

Chapter 1

Introduction

A note on the use of these PowerPoint slides:

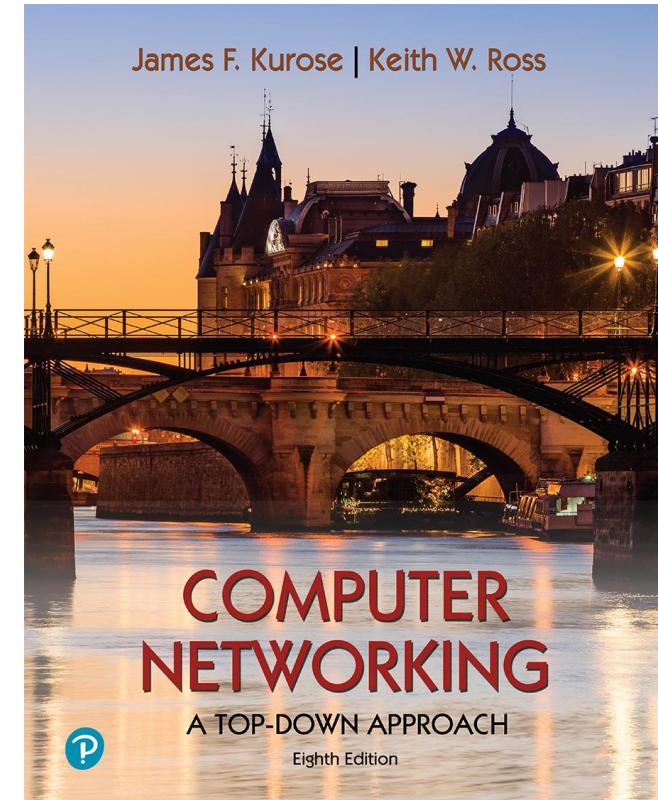
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, 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

All material copyright 1996-2023
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer Networking: A
Top-Down Approach*
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Chapter 1: introduction

Chapter goal:

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

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 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)

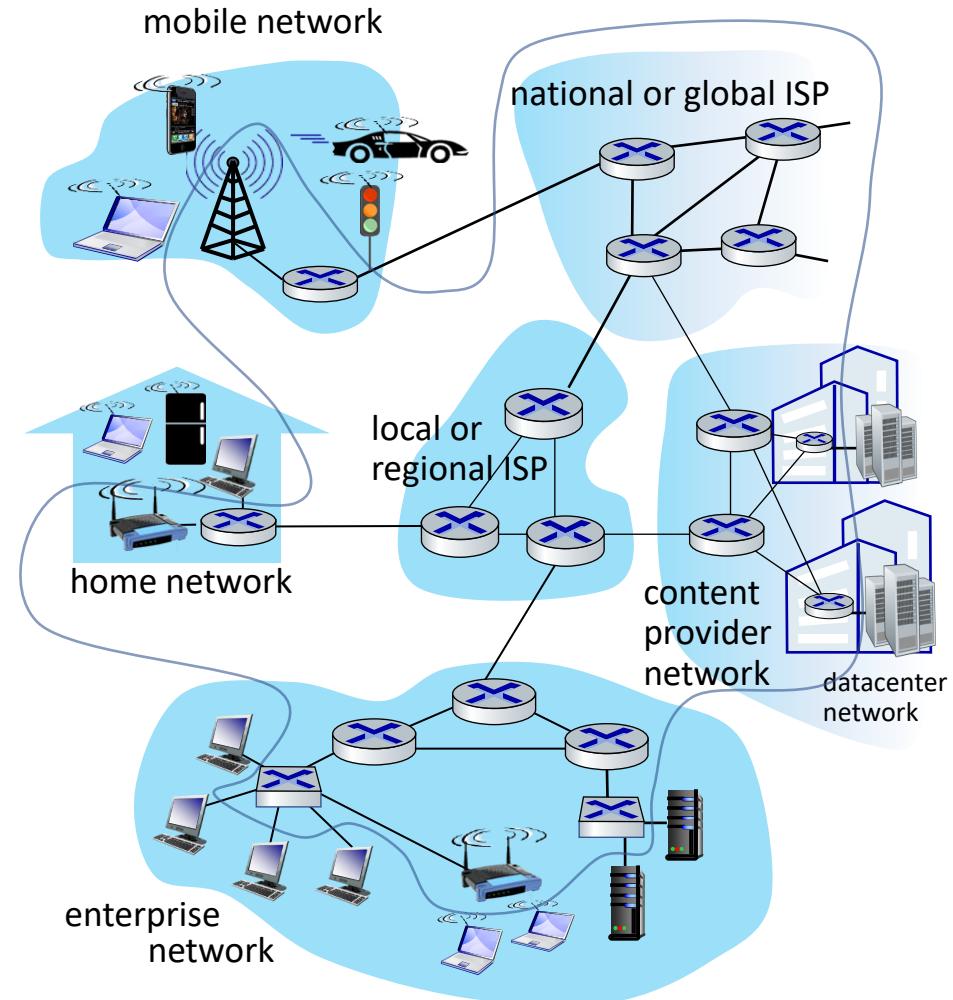
- routers, switches

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



Security Camera



Internet phones



IP picture frame



Slingbox: remote control cable TV



Gaming devices



Pacemaker & Monitor



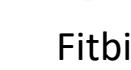
Web-enabled toaster + weather forecaster



sensorized, bed mattress



AR devices



Fitbit



diapers



Tweet-a-watt:
monitor energy use

bikes



cars

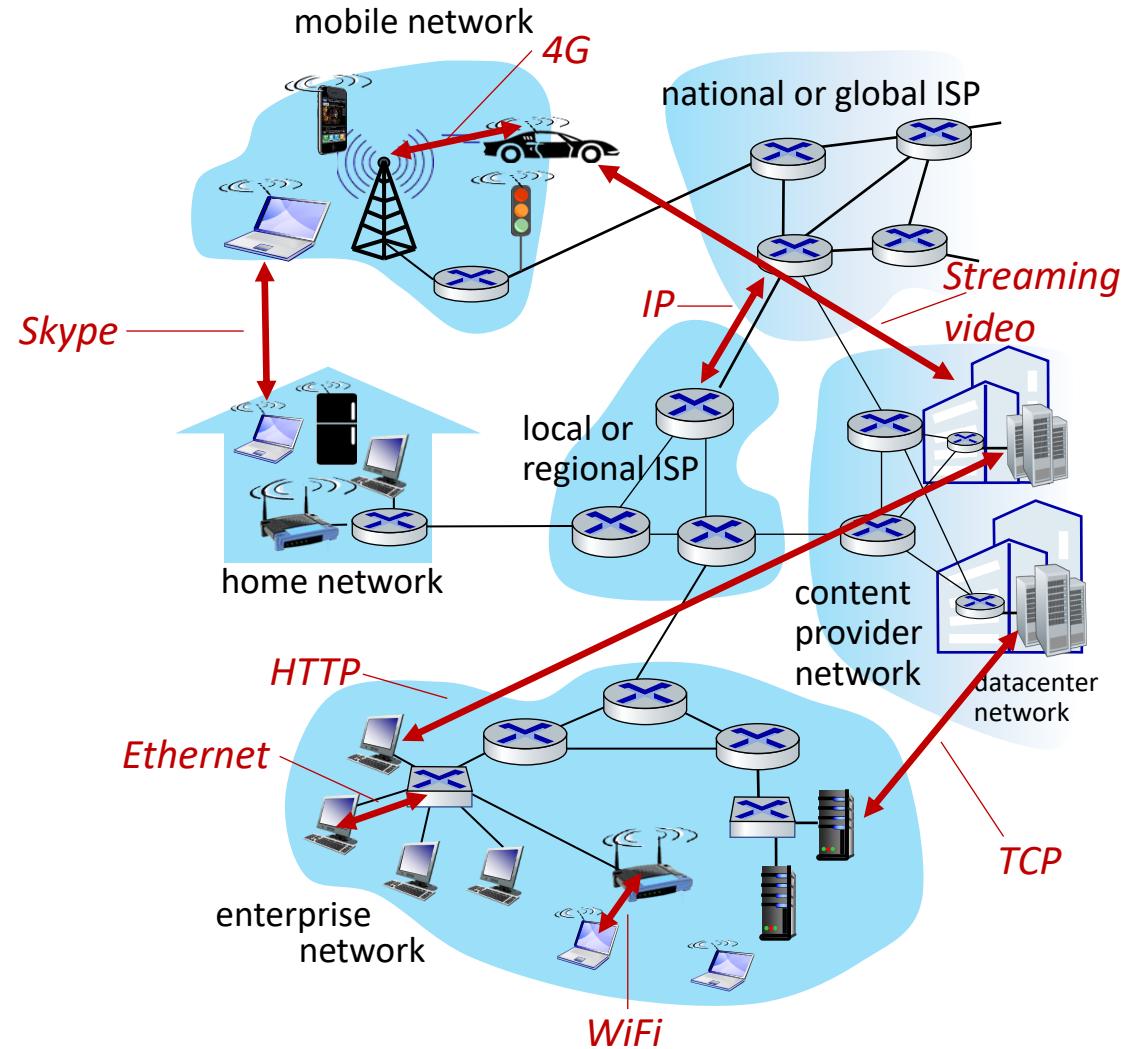


scooters

Others?

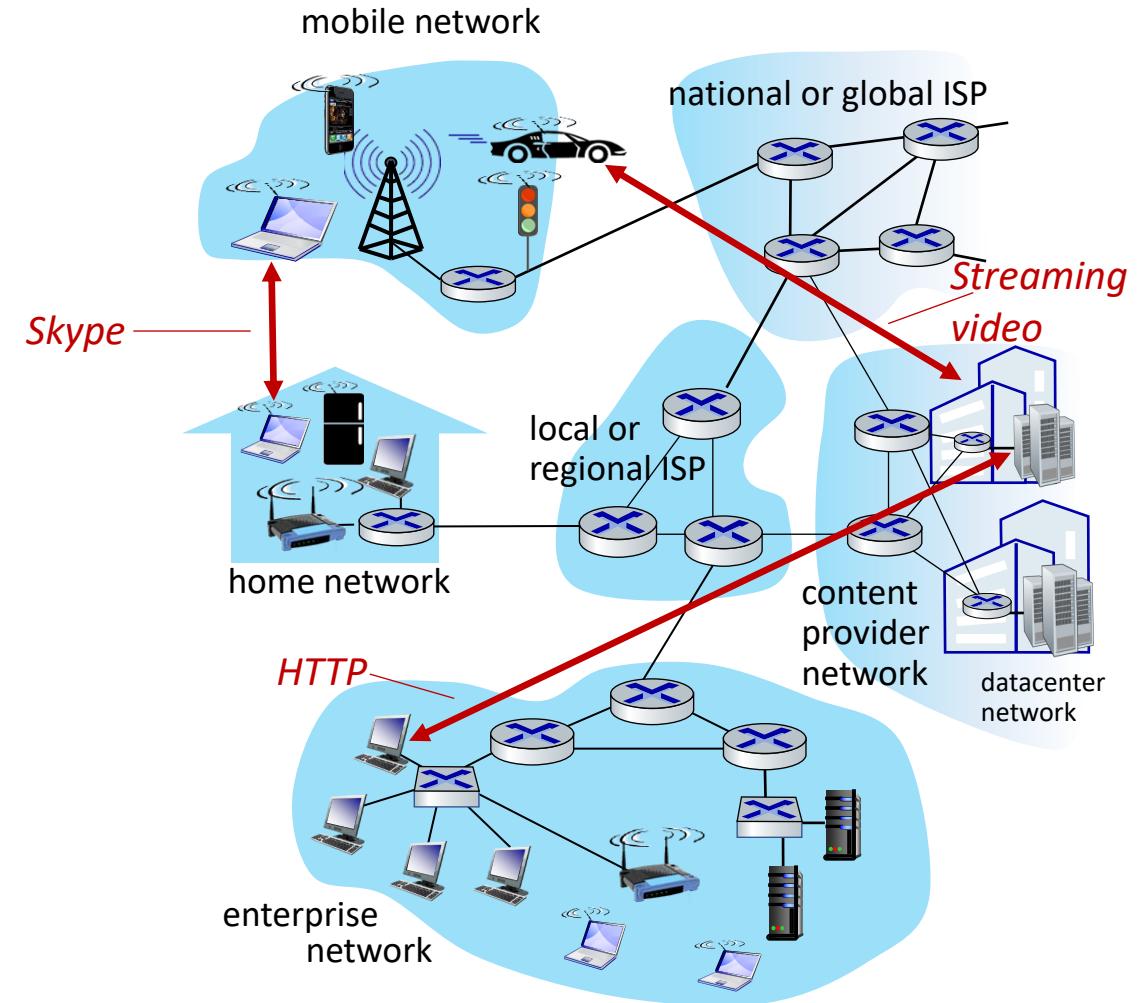
The Internet: a “nuts and bolts” view

- *Internet: “network of networks”*
 - Interconnected ISPs
- *protocols are everywhere*
 - control sending, receiving of messages
 - e.g., HTTP (Web), streaming video, Skype, TCP, IP, WiFi, 4/5G, Ethernet
- *Internet standards*
 - RFC: Request for Comments
 - IETF: Internet Engineering Task Force



The Internet: a “services” view

- *Infrastructure* that provides services to applications:
 - Web, streaming video, multimedia teleconferencing, email, games, e-commerce, social media, interconnected 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



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

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

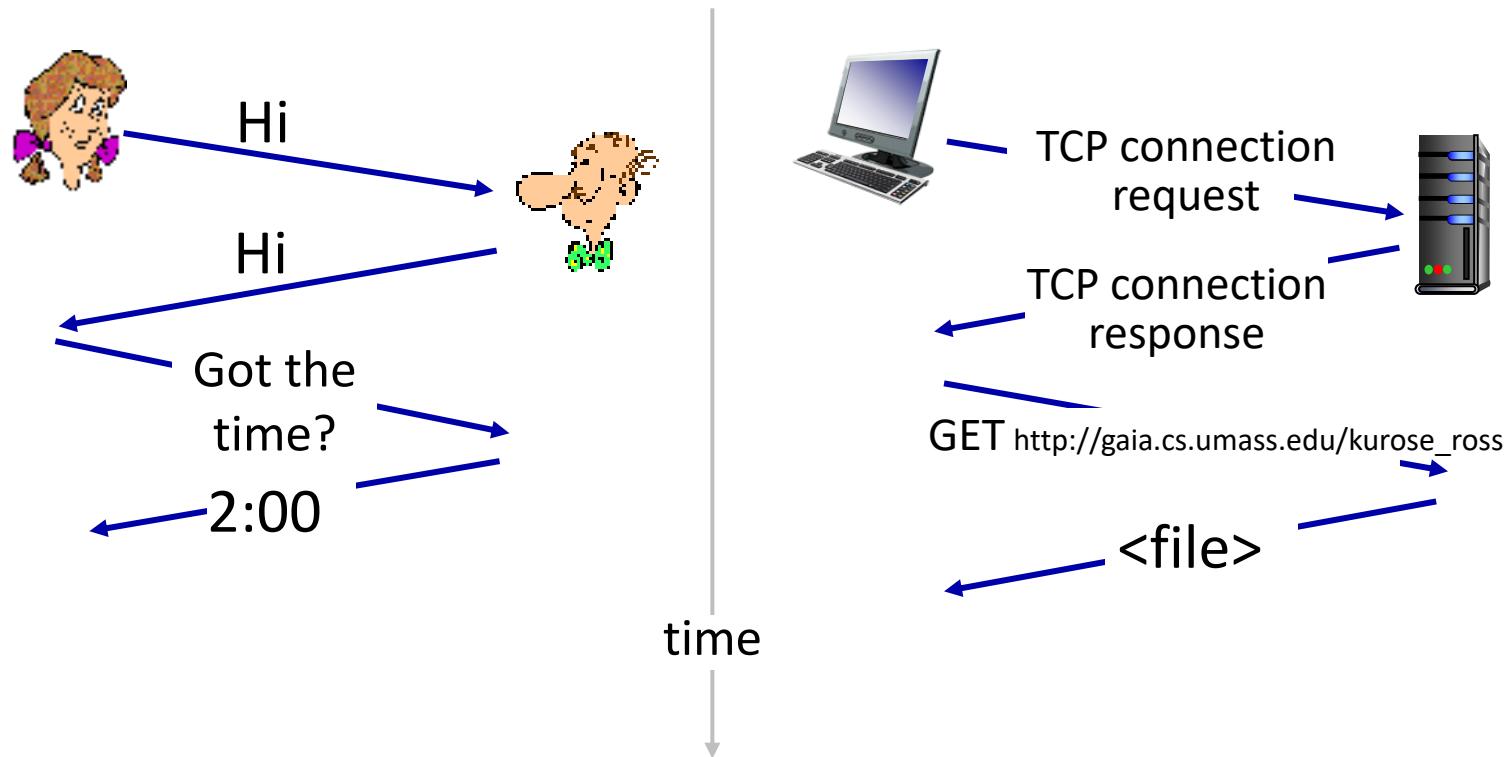
Network protocols:

- 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, and/or receipt of a message.*

What's a protocol?

A human protocol and a computer network protocol:



Q: other human protocols?

