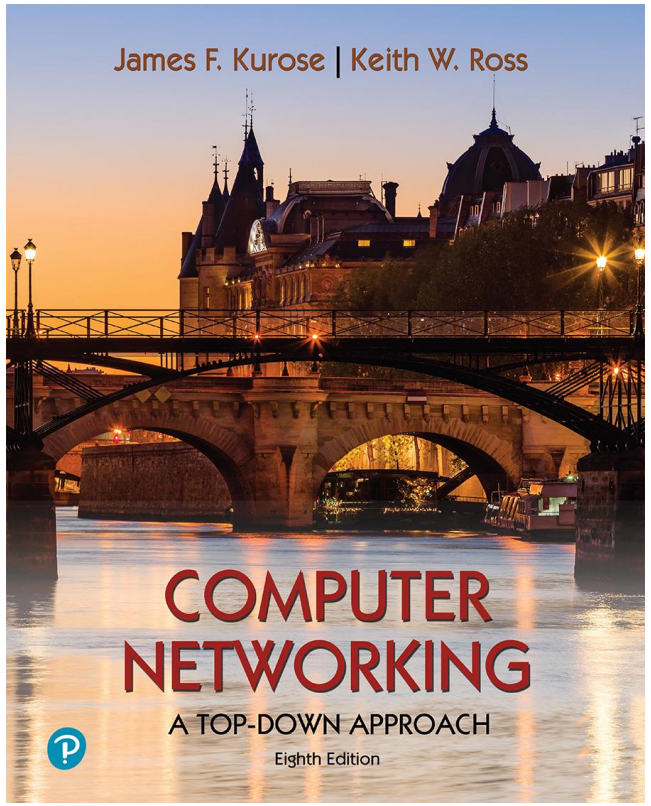
Chapter 1 Introduction



Computer Networking: A Top-Down Approach

8th edition
Jim Kurose, Keith Ross
Pearson, 2020

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

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)

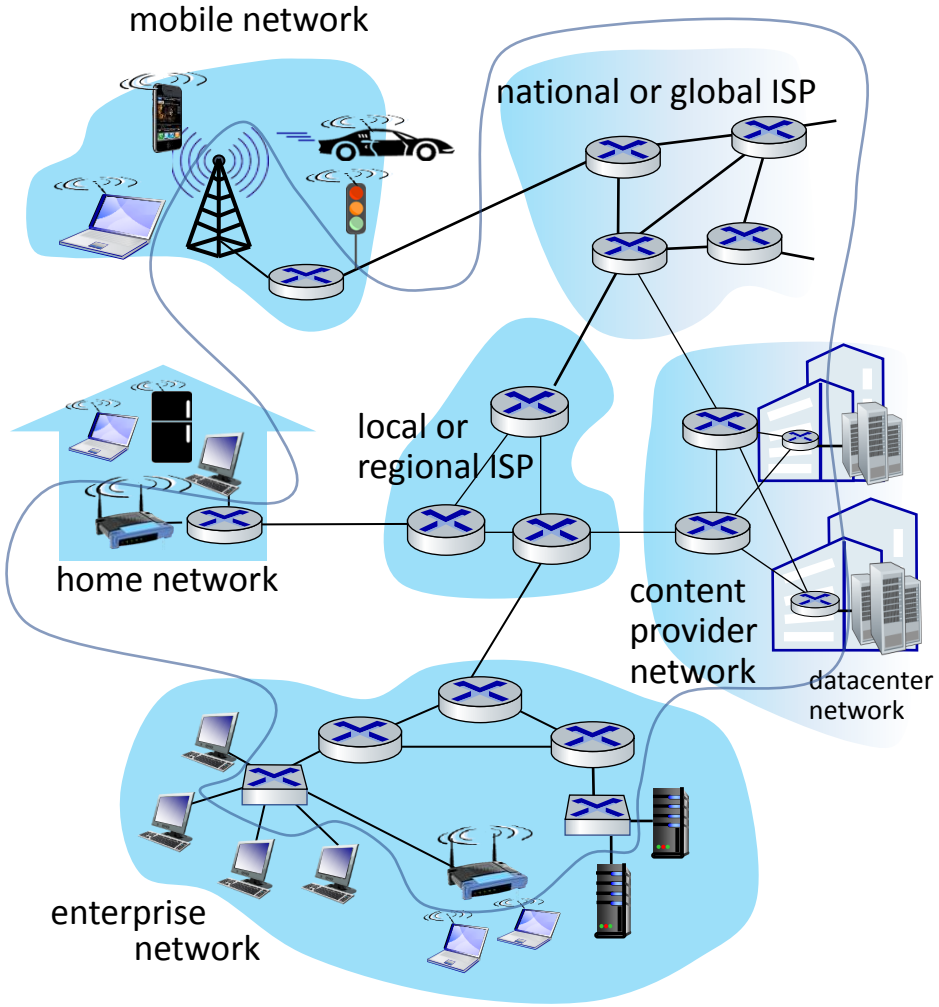
- *routers, switches*

Communication links

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

Networks

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



“Fun” Internet-connected devices

Amazon Echo

Internet refrigerator

IP picture frame

Pacemaker & Monitor

Tweet-a-watt: monitor energy use

bikes

cars

scooters

Security Camera

Slingbox: remote control cable TV

Web-enabled toaster + weather forecaster

AR devices

Internet phones

Gaming devices

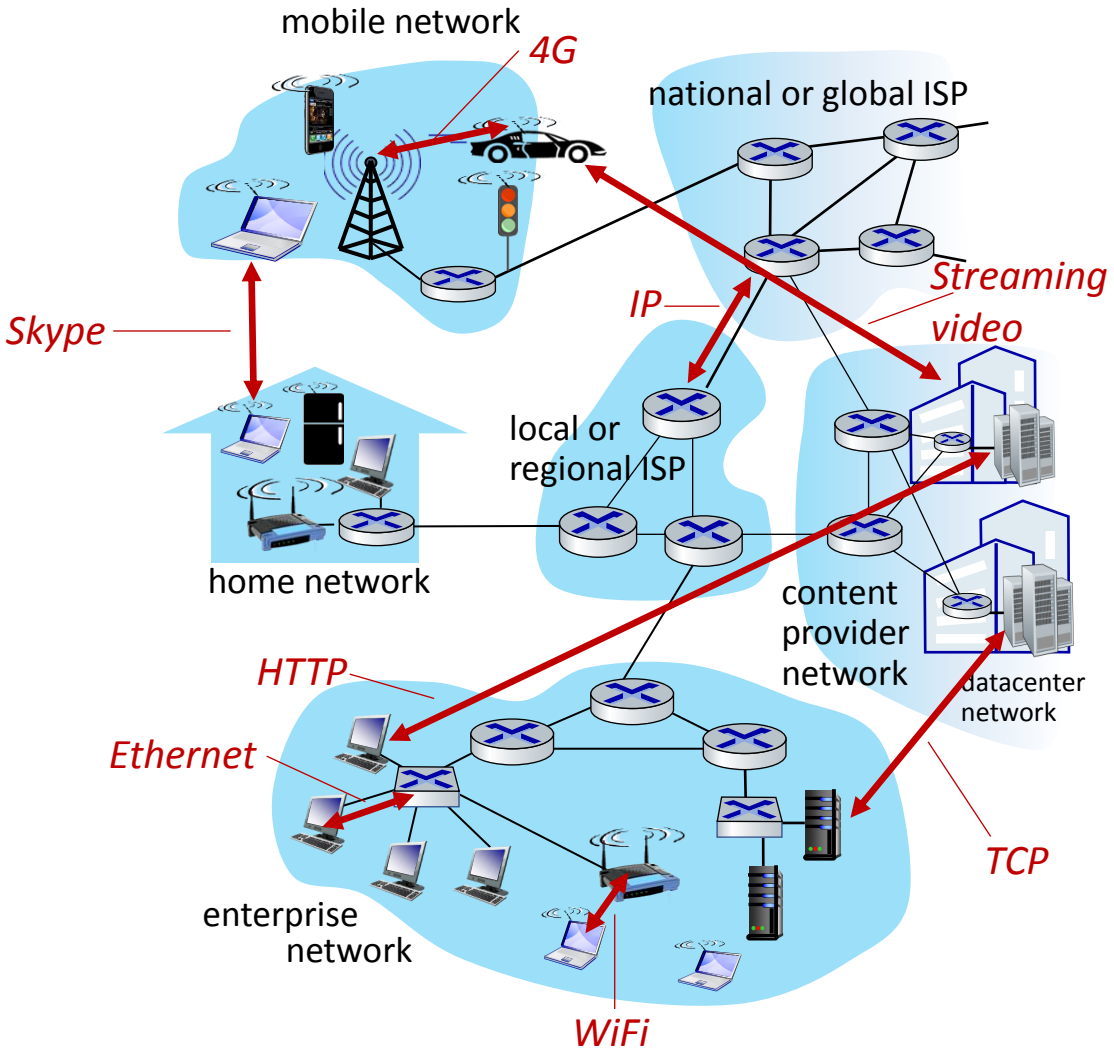
sensorized, bed mattress

Fitbit

Others?

