IGRUS Winter Bootcamp
Computer Network Ch01

전기전자공학부 22 윤여준

Chapter 1: introduction

Chapter goal:

- Get "feel," "big picture," introduction to terminology
 - more depth, detail *later* in course

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
- Protocol layers, service models
- Security
- History

The Internet: a "nuts and bolts" view

Billions of connected computing *devices*:

- *hosts = end systems*
- running *network apps* at Internet's "edge"
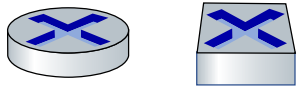
Packet switches: forward packets (chunks of data)
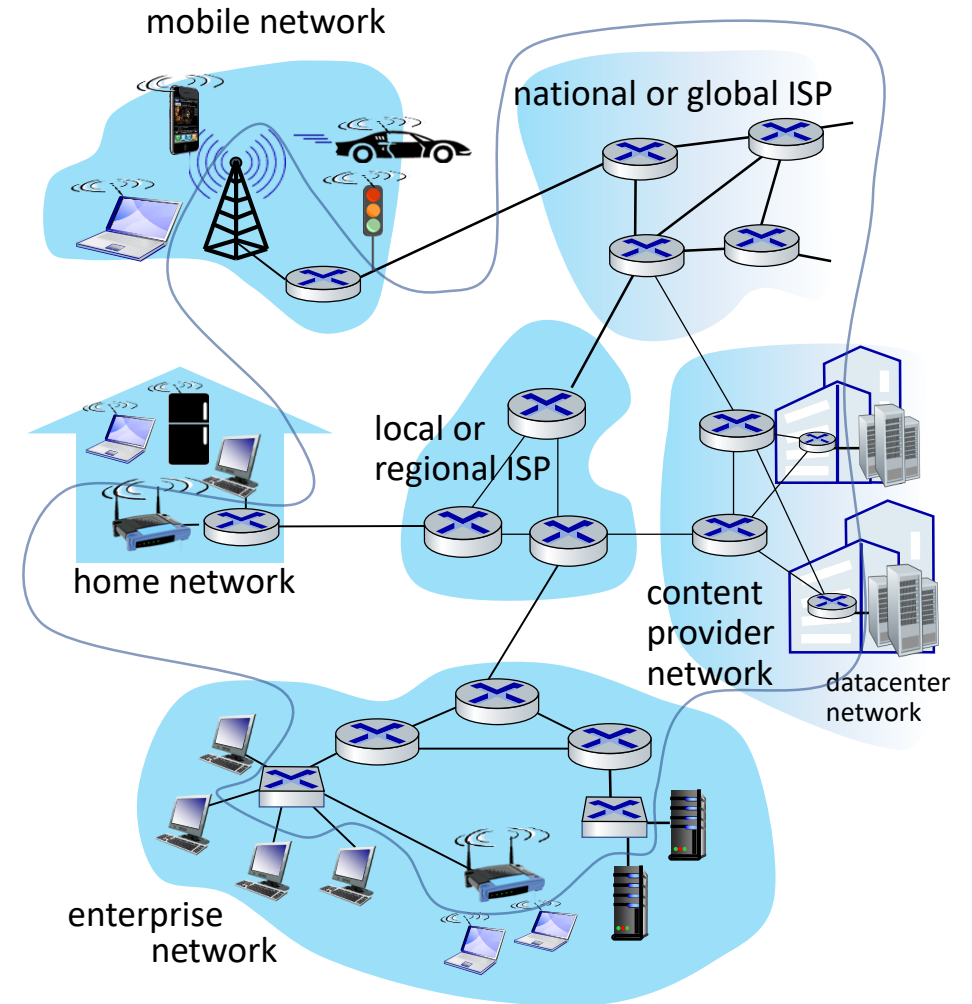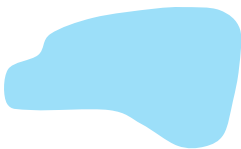
- *routers, switches*

Communication links

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

Networks

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

mobile network

national or global ISP

local or regional ISP

home network

content provider network

datacenter network

enterprise network

"Fun" Internet-connected devices

Amazon Echo

Internet refrigerator

IP picture frame

Pacemaker & Monitor

Tweet-a-watt: monitor energy use

bikes

Web-enabled toaster + weather forecaster

Security Camera

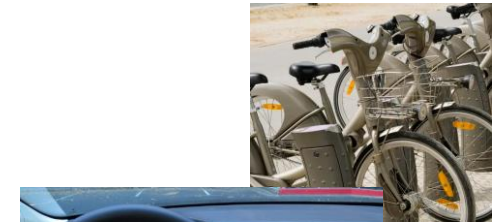Slingbox: remote control cable TV

cars

scooters

AR devices

Internet phones

Gaming devices

sensorized, bed mattress

Fitbit

diapers
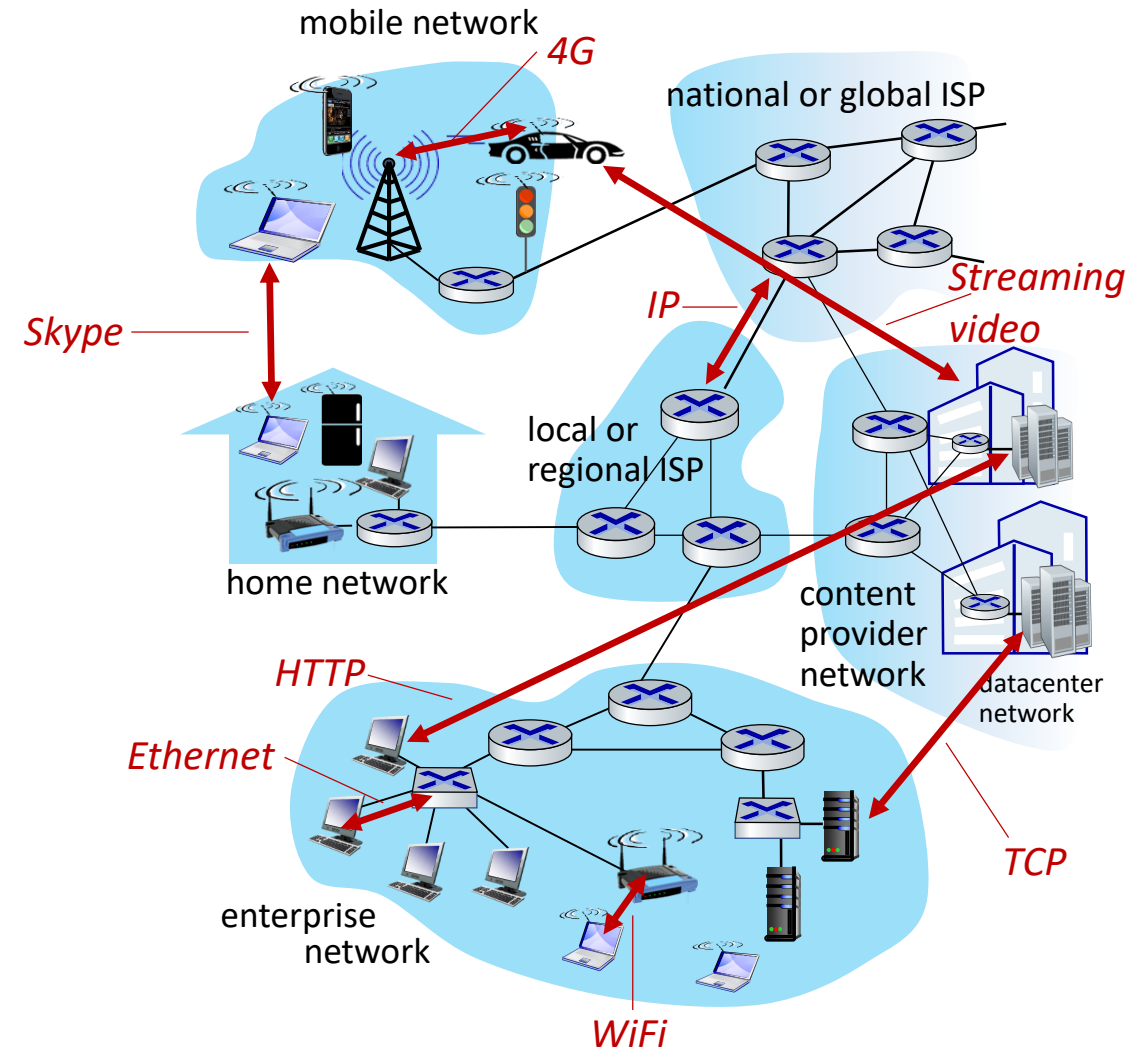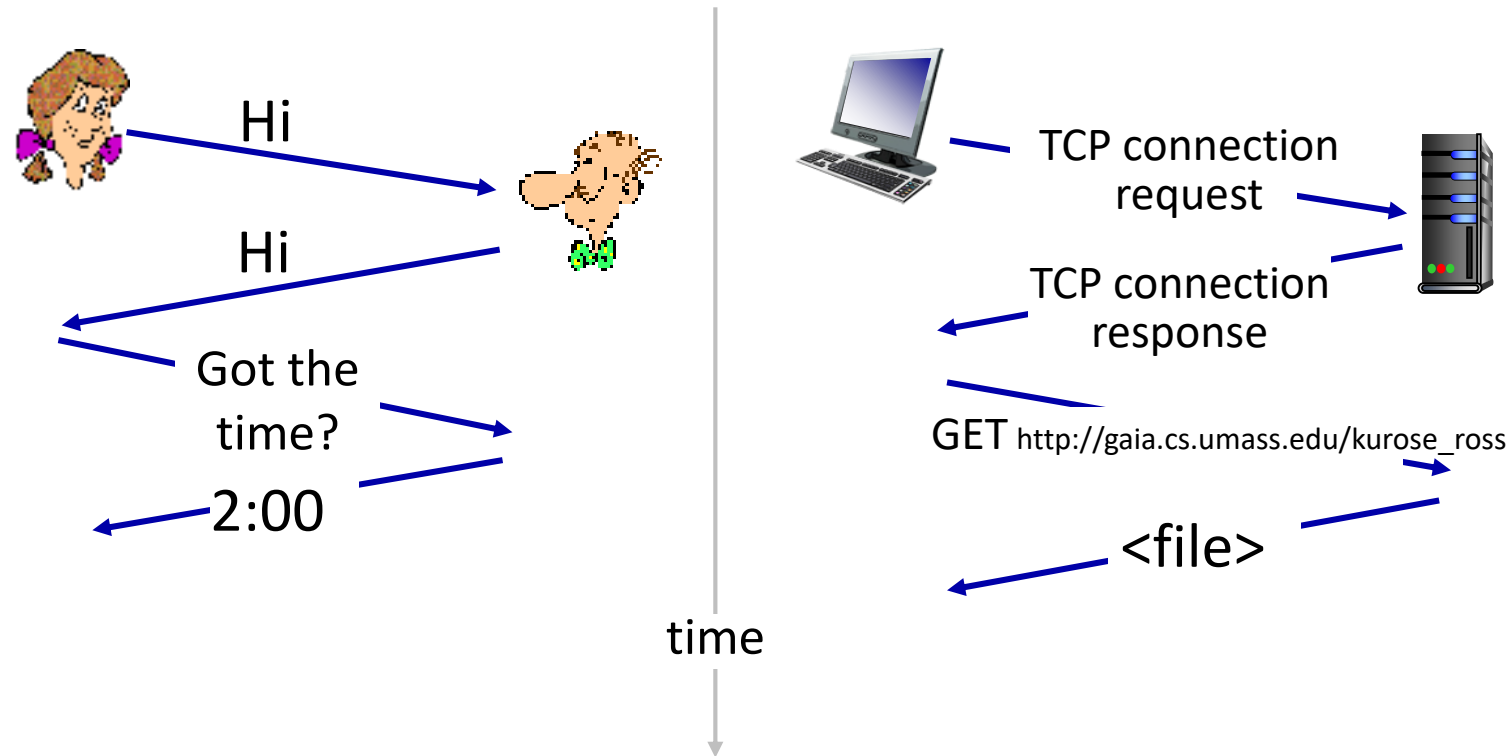
The Internet: a "nuts and bolts" view

- *Internet:* "network of networks"
  - Interconnected ISPs

- *protocols* are *everywhere*
  - control sending, receiving of messages
  - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4/5G, Ethernet

- *Internet standards*
  - RFC: Request for Comments
  - IETF: Internet Engineering Task Force

What's a protocol?

A human protocol and a computer network protocol:

Hi

Hi

Got the time?

2:00

TCP connection request

TCP connection response

GET http://gaia.cs.umass.edu/kurose_ross

<file>

time

Q: other human protocols?

What's a protocol?

