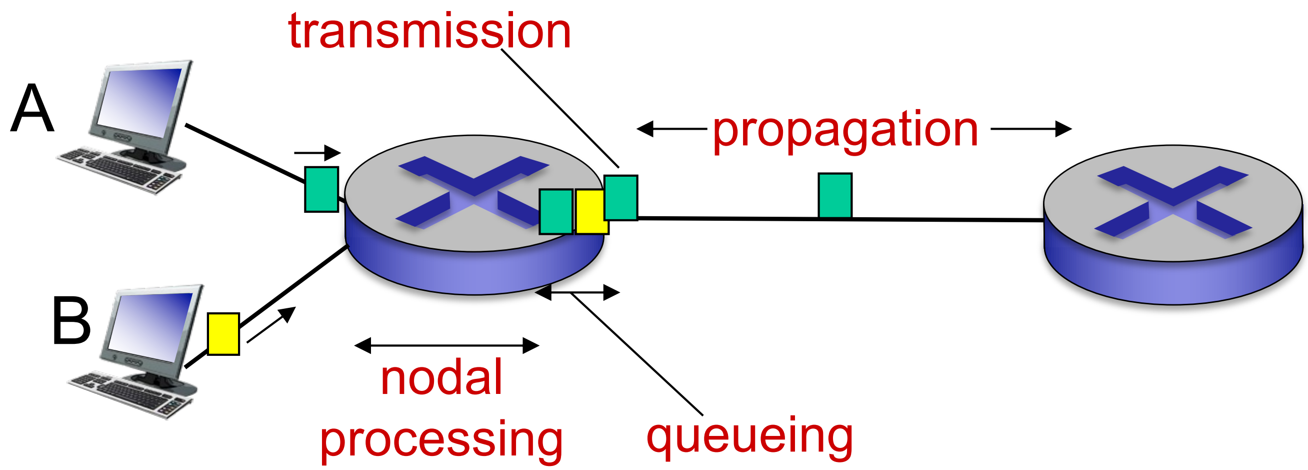
- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay

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


PLAZA Floor. 38
FOR RENT
59

Four sources of packet delay



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

d_{trans}: transmission delay:

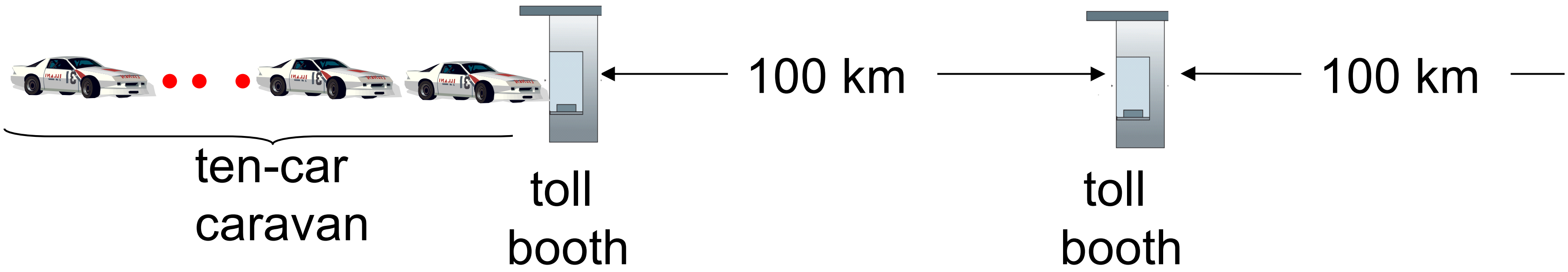
- L: packet length (bits)
- R: link *bandwidth (bps)*
- $d_{trans} = L/R$

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

d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed (~2×10^8 m/sec)
- $d_{prop} = d/s$

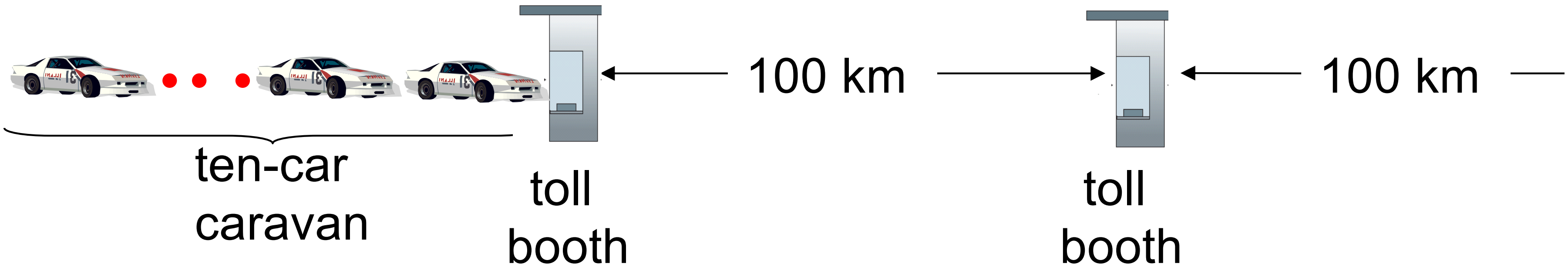
Caravan analogy



- 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

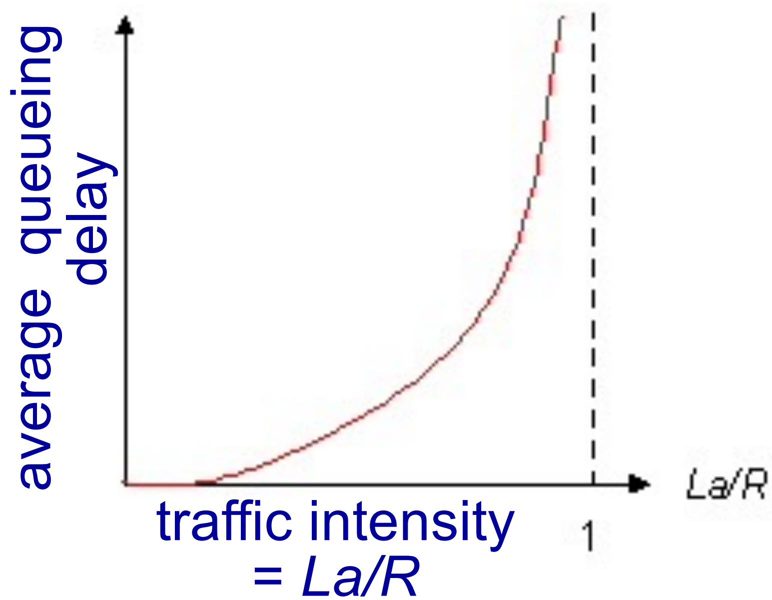
Caravan analogy (more)



- 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

Queueing delay (revisited)

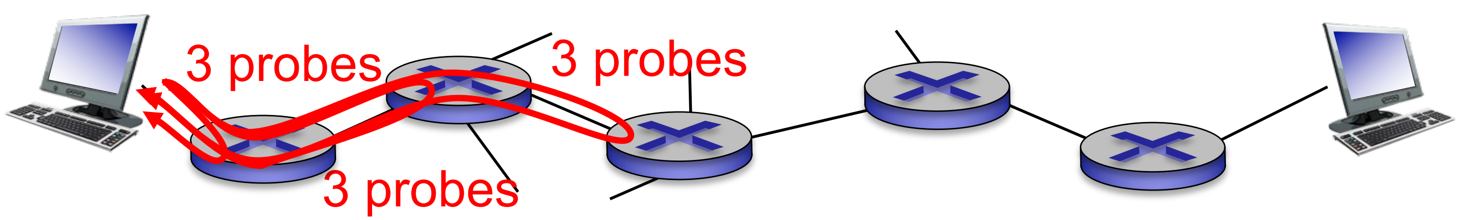
- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- a: average packet arrival rate