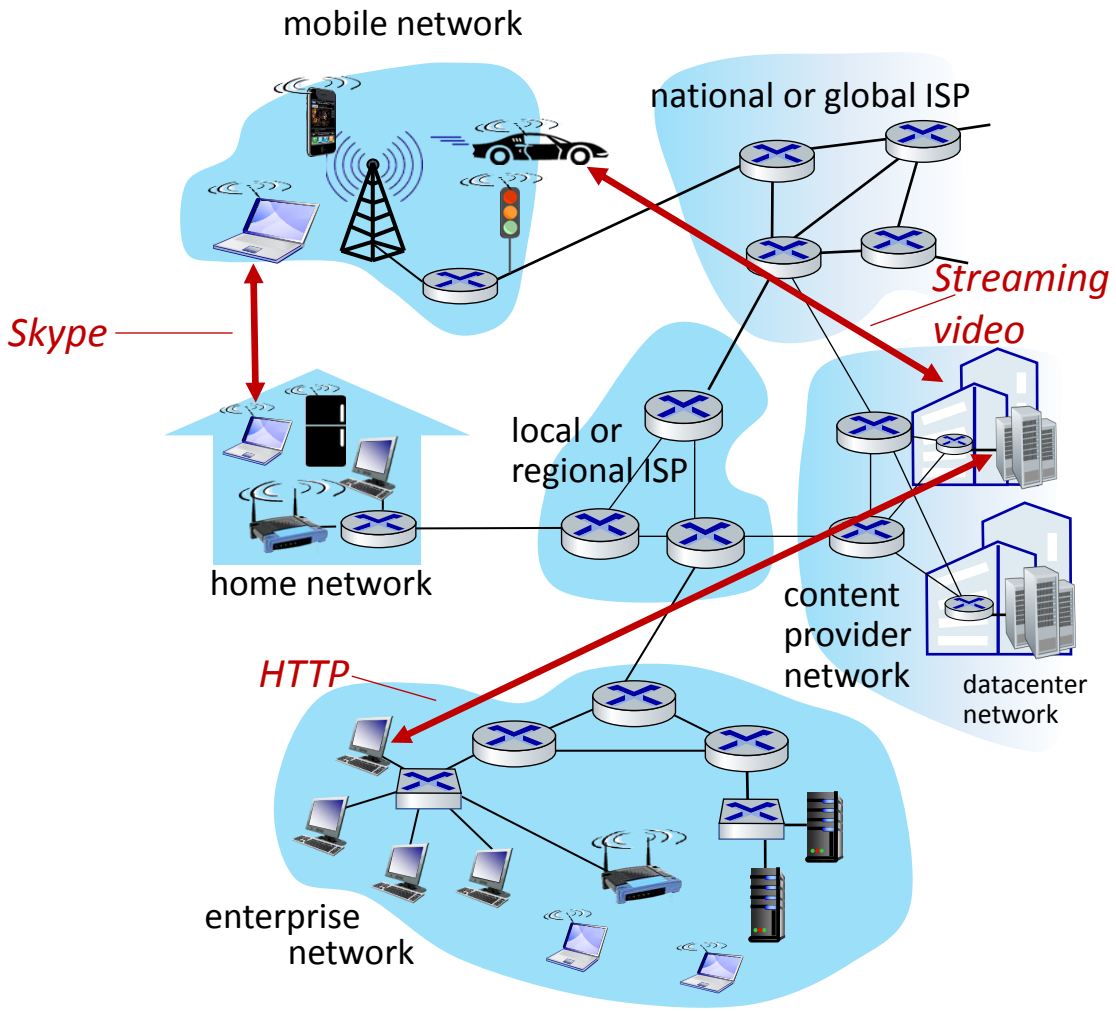
The Internet: a "nuts and bolts" view

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



The Internet: a "services" view

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



What's a protocol?

Human protocols:

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

Rules for:

… specific messages sent

… specific actions taken when message received, or other events

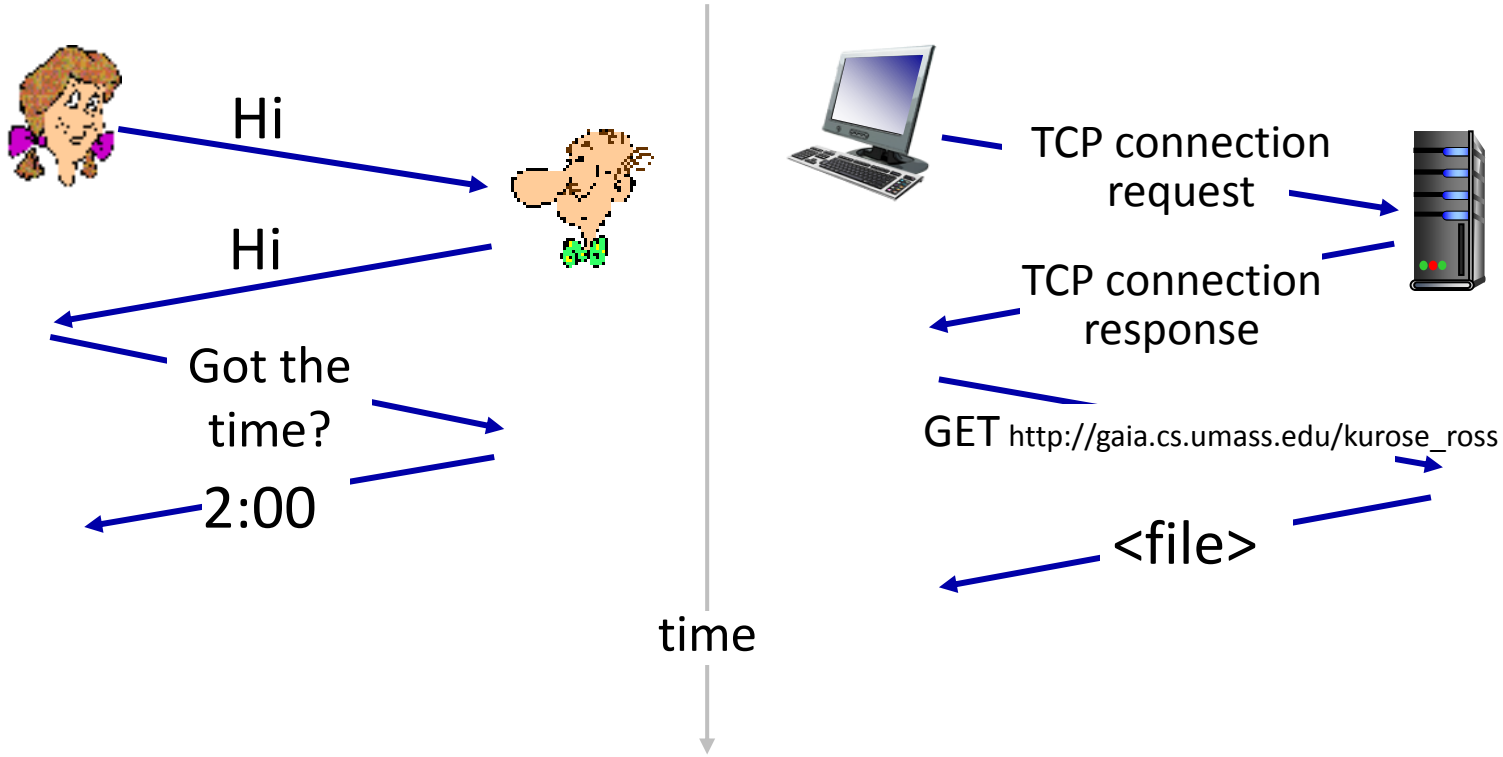
Network protocols:

- 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

What's a protocol?

A human protocol and a computer network protocol:



Q: other human protocols?