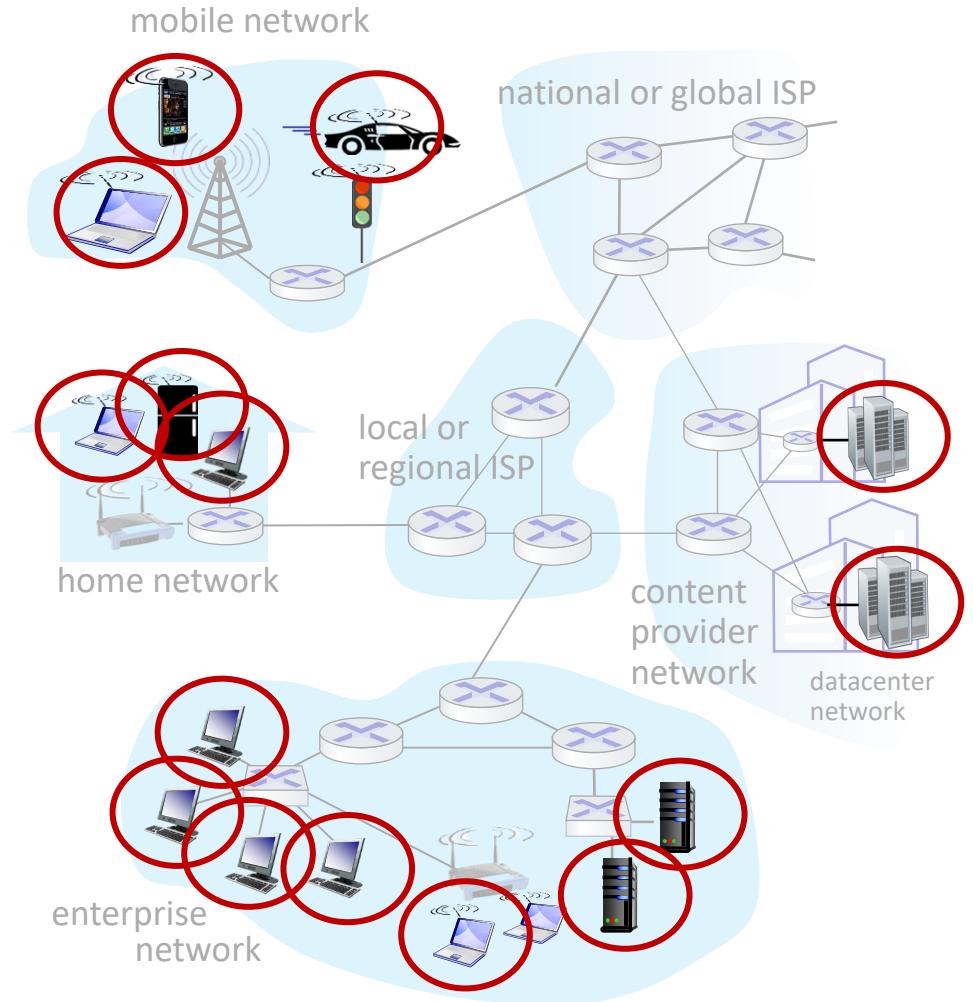
Chapter 1: roadmap

- 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



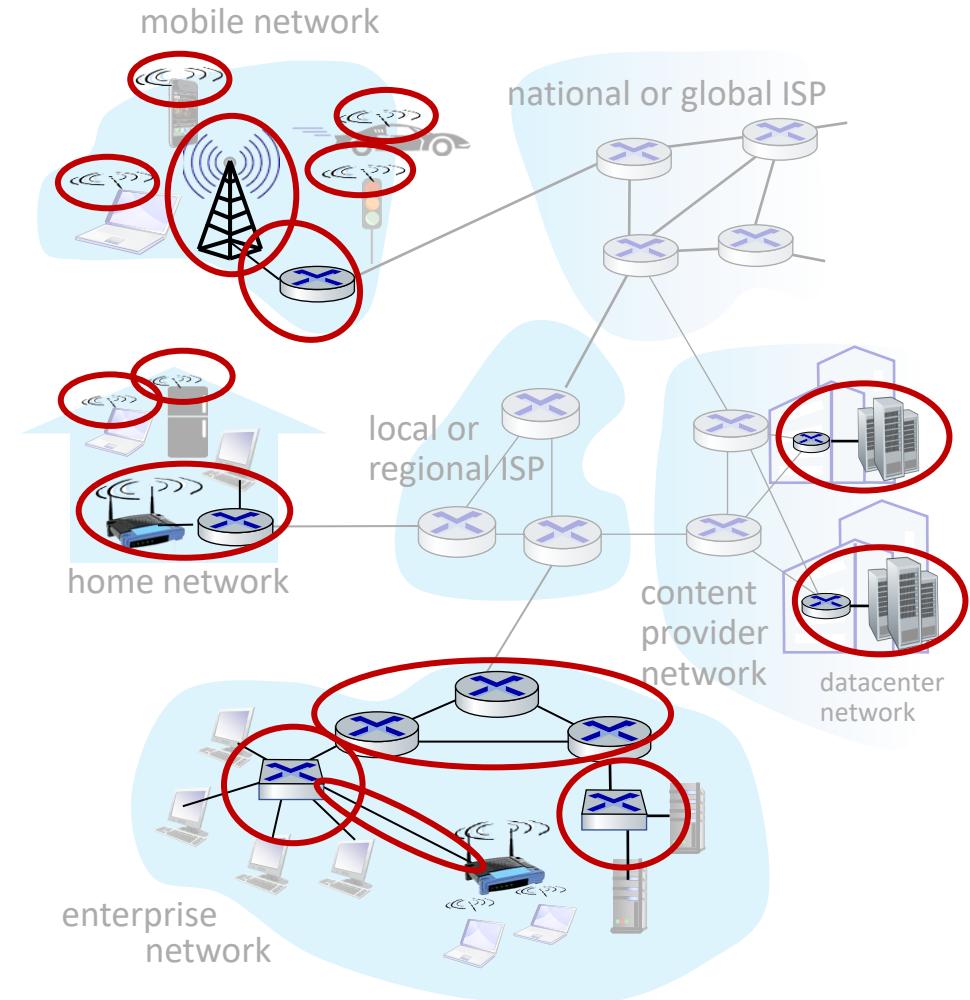
A closer look at Internet structure

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:

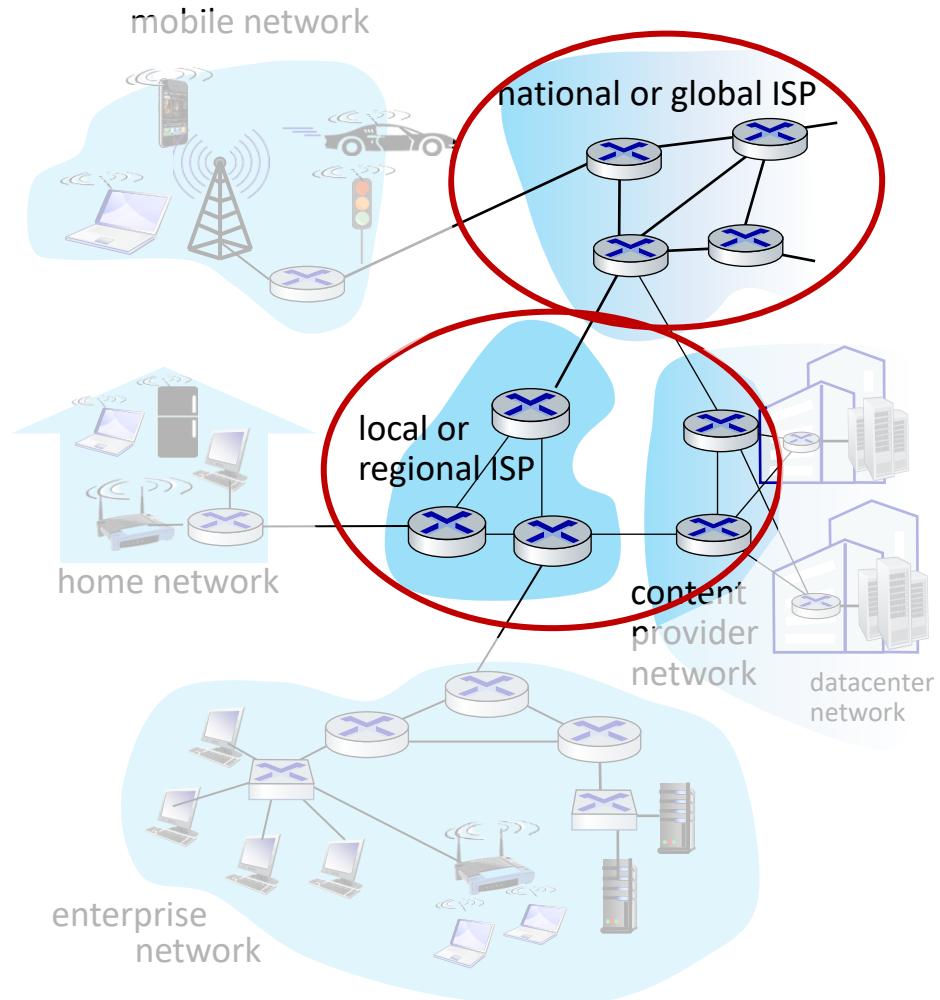
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links

Network core:

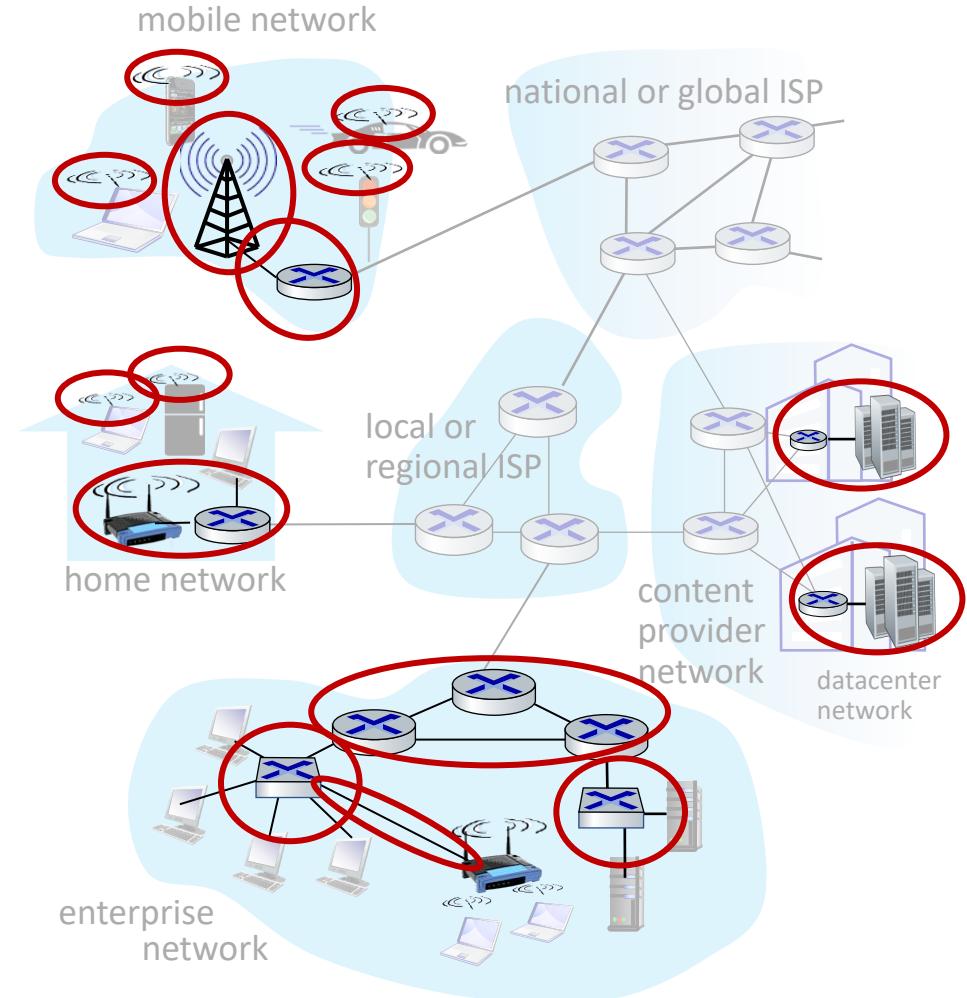
- interconnected routers
- network of networks



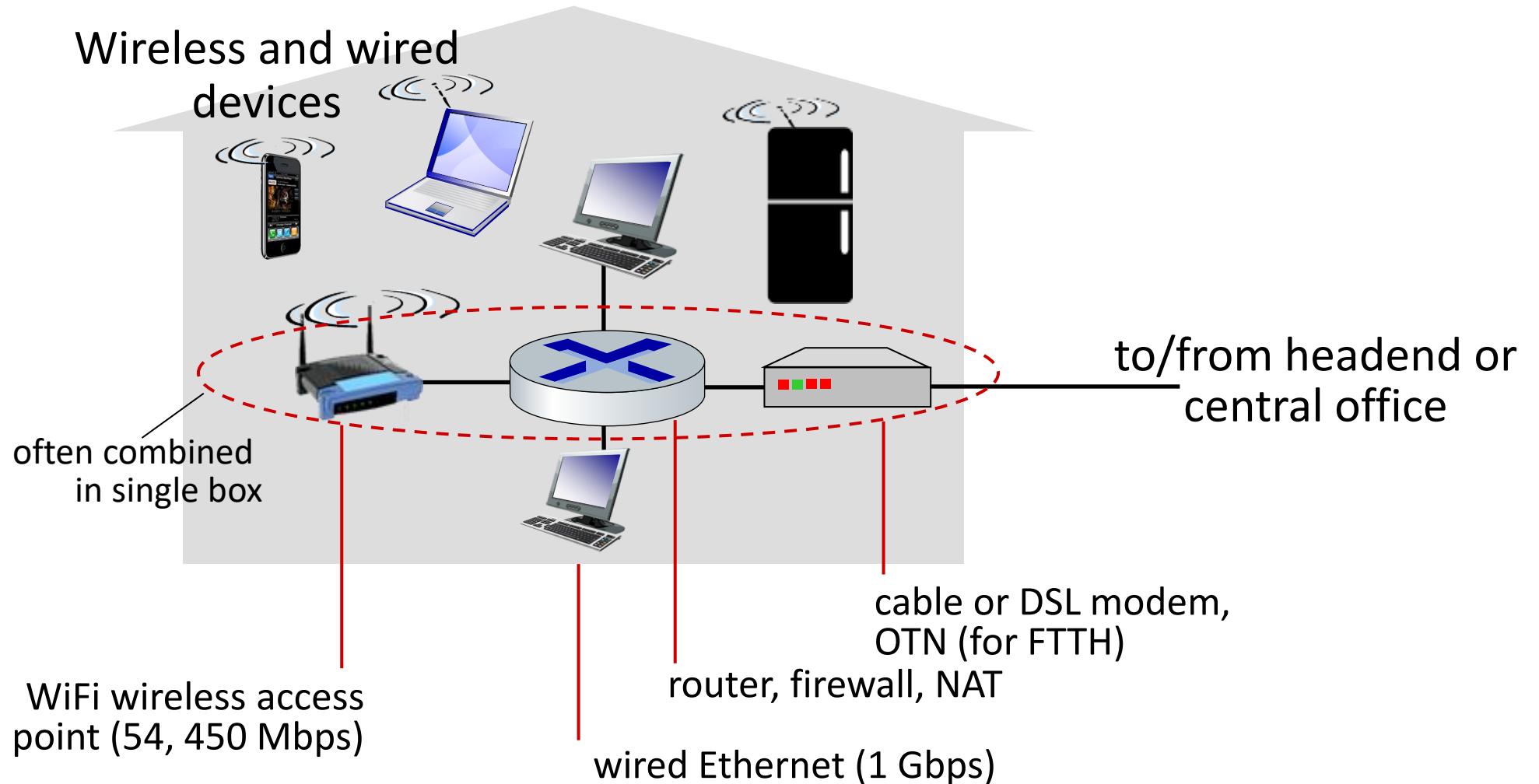
Access networks and physical media

*Q: How to connect end systems
to edge router?*

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



Access networks: home networks



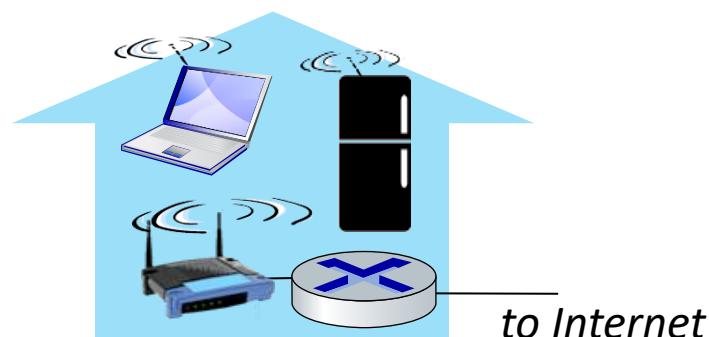
Wireless access networks

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

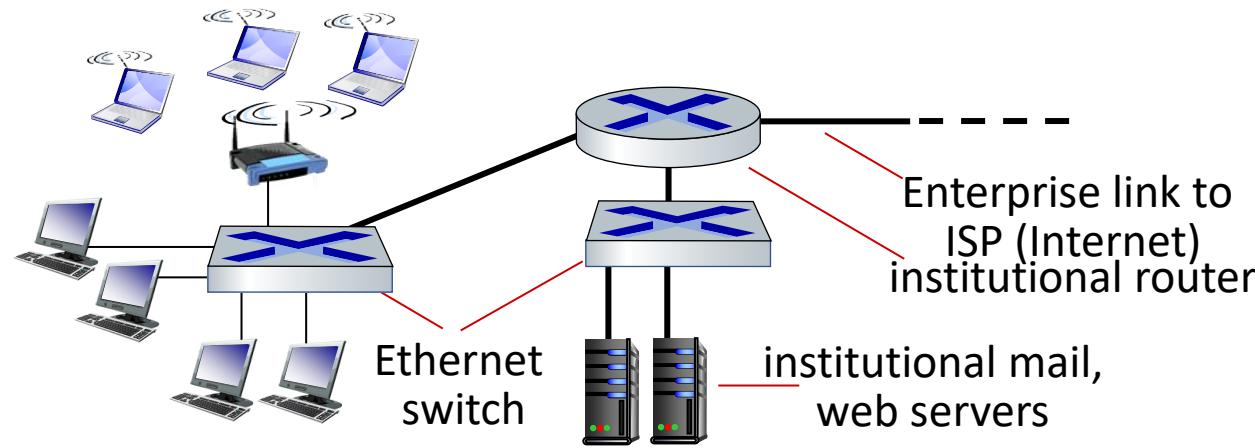


Wide-area cellular access networks

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



Access networks: enterprise networks



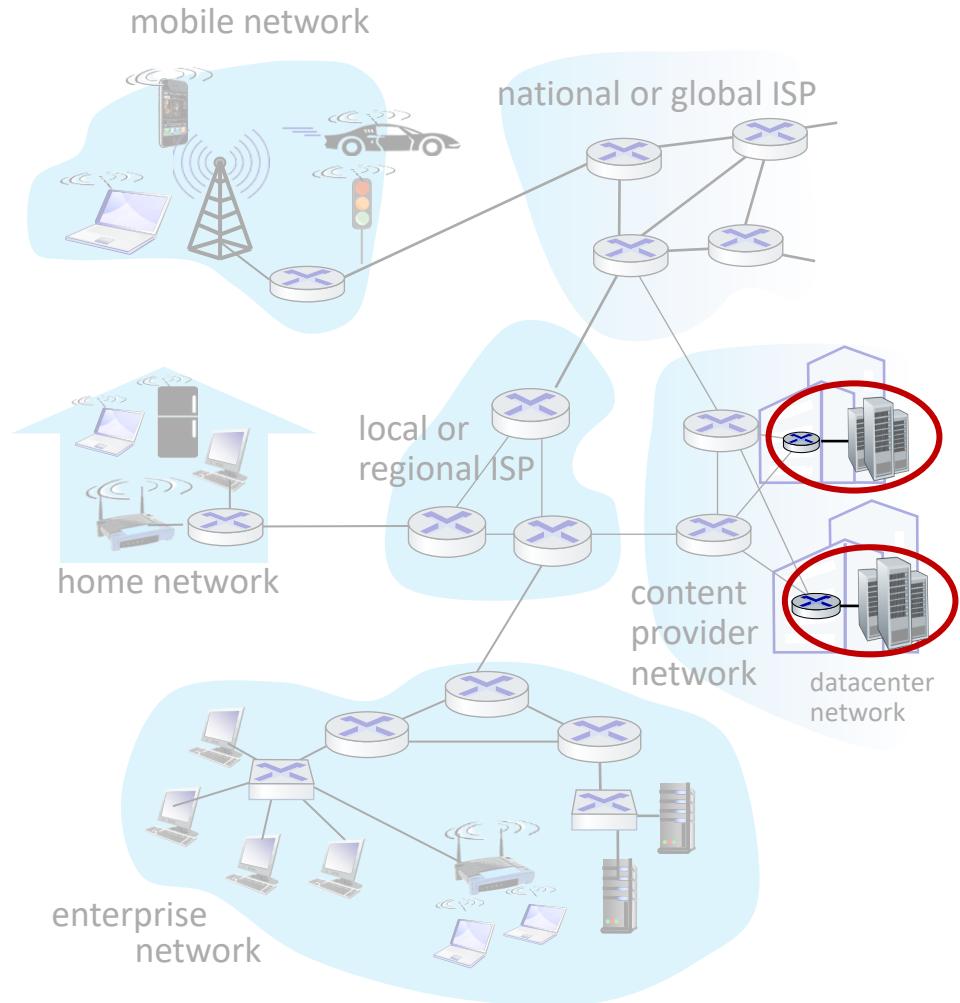
- companies, universities, etc.
- mix of wired, wireless link technologies, connecting a mix of switches and routers (we'll cover differences shortly)
 - Ethernet: wired access at 100Mbps, 1Gbps, 10Gbps
 - WiFi: wireless access points at 11, 54, 450 Mbps

Access networks: data center networks

- high-bandwidth links (10s to 100s Gbps) connect hundreds to thousands of servers together, and to Internet