- *La/R* ~ 0: avg. queueing delay small
- *La/R* → 1: avg. queueing delay large
- *La/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 toward destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply



"Real" Internet delays, 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
```

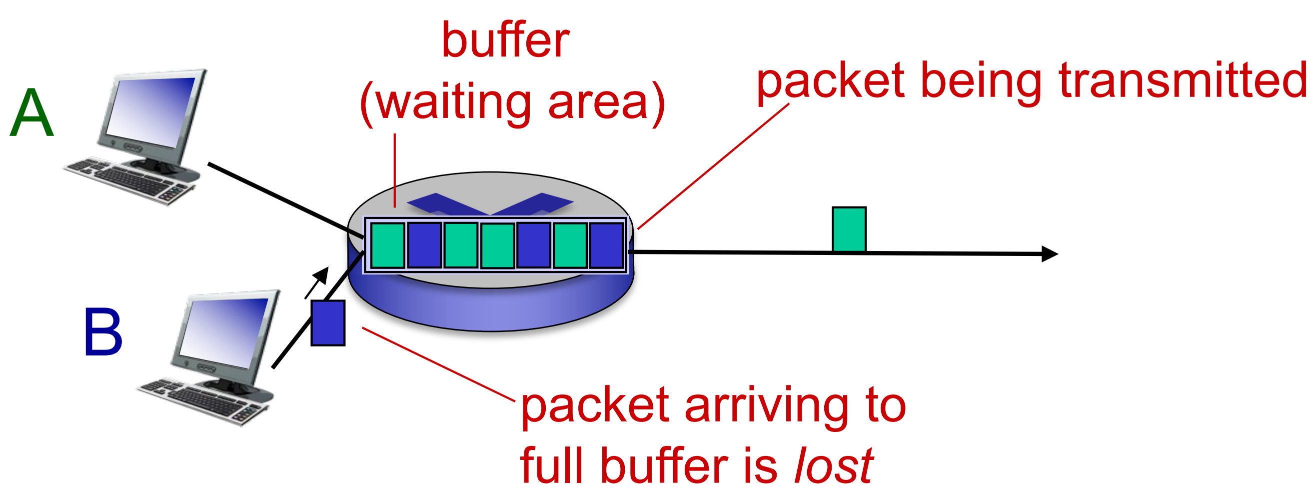
trans-oceanic link

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

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

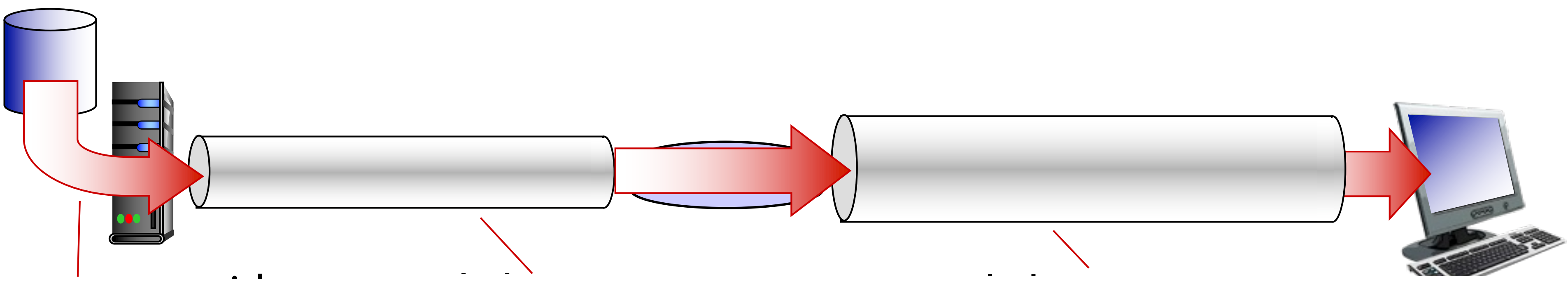
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packets arriving at a full queue get dropped (aka lost)
- How to handle lost packets?
  - 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 a given point in time