Chapter 1: roadmap

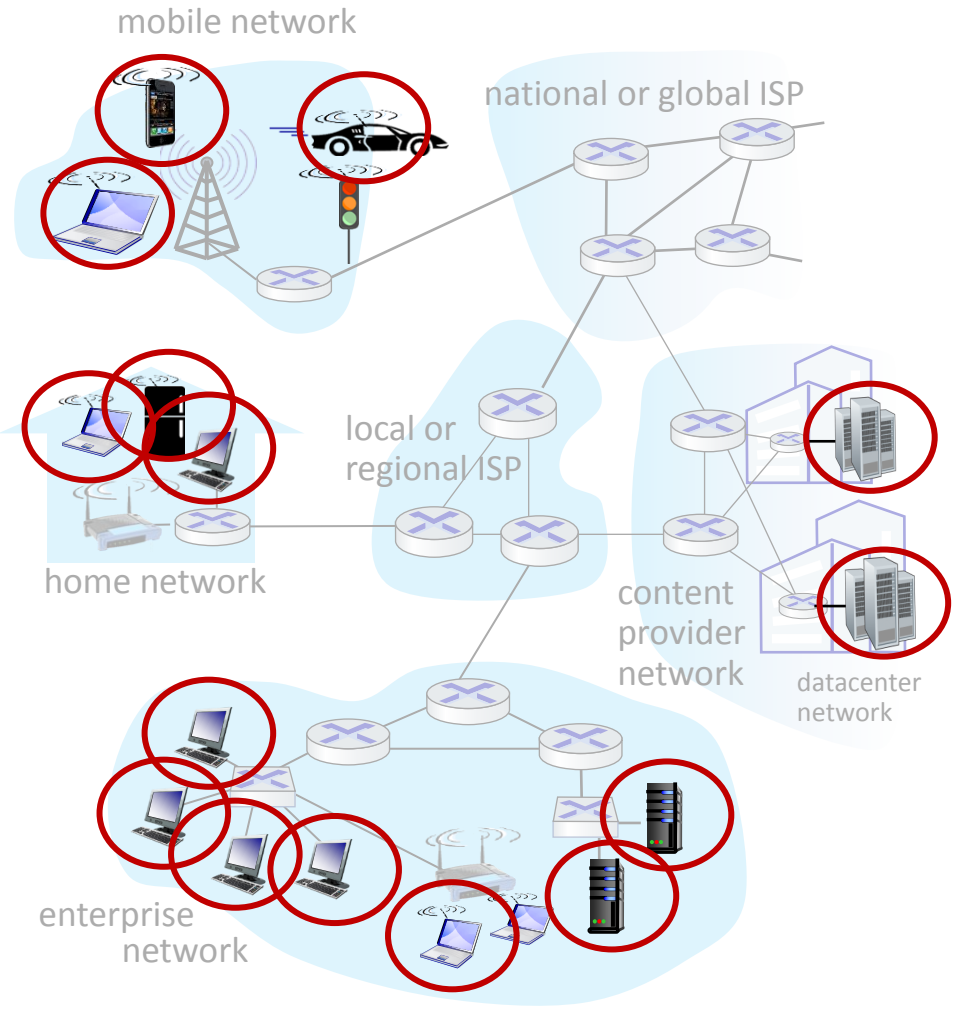
- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

A closer look at Internet structure

Network edge:

End-systems/hosts:

- clients and servers
- servers often in data centers



A closer look at Internet structure

Network edge:

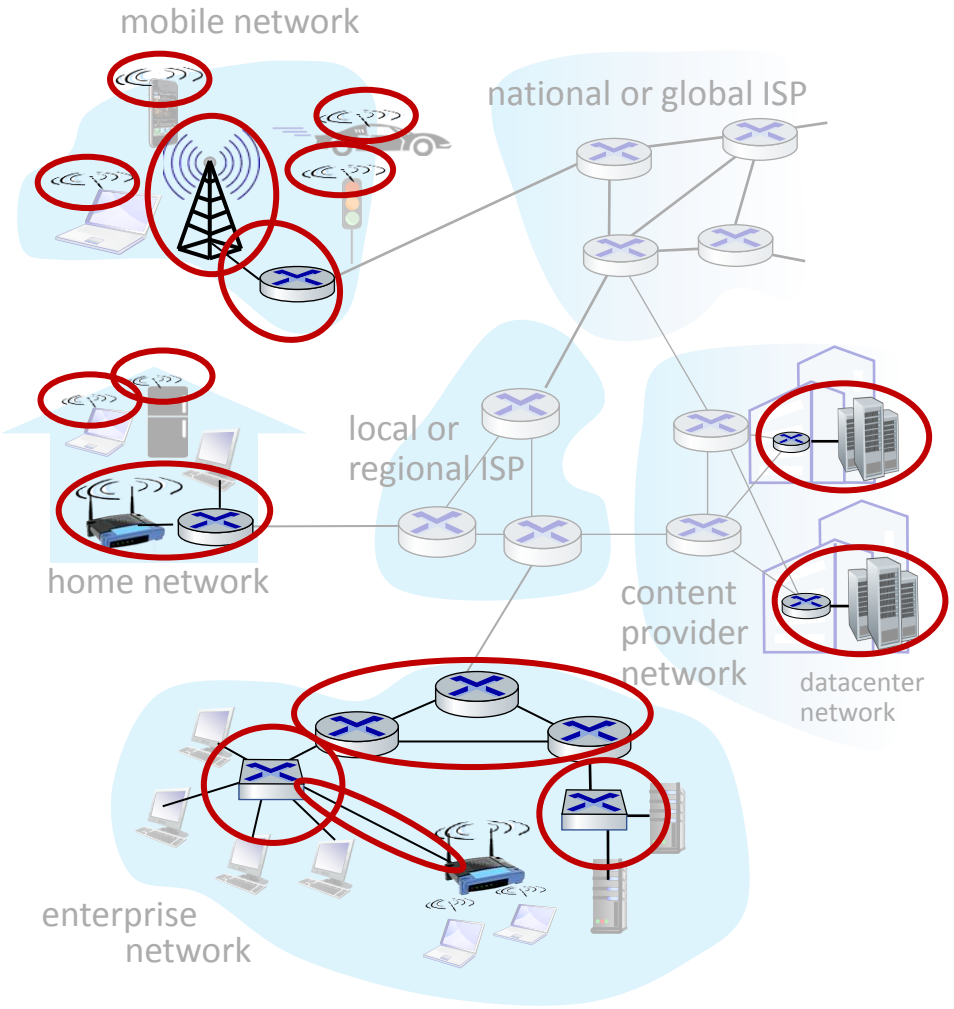
End-systems/hosts

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

Access networks: connects hosts to the “edge router”

wired, wireless communication links

Switches



A closer look at Internet structure

Network edge:

End-systems/hosts

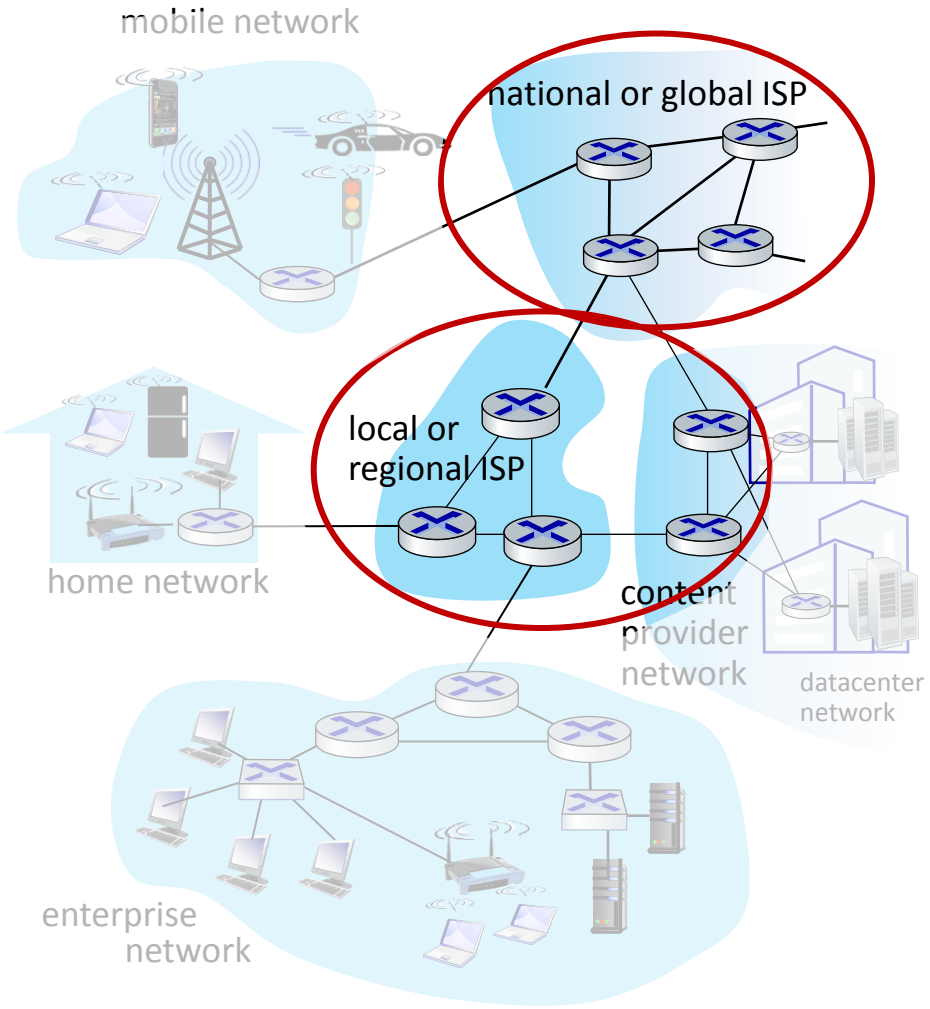
- hosts: clients and servers
- servers often in data centers