Human protocols:

- "what's the time?"
- "I have a question"
- introductions

Rules for:

... 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 message transmission, receipt
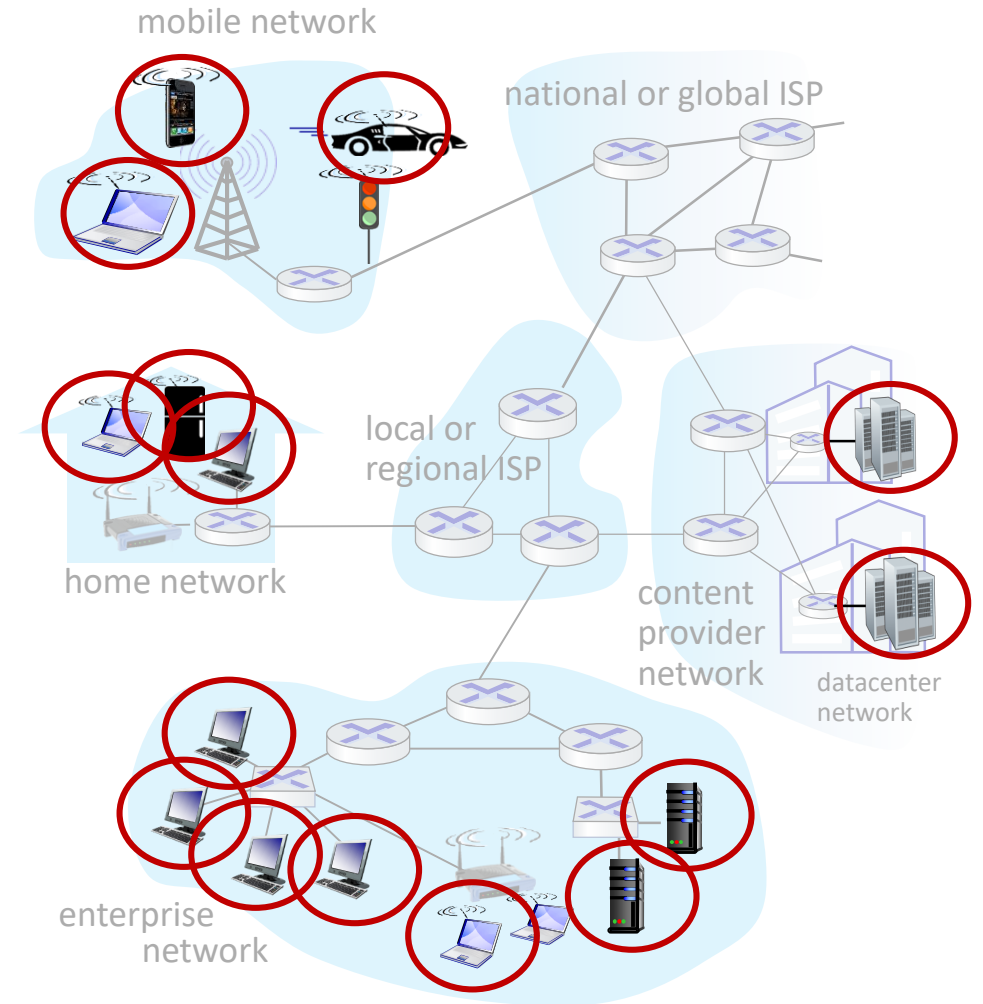
Chapter 1: roadmap

- 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

mobile network

national or global ISP

local or regional ISP

home network

content provider network

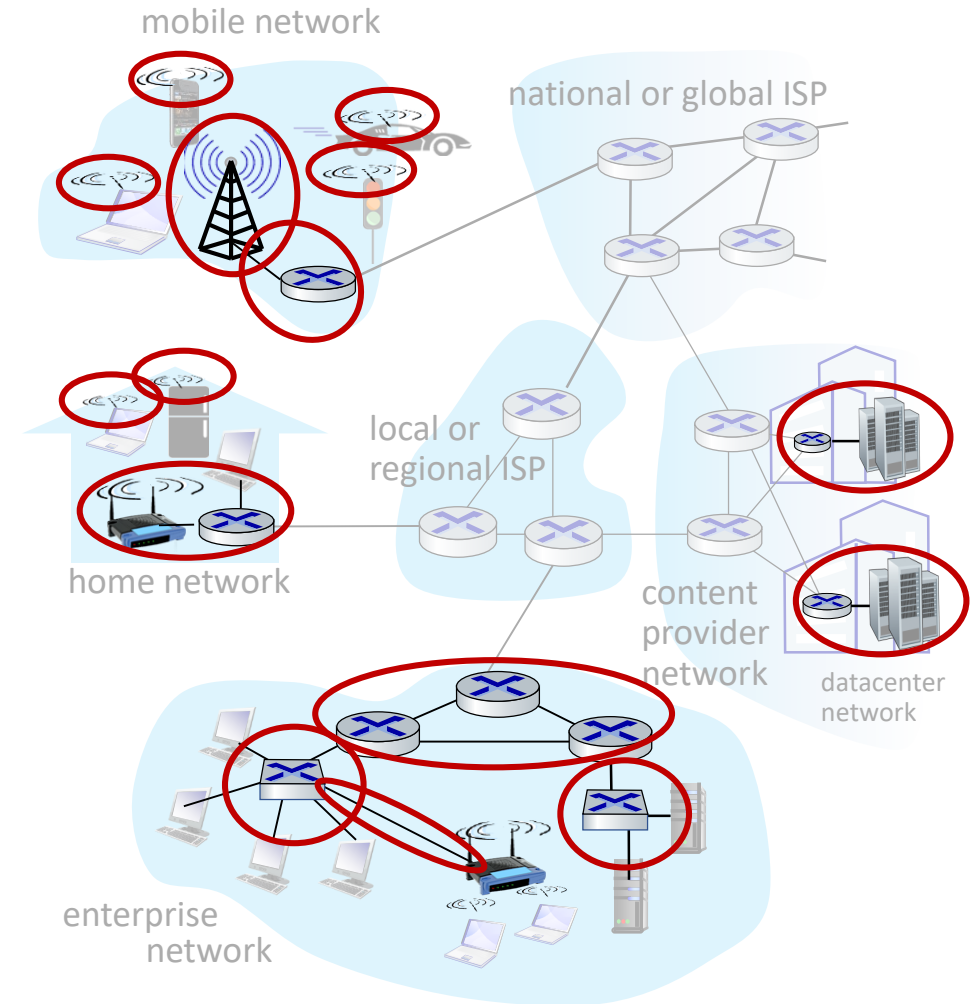datacenter network

enterprise network

A closer look at Internet structure
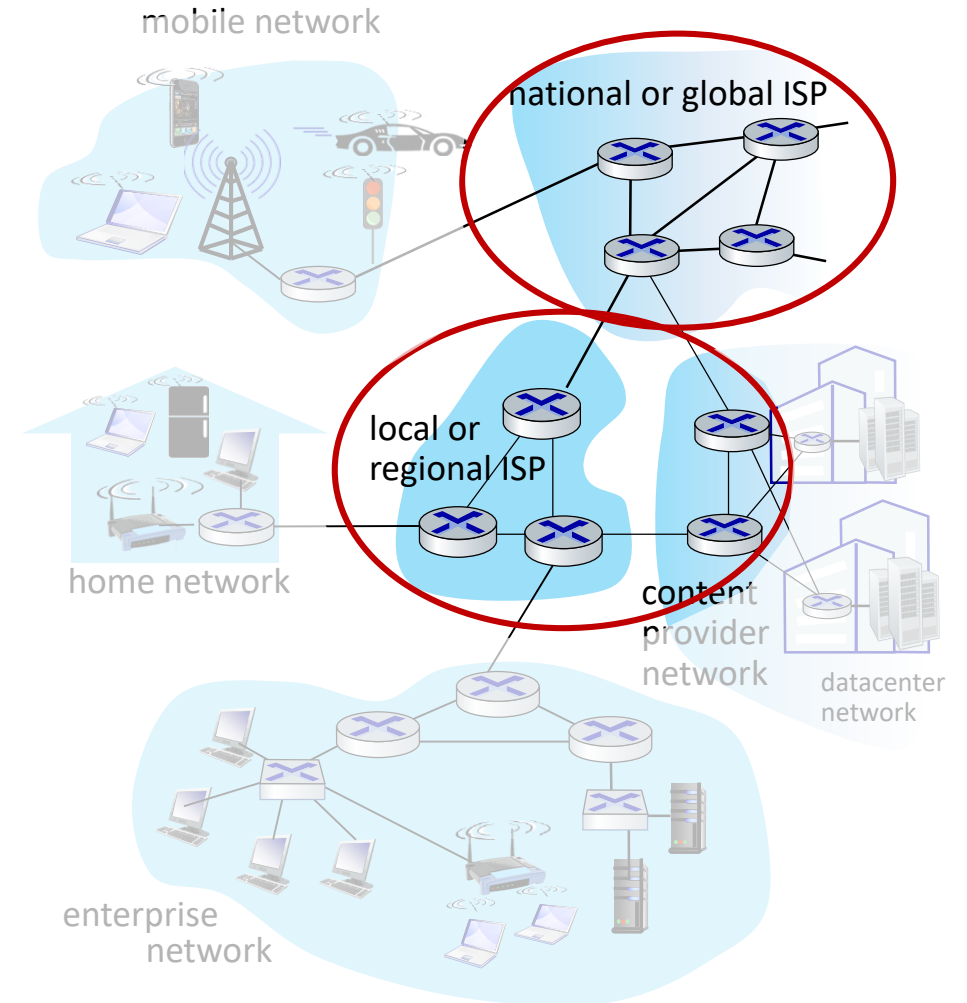
Network edge:
- hosts: clients and servers
- servers often in data centers

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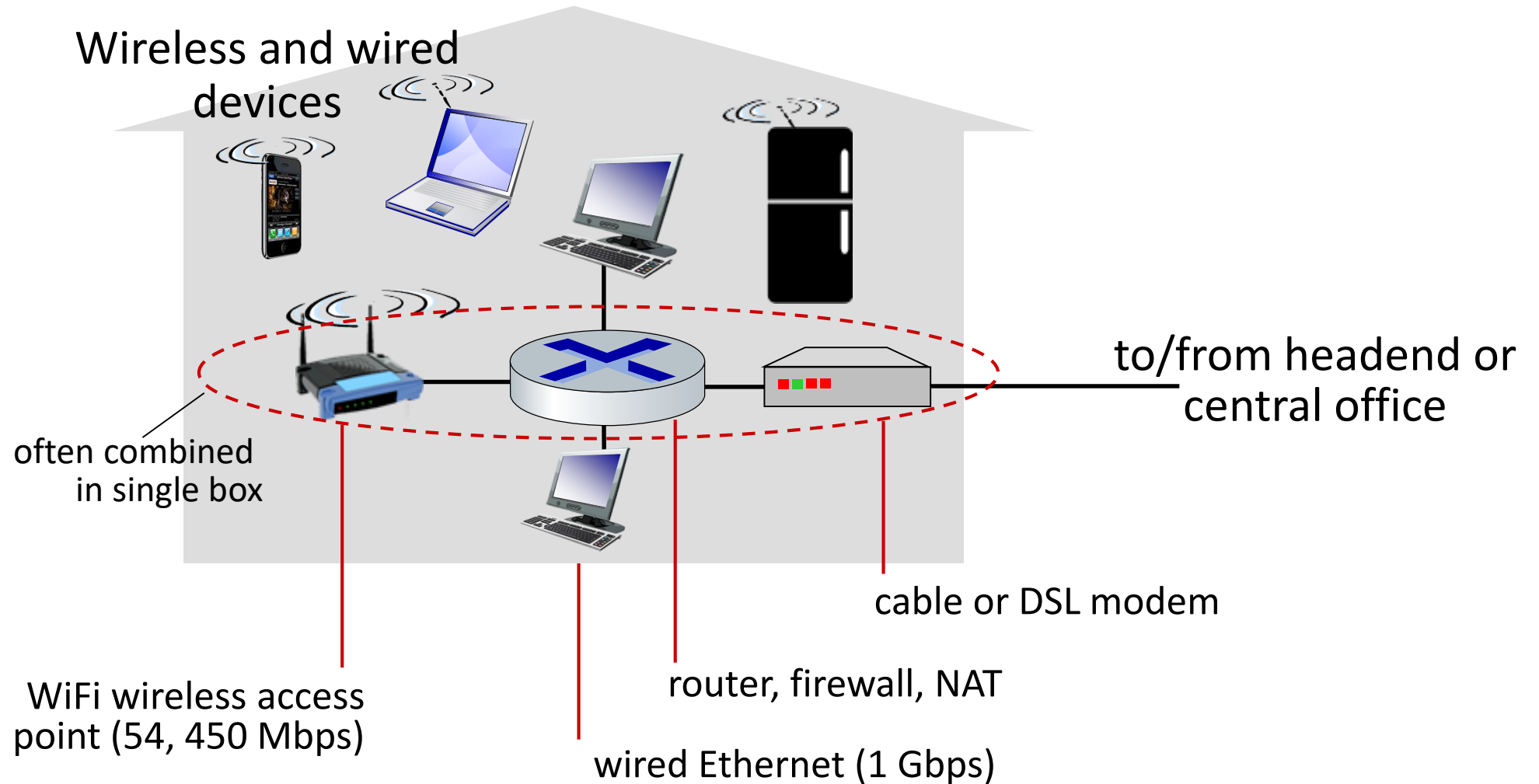- wired, wireless communication links

Network core:
- interconnected routers
- network of networks

mobile network

national or global ISP

local or regional ISP

home network

content provider network

datacenter network

enterprise network

Access networks: home networks

Wireless and wired devices

often combined in single box

to/from headend or central office

WiFi wireless access point (54, 450 Mbps)

router, firewall, NAT

wired Ethernet (1 Gbps)

cable or DSL modem

Access networks: enterprise networks

Enterprise link to
ISP (Internet)
institutional router

Ethernet switch

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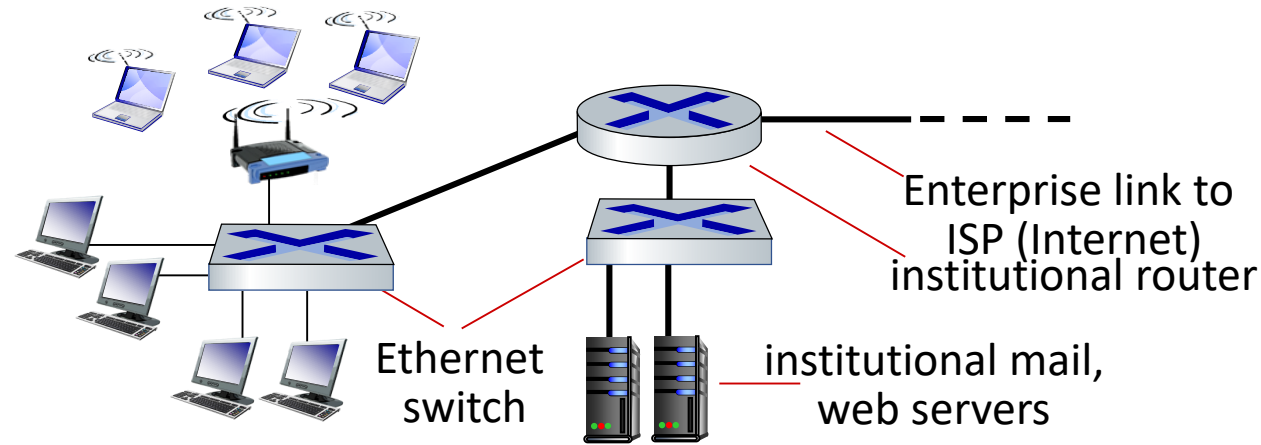
- companies, universities, etc.
- mix of wired, wireless link technologies, connecting a mix of switches and routers (we'll cover differences shortly)
  - Ethernet: wired access at 100Mbps, 1Gbps, 10Gbps
  - WiFi: wireless access points at 11, 54, 450 Mbps