  - *average:* rate over a longer period of time



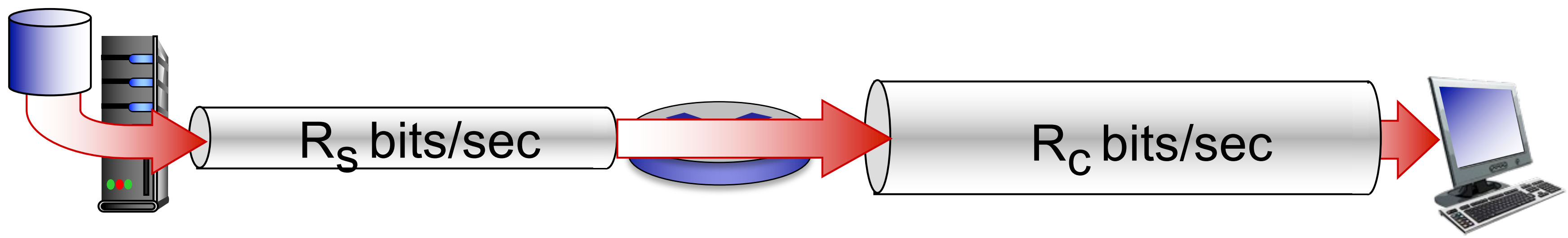
server sends bits (fluid) into pipe

pipe that can carry fluid at rate R_s bits/sec

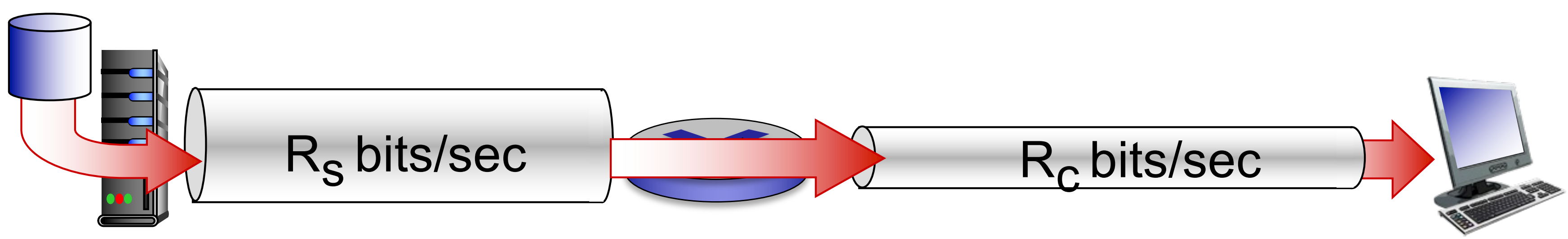
pipe that can carry fluid at rate R_c bits/sec

Throughput (more)

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



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

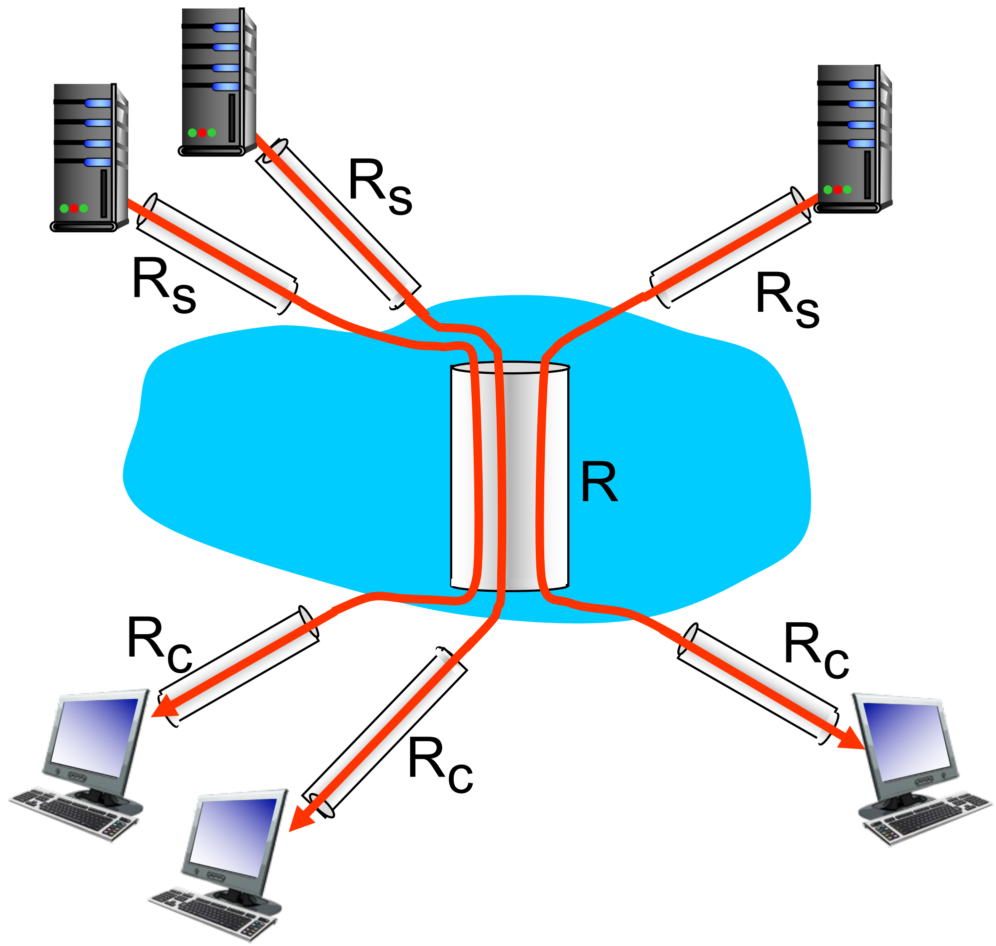


bottleneck link

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

Throughput: Internet scenario

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

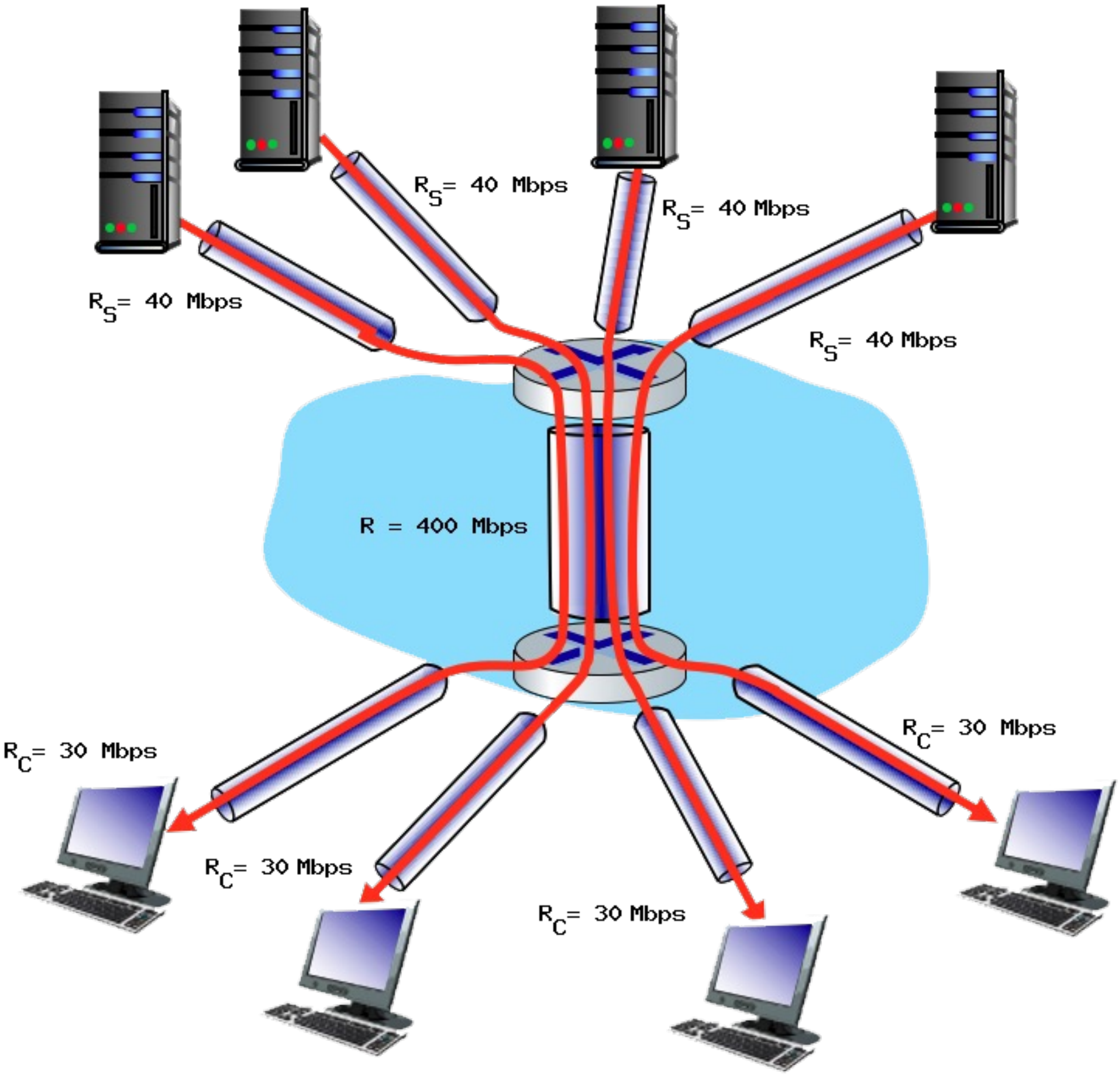


n connections (fairly) share backbone bottleneck link *R* bits/sec

Example