Access networks, physical media:

- wired, wireless communication links

Network core:

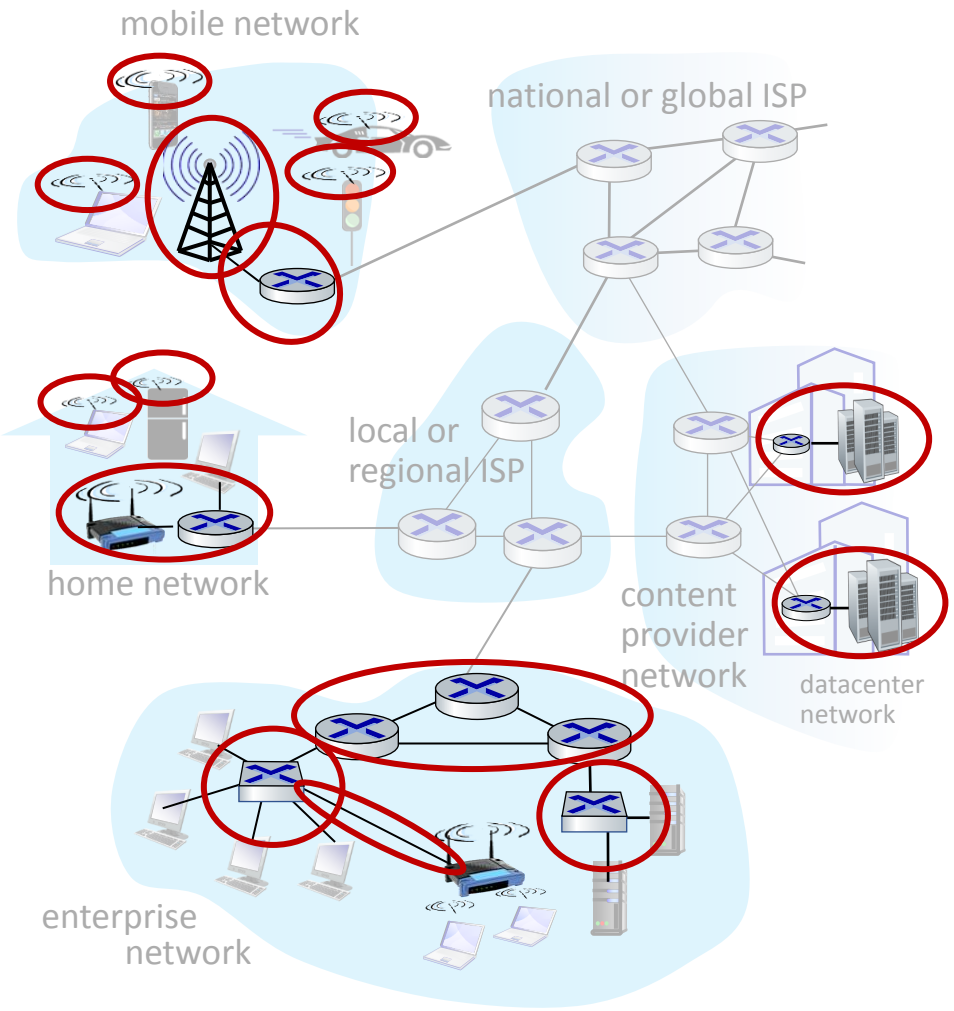
- interconnected routers
- network of networks



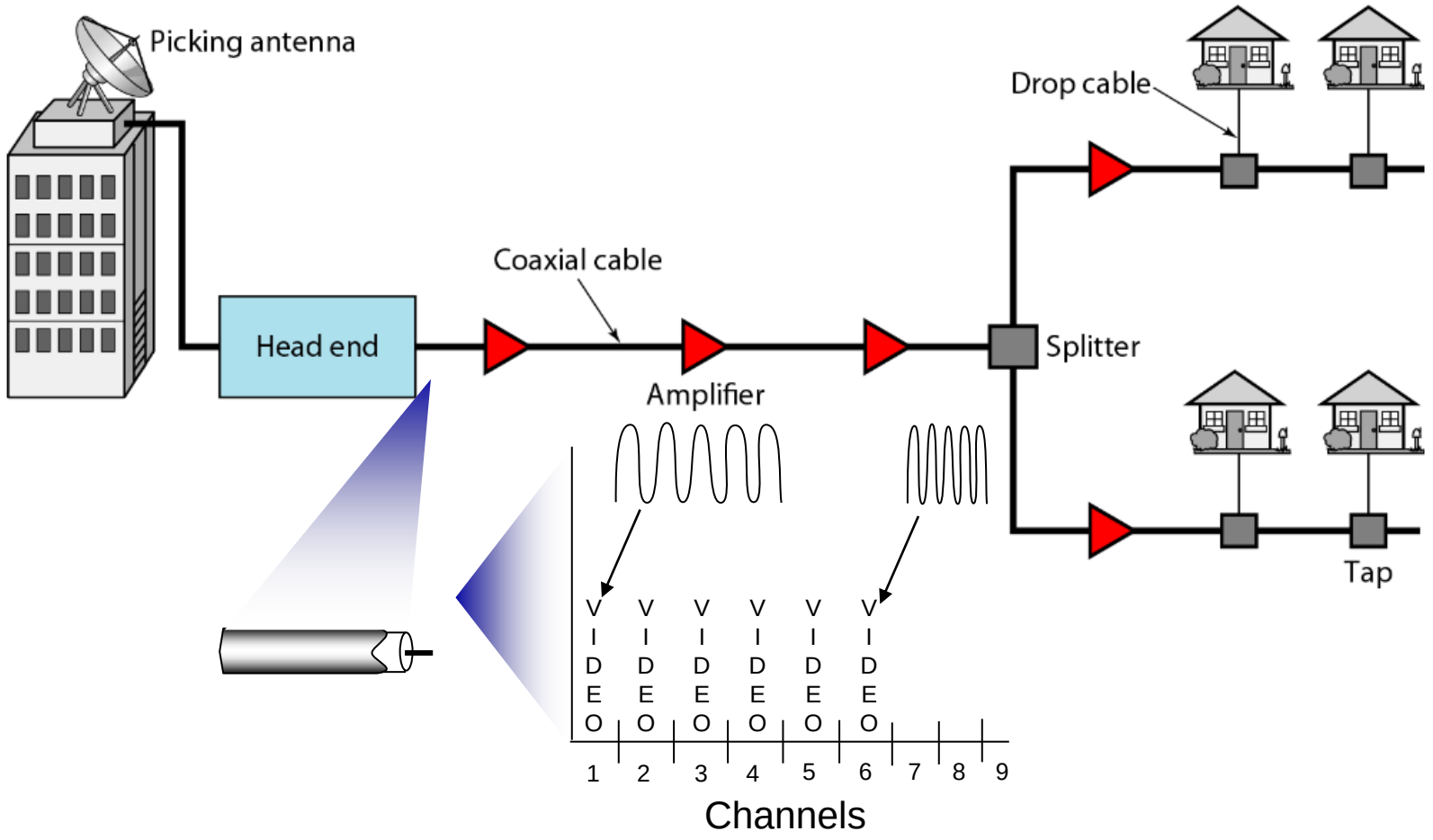
Access networks and physical media

Q: How to connect end systems to edge router?