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*

two packets, *L* bits each

host

R: link transmission rate
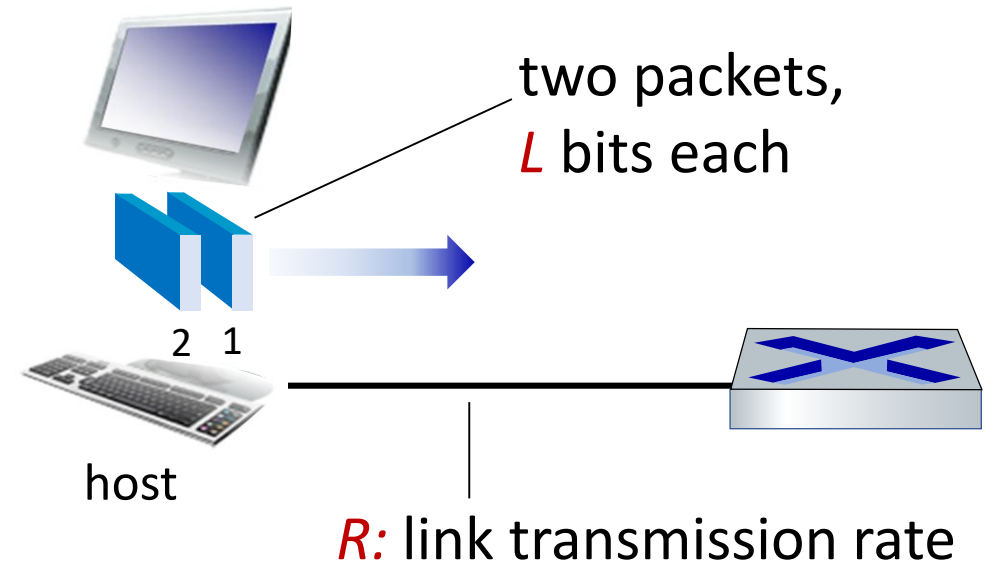
$$\text{packet transmission delay} = \text{time needed to transmit } L\text{-bit packet into link} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$
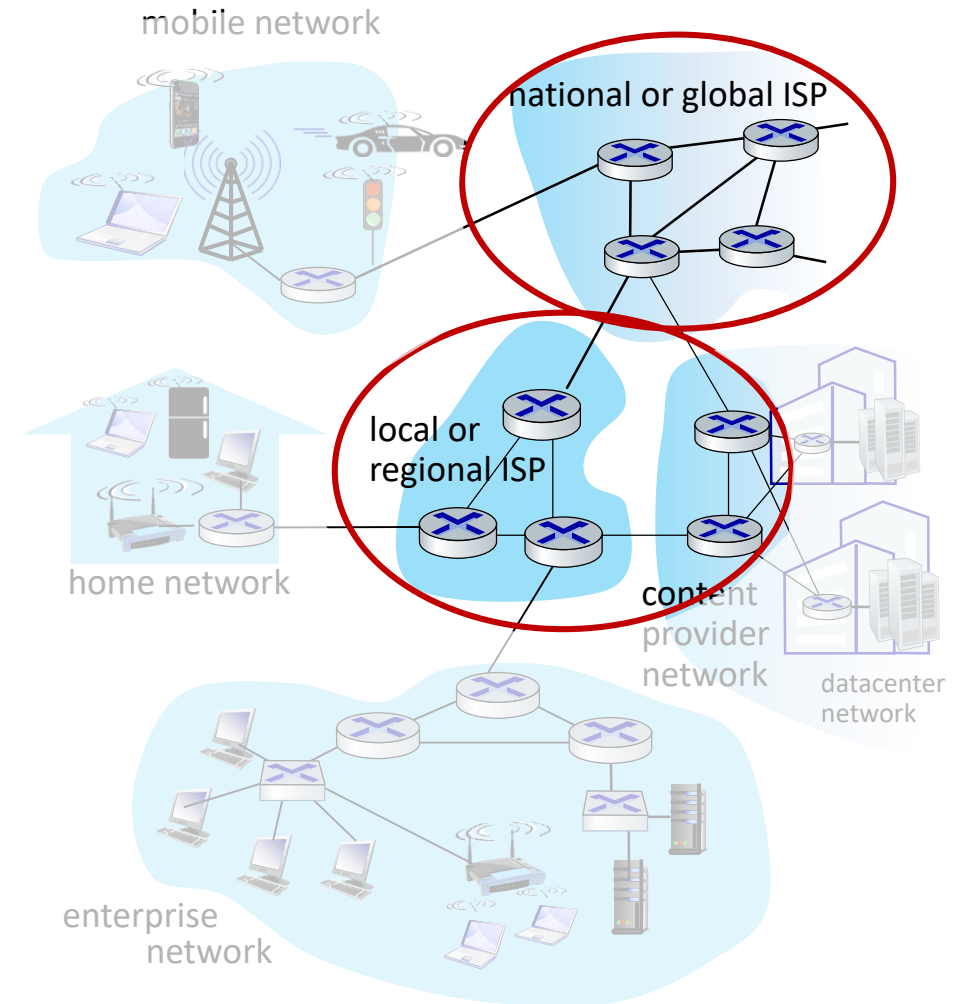
Chapter 1: roadmap

- 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
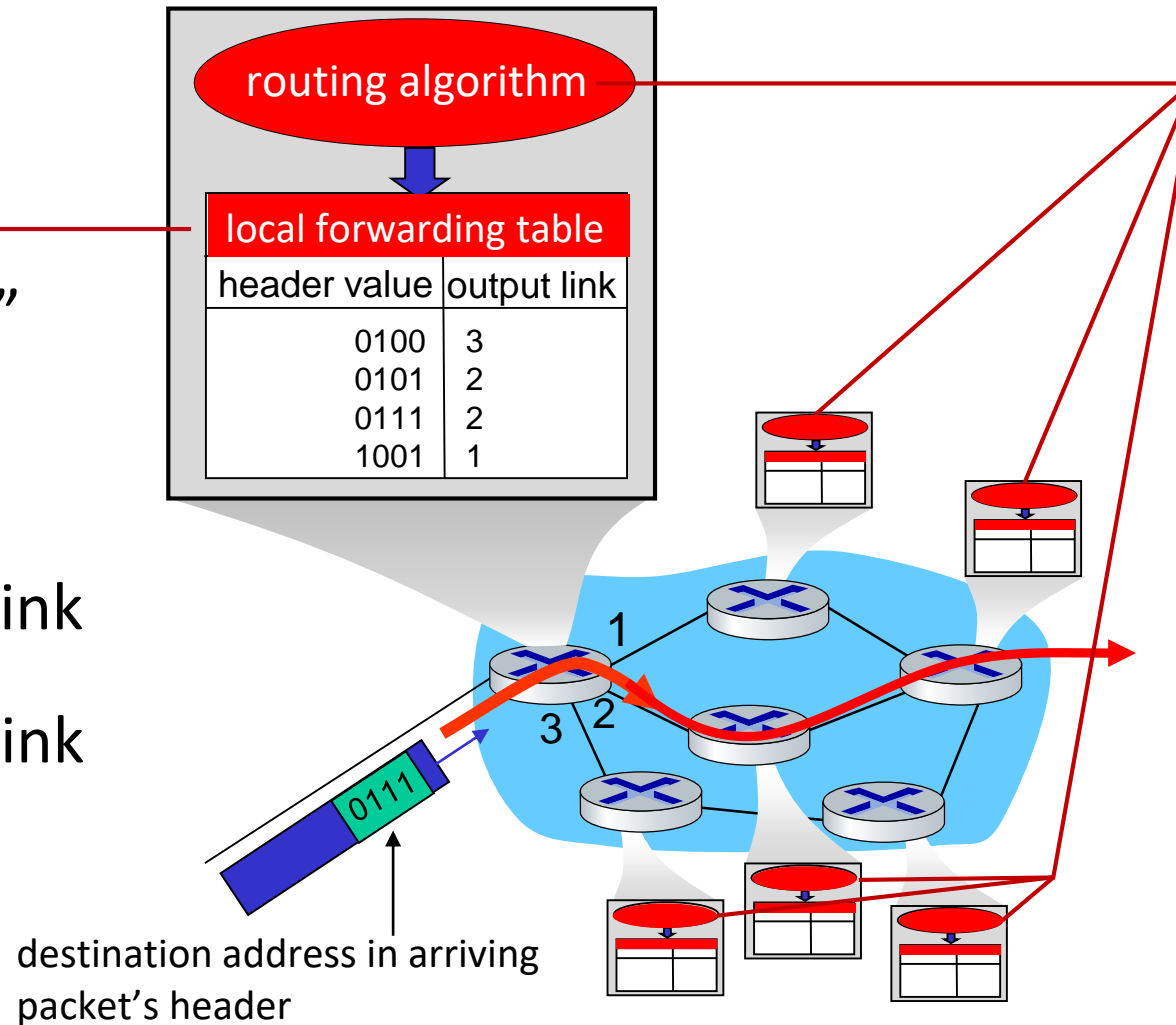- History

The network core

- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
  - network forwards packets from one router to the next, across links on path from source to destination

mobile network

national or global ISP

local or regional ISP

home network

content provider network

datacenter network

enterprise network

Two key network-core functions

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

routing algorithm

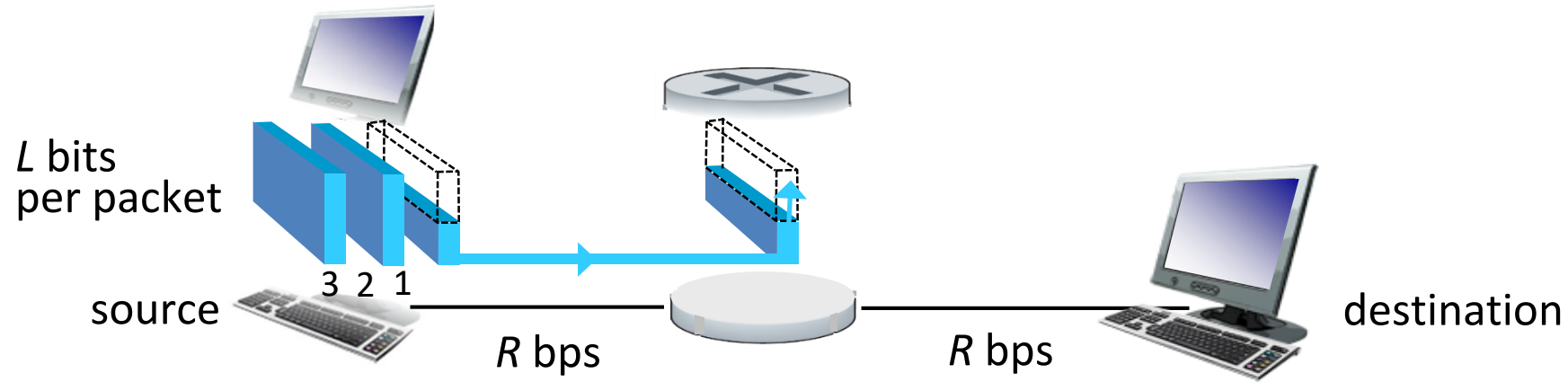| local forwarding table	
header value	output link
0100	3
0101	2
0111	2
1001	1

destination address in arriving packet's header

0111

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

routing

forwarding

forwarding

Packet-switching: store-and-forward

- **packet 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

One-hop numerical example:
- L = 10 Kbits
- R = 100 Mbps
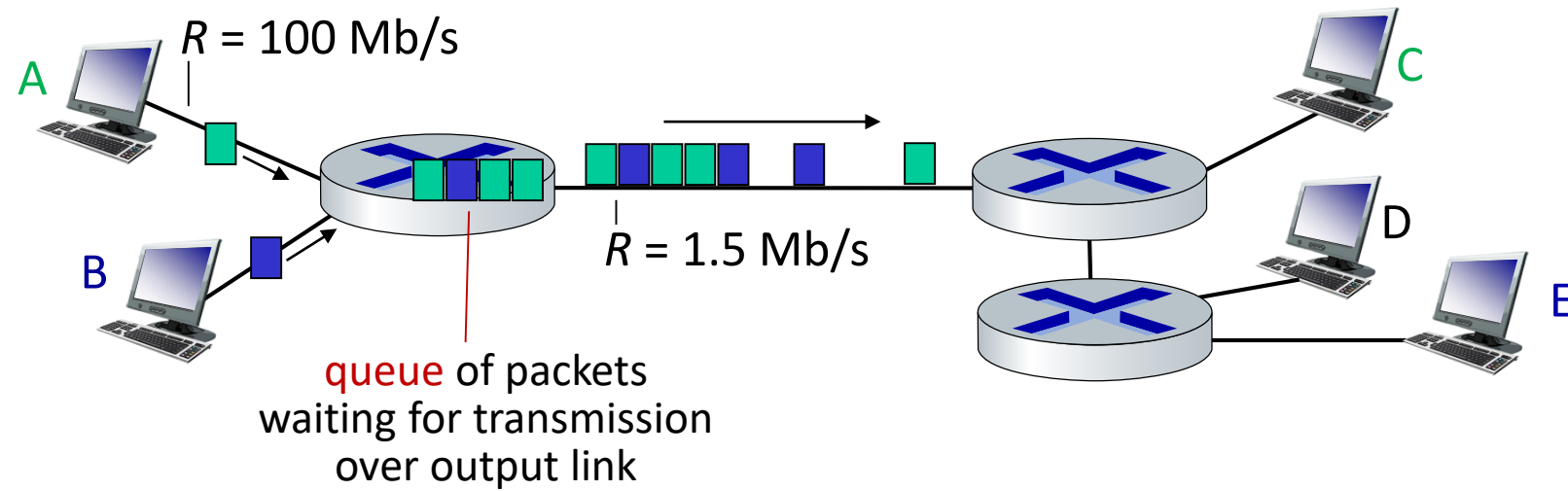- one-hop transmission delay = 0.1 msec