Q1: What is the max end-end throughput for each client-server connection?

Ans: $min(R_c, R_s, R/4)$
$= min(30, 40, 100)$
$= 30$ Mbps

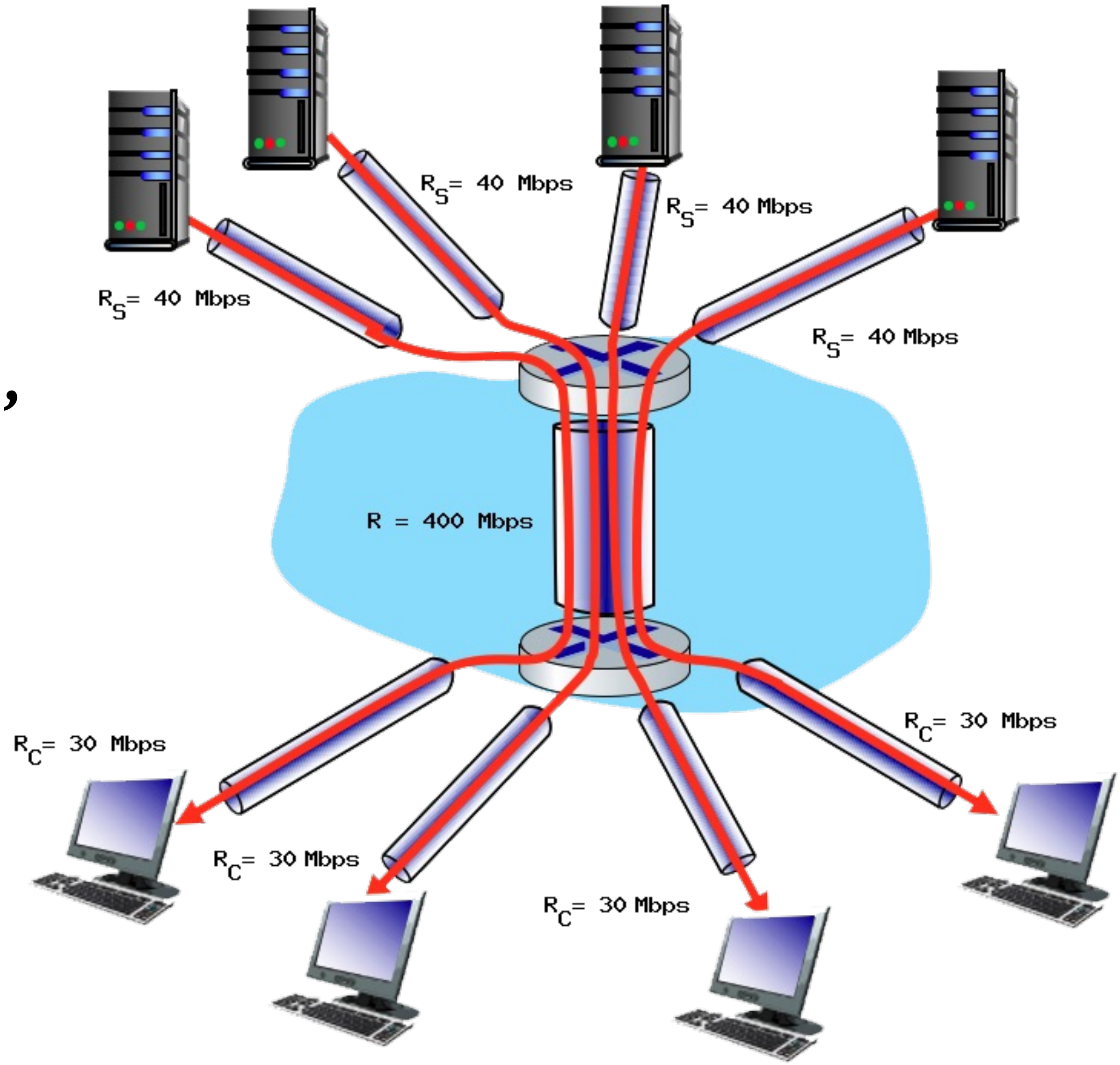


- *$R_c = 30$, $R_s = 40$, $R = 400$ (Mbps)*
- *Four client-to-server pairs, fairly shared*

Example

Q2: Which link is the bottleneck link? R_c, R_s, or R?

Ans: R_c

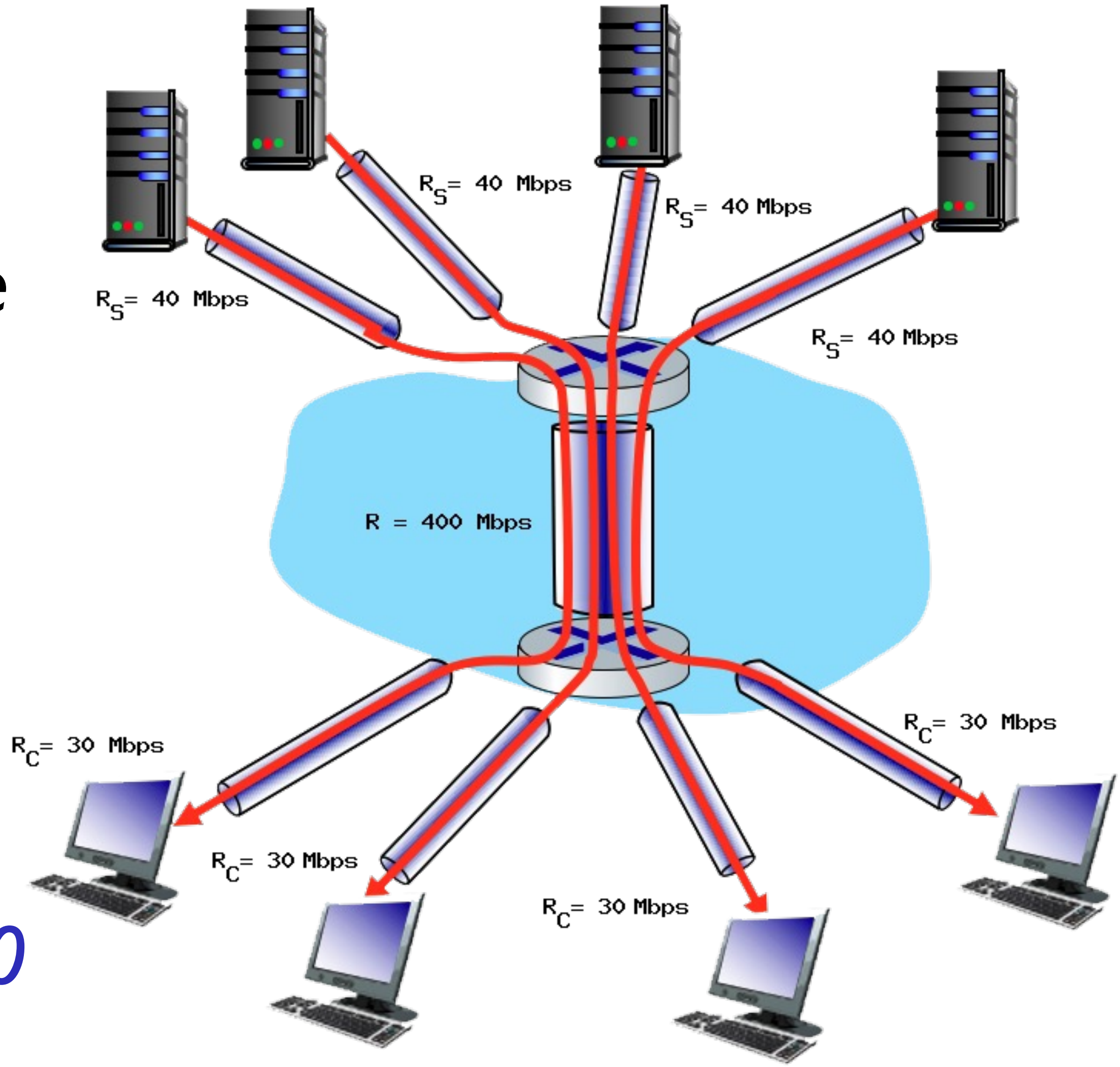


- *R_c = 30, R_s = 40, R = 400 (Mbps)*
- *Four client-to-server pairs, fairly shared*

Example

Q3: Assuming that the servers are sending at the maximum rate possible, what are the link utilizations for the server links (R_S)?

Ans: R_c / R_s = 30 / 40 = 0.75

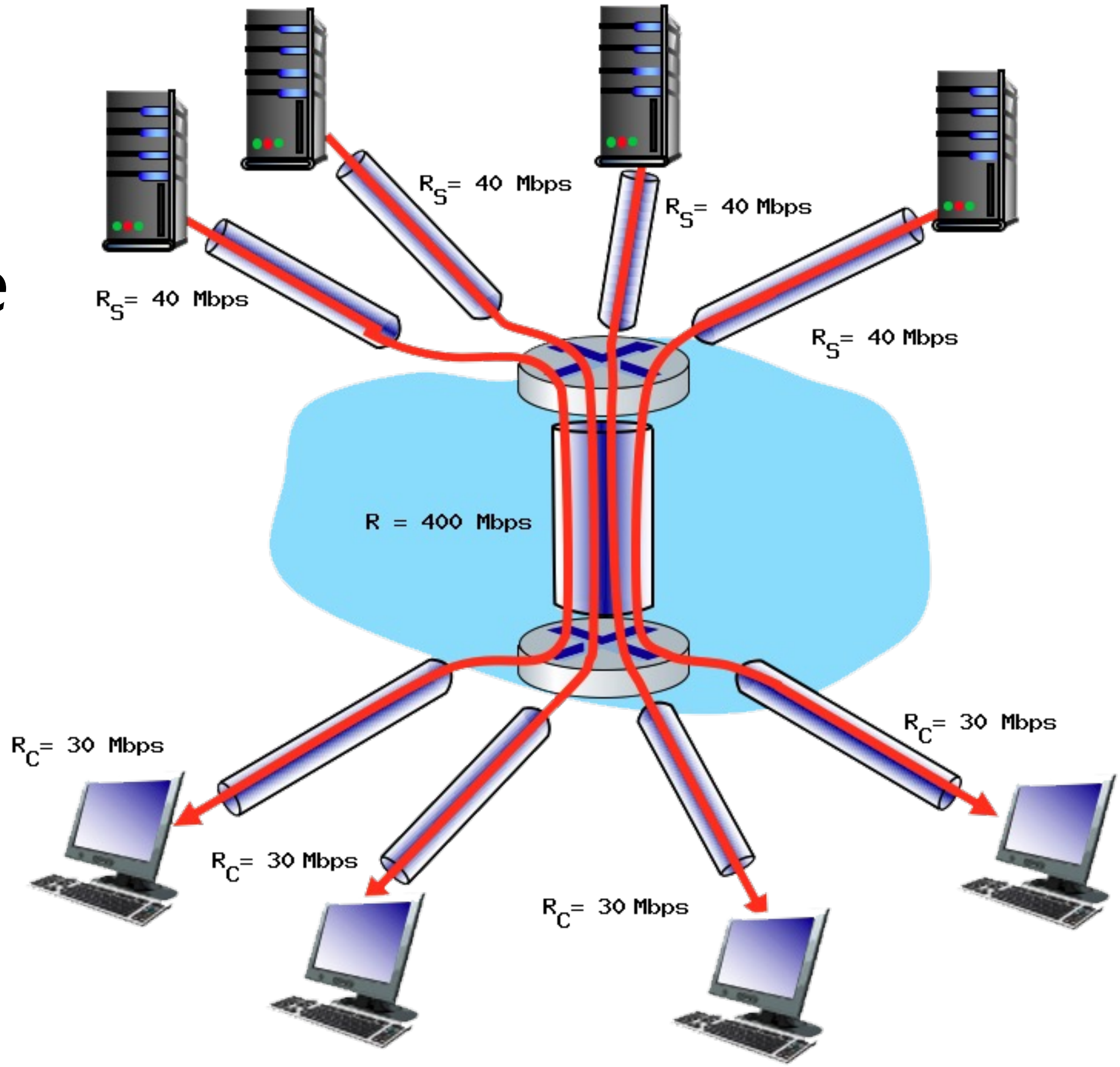


- *R_c = 30, R_s = 40, R = 400 (Mbps)*
- *Four client-to-server pairs, fairly shared*