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



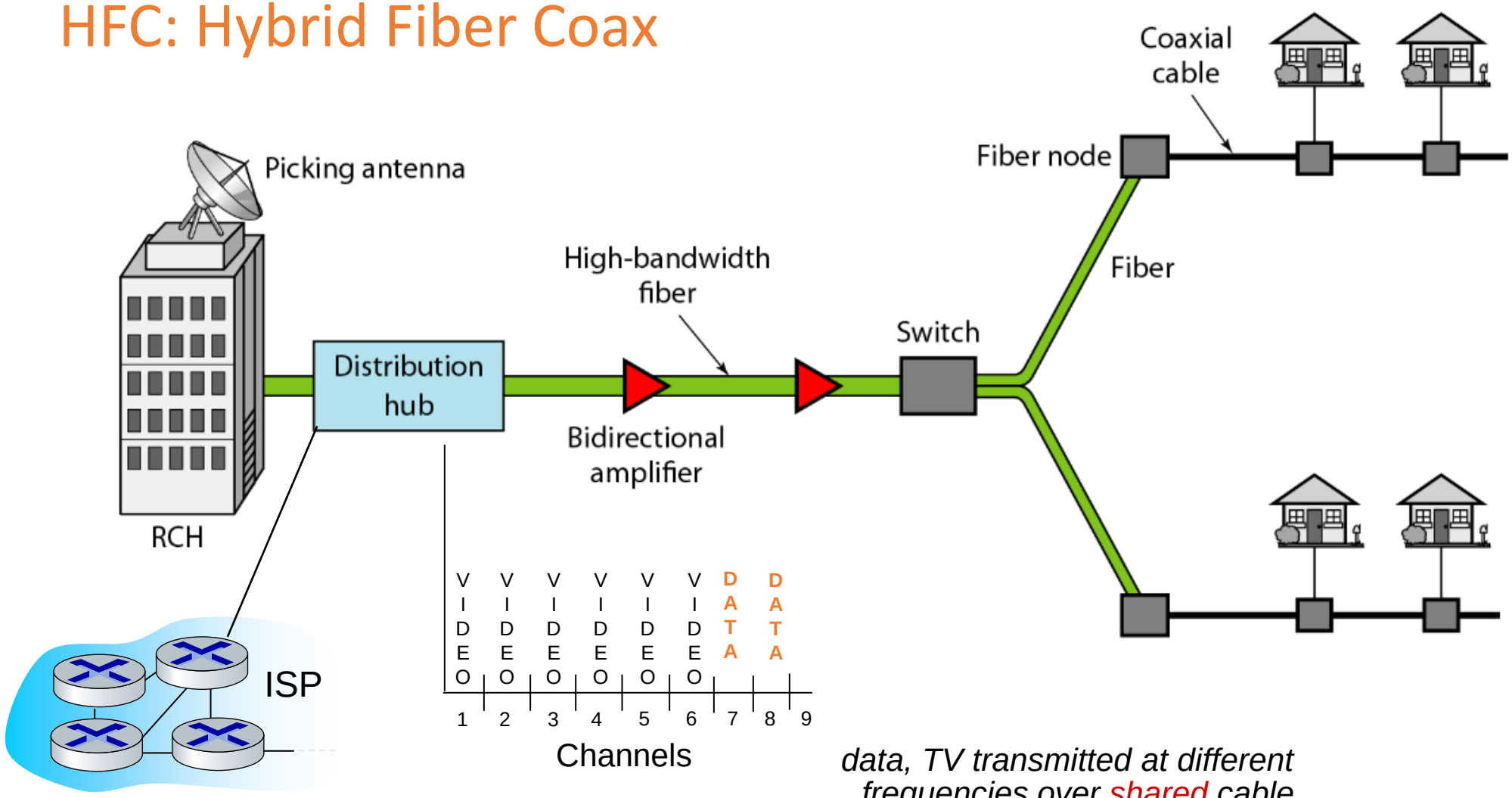
Access networks: cable-based access



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

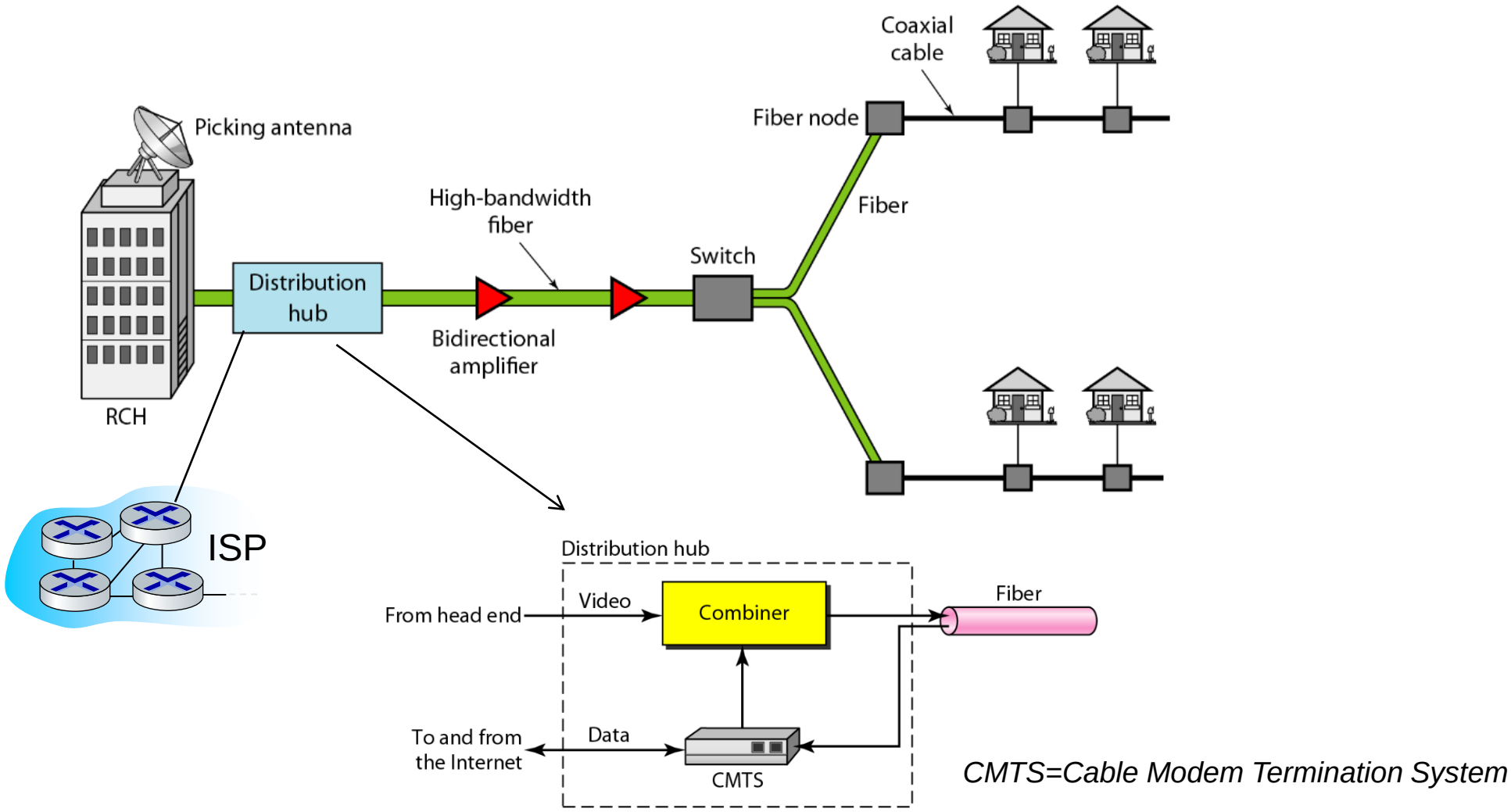
Access networks: cable-based access

HFC: Hybrid Fiber Coax



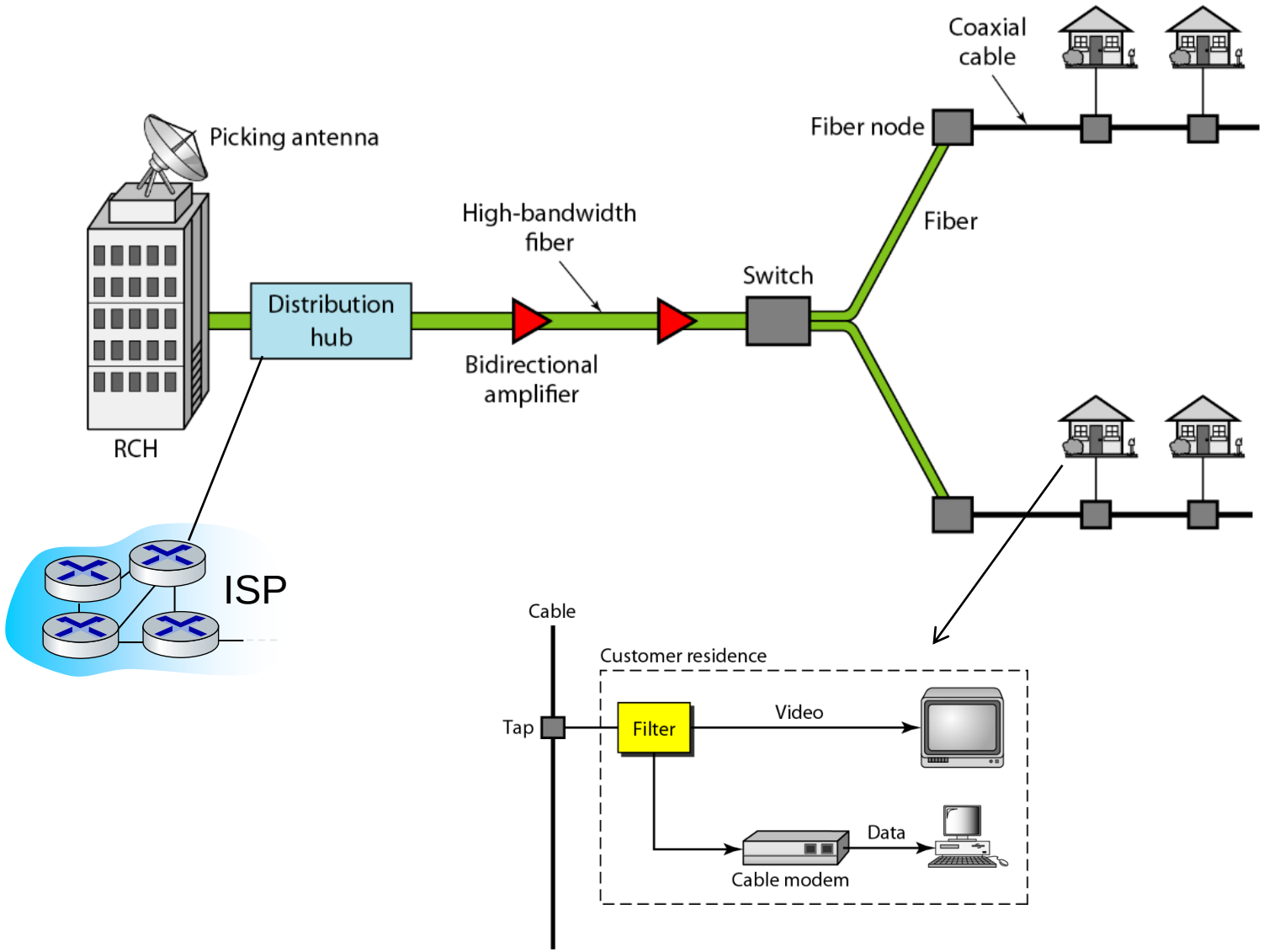
data, TV transmitted at different frequencies over shared cable distribution network