Packet-switching: queueing

R = 100 Mb/s

A

R = 1.5 Mb/s

B

C

D

E

queue of packets
waiting for transmission
over output link
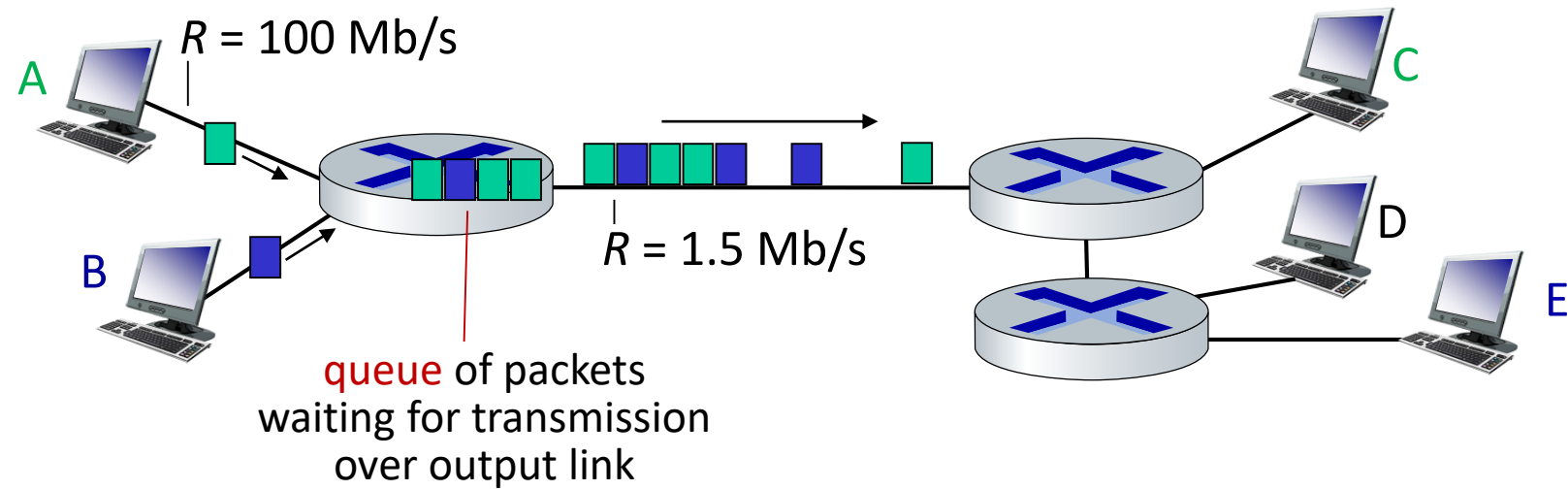
Queueing occurs when work arrives faster than it can be serviced:

Packet-switching: queueing

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

Internet structure: a "network of networks"

- hosts connect to Internet via access Internet Service Providers (ISPs)

- access ISPs in turn must be interconnected
  - so that *any* two hosts *(anywhere!)* can send packets to each other

- resulting network of networks is very complex
  - evolution driven by economics, national policies
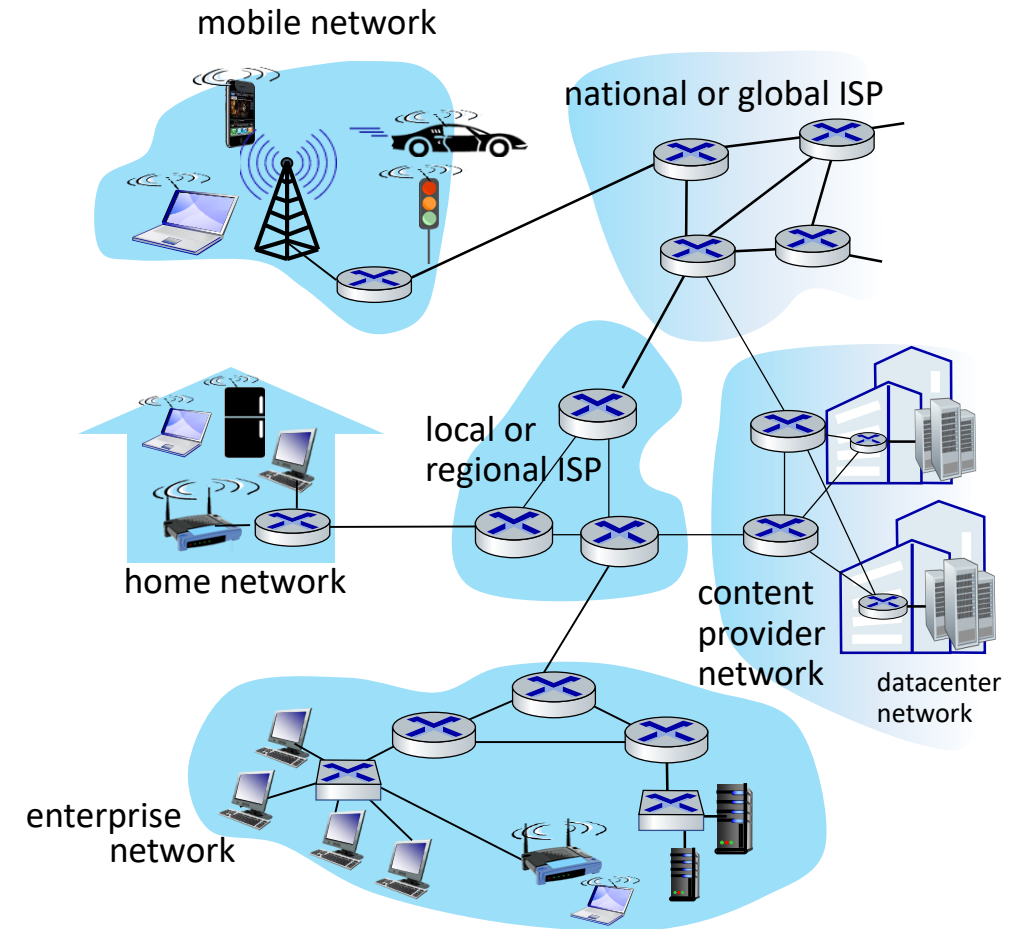
Let's take a stepwise approach to describe current Internet structure

Internet structure: a "network of networks"
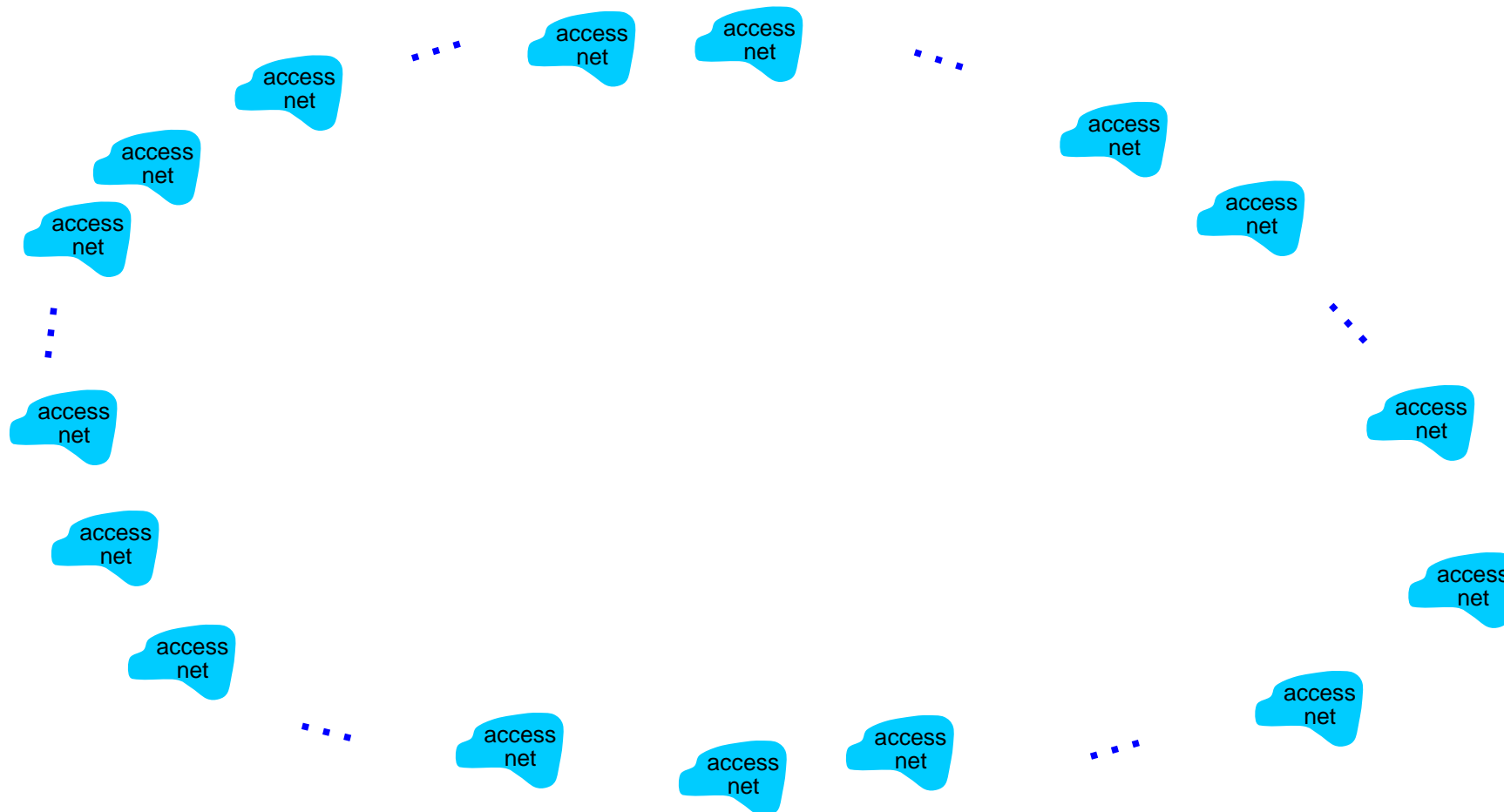
Question: given *millions* of access ISPs, how to connect them together?

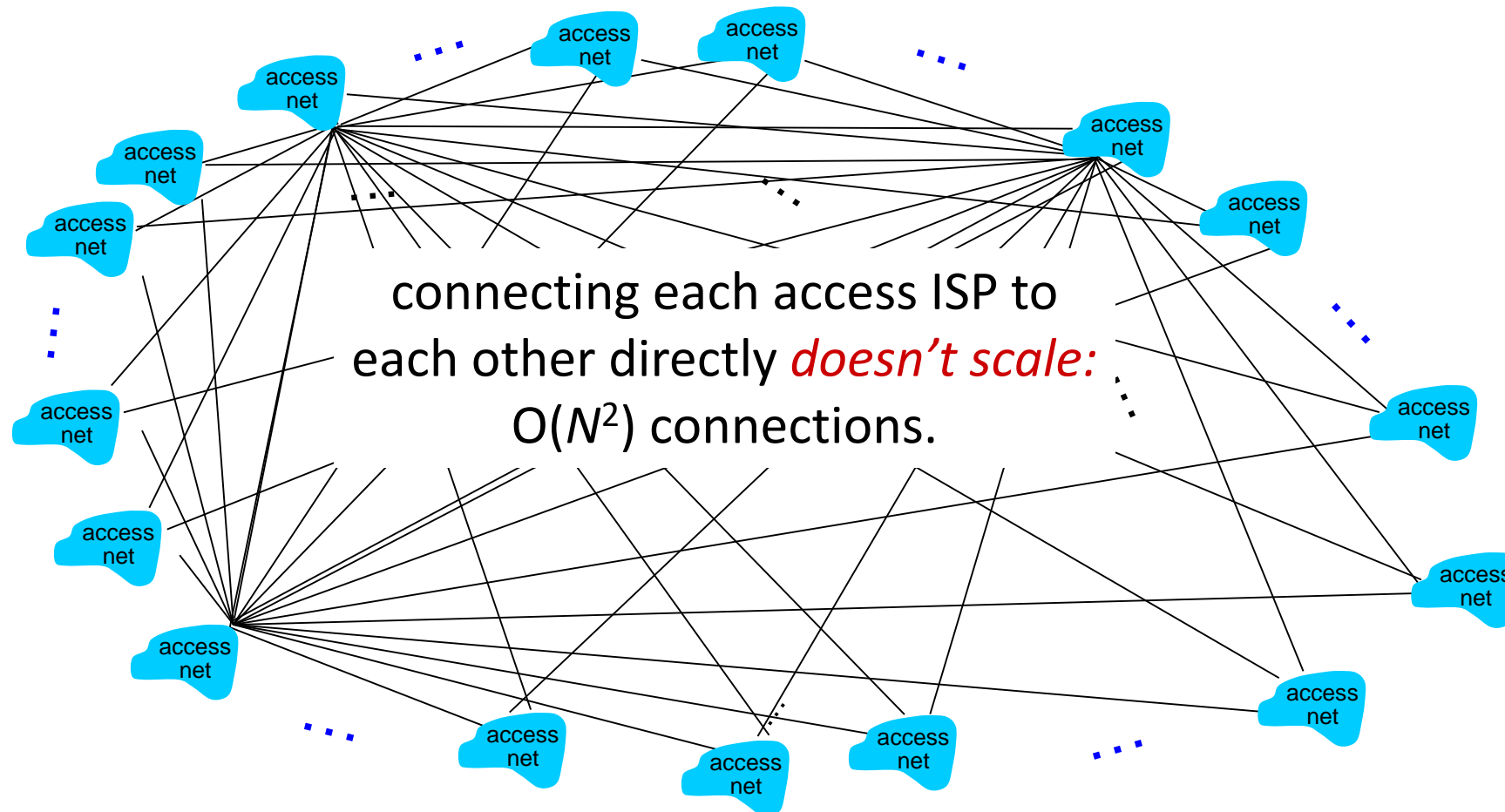# Internet structure: a "network of networks"

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

connecting each access ISP to each other directly *doesn't scale:* $O(N^2)$ connections.