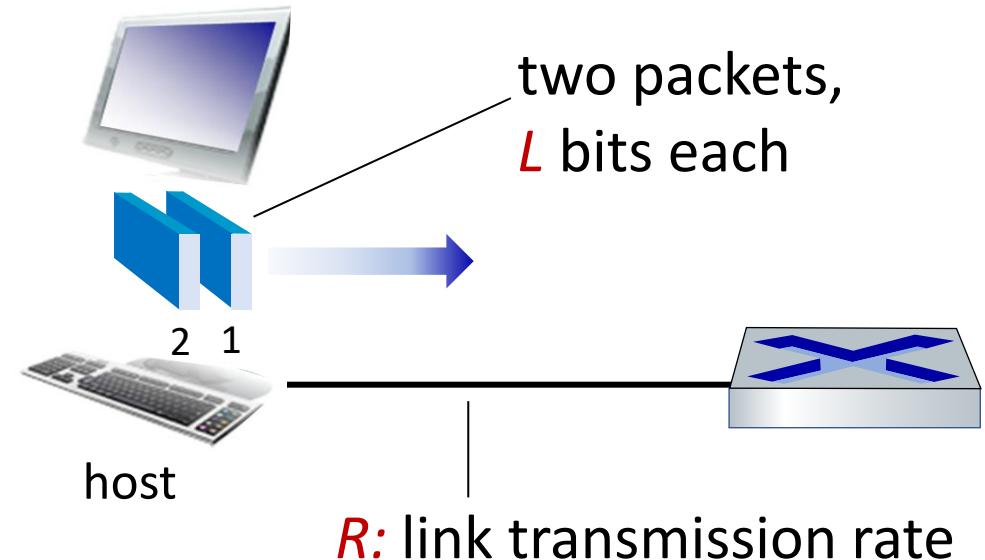
Courtesy: Massachusetts Green High Performance Computing Center (mghpcc.org)



Host: sends *packets* of data

host sending function:

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



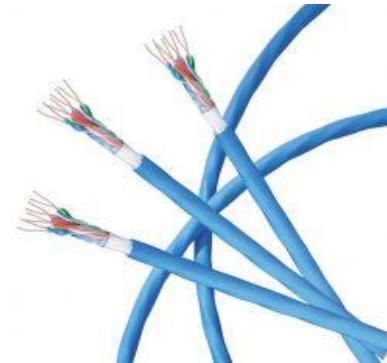
$$\text{packet transmission delay} = \frac{\text{time needed to transmit } L\text{-bit packet into link}}{R \text{ (bits/sec)}} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

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

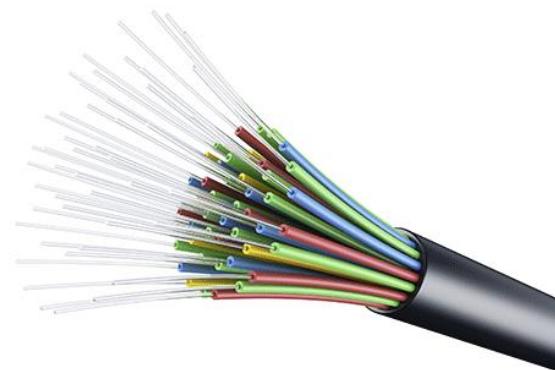
Coaxial cable:

- 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 various “bands” in electromagnetic spectrum
- no physical “wire”
- broadcast, “half-duplex” (sender to receiver)
- propagation environment effects:
 - reflection
 - obstruction by objects
 - Interference/noise

Radio link types:

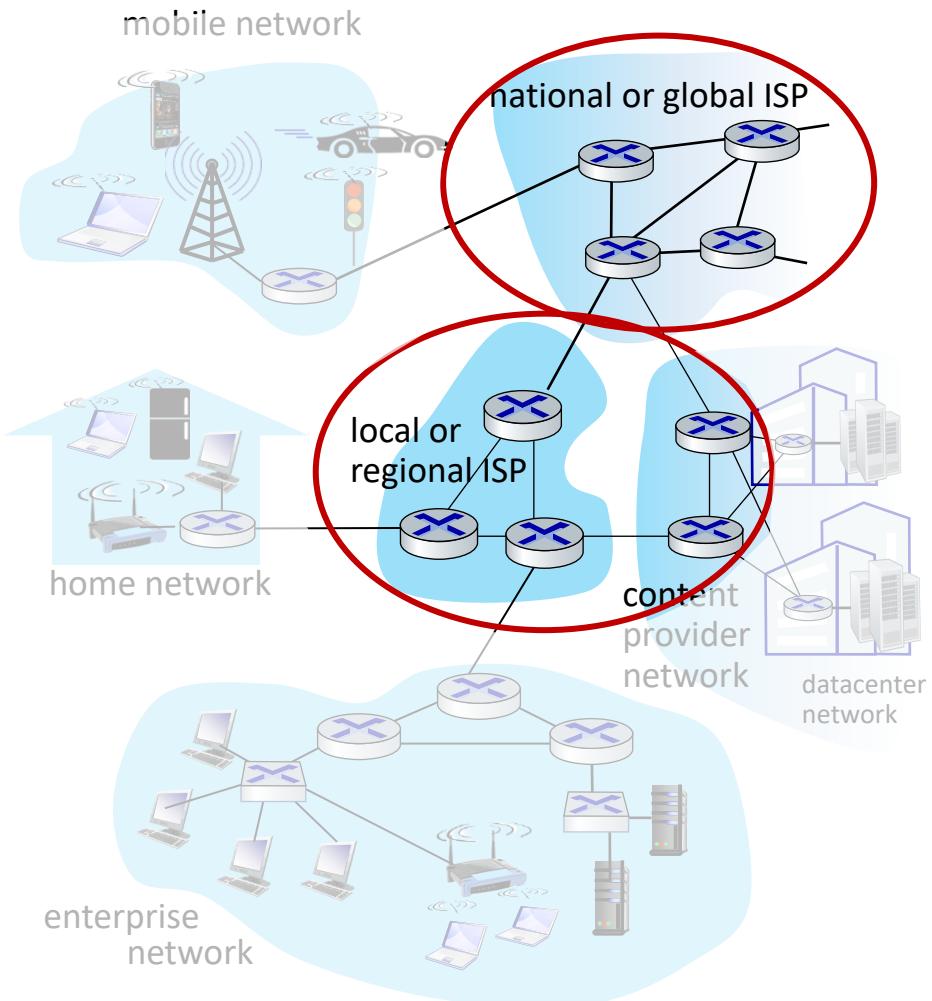
- **Wireless LAN (WiFi)**
 - 10-100's Mbps; 10's of meters
- **wide-area** (e.g., 4G/5G cellular)
 - 10's Mbps (4G) over ~10 Km
- **Bluetooth:** cable replacement
 - short distances, limited rates
- **terrestrial microwave**
 - point-to-point; 45 Mbps channels
- **satellite**
 - up to < 100 Mbps (Starlink) downlink
 - 270 msec end-end delay (geostationary)

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

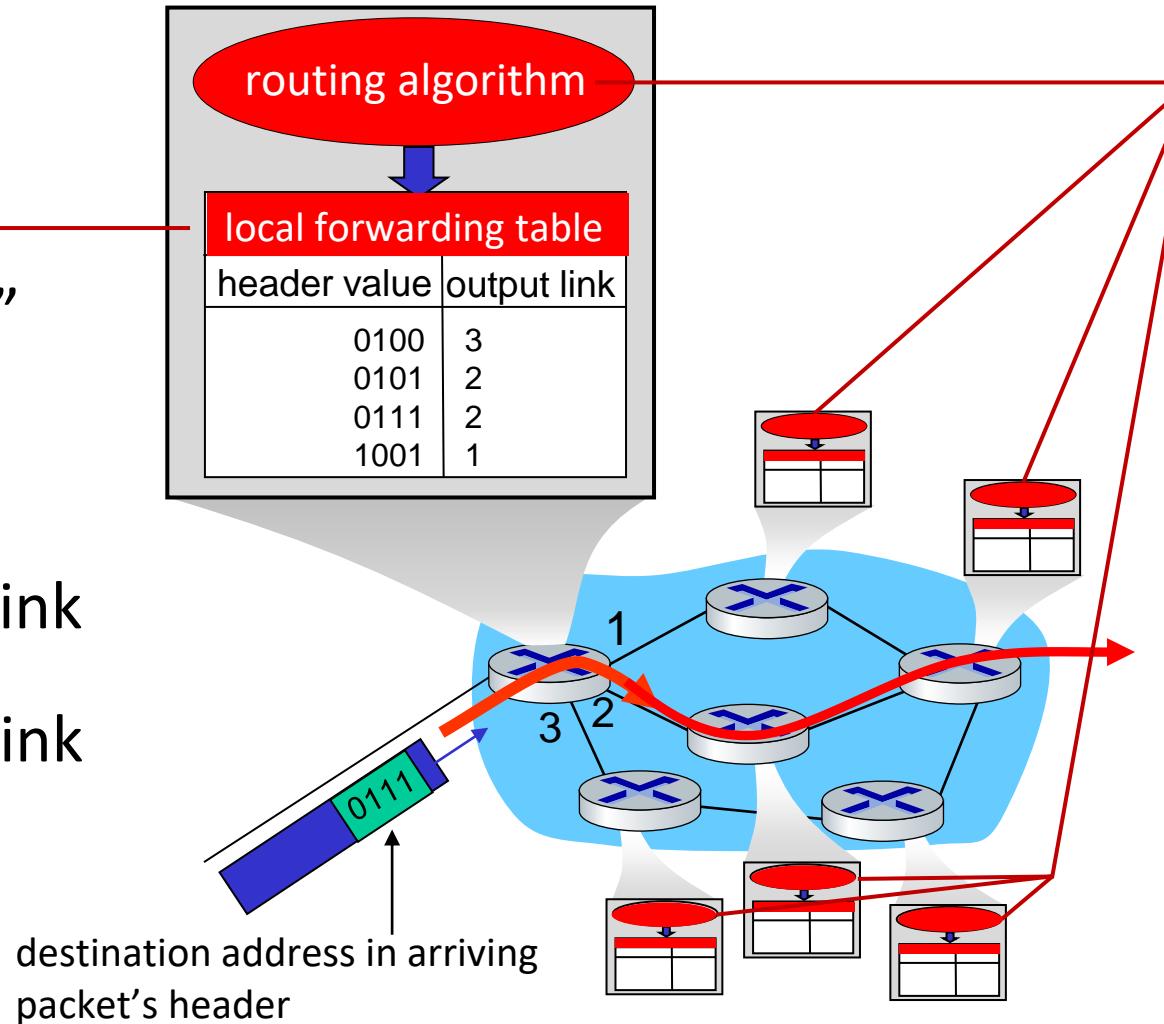
- mesh of interconnected routers
- **packet-switching**: hosts break application-layer messages into *packets*
 - network **forwards** packets from one router to the next, across links on path from **source to destination**
 - each packet transmitted at full link capacity



Two key network-core functions

Forwarding:

- aka “switching”
- *local* action:
move arriving
packets from
router’s input link
to appropriate
router output link



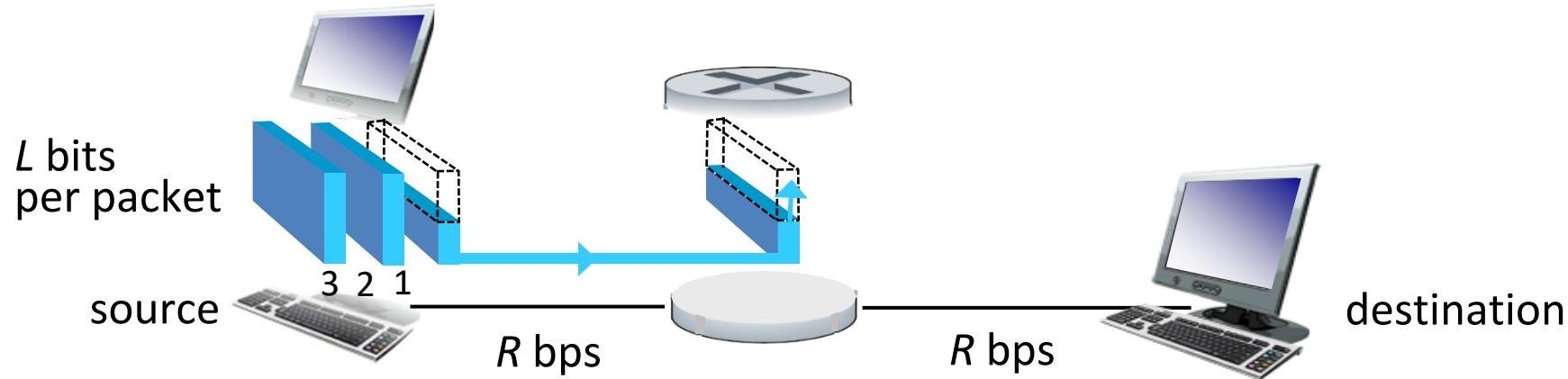
Routing:

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





Packet-switching: store-and-forward

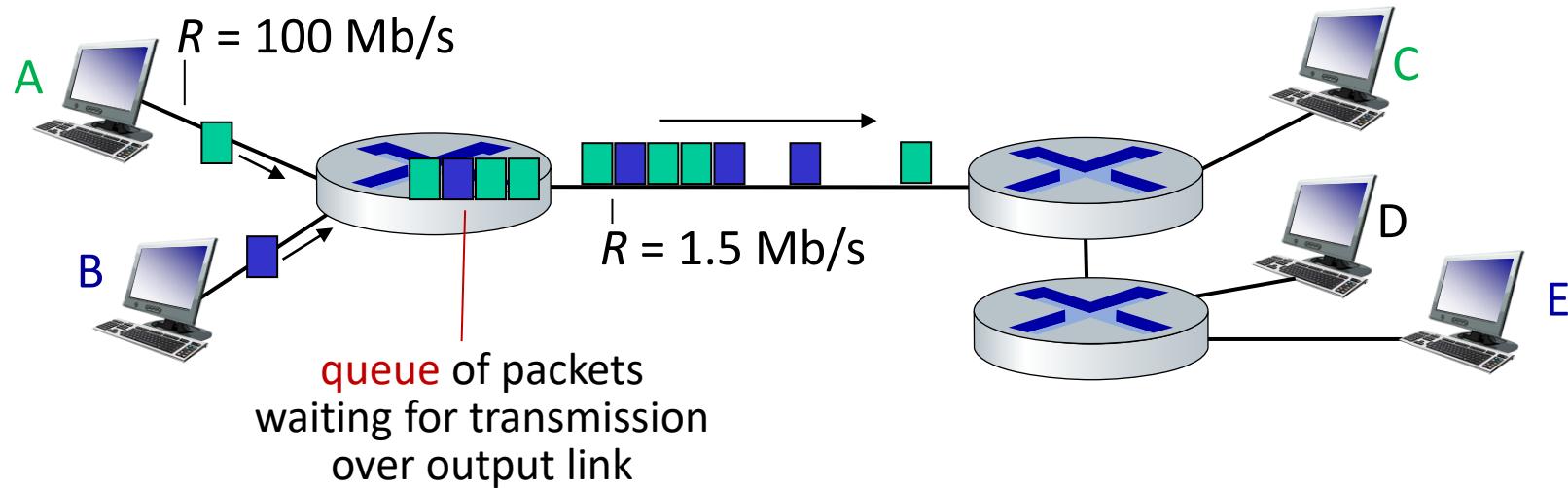


- **packet transmission delay:** takes L/R seconds to transmit (push out) L -bit packet into link at R bps
- **store and forward:** entire packet must arrive at router before it can be transmitted on next link

One-hop numerical example:

- $L = 10 \text{ Kbits}$
- $R = 100 \text{ Mbps}$
- one-hop transmission delay = 0.1 msec

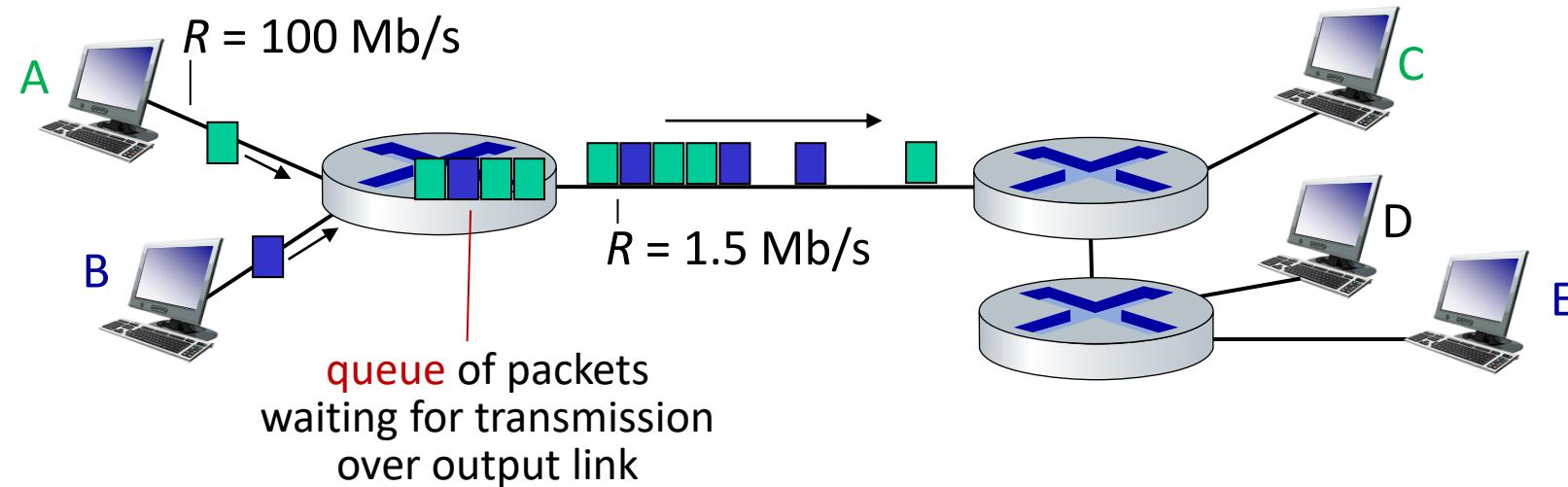
Packet-switching: queueing



Queueing occurs when work arrives faster than it can be serviced:



Packet-switching: queueing



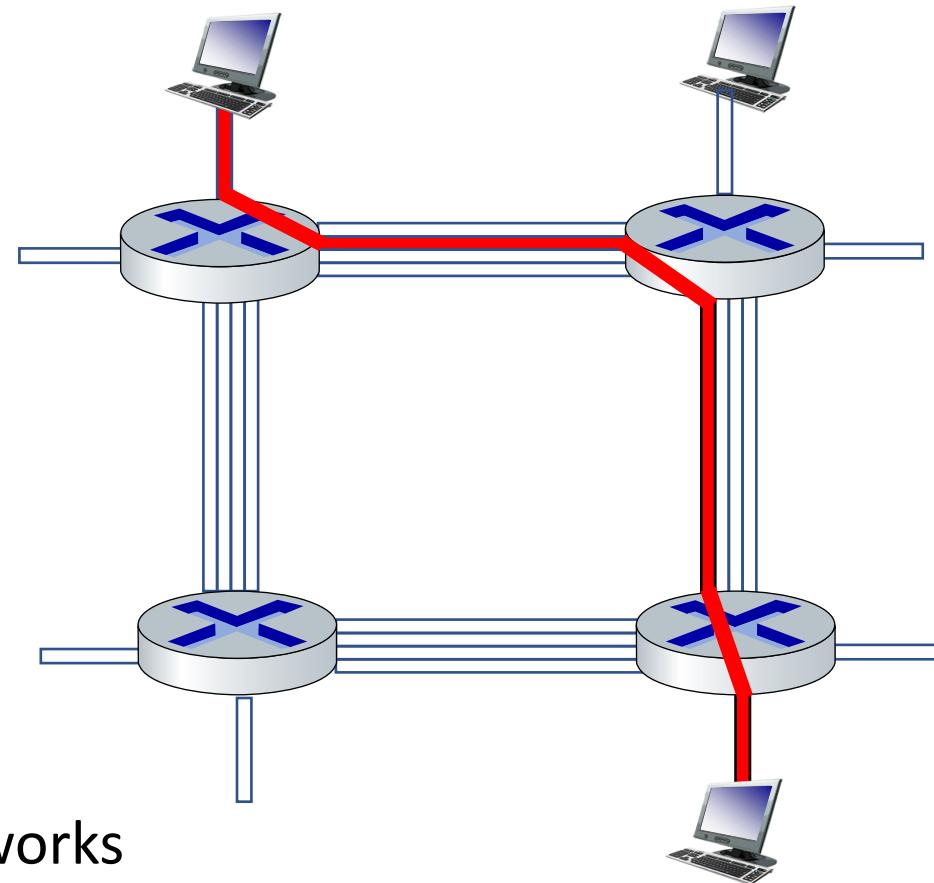
Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