Access networks: cable-based access

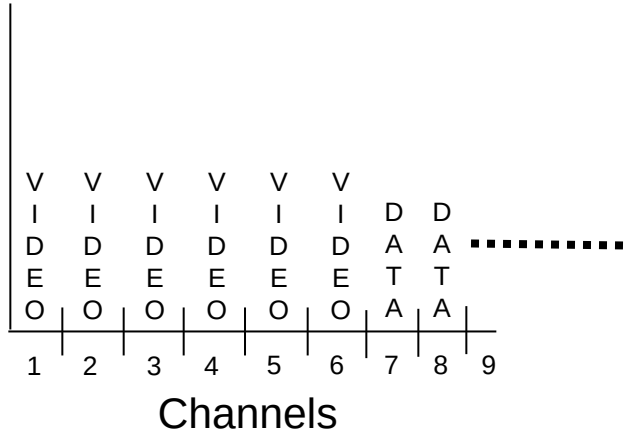


CMTS=Cable Modem Termination System

Access networks: cable-based access



Access networks: cable-based access



Each channel is ***shared*** among a number of subscribers

Downlink: multicasting

Uplink: multiple-access

Downlink faster than uplink

DL: 40 Mbps-1.2 Gbps, UL: 30-100 Mbps (DOCSIS 2.0, 3.0)

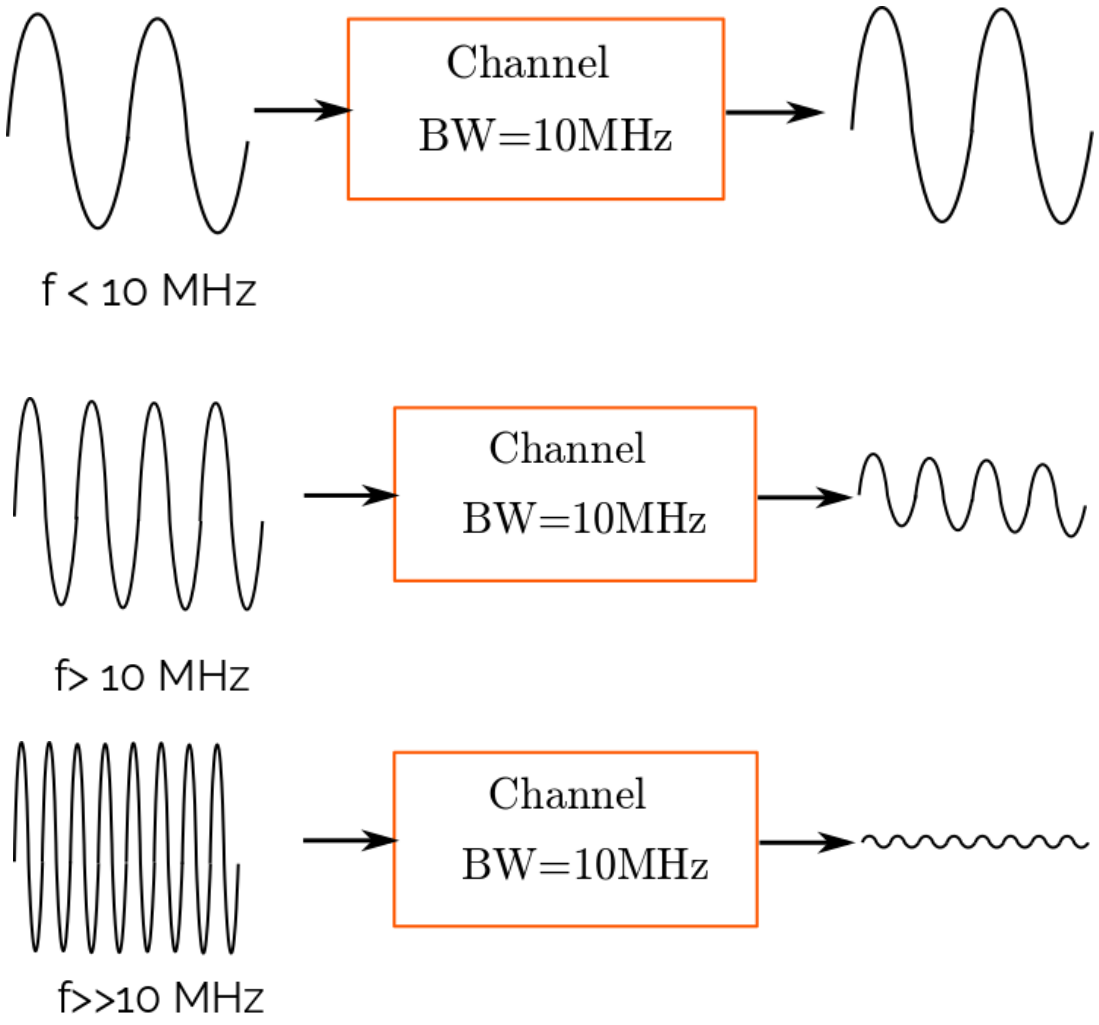
Because of different channel bandwidths/modulations

Aside: "Bandwidth" , "modulation", and "bit rate"

Bandwidth:

Unit: Hertz (Hz)

Limits the max frequency of passing sines



Aside: "Bandwidth" , "modulation", and "bit rate"

A pulse (carries a 0/1) can be written as the sum of infinite sines

- First harmony, second harmony, …

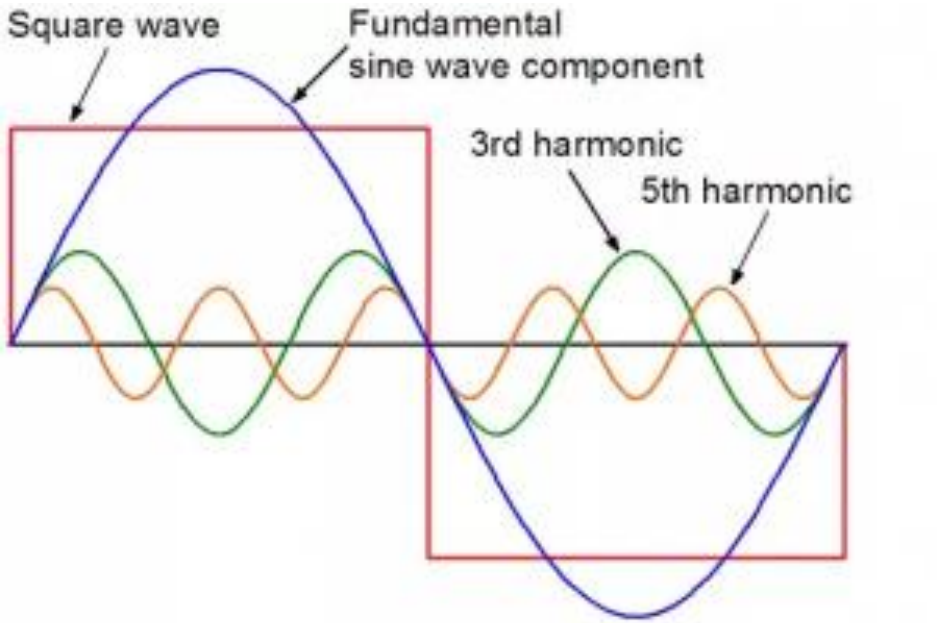
As the pulse's width decreases, the frequency of the first harmony increases.