Internet structure: a "network of networks"

Option: connect each access ISP to one global transit ISP?
Customer and *provider* ISPs have economic agreement.
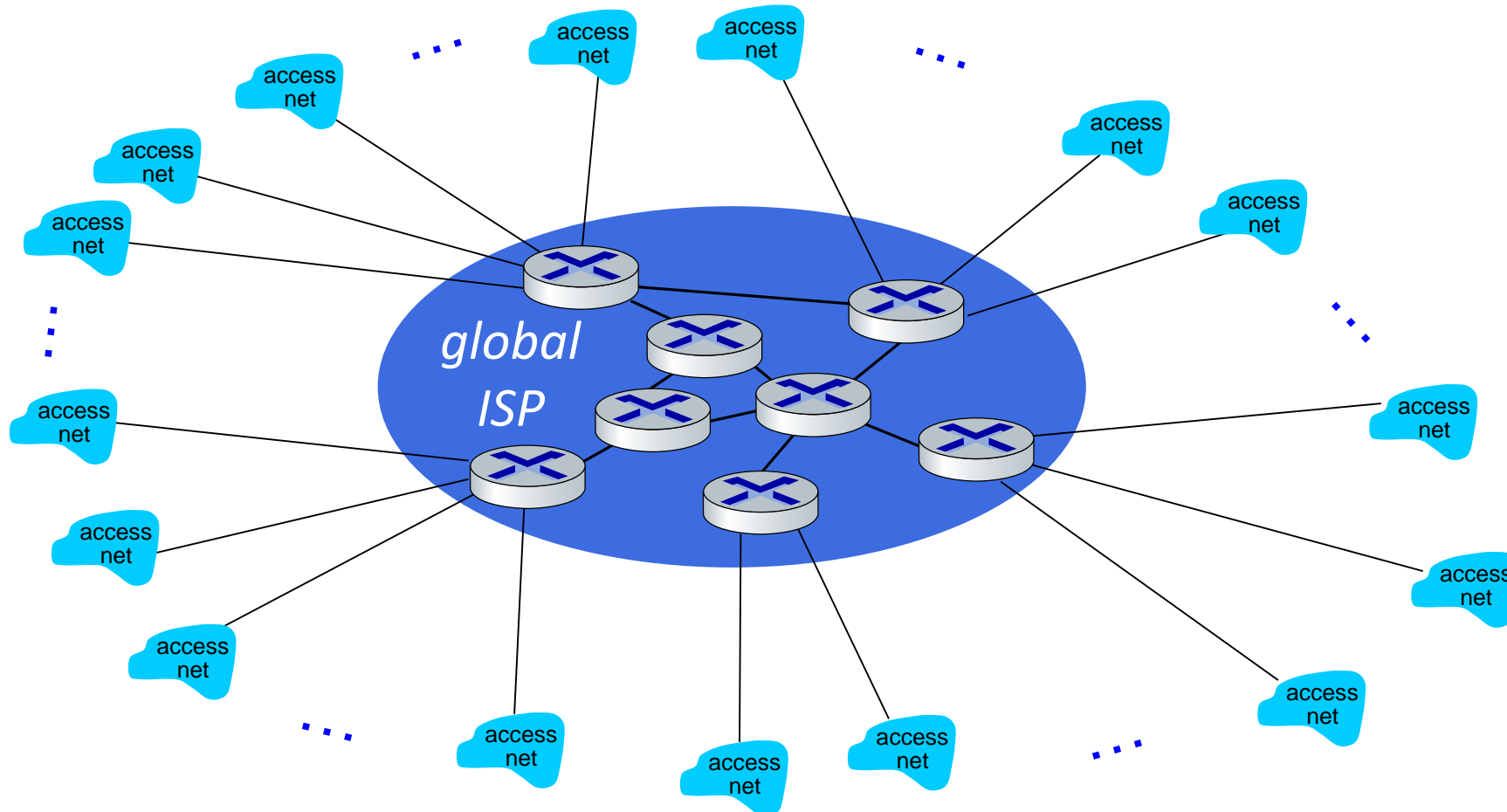
Internet structure: a "network of networks"

But if one global ISP is viable business, there will be competitors ….
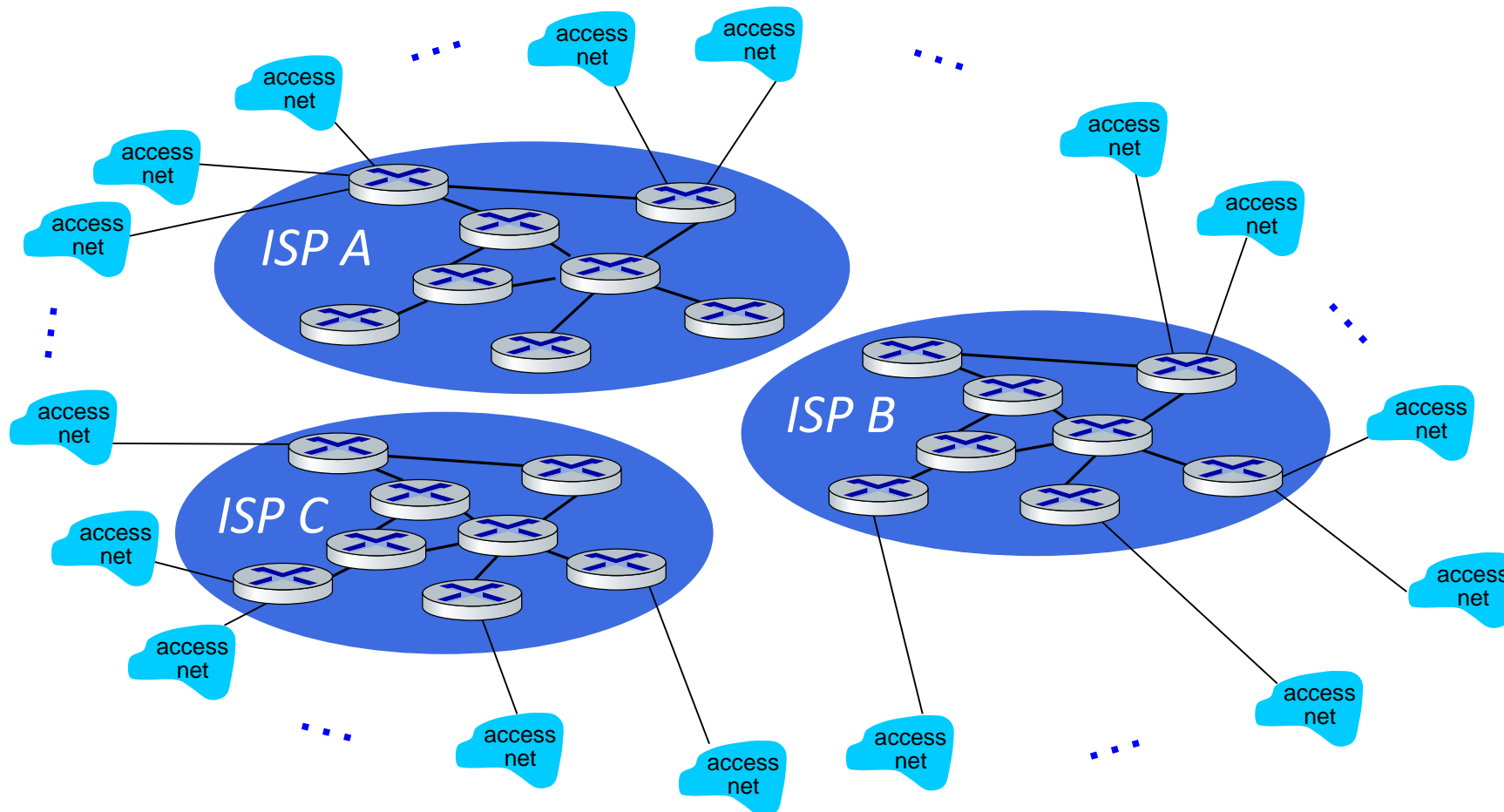
Internet structure: a "network of networks"

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

Internet exchange point

peering link
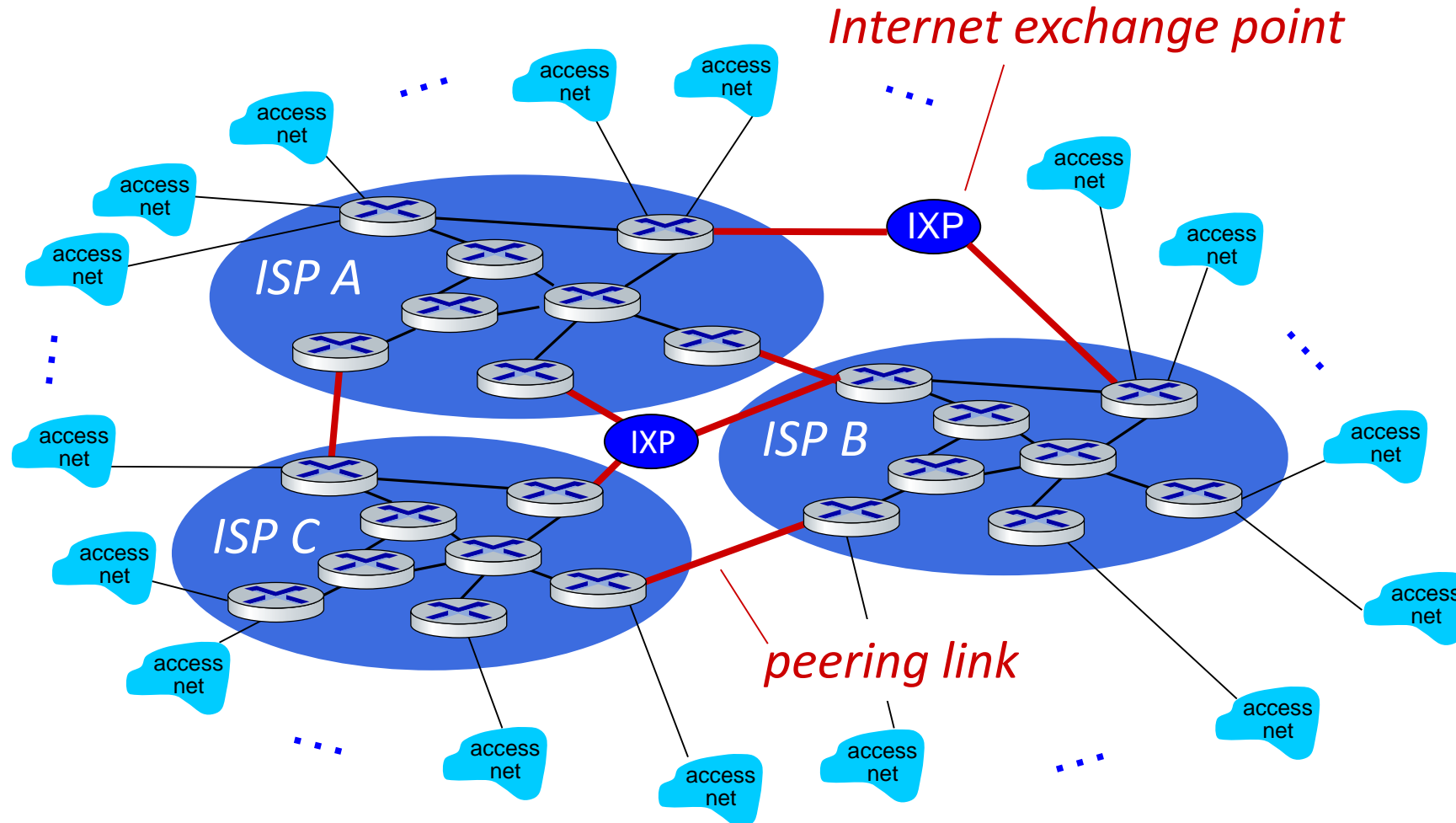
Internet structure: a "network of networks"

... and regional networks may arise to connect access nets to ISPs
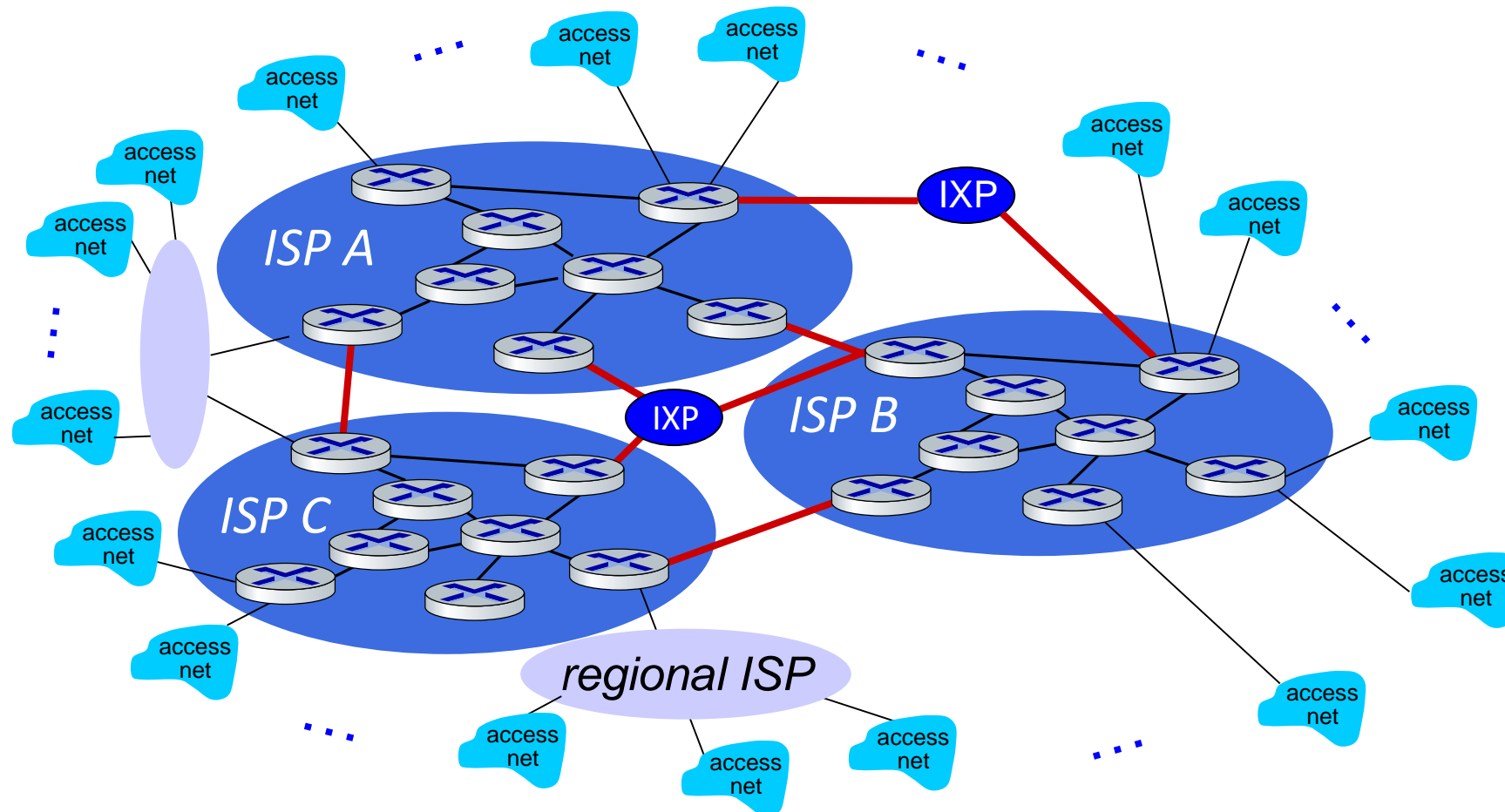
Internet structure: a "network of networks"