Example

Q3: Assuming that the servers are sending at the maximum rate possible, what are the link utilizations for the client links (R_c)?

Ans: R_c / R_c = 30 / 30 = 1.0

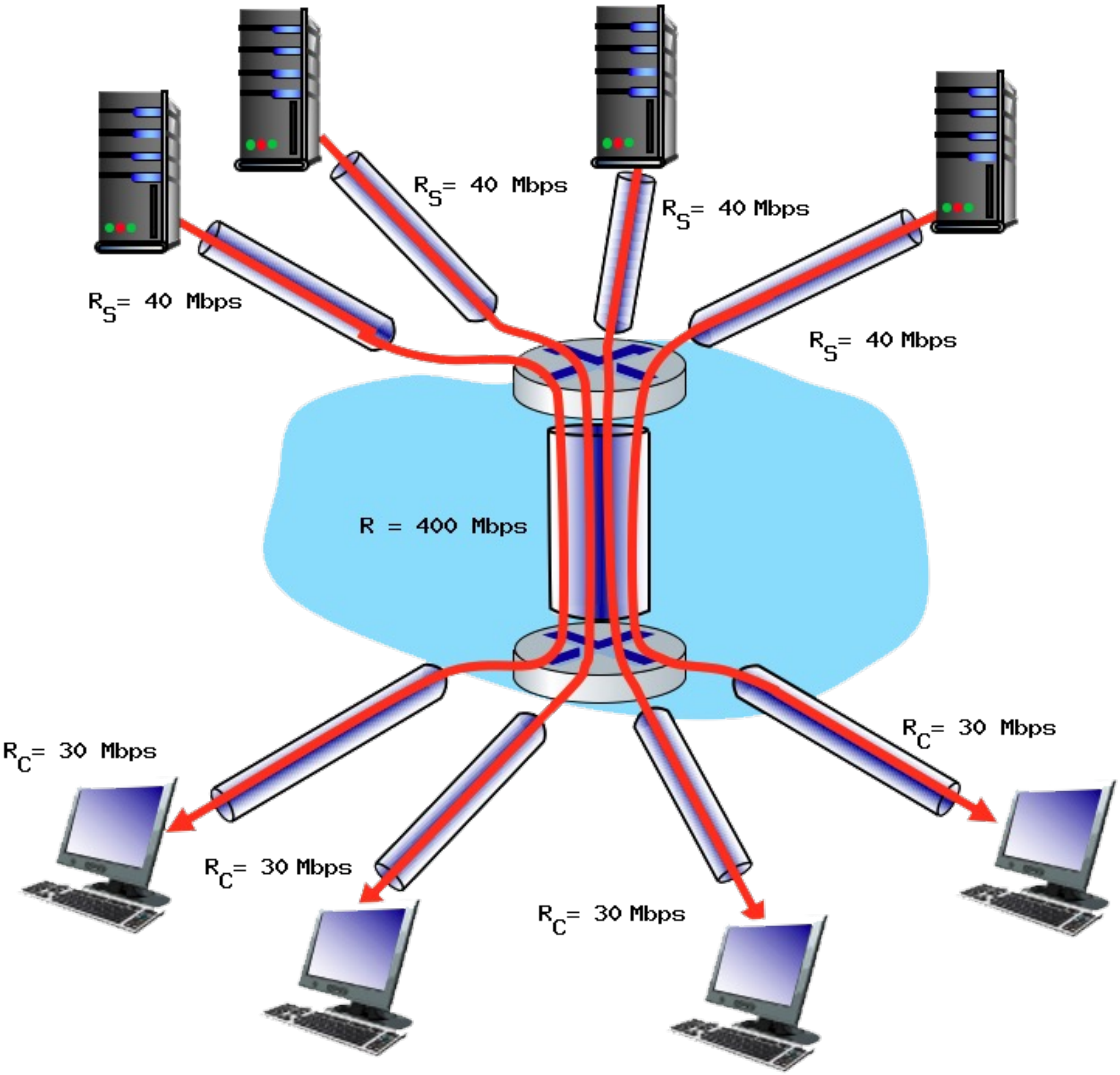


- *R_c = 30, R_s = 40, R = 400 (Mbps)*
- *Four client-to-server pairs, fairly shared*

Example

Q3: Assuming that the servers are sending at the maximum rate possible, what is the link utilization for the shared link (R)?

Ans: R_c / R = 30 / 100 = 0.3



- *R_c = 30, R_s = 40, R = 400 (Mbps)*
- *Four client-to-server pairs, fairly shared*

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

Protocol "layers"

Networks are complex, with many "pieces":

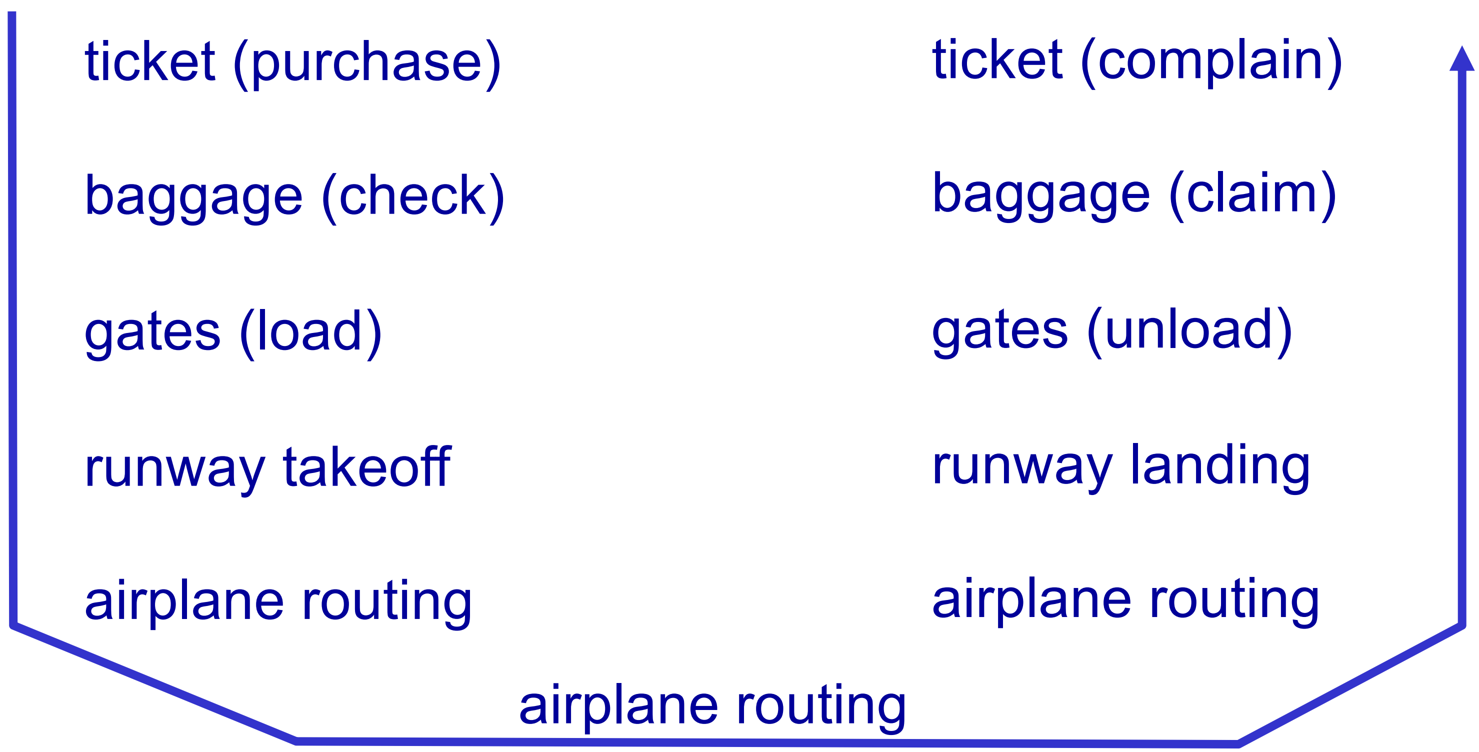