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

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



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

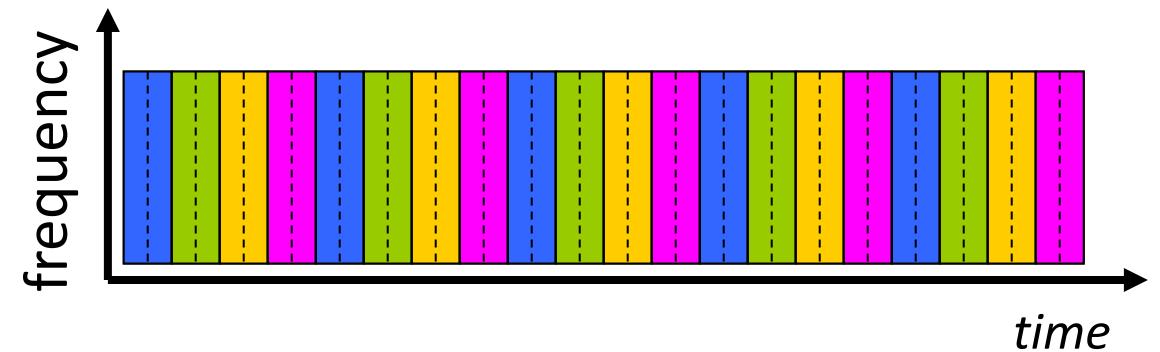
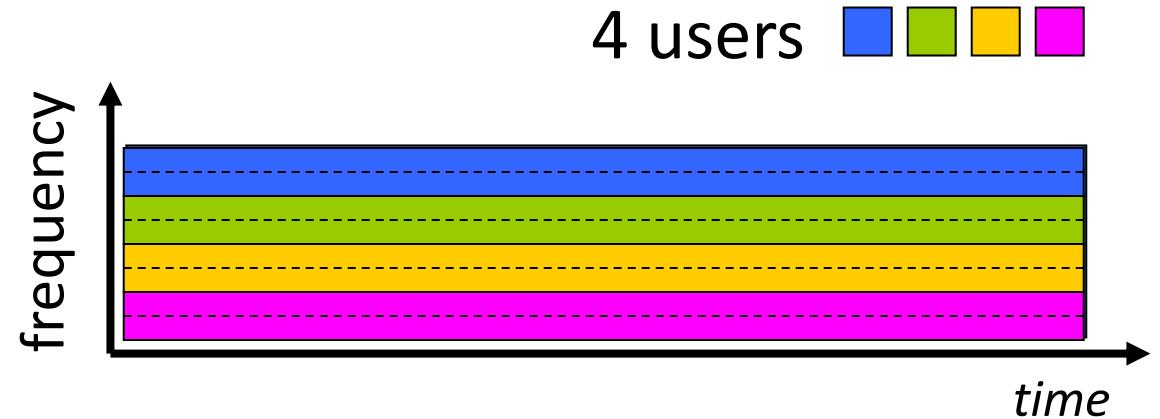
Circuit switching: FDM and TDM

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)

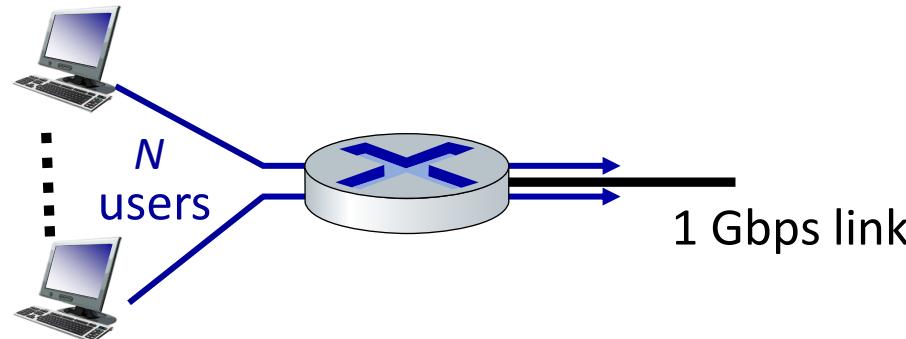
- time divided into slots
- each call allocated periodic slot(s), can transmit at maximum rate of (wider) frequency band (only) during its time slot(s)



Packet switching versus circuit switching

example:

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



Q: 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 .0004 *

Q: how did we get value 0.0004?
A: HW problem (for those with
course in probability only)

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

Packet switching versus circuit switching

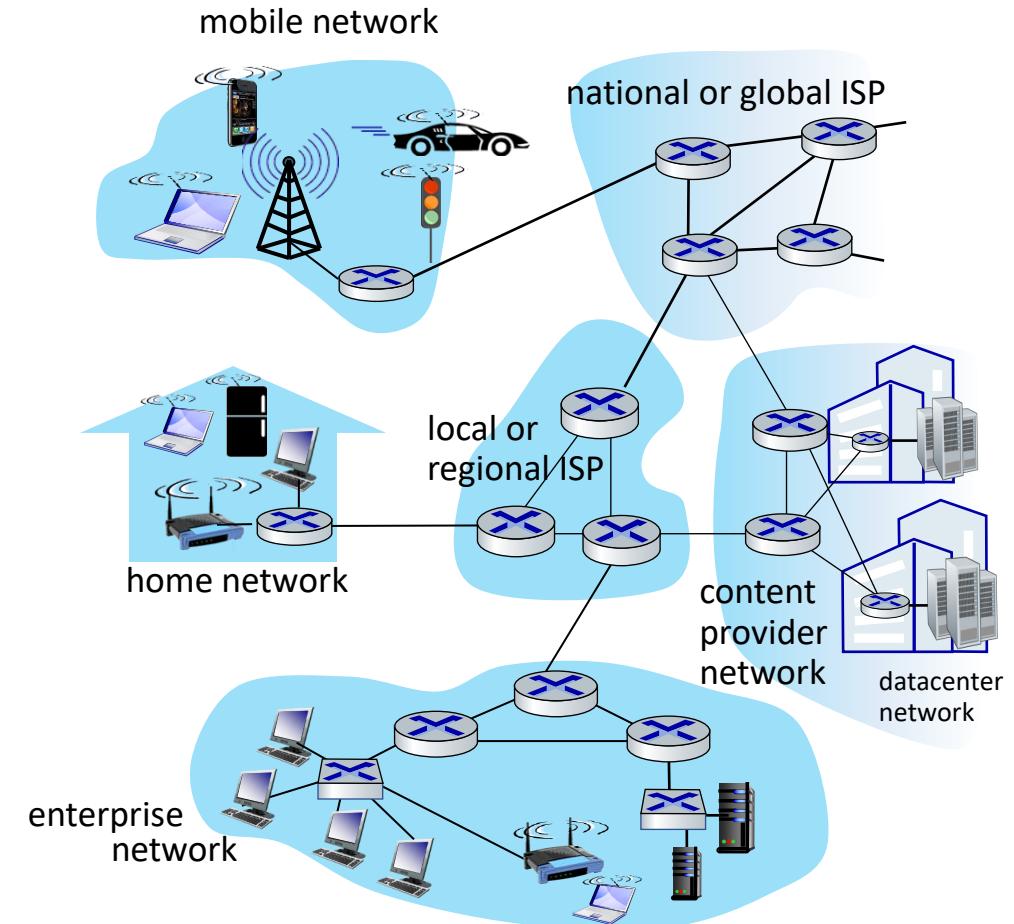
Is packet switching a “slam dunk winner”?

- 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 with packet-switching?**
 - “It’s complicated.” We’ll study various techniques that try to make packet switching as “circuit-like” as possible.

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

Internet structure: a “network of networks”

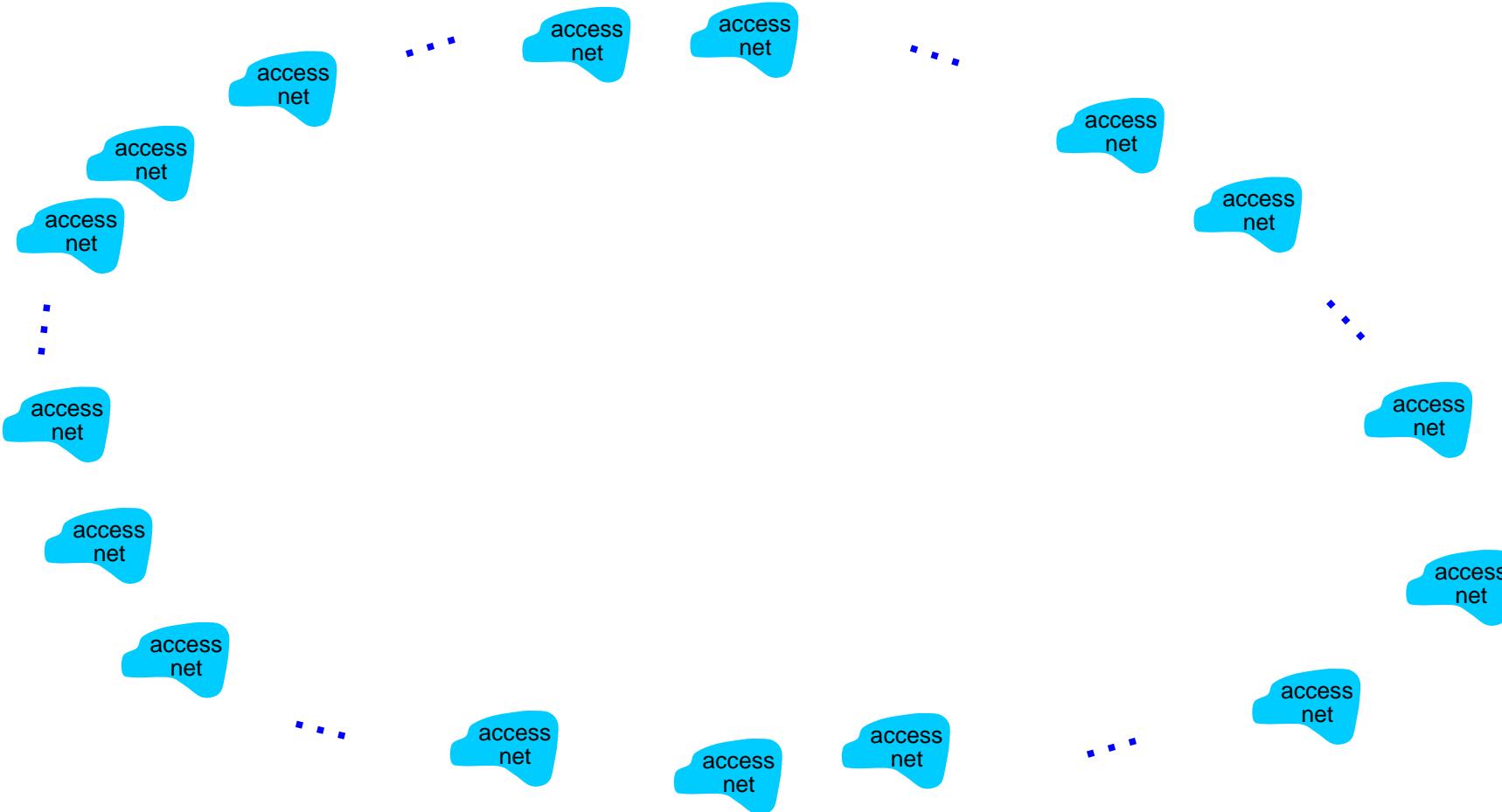
- hosts connect to Internet via **access** Internet Service Providers (ISPs)
- access ISPs in turn must be interconnected
 - so that *any* two hosts (*anywhere!*) can send packets to each other
- resulting network of networks is very complex
 - evolution driven by **economics, national policies**



Let's take a stepwise approach to describe current Internet structure

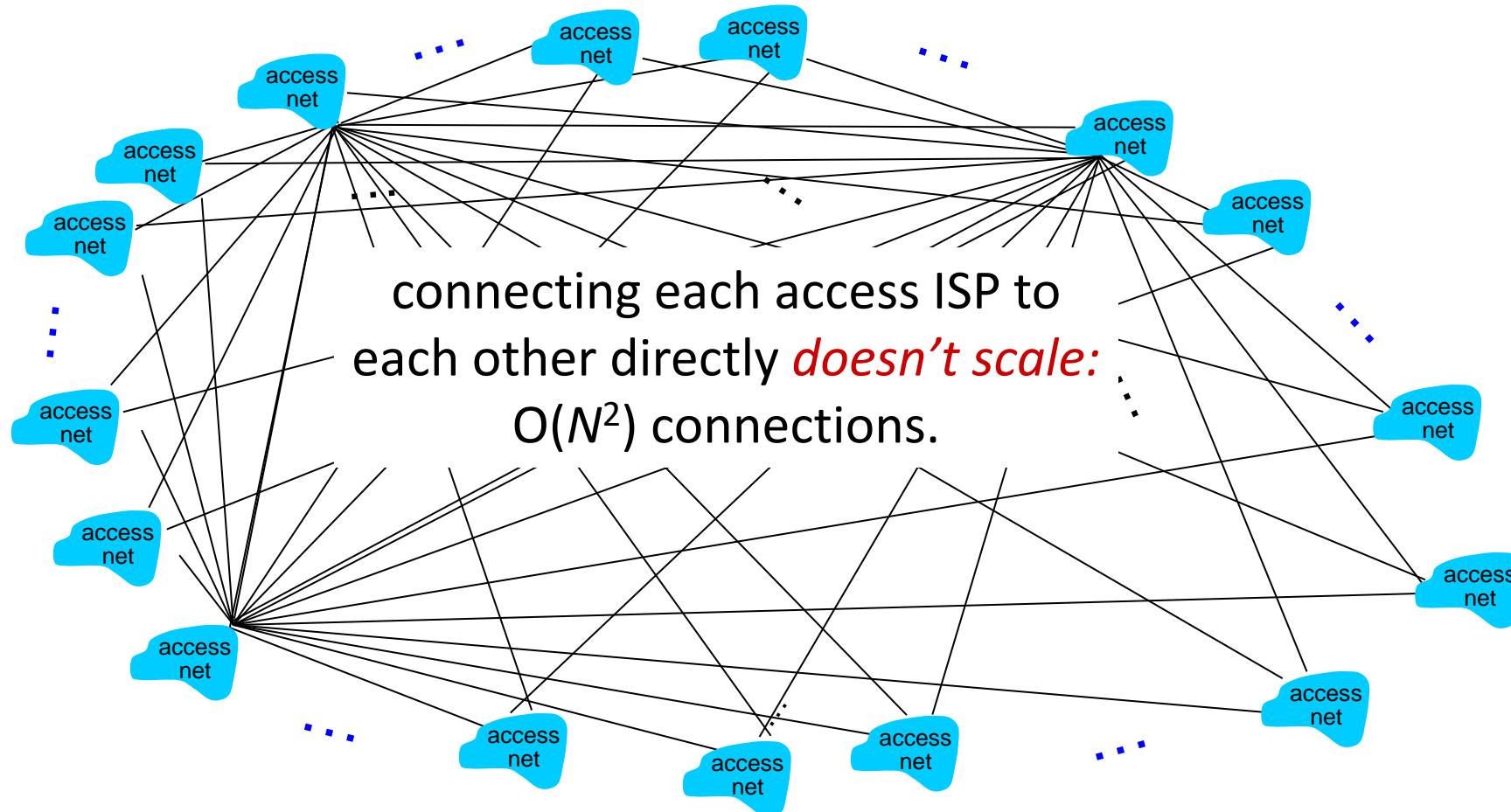
Internet structure: a “network of networks”

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



Internet structure: a “network of networks”

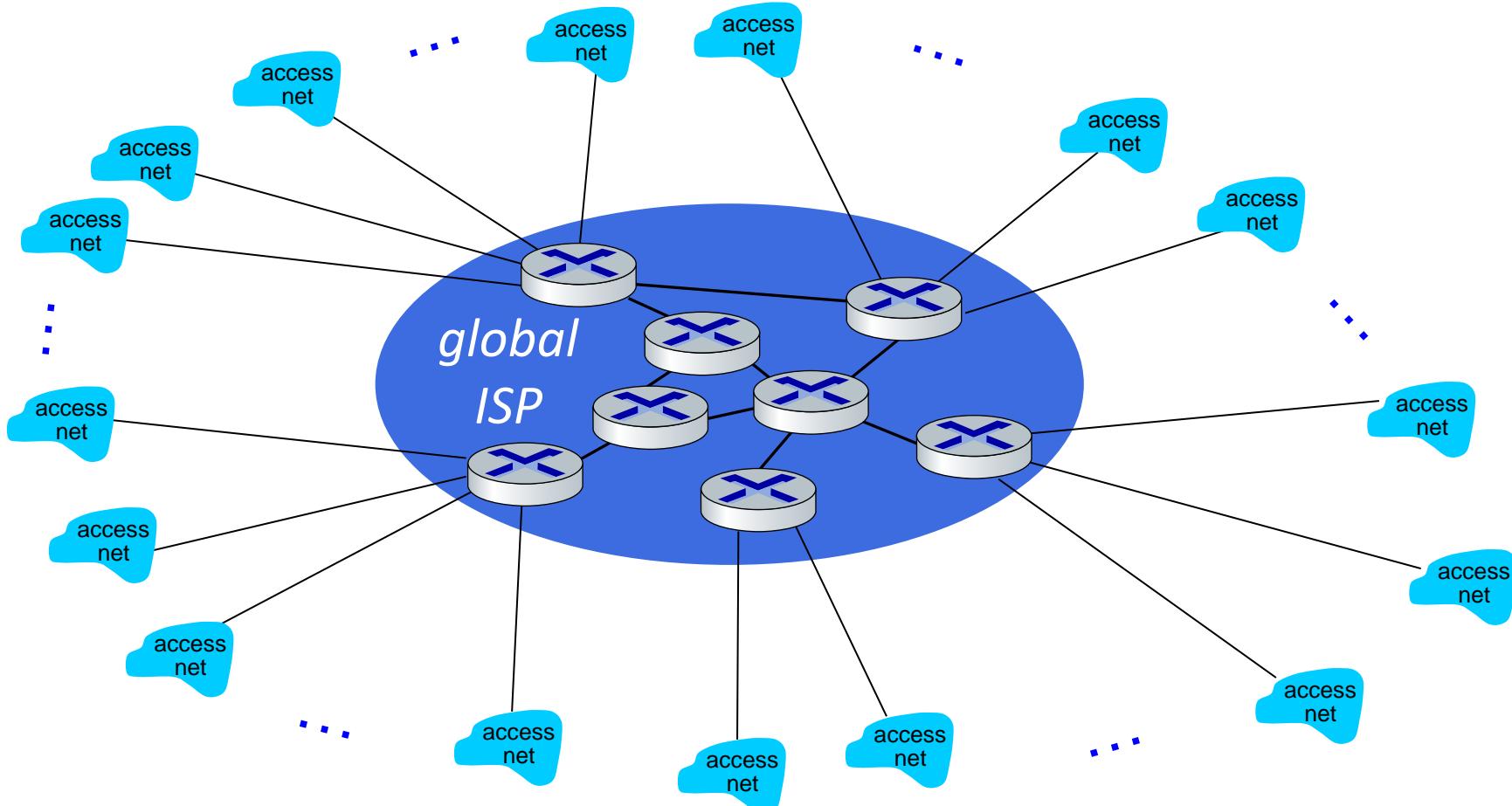
Question: given *millions* of access ISPs, how to connect them together?