... and content provider networks (e.g., Google, Microsoft, Akamai) may run their own network, to bring services, content close to end users
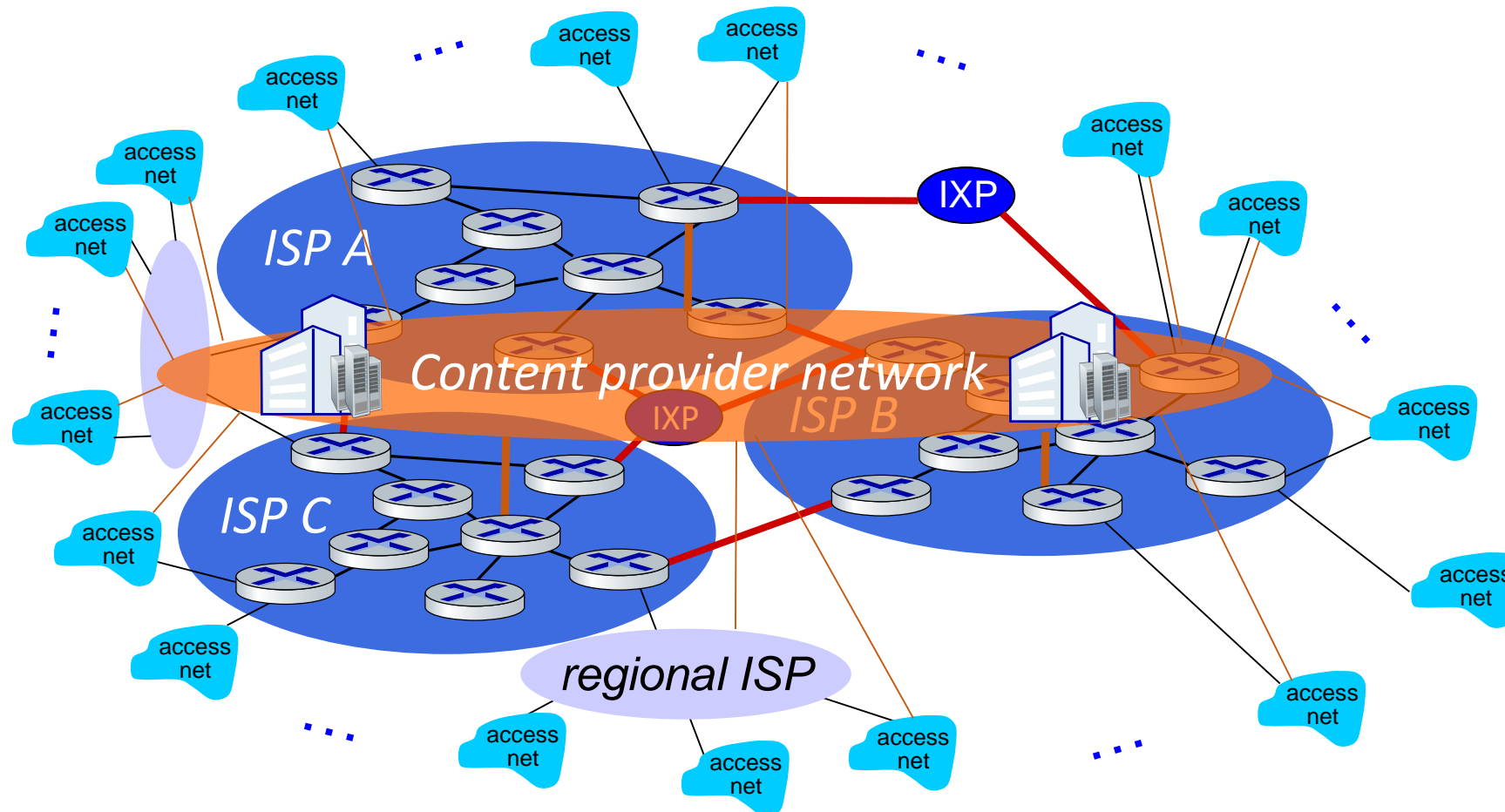
Internet structure: a "network of networks"
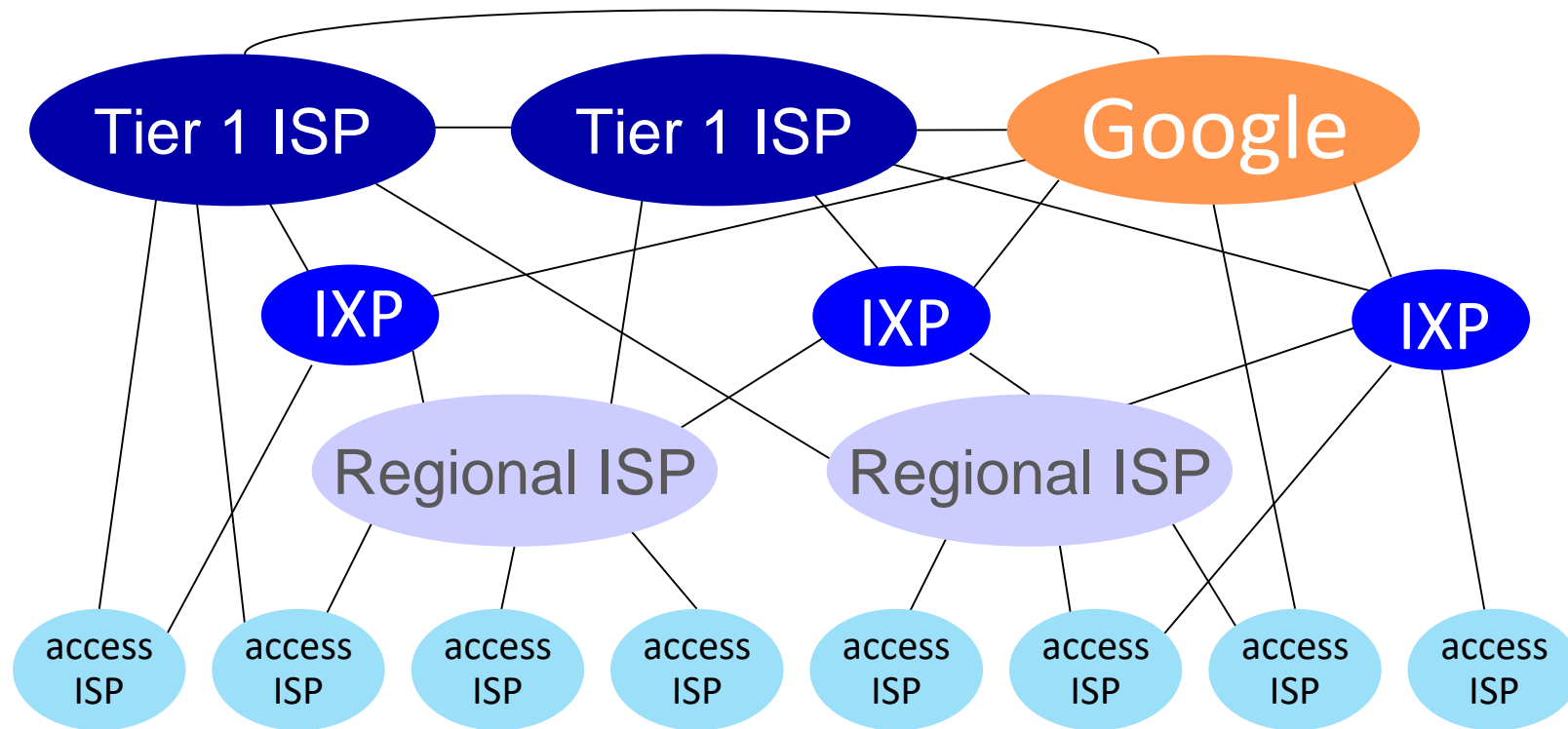
At "center": small # of well-connected large networks

- "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

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 delay and loss occur?

- packets *queue* in router buffers, waiting for turn for transmission
  - queue length grows when arrival rate to link (temporarily) exceeds output link capacity

- packet *loss* occurs when memory to hold queued packets fills up

packet being transmitted (transmission delay)

packets in buffers (queueing delay)

free (available) buffers: arriving packets dropped (loss) if no free buffers
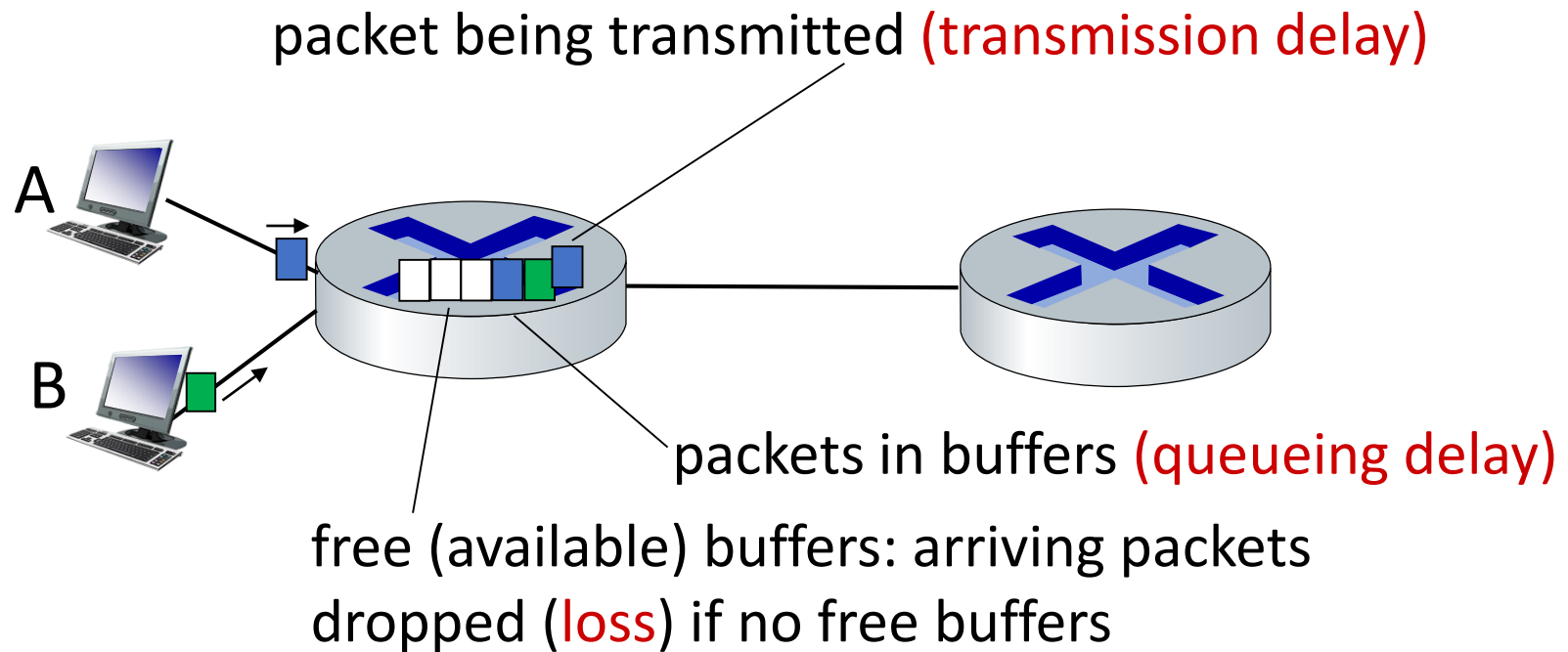
Packet delay: four sources
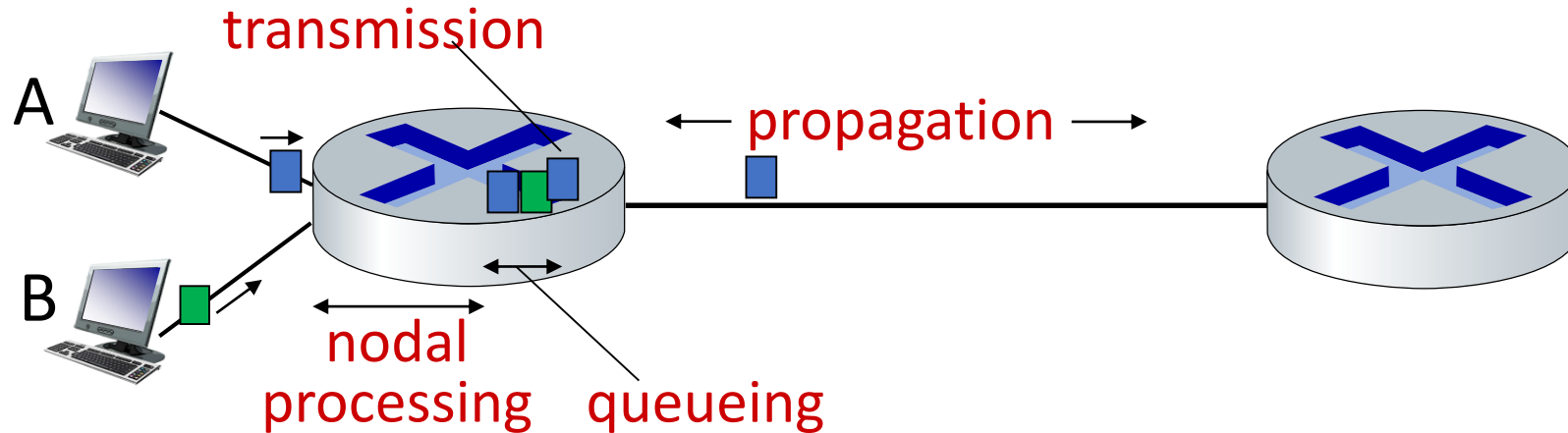
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