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

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

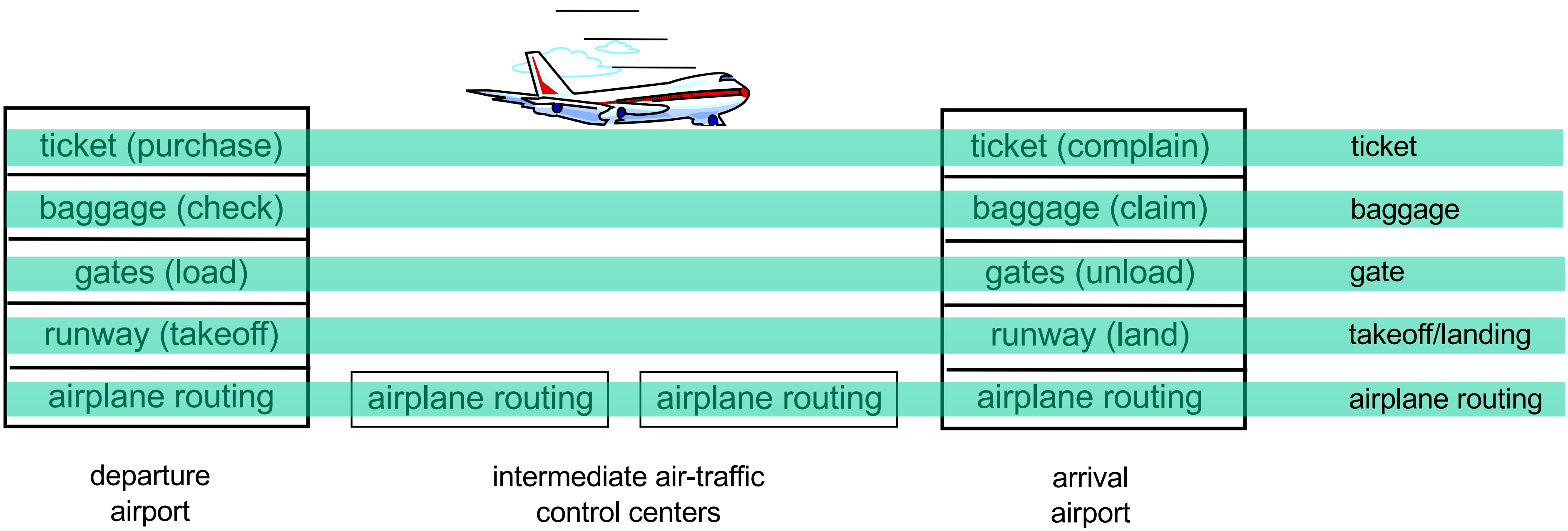
…. or at least our discussion of networks?

Organization of air travel



- a series of steps

Layering of airline functionality



layers: each layer implements a service

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

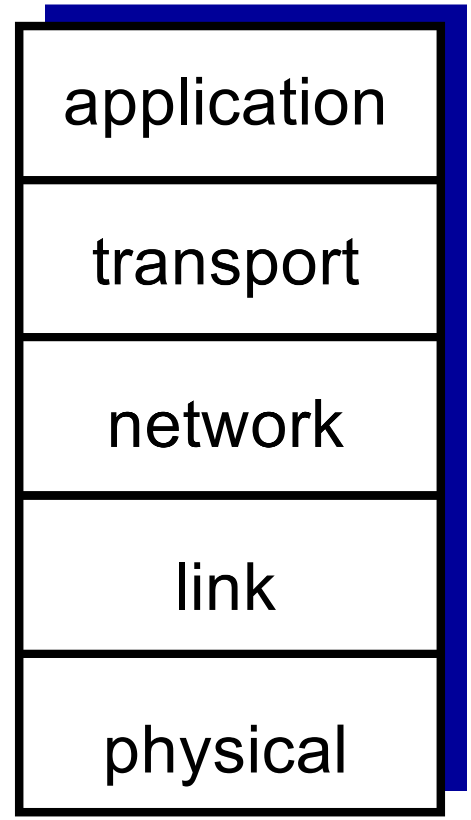
Why layering?

dealing with complex systems:

- explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, and updating of system
  - change of implementation of layer's service transparent to rest of system
  - e.g., a change in gate procedure doesn't affect rest of the system
- layering considered harmful?