Internet structure: a “network of networks”

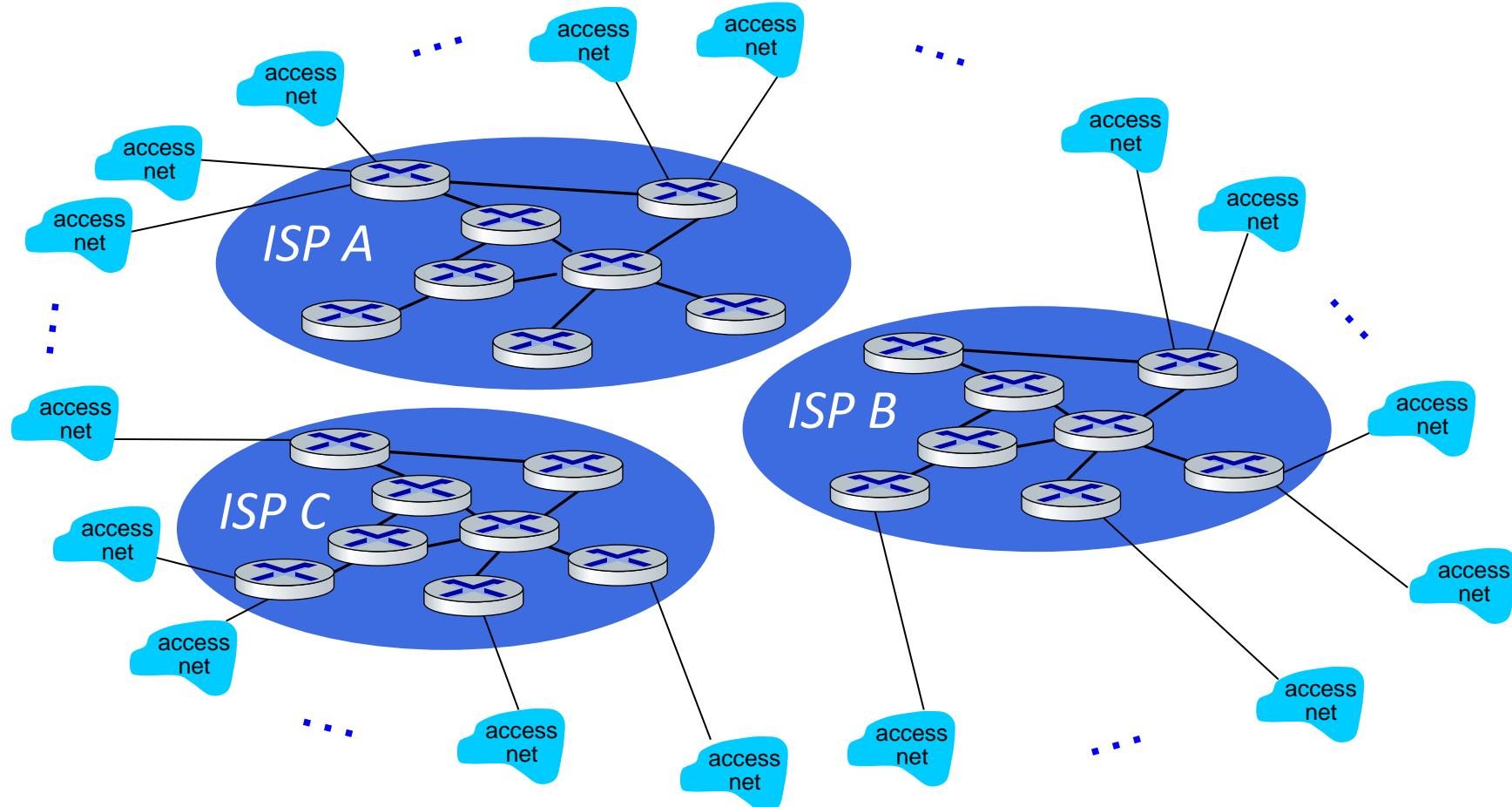
Option: connect each access ISP to one global transit ISP?

Customer and provider ISPs have economic agreement.



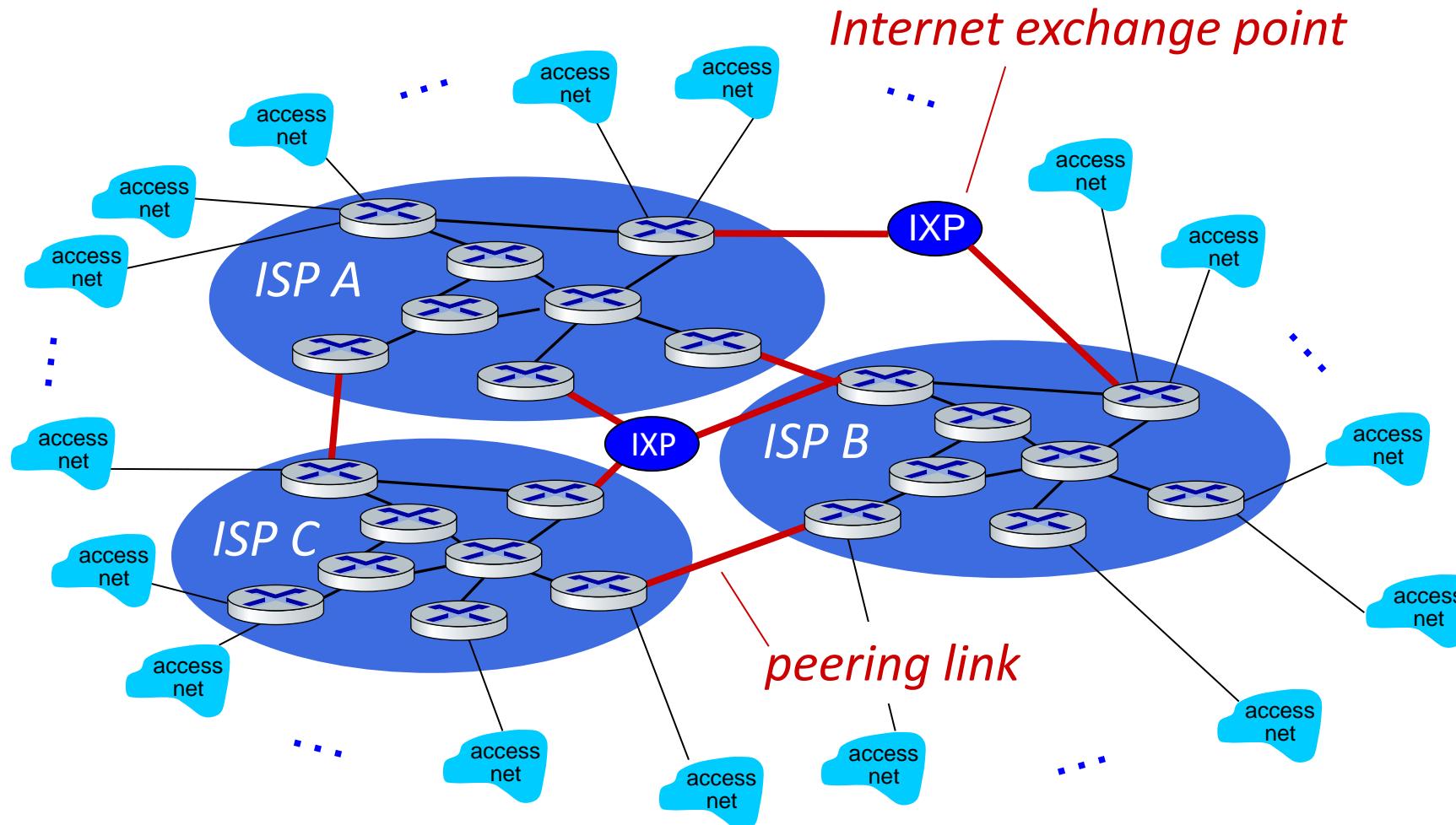
Internet structure: a “network of networks”

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



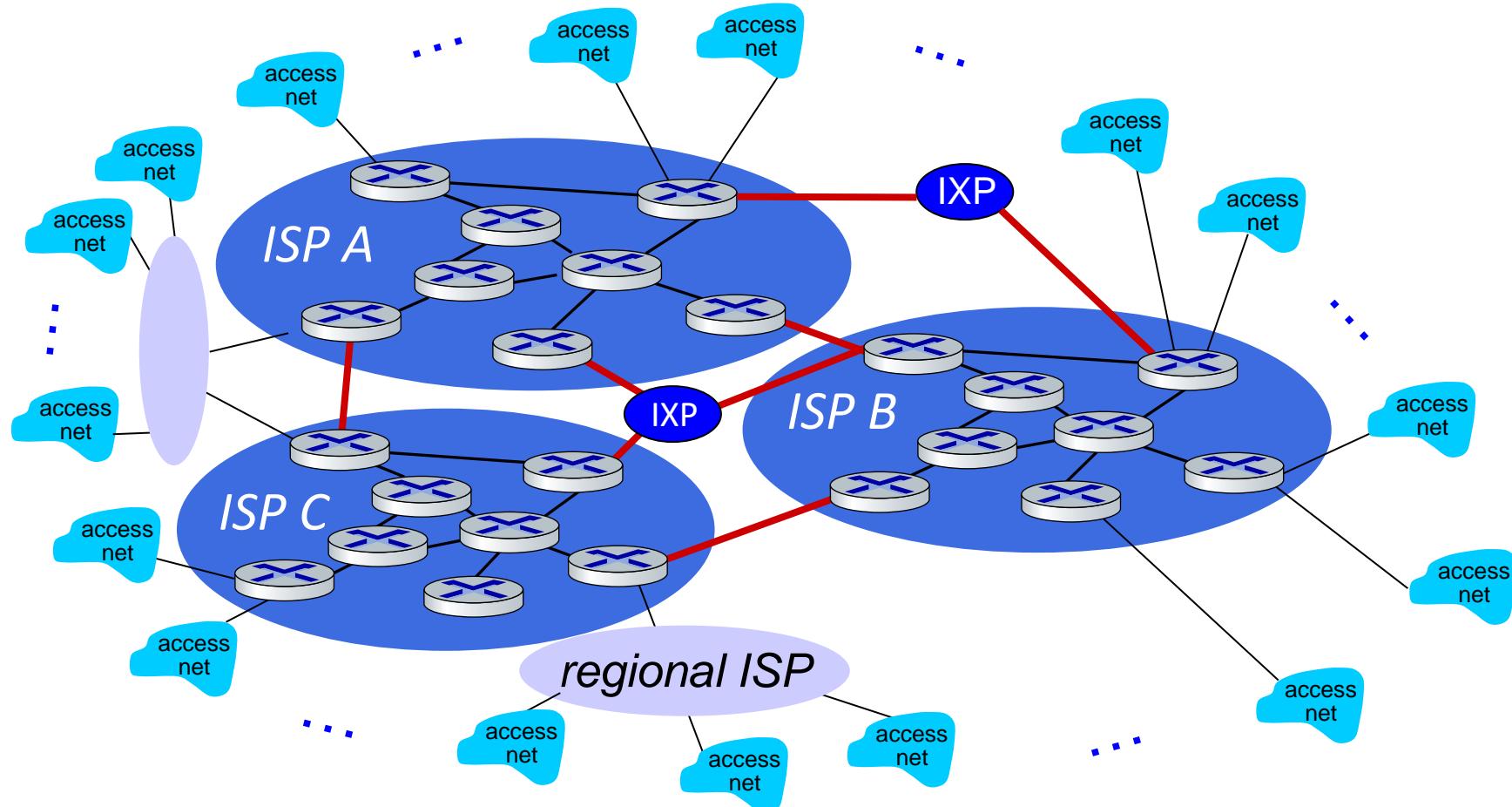
Internet structure: a “network of networks”

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



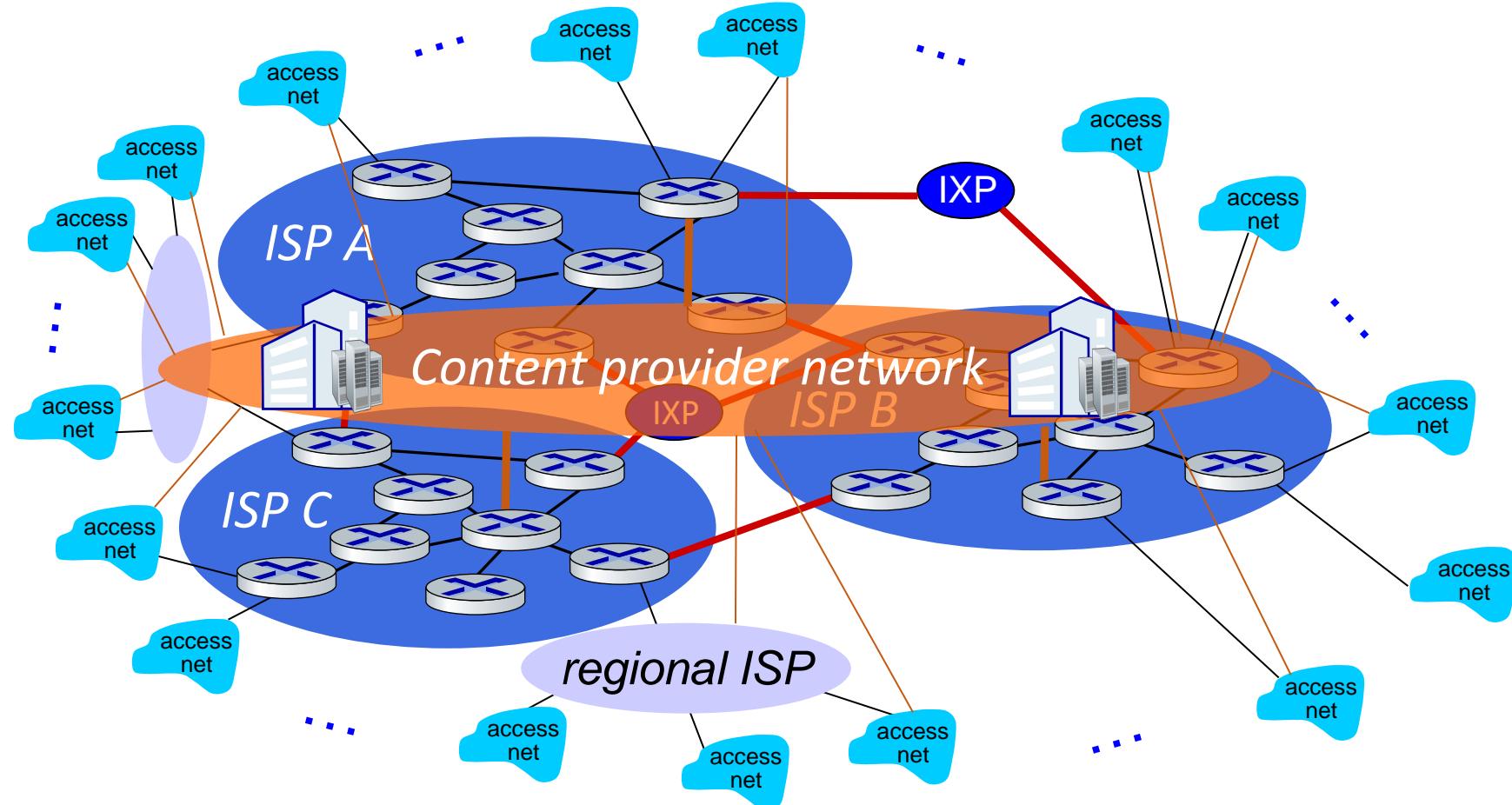
Internet structure: a “network of networks”

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

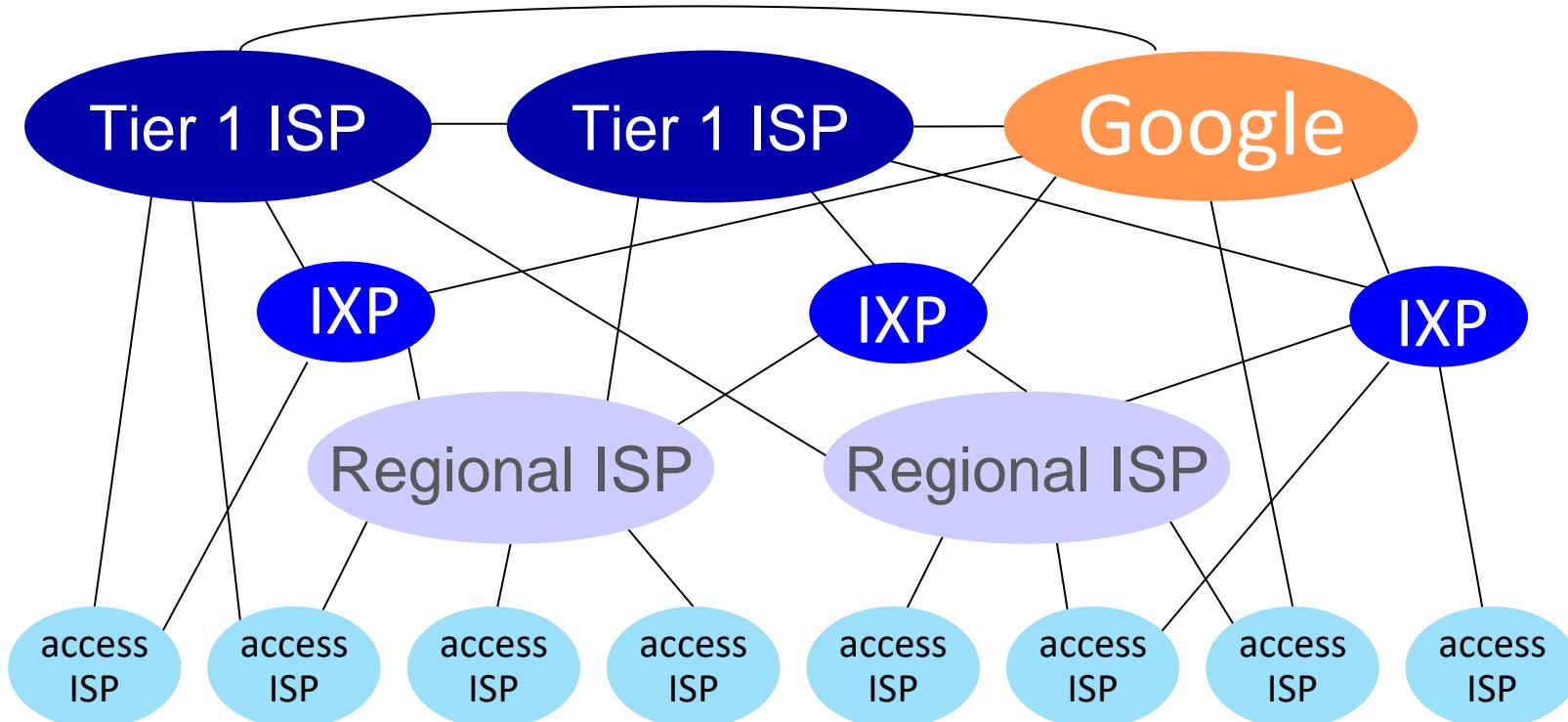


Internet structure: a “network of networks”