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

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

Packet delay: four sources
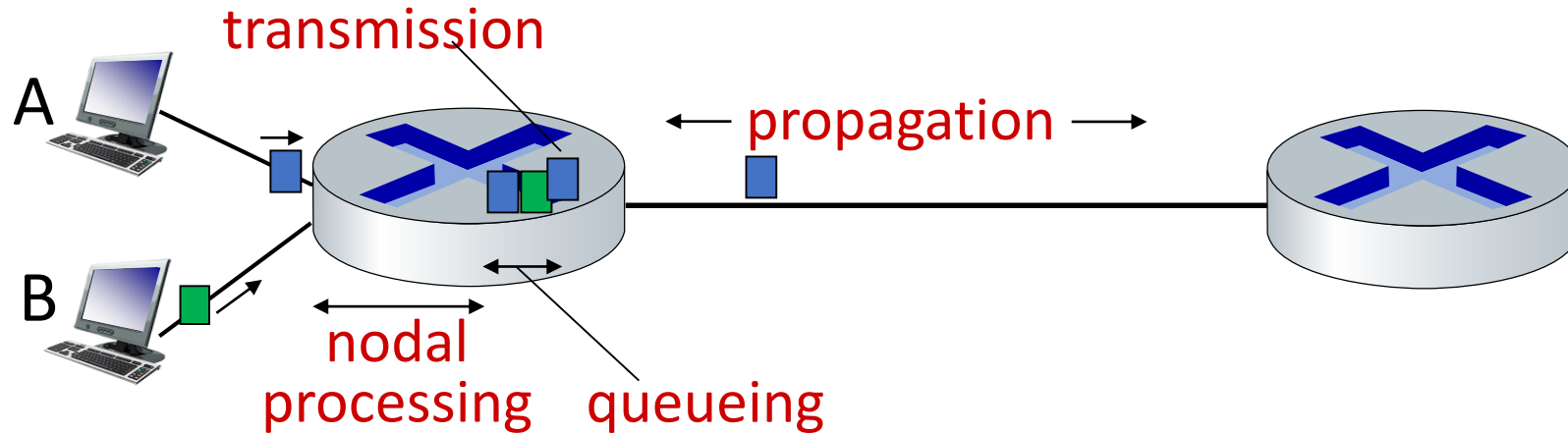
$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

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 (~2×10^8 m/sec)
- $d_{prop} = d/s$

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

Caravan analogy

ten-car caravan
(aka 10-bit packet)

toll booth
(aka link)

100 km

toll booth
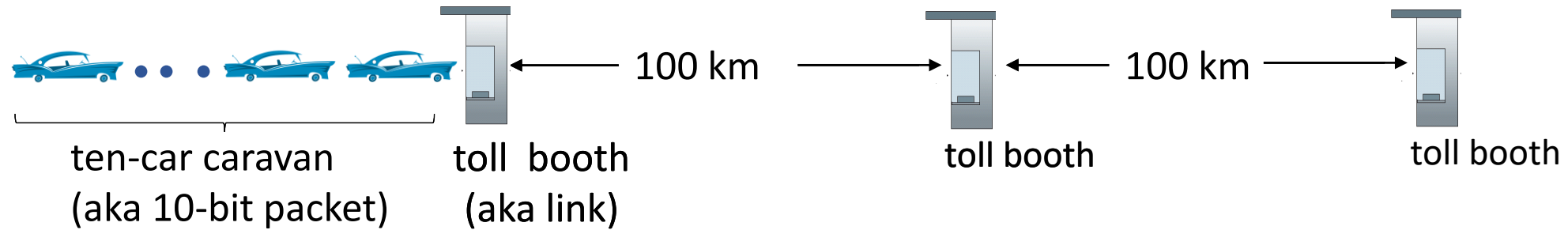
100 km

toll booth

- car ~ bit; caravan ~ packet; toll service ~ link transmission
- toll booth takes 12 sec to service car (bit transmission time)
- "propagate" at 100 km/hr
- *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

ten-car caravan
(aka 10-bit packet)

toll booth
(aka router)

toll booth

100 km

100 km
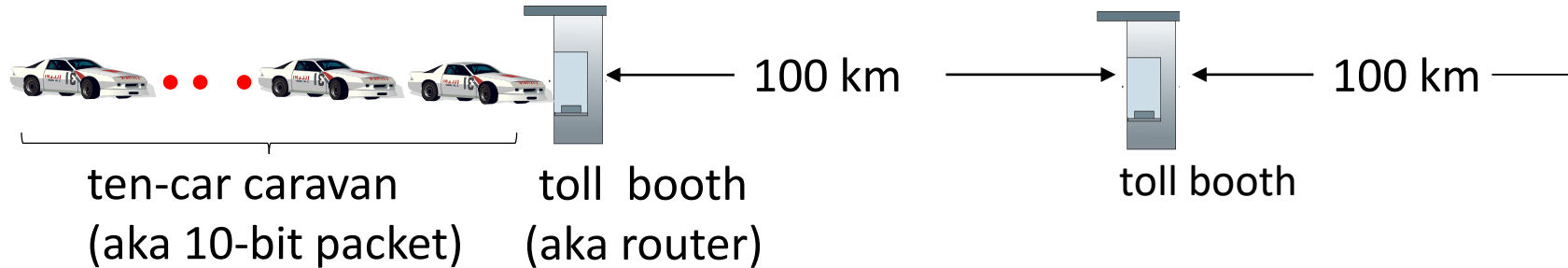
- suppose cars now "propagate" at 1000 km/hr

- 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 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

buffer (waiting area)

packet being transmitted

A

B

packet arriving to full buffer is *lost*
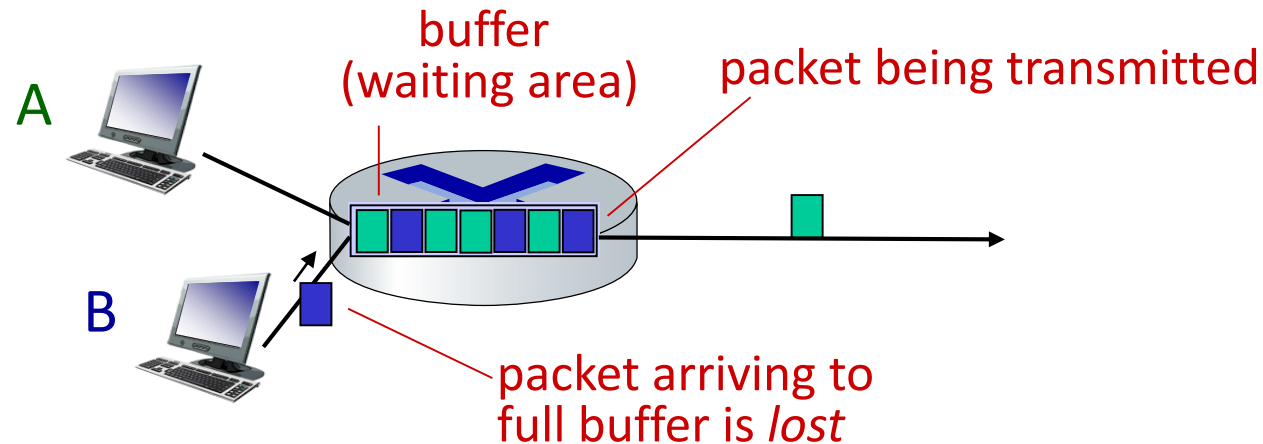
* Check out the Java applet for an interactive animation (on publisher's website) of queuing and loss

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

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)
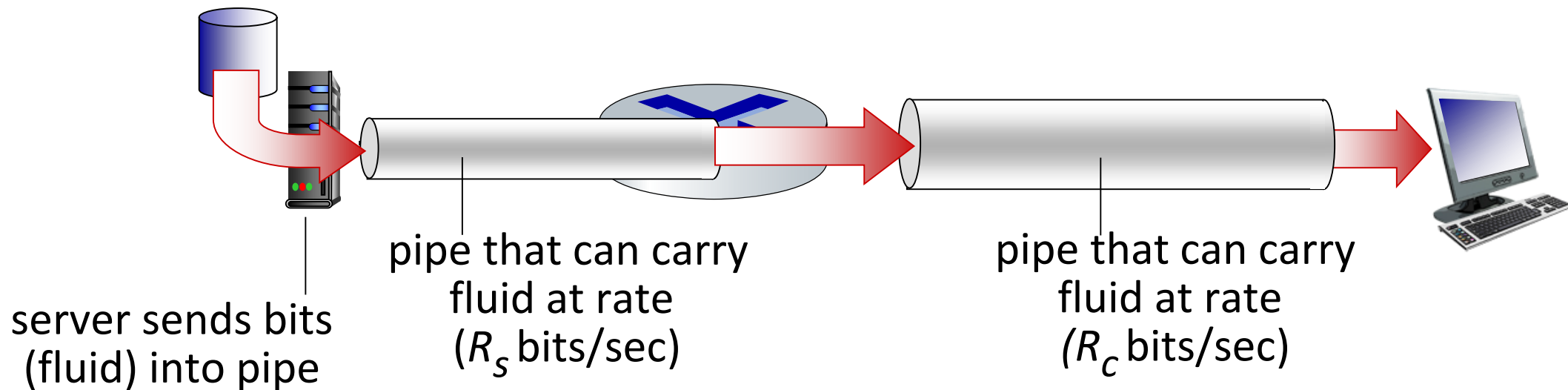
Throughput

$R_s < R_c$ What is average end-end throughput?
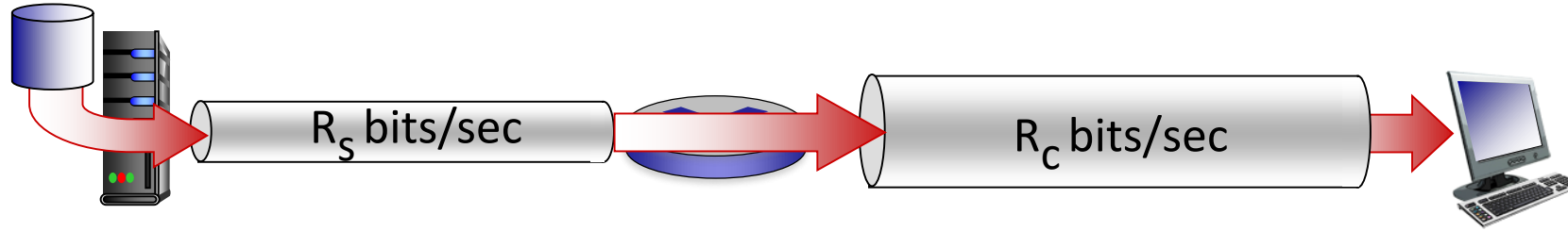
$R_s > R_c$ What is average end-end throughput?
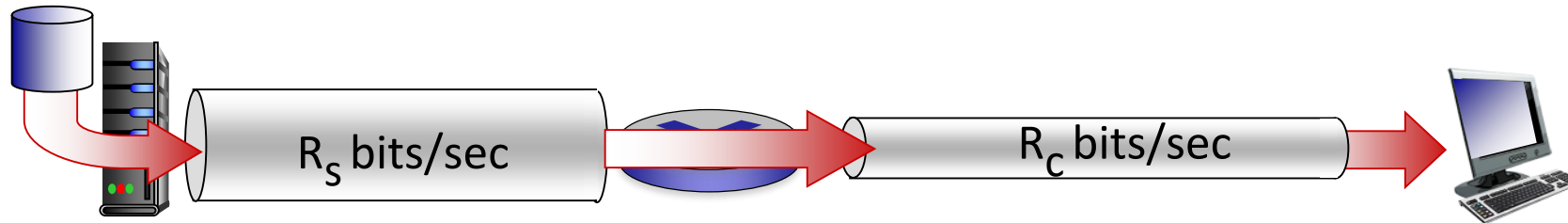
bottleneck link

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

Throughput: network scenario
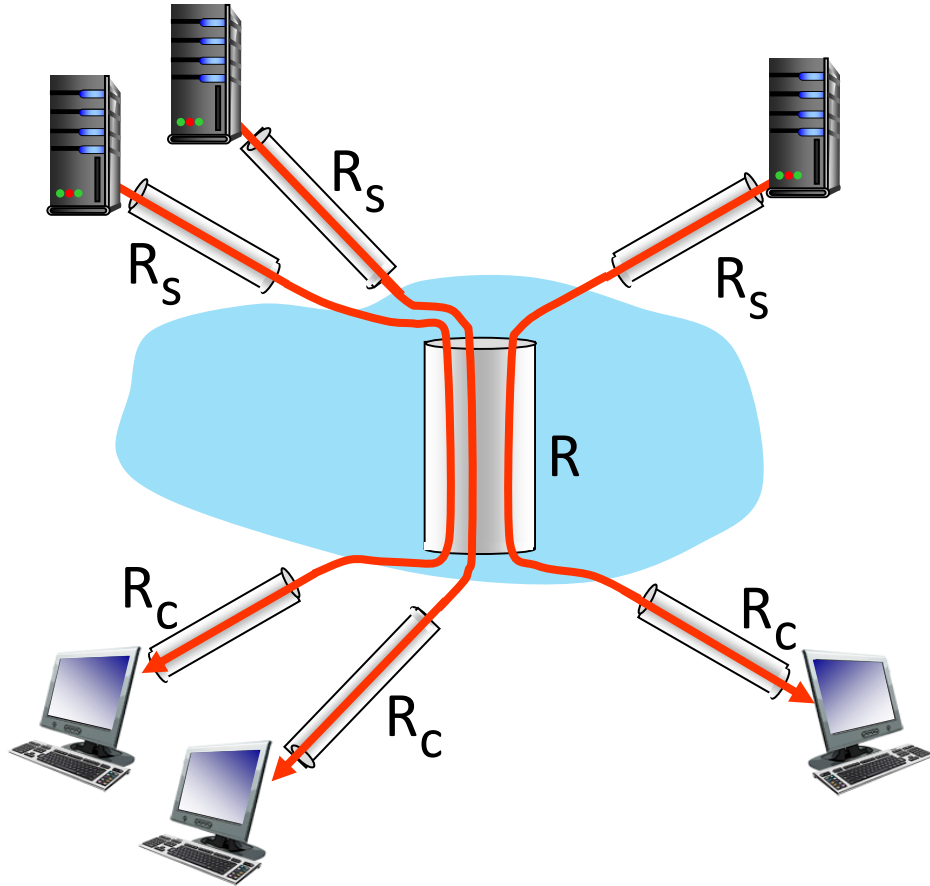
10 connections (fairly) share backbone bottleneck link R bits/sec

- 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/

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?
- and/or our *discussion* of networks?