The BW of the channel *limits* the pulse rate

aka symbol rate

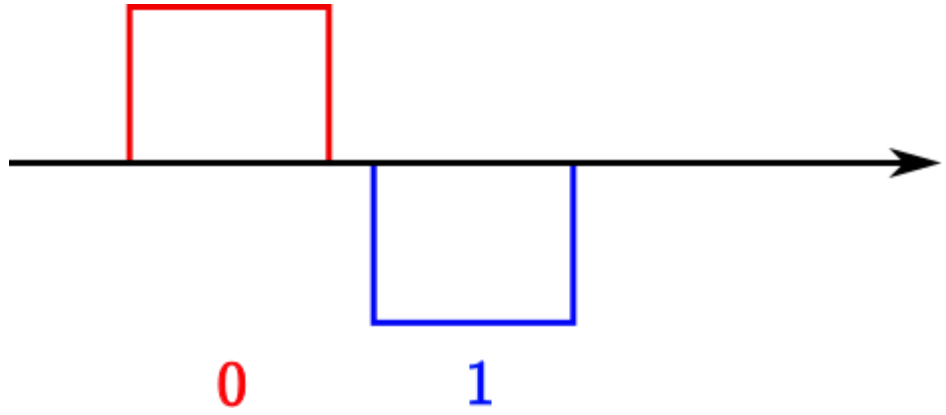
aka baud-rate

Baud rate ≈ BW



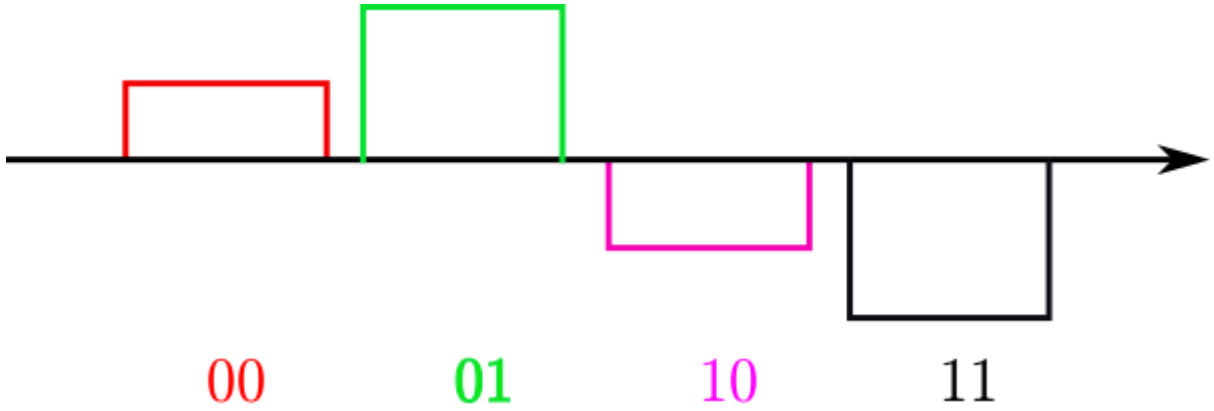
Aside: “Bandwidth” , “modulation”, and “bit rate”

A pulse can carry a single bit or more than one bit



Modulation: the number of levels

M= Number of bits per pulse = log (number of levels)



Aside: “Bandwidth” , “modulation”, and “bit rate”

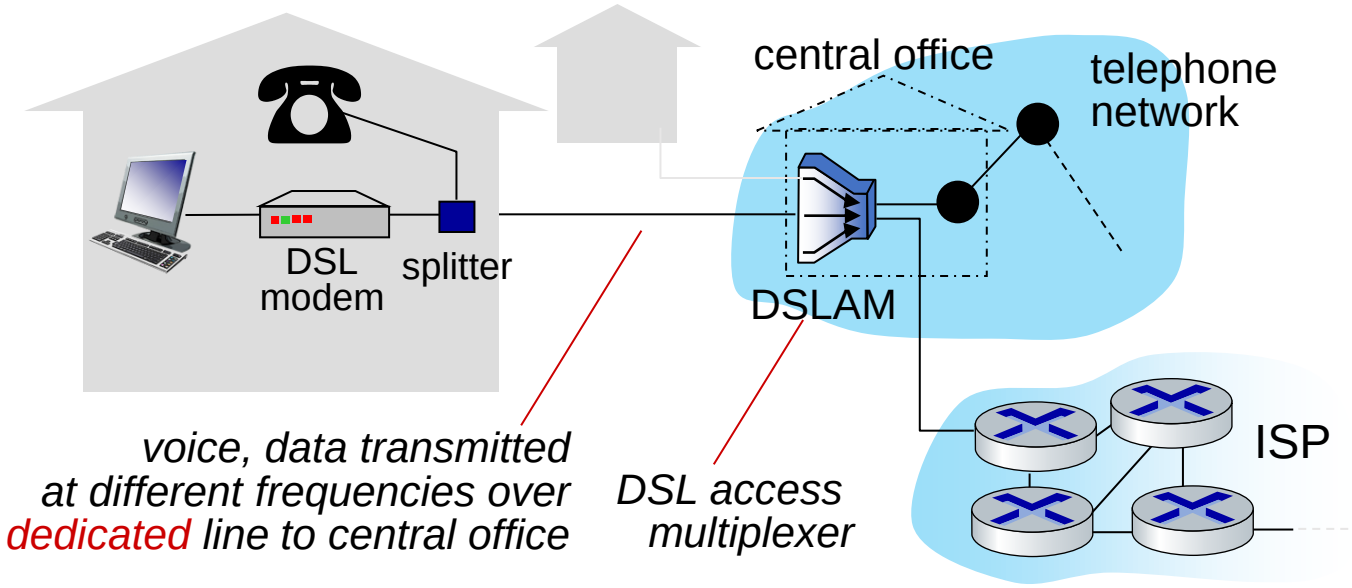
Bit rate = M $\times$ Baud-rate

Aside: “Bandwidth” , “modulation”, and “bit rate”

But in Network:

Bandwidth = Bit-rate

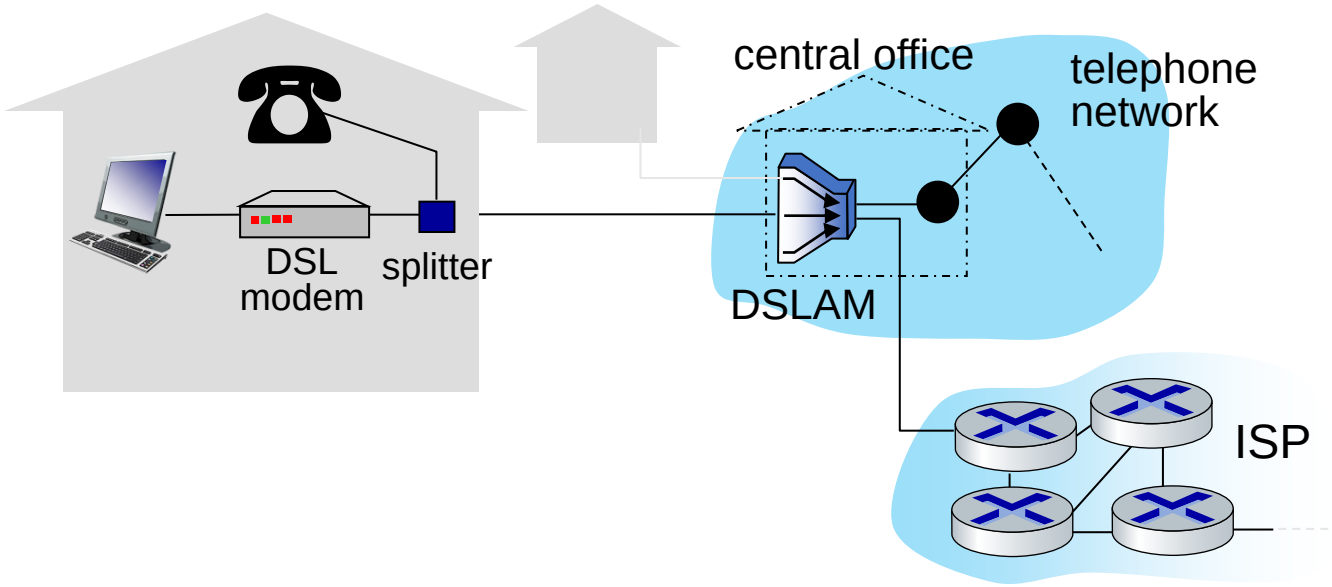
Access networks: digital subscriber line (DSL)



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

Asymmetric -> **A**DSL

Access networks: digital subscriber line (DSL)



>> ADSL is dedicated but not independent!