... 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

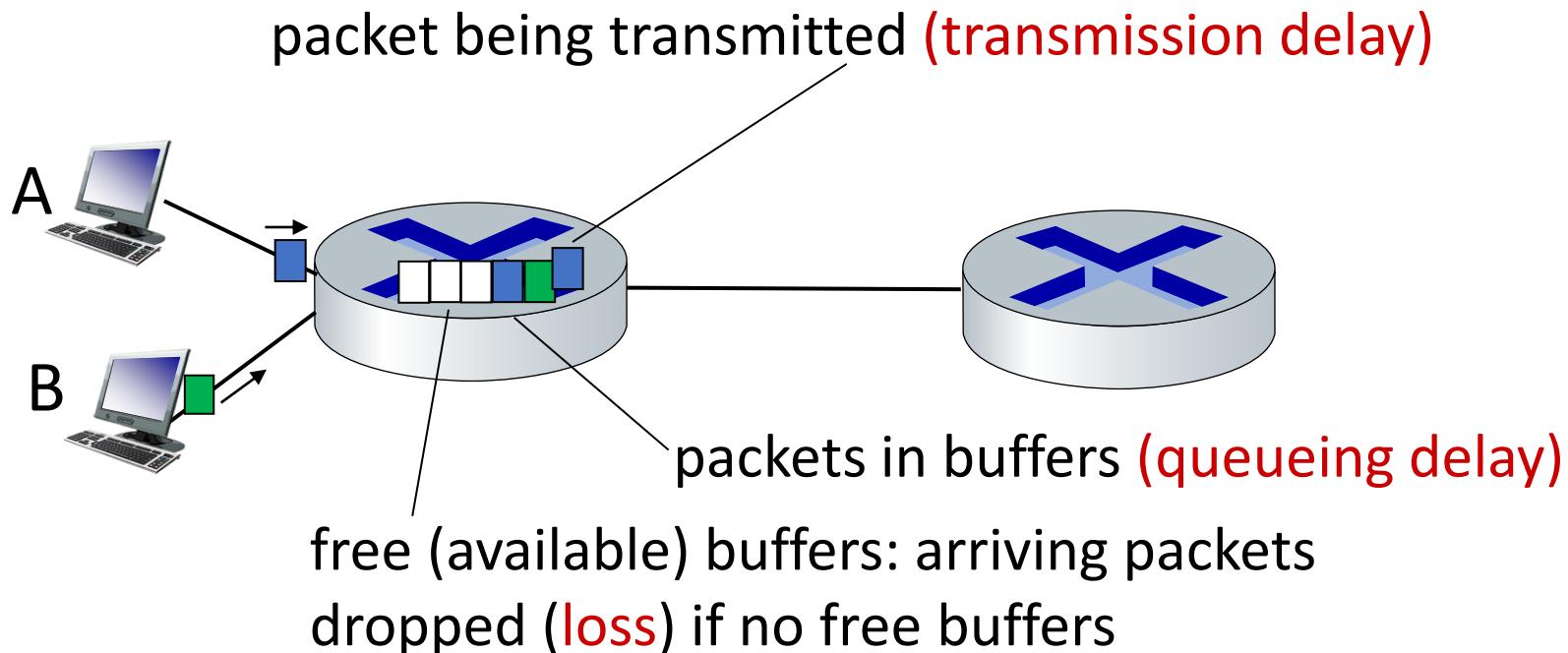
- **“tier-1” commercial ISPs** (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage
- **content provider networks** (e.g., Google, Facebook): private network that connects its data centers to Internet, often bypassing tier-1, regional ISPs

Chapter 1: roadmap

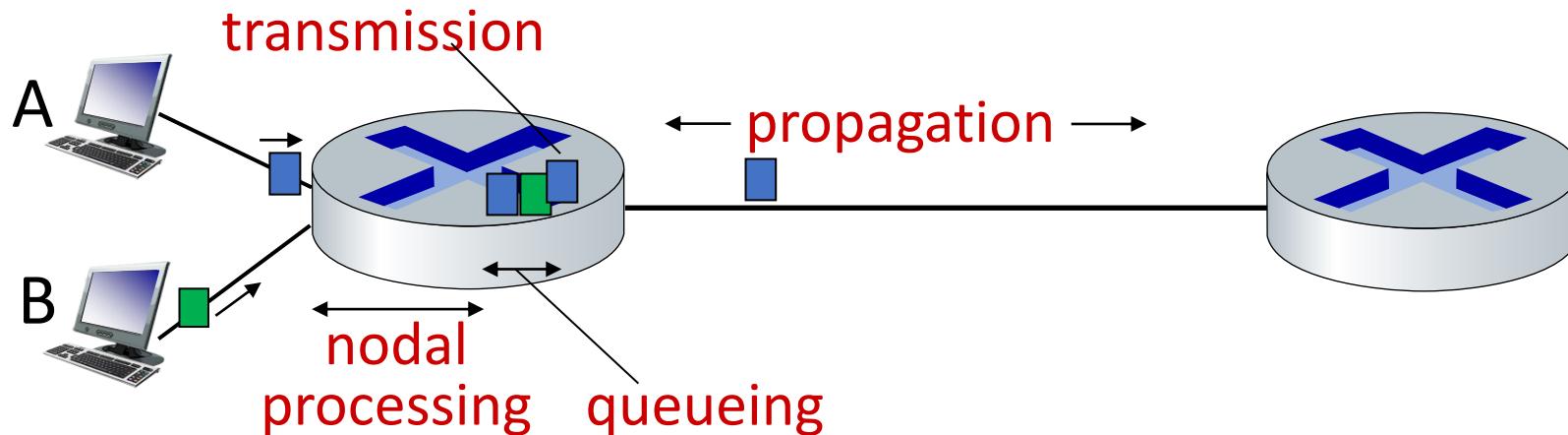
- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- **Performance: loss, delay, throughput**
- Security
- Protocol layers, service models
- History

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 delay: four sources



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

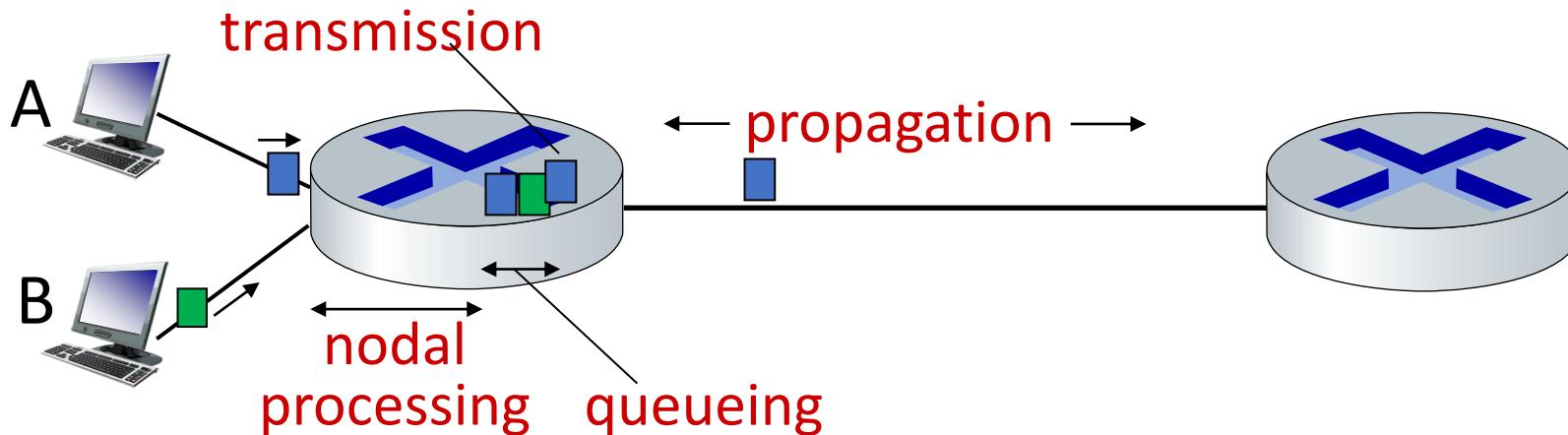
d_{proc} : nodal processing

- 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



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

d_{trans} : transmission delay:

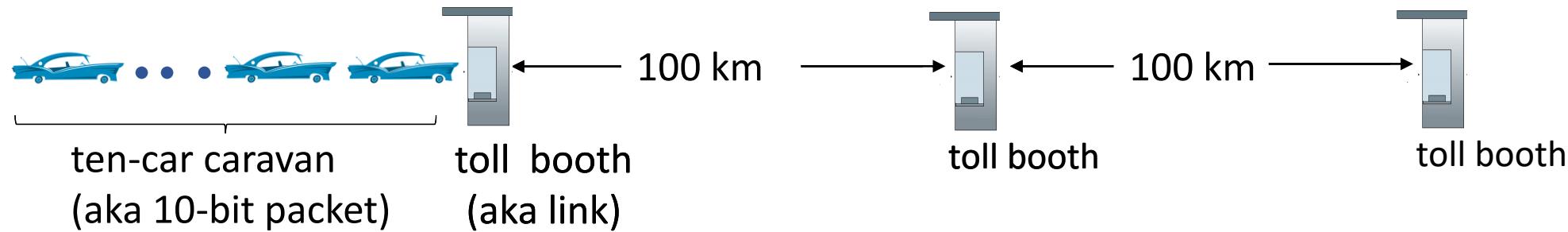
- L : packet length (bits)
- R : link *transmission rate (bps)*
- $d_{\text{trans}} = L/R$

d_{trans} and d_{prop}
very different

d_{prop} : propagation delay:

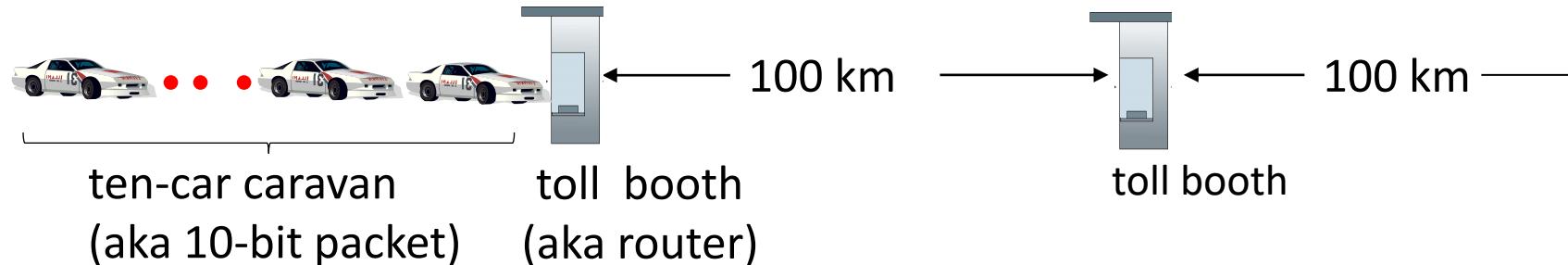
- d : length of physical link
- s : propagation speed ($\sim 2 \times 10^8$ m/sec)
- $d_{\text{prop}} = d/s$

Caravan analogy



- 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: $100\text{km}/(100\text{km/hr}) = 1$ hr
- **A: 62 minutes**

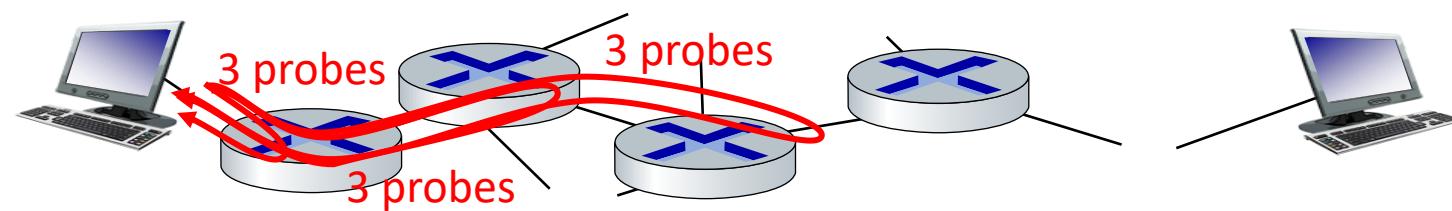
Caravan analogy



- 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

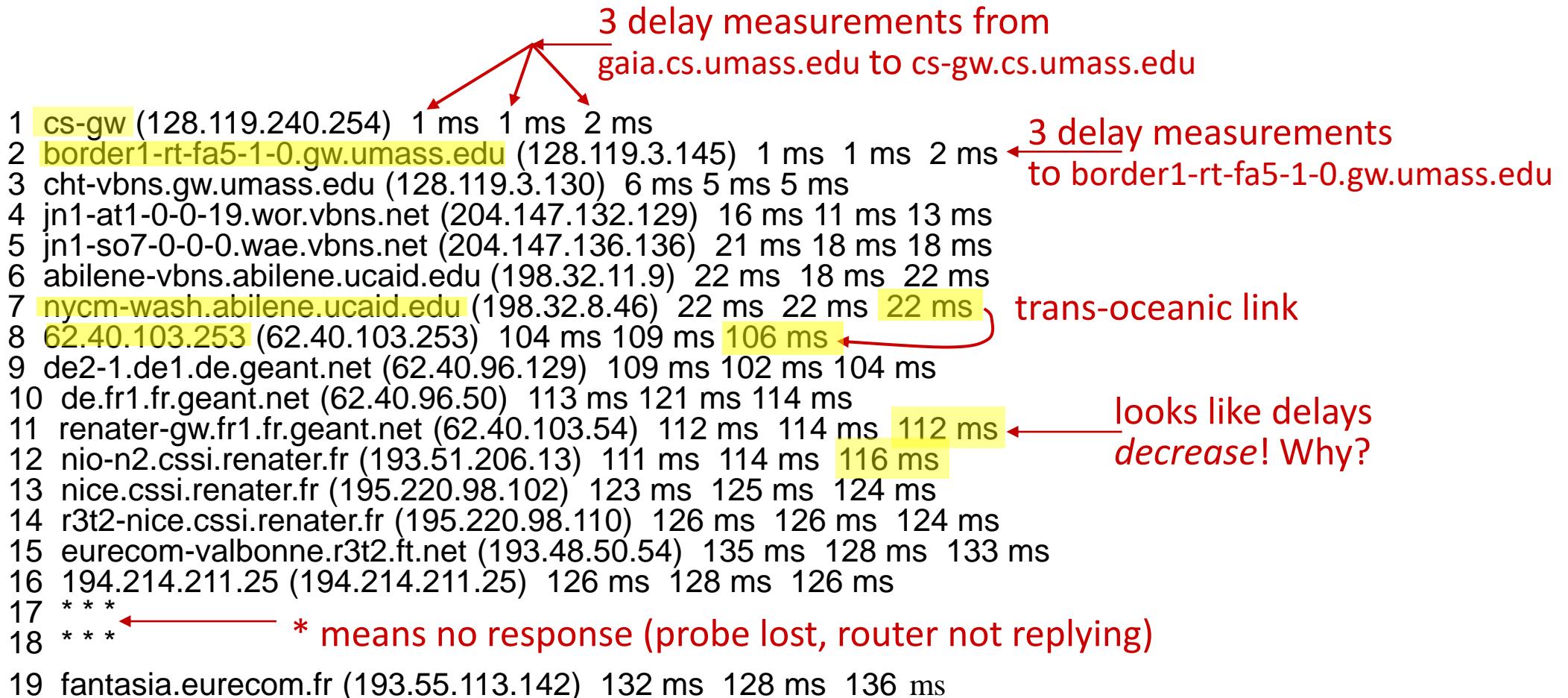
“Real” Internet delays and routes

- 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

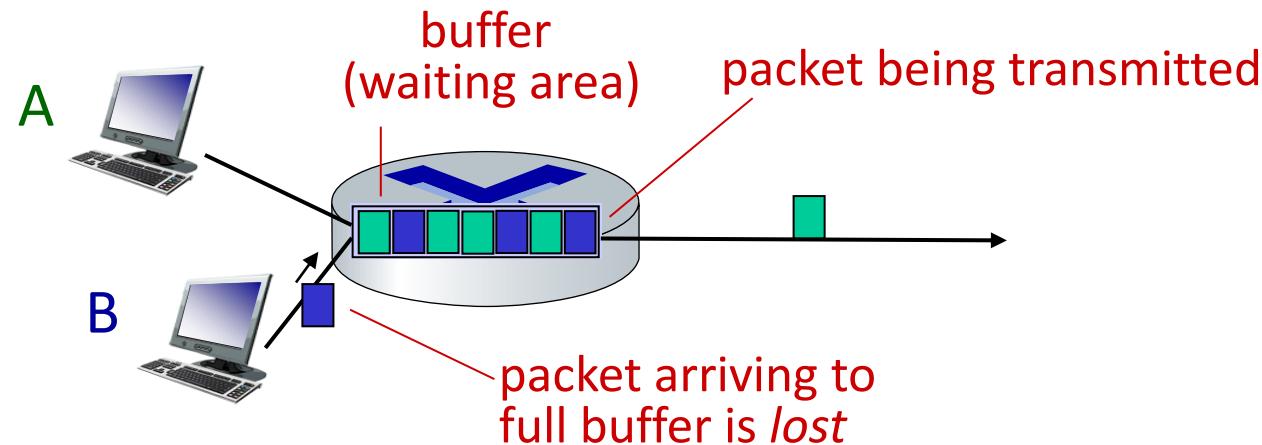
traceroute: gaia.cs.umass.edu to www.eurecom.fr



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

Packet loss

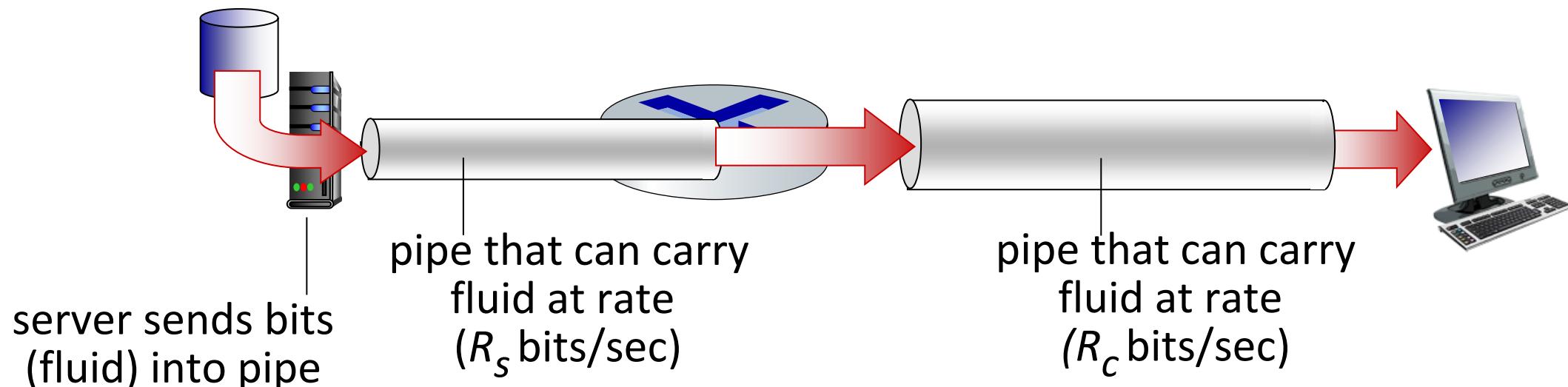
- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



* Check out the Java applet for an interactive animation (on publisher's website) of queuing and loss