Example: organization of air travel
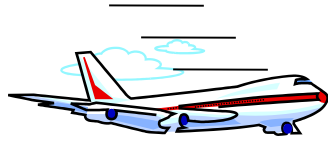
end-to-end transfer of person plus baggage

ticket (purchase)   ticket (complain)

baggage (check)   baggage (claim)

gates (load)   gates (unload)

runway takeoff   runway landing

airplane routing   airplane routing

airplane routing

How would you *define/discuss* the *system* of airline travel?

- a series of steps, involving many services

Example: organization of air travel

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

layers: each layer implements a service
- via its own internal-layer actions
- relying on services provided by layer below

Why layering?

Approach to designing/discussing complex systems:

- explicit structure allows identification, relationship of 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

Layered Internet protocol stack

- *application:* supporting network applications
  - HTTP, IMAP, SMTP, DNS
- *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"

application
transport
network
link
physical

Services, Layering and Encapsulation

application

transport

network

link

physical

source

M

Application exchanges messages to implement some application service using *services* of transport layer

H_t | M

Transport-layer protocol transfers M (e.g., reliably) from one *process* to another, using services of network layer

- transport-layer protocol encapsulates application-layer message, M, with *transport* layer-layer header H_t to create a transport-layer segment
  - H_t used by transport layer protocol to implement its service

application

transport

network

link

physical

destination

Services, Layering and Encapsulation

application
transport
network
link
physical

source

M

H_t M

Transport-layer protocol transfers M (e.g., reliably) from one *process* to another, using services of network layer

H_n H_t M

Network-layer protocol transfers transport-layer segment [H_t | M] from one *host* to another, using link layer services

application
transport
network
link
physical

destination

- network-layer protocol encapsulates transport-layer segment [H_t | M] with network layer-layer header H_n to create a network-layer datagram
  - H_n used by network layer protocol to implement its service

Services, Layering and Encapsulation

Network-layer protocol transfers transport-layer segment [H_t | M] from one *host* to another, using link layer services

Link-layer protocol transfers datagram [H_n| [H_t |M] from *host* to neighboring host, using network-layer services

- link-layer protocol encapsulates network datagram [H_n| [H_t |M], with link-layer header H_l to create a link-layer frame

Encapsulation

Matryoshka dolls (stacking dolls)

message   segment   datagram   frame
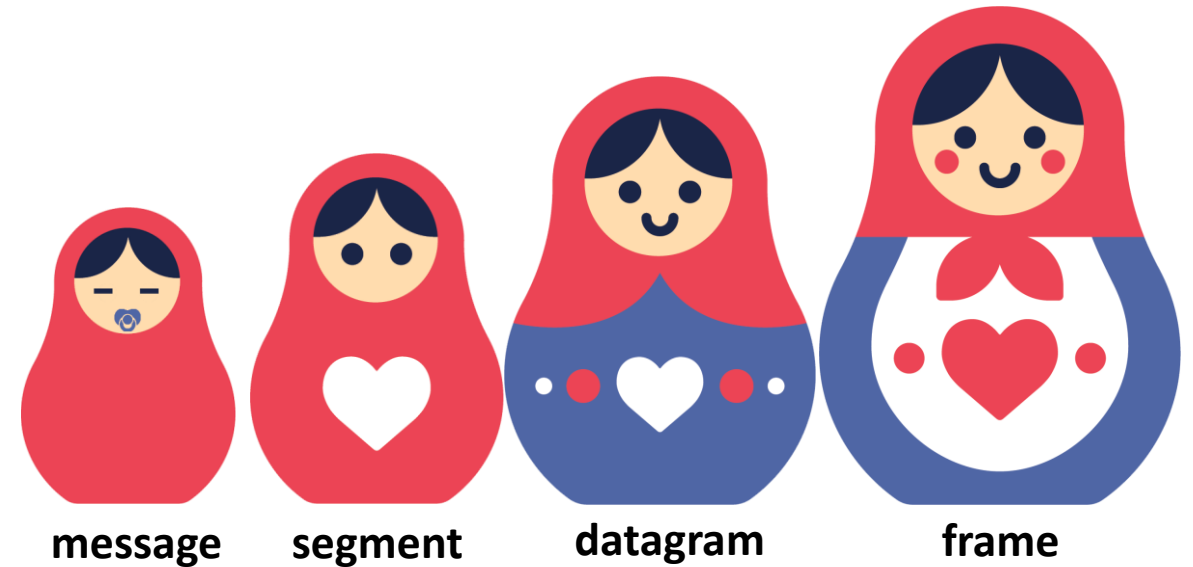
Services, Layering and Encapsulation
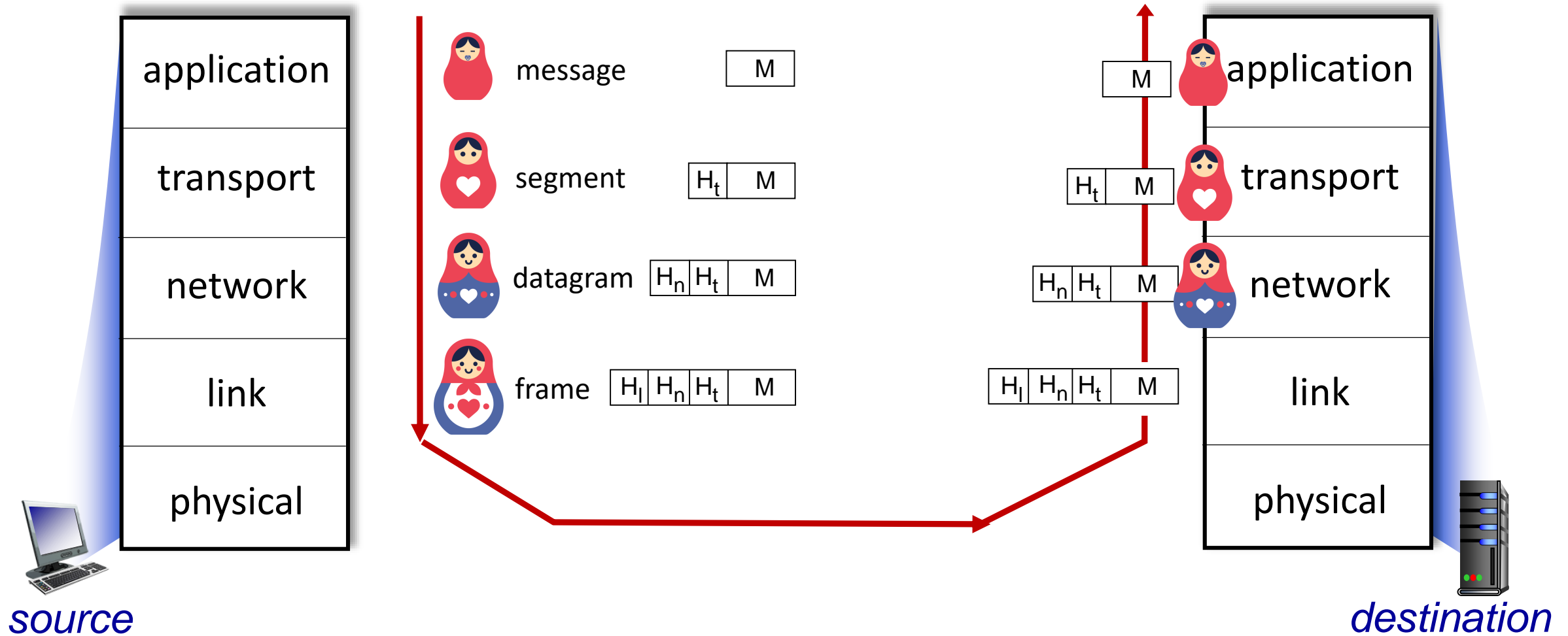
Encapsulation: an end-end view

message   M

segment H_t M

datagram H_n H_t M

frame H_l H_n H_t M

source

| application |
| transport |
| network |
| link |
| physical |

| link |
| physical |

switch

destination

M

H_t M

H_n H_t M

H_l H_n H_t M

| application |
| transport |
| network |
| link |
| physical |

H_n H_t M

H_l H_n H_t M

| network |
| link |
| physical |

H_n H_t M

router
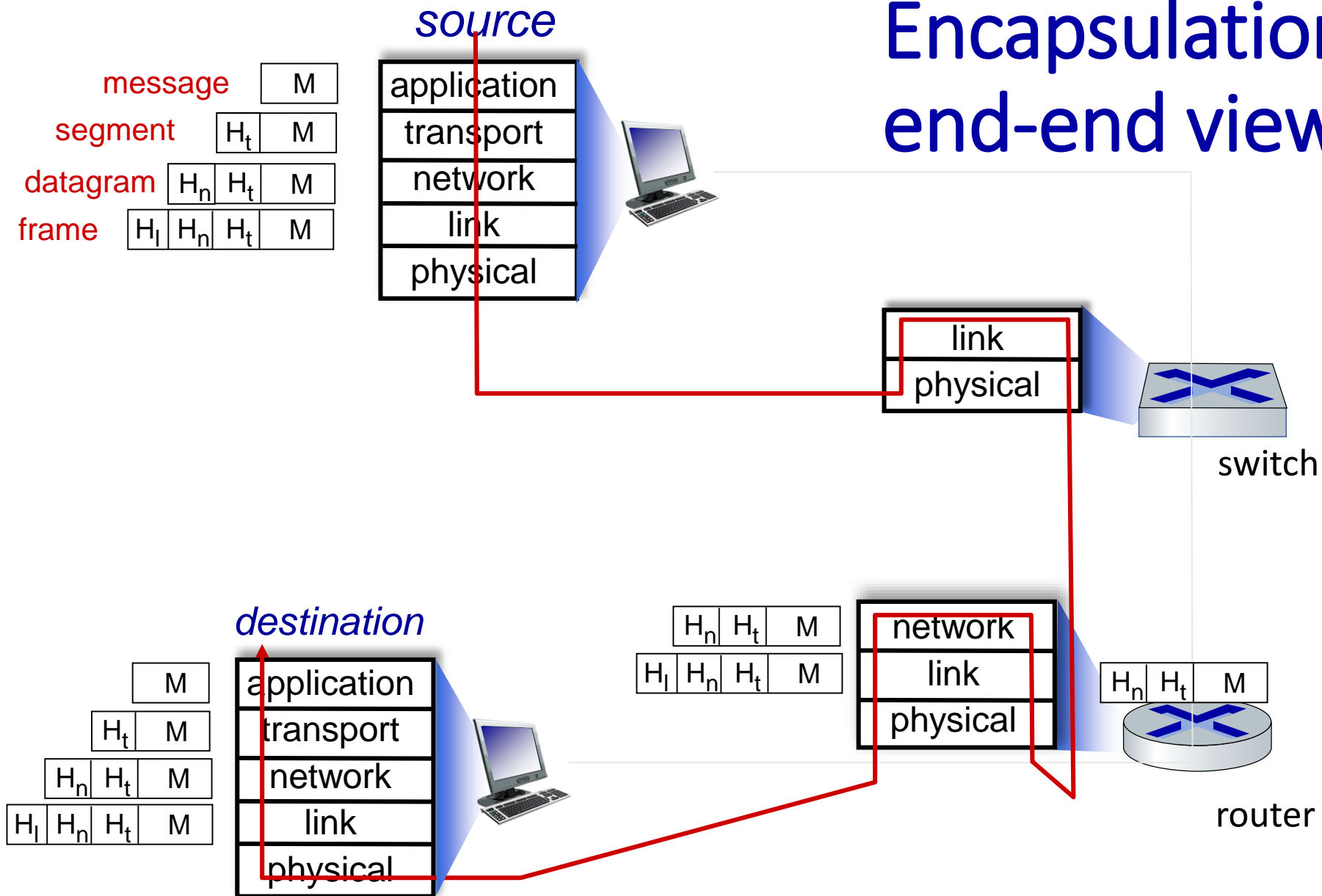
Chapter 1: summary

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

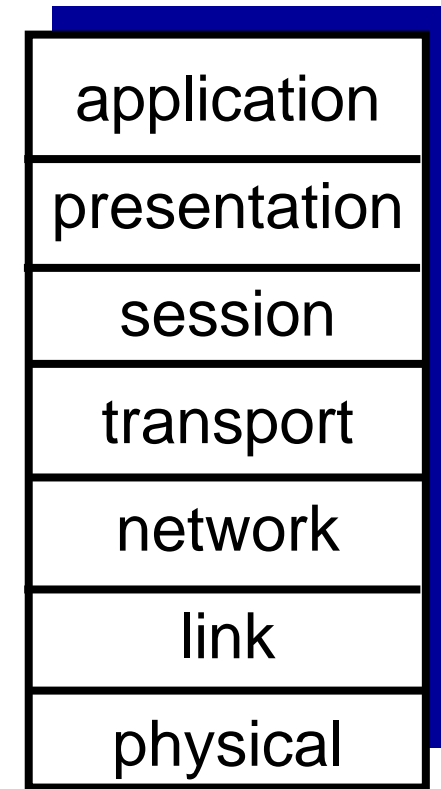- layering, service models

- security

- history

You now have:

- context, overview, vocabulary, "feel" of networking

- more depth, detail, *and fun* to follow*!*

Additional Chapter 1 slides

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?

application
presentation
session
transport
network
link
physical

The seven layer OSI/ISO reference model

Services, Layering and Encapsulation