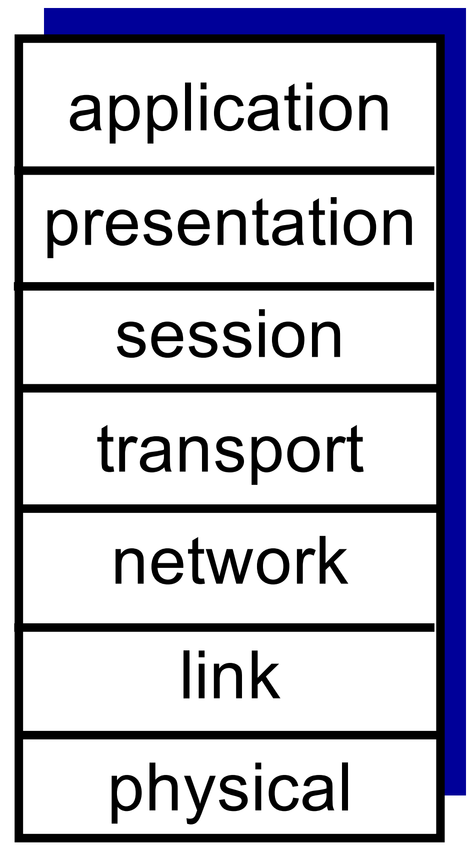
Internet protocol stack

- *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
  - Ethernet, 802.111 (WiFi), PPP
- *physical:* bits "on the wire"

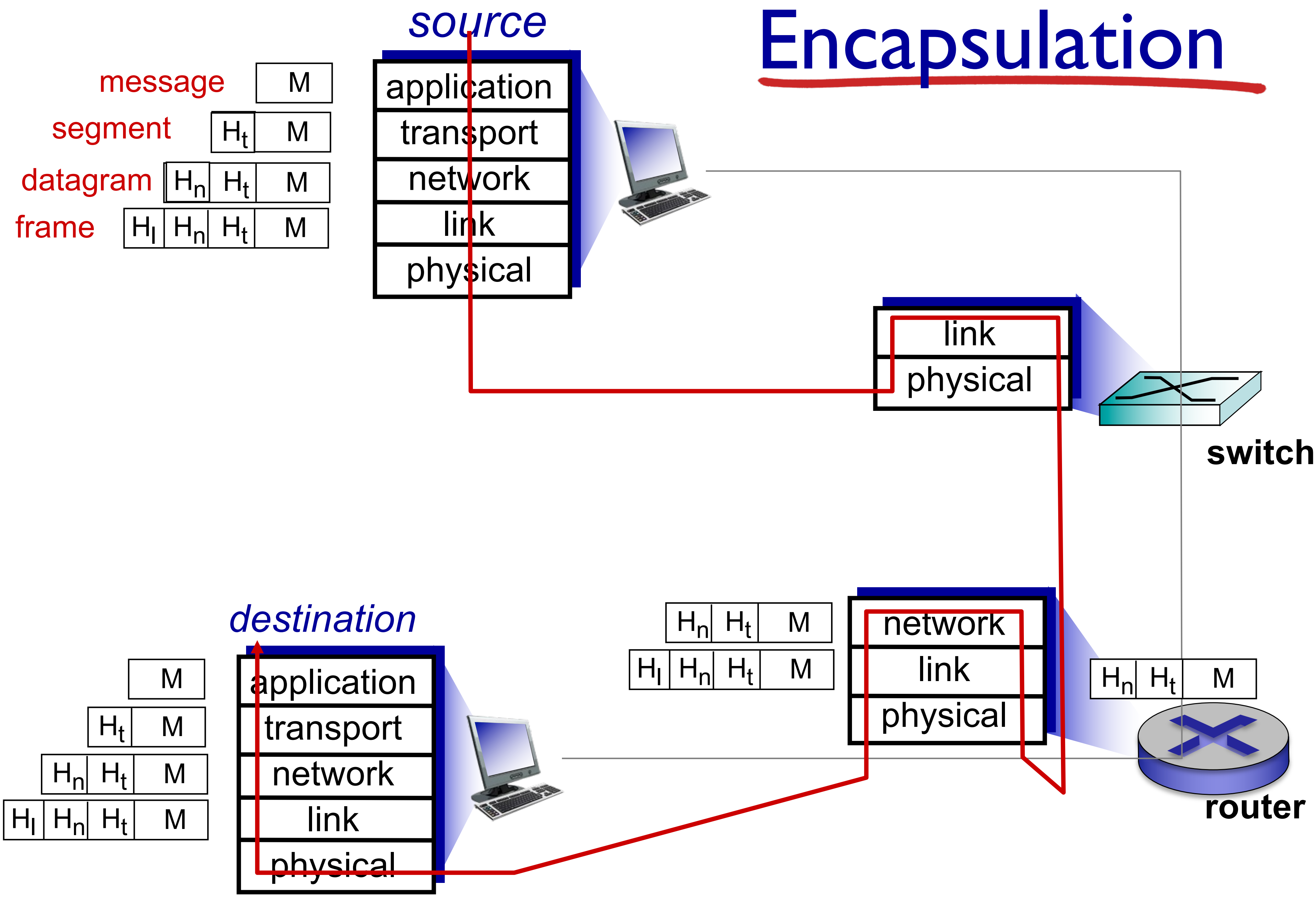


ISO/OSI reference model

- *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?



Encapsulation



Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

Network security

- field of network security:
  - how bad guys can attack computer networks
  - how we can defend networks against attacks
  - how to design architectures that are immune to attacks
- 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!

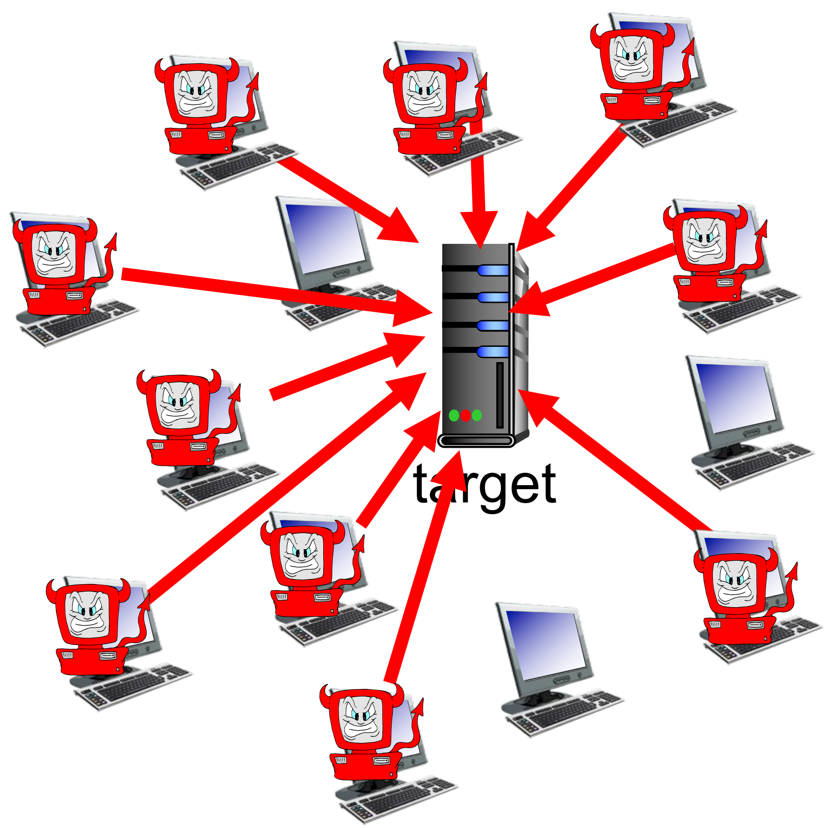
Bad guys: put malware into hosts via Internet