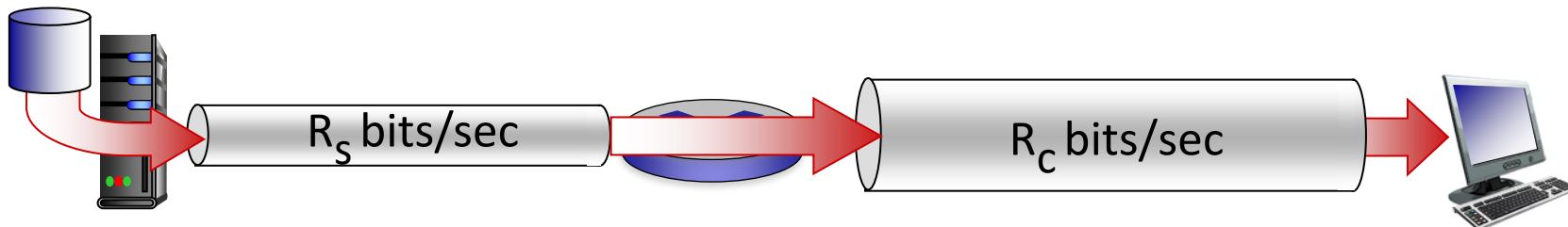
Throughput

- *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

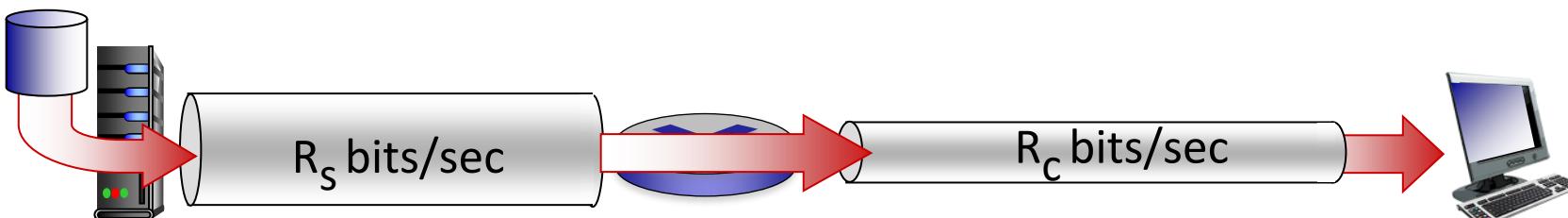


Throughput

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



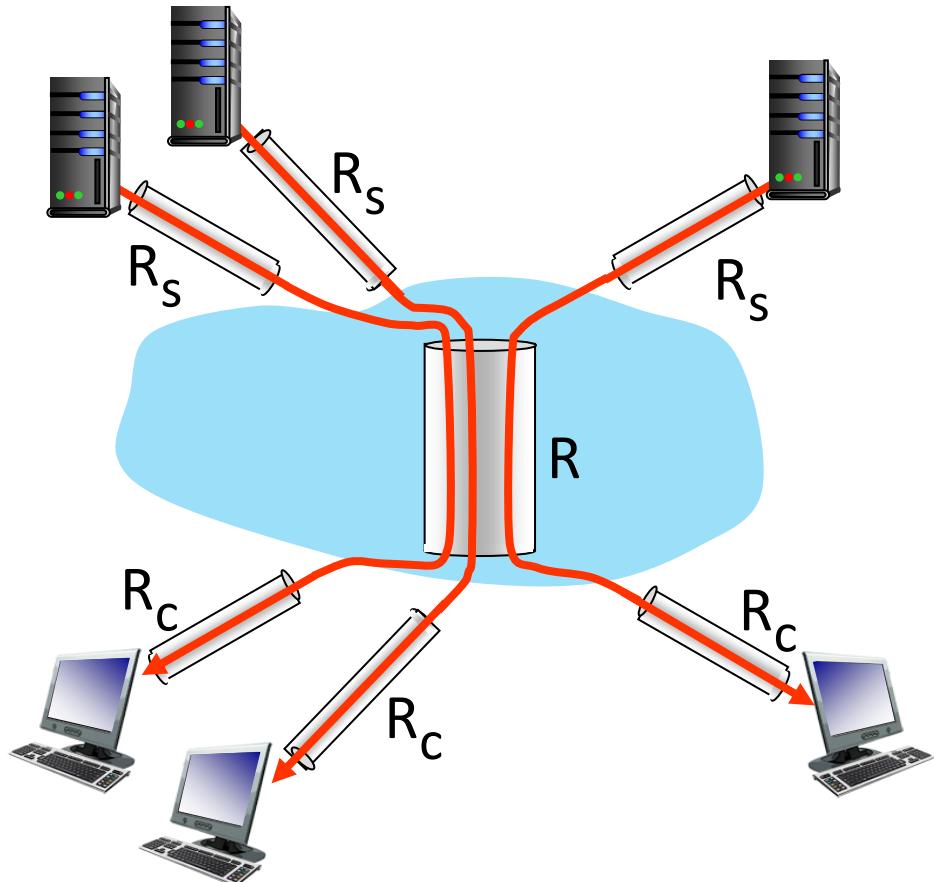
$R_s > R_c$ What is average end-end throughput?



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



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

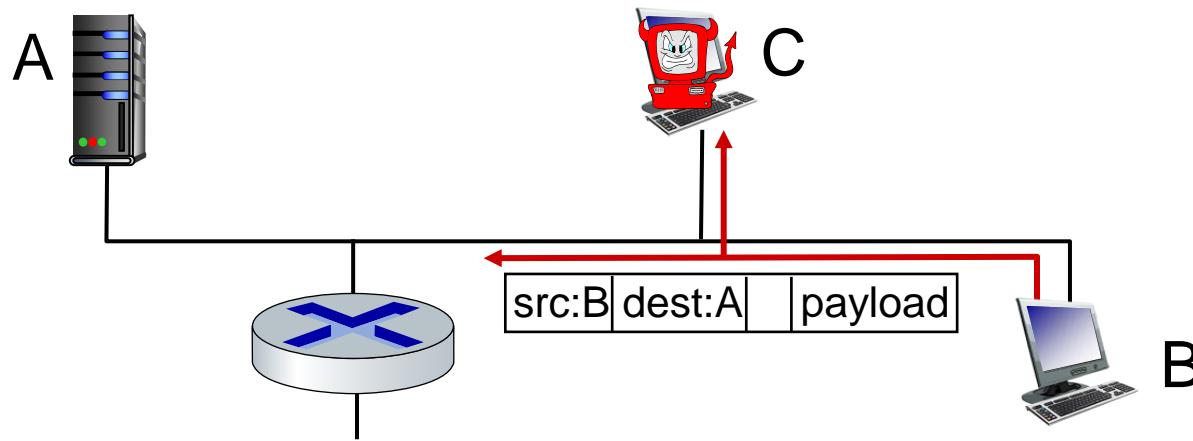
Network security

- 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: packet interception

packet “sniffing”:

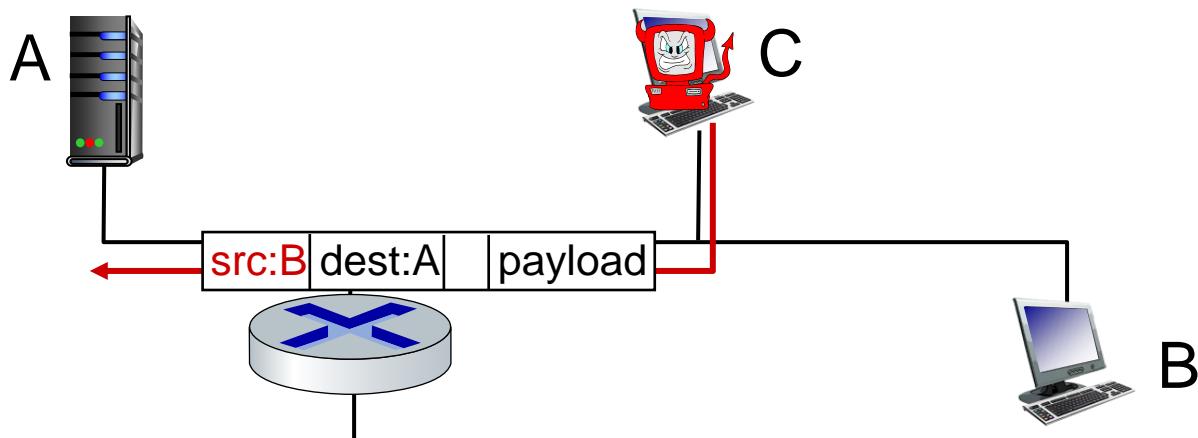
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



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

Bad guys: fake identity

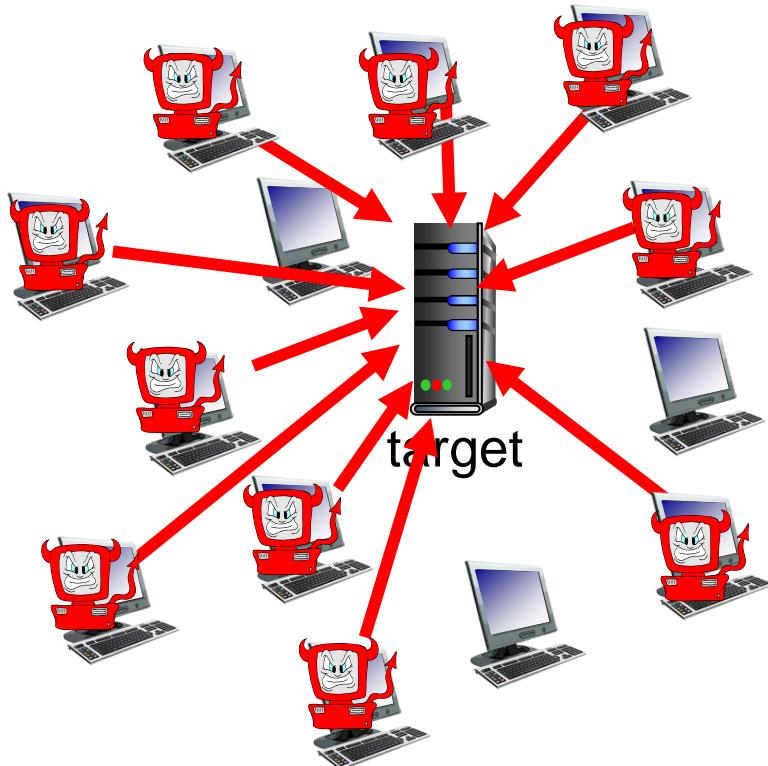
IP spoofing: injection of packet with false source address



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



Lines of defense:

- **authentication**: proving you are who you say you are
 - cellular networks provides hardware identity via SIM card; no such hardware assist in traditional Internet
- **confidentiality**: via encryption
- **integrity checks**: digital signatures prevent/detect tampering
- **access restrictions**: password-protected VPNs
- **firewalls**: specialized “middleboxes” in access and core networks:
 - off-by-default: filter incoming packets to restrict senders, receivers, applications
 - detecting/reacting to DOS attacks

... 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

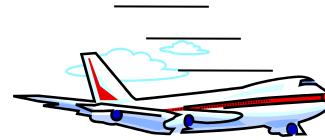
Networks are complex,
with many “pieces”:

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

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

- and/or our *discussion*
of networks?

Example: organization of air travel



— *end-to-end transfer of person plus baggage* —→

ticket (purchase)

baggage (check)

gates (load)

runway takeoff

airplane routing

ticket (complain)

baggage (claim)

gates (unload)

runway landing

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



layers: each layer implements a service

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

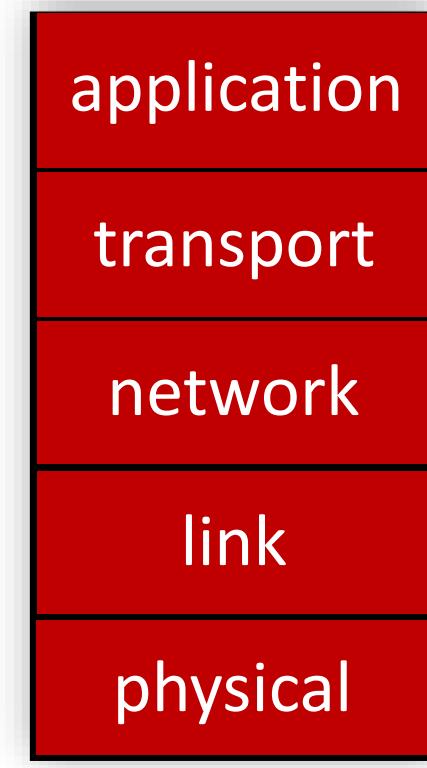
Why layering?

Approach to designing/discussing complex systems:

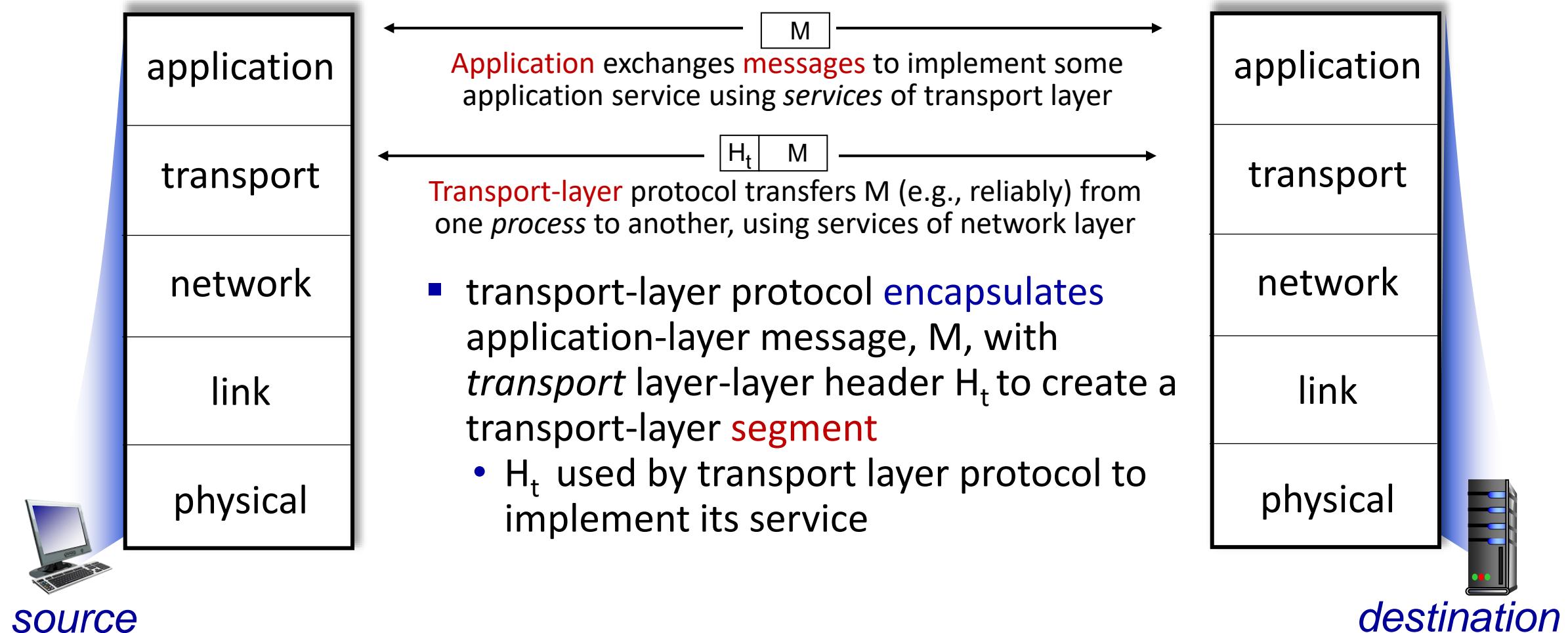
- explicit structure allows identification, relationship of system's pieces
 - layered *reference model* for discussion
- modularization eases maintenance, updating of system
 - change in layer's service *implementation*: transparent to rest of system
 - e.g., change in gate procedure doesn't affect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

Layered Internet protocol stack

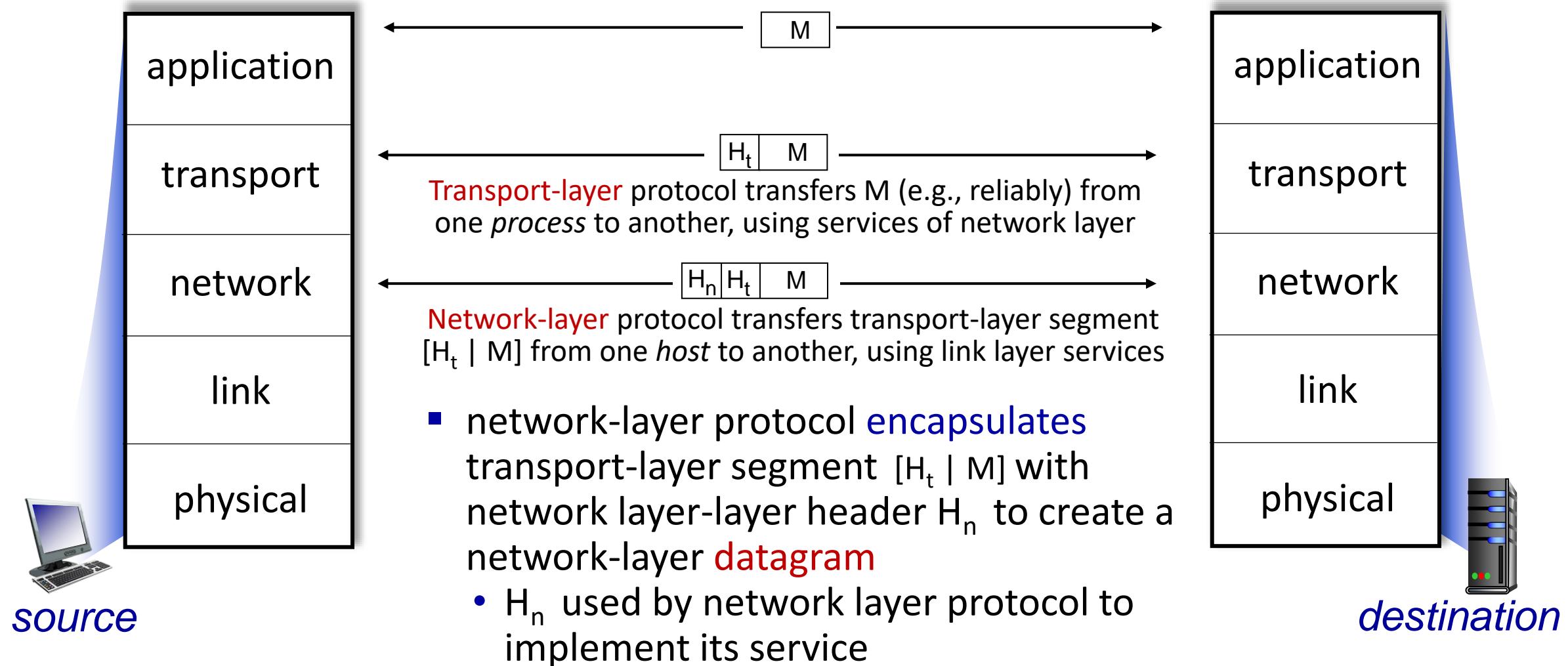
- *application*: supporting network applications
 - HTTP, IMAP, SMTP, DNS
- *transport*: process-process data transfer
 - TCP, UDP
- *network*: routing of datagrams from source to destination
 - IP, routing protocols
- *link*: data transfer between neighboring network elements
 - Ethernet, 802.11 (WiFi),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



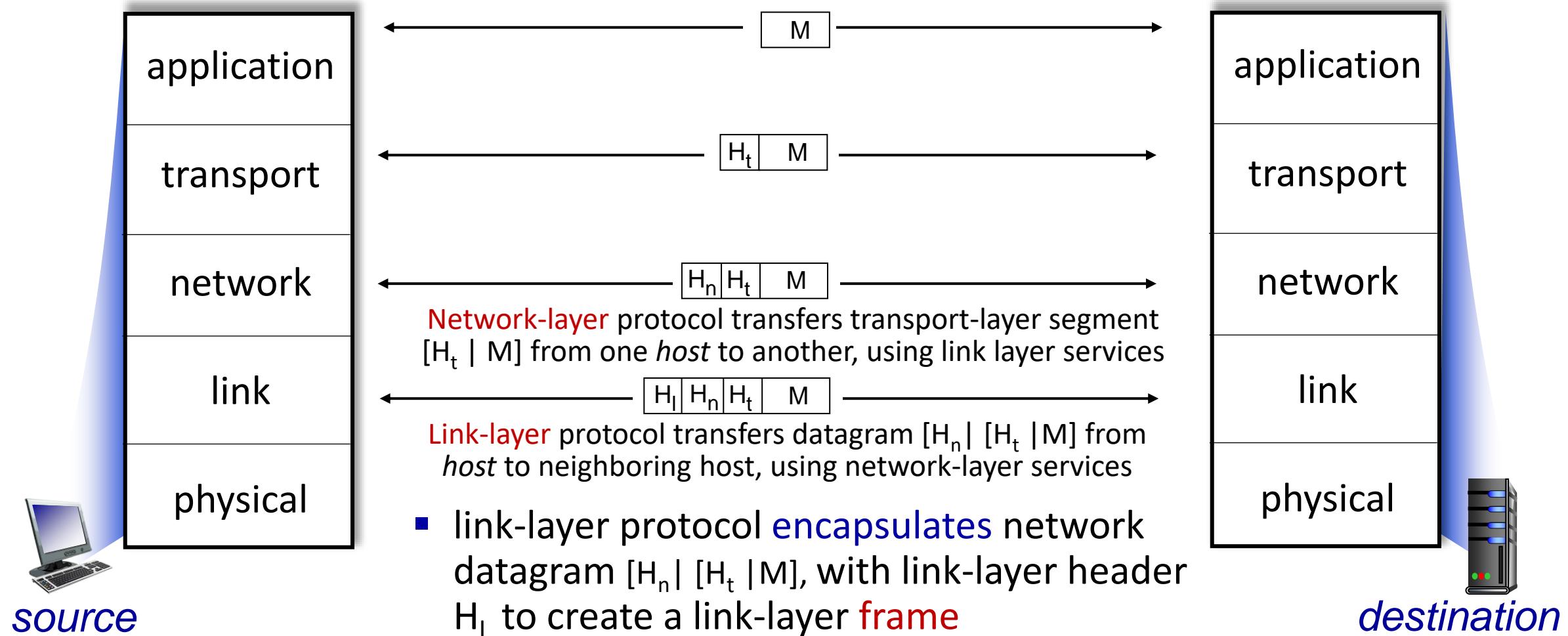
Services, Layering and Encapsulation



Services, Layering and Encapsulation

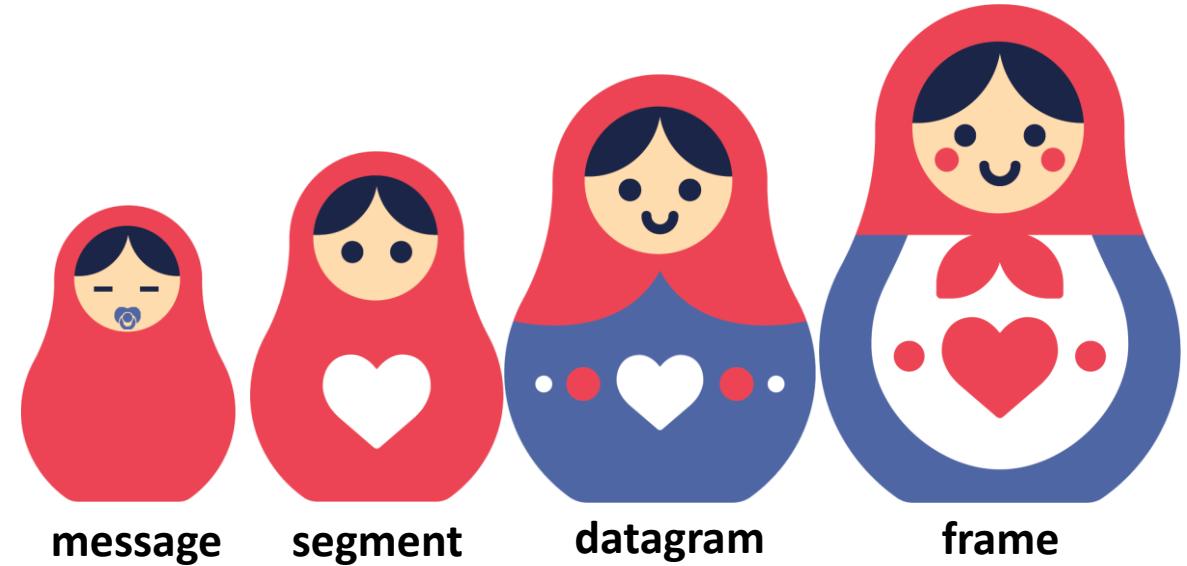


Services, Layering and Encapsulation

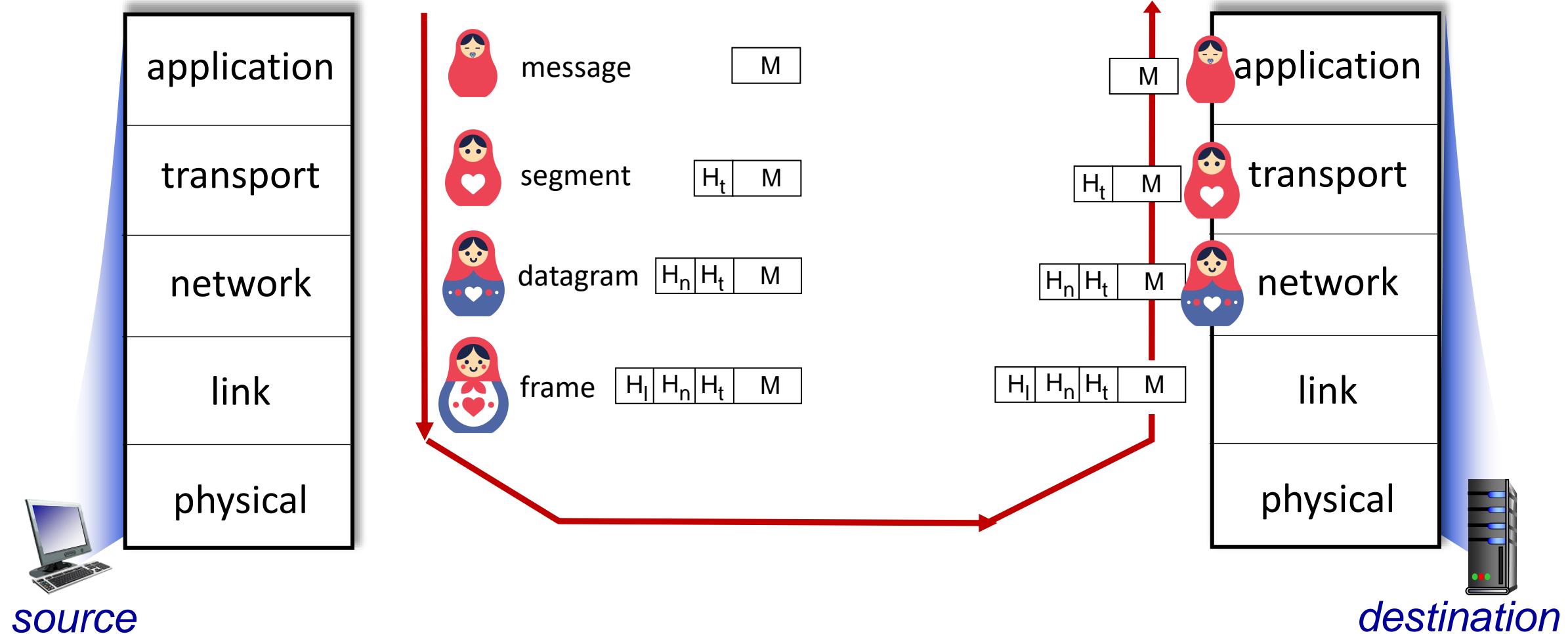


Encapsulation

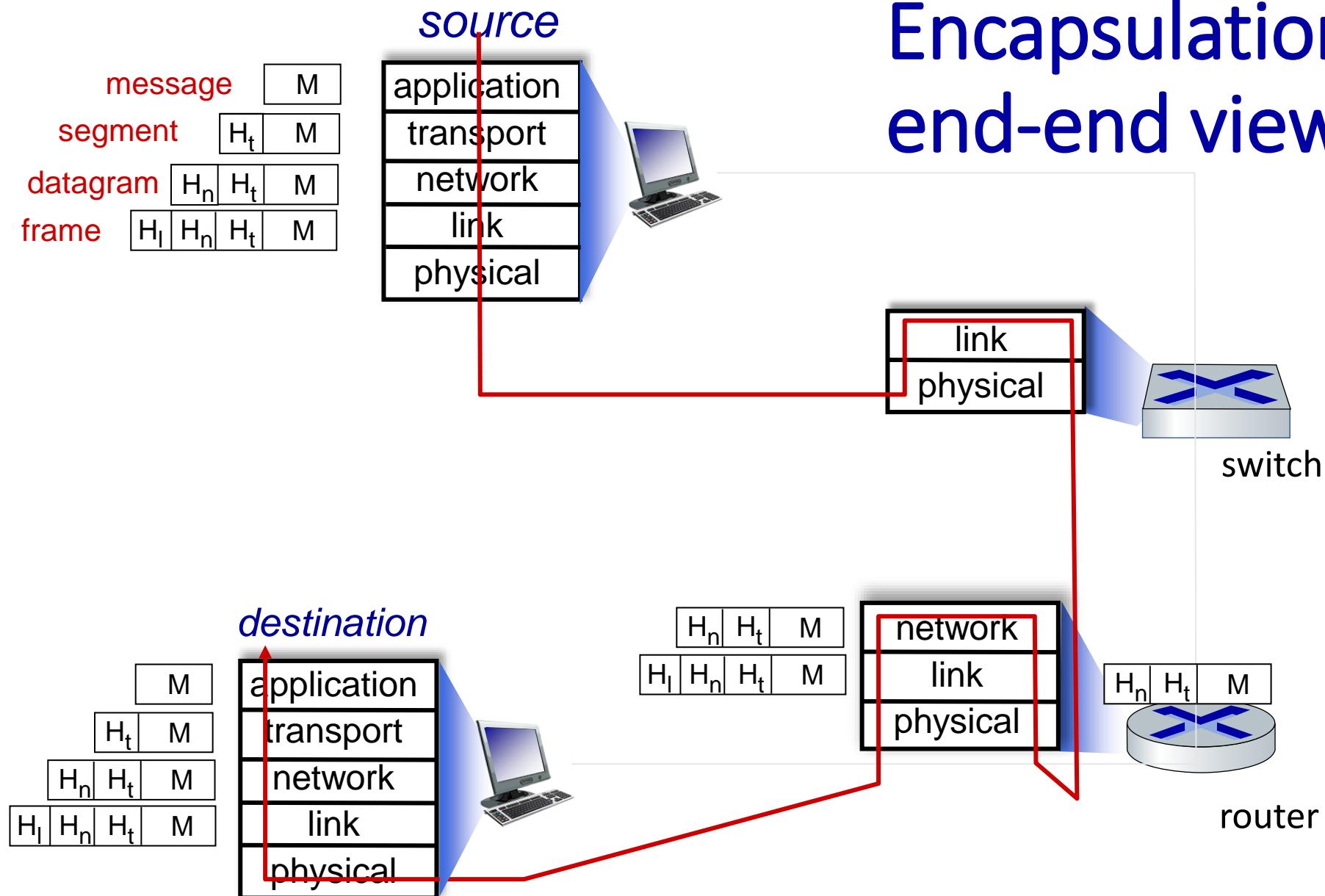
Matryoshka dolls (stacking dolls)



Services, Layering and Encapsulation



Encapsulation: an end-end view



Chapter 1: roadmap

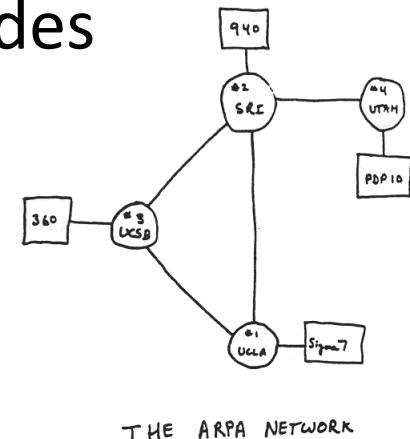
- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History



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 networks

- 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
- 1979: ARPAnet has 200 nodes

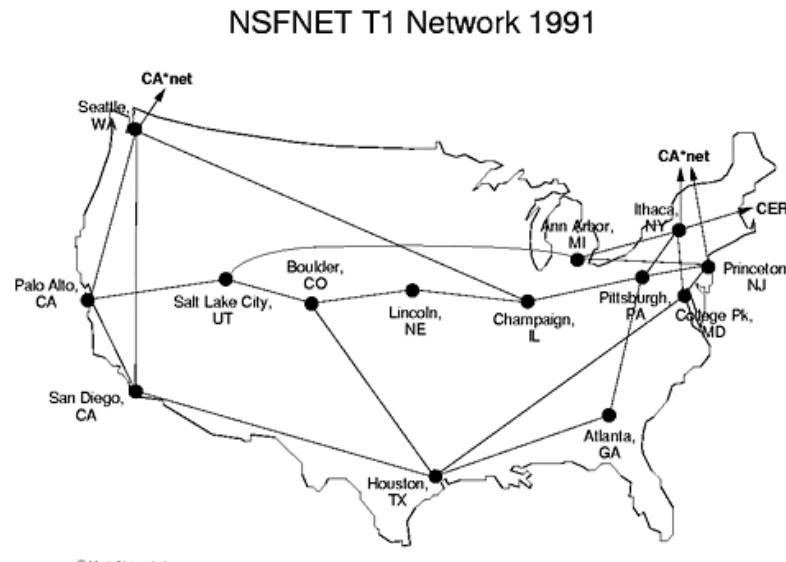
Cerf and Kahn's internetworking principles:

- 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



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: scale, SDN, mobility, cloud

- aggressive deployment of broadband home access (10-100's Mbps)
- 2008: software-defined networking (SDN)
- increasing ubiquity of high-speed wireless access: 4G/5G, WiFi
- service providers (Google, FB, Microsoft) create their own networks
 - bypass commercial Internet to connect “close” to end user, providing “instantaneous” access to social media, search, video content, ...
- enterprises run their services in “cloud” (e.g., Amazon Web Services, Microsoft Azure)
- rise of smartphones: more mobile than fixed devices on Internet (2017)
- ~15B devices attached to Internet (2023, statista.com)

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:

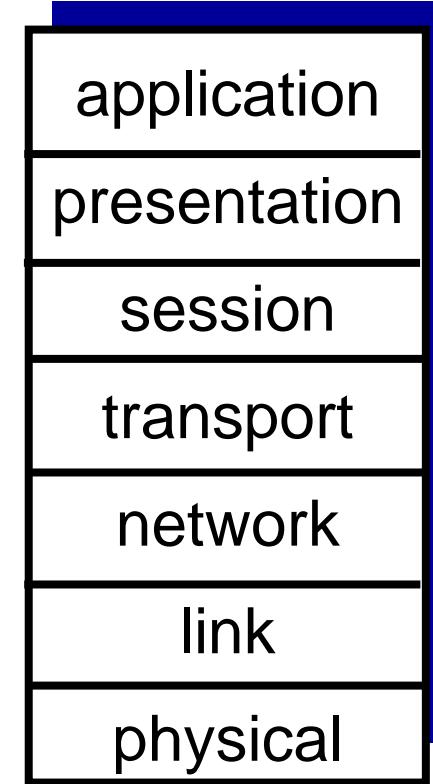
- context, overview, vocabulary, "feel" of networking
- more depth, detail, *and fun* to follow!

Additional Chapter 1 slides

ISO/OSI reference model

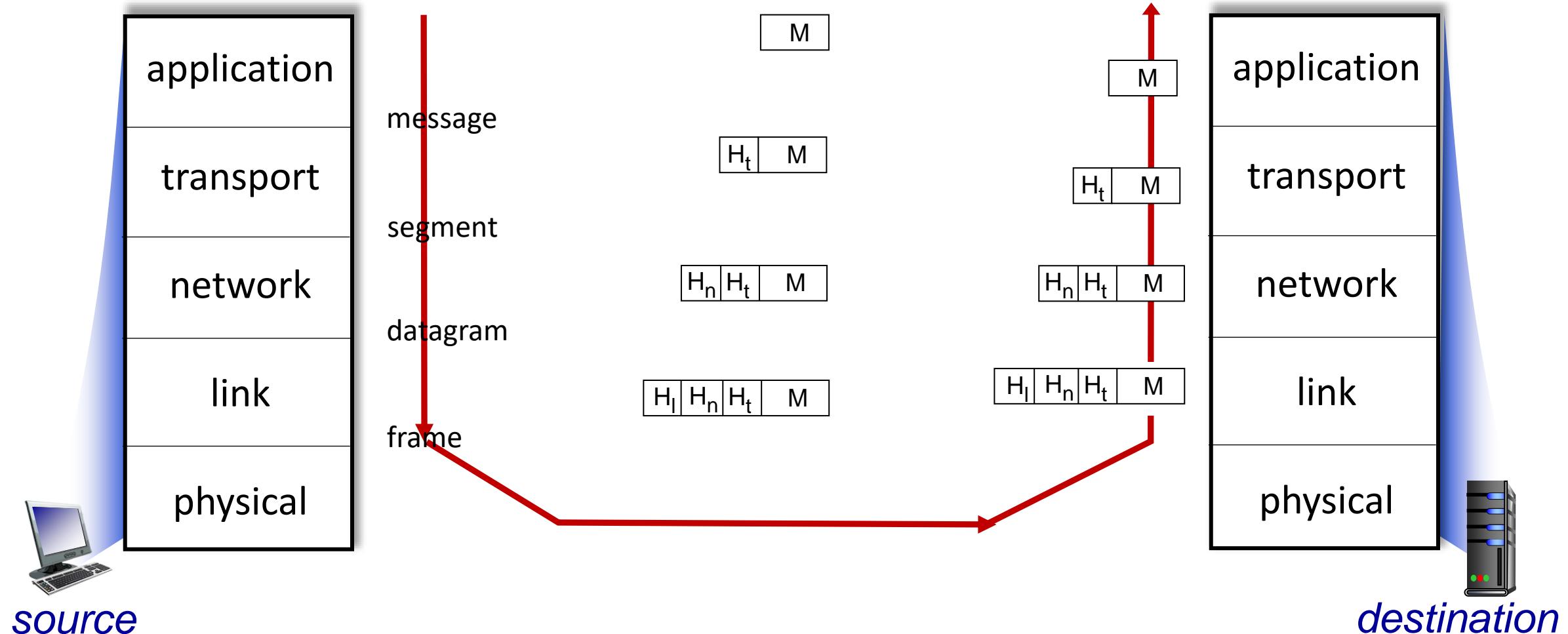
Two layers not found in Internet protocol stack!

- *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?



The seven layer OSI/ISO reference model

Services, Layering and Encapsulation



Wireshark