- 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, DDoS attacks

Bad guys: attack server, network infrastructure

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

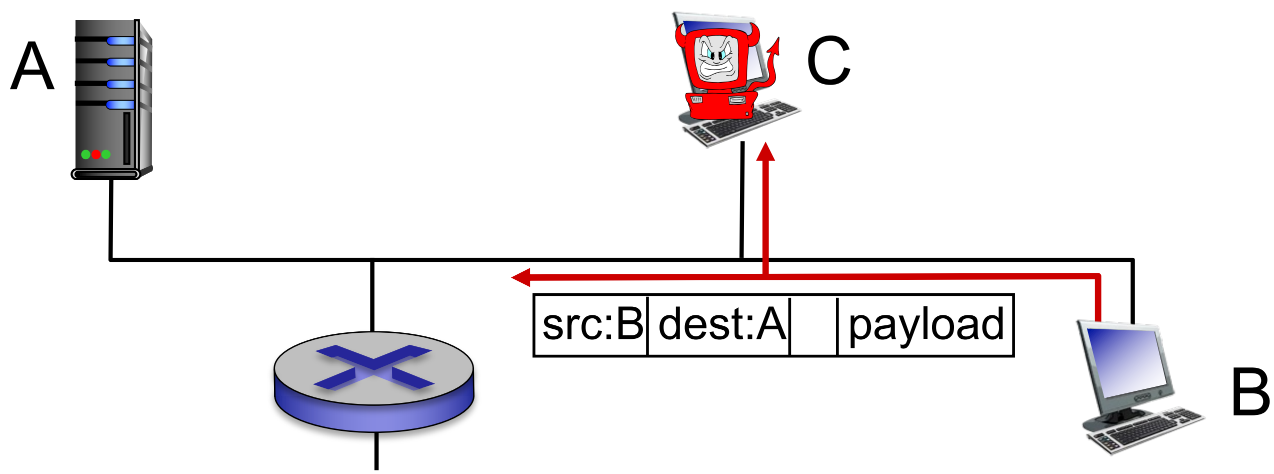
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Bad guys can sniff packets

packet "sniffing":

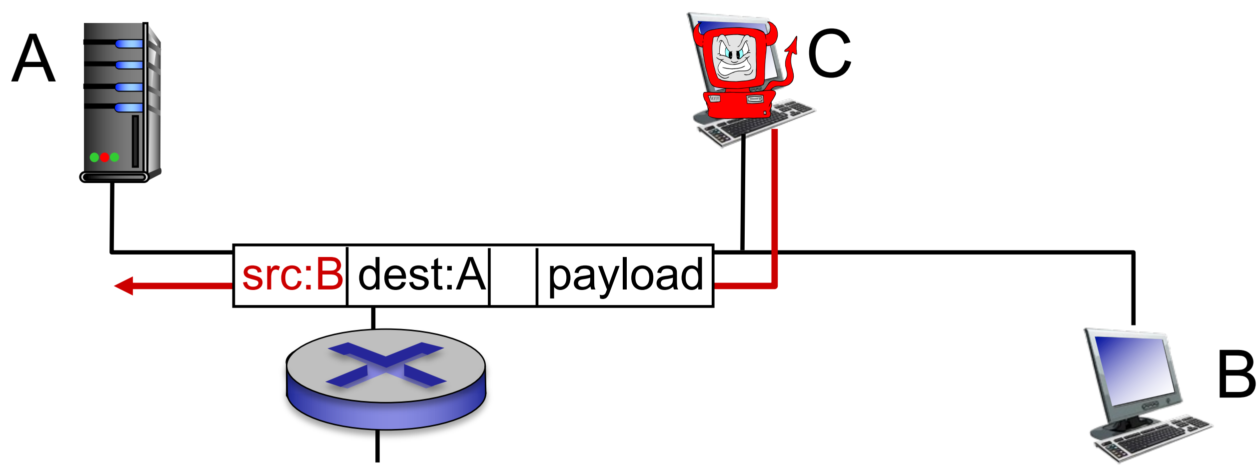
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- wireshark software used for labs is a (free) packet-sniffer

Bad guys can use fake addresses

IP spoofing: send packet with false source address



… lots more on security

Introduction: summary

covered a "ton" 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
- security

you now have:

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