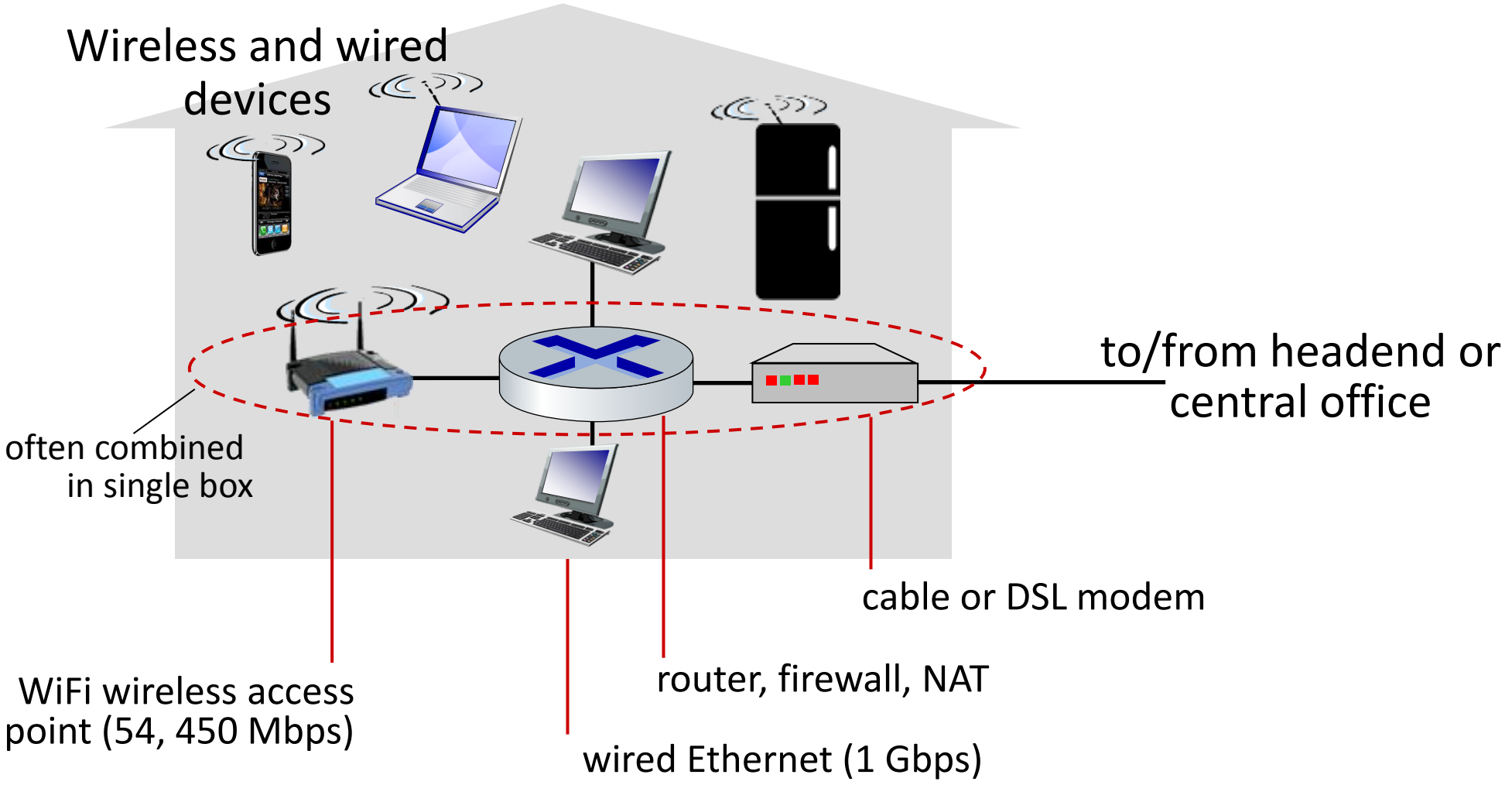
DQ: Why?

>> ADSL channels vary -> ADSL modems work adaptively

time of the day

Distance to CO

Access networks: home networks



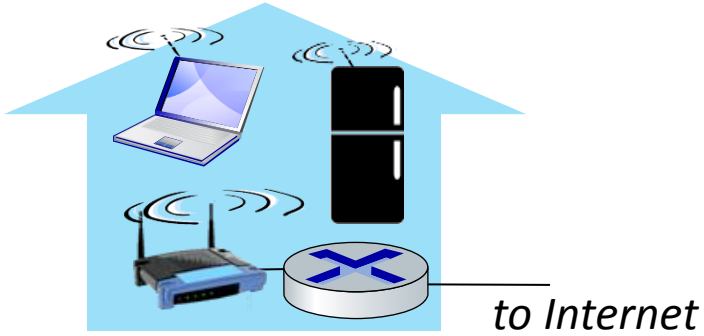
Wireless access networks

Shared *wireless* access network connects end system to router

- via base station aka “access point”

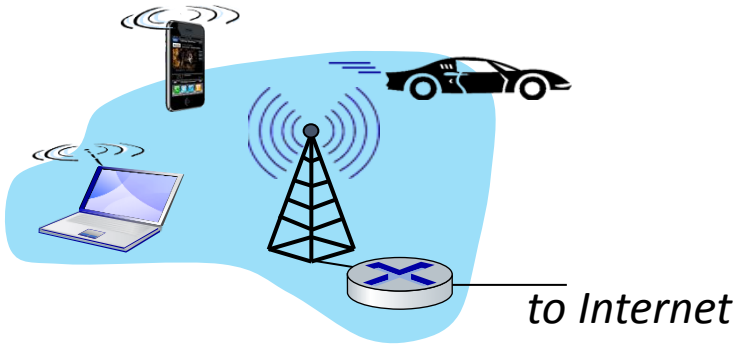
Wireless local area networks (WLANs)

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

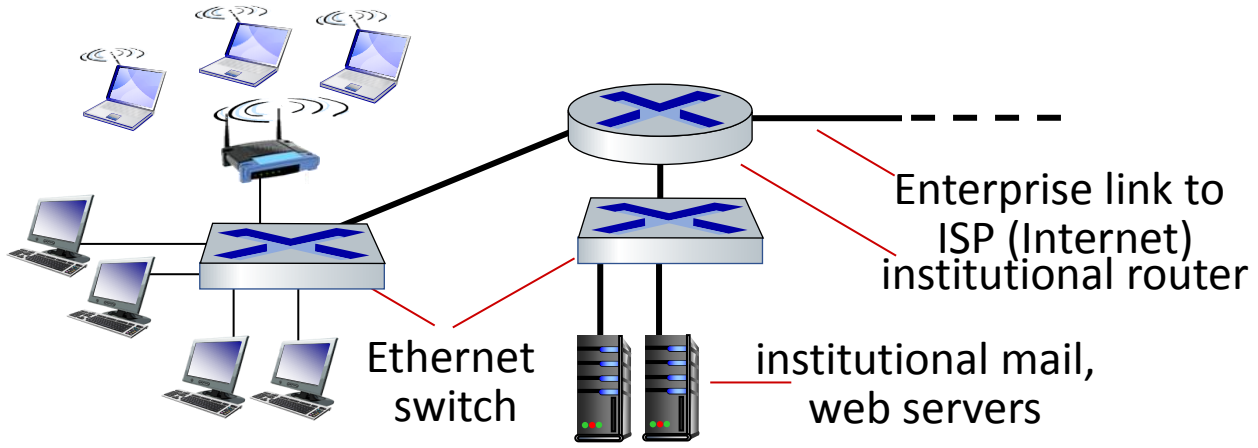


Wide-area cellular access networks

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



Access networks: enterprise networks



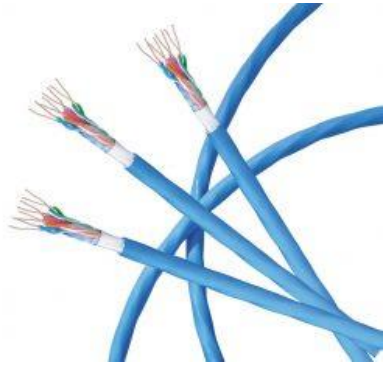
- companies, universities, etc.
- mix of wired, wireless link technologies, connecting a mix of switches and routers (we'll cover differences shortly)
  - Ethernet: wired access at 100Mbps, 1Gbps, 10Gbps
  - WiFi: wireless access points at 11, 54, 450 Mbps

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



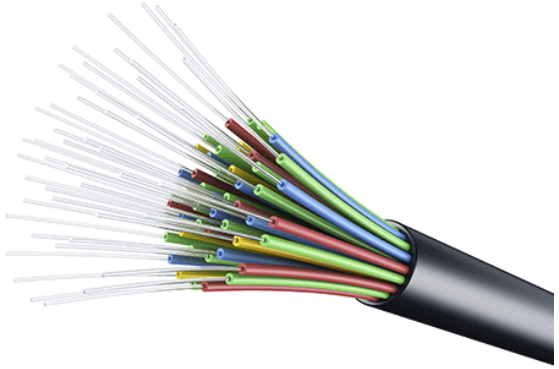
Links: physical media

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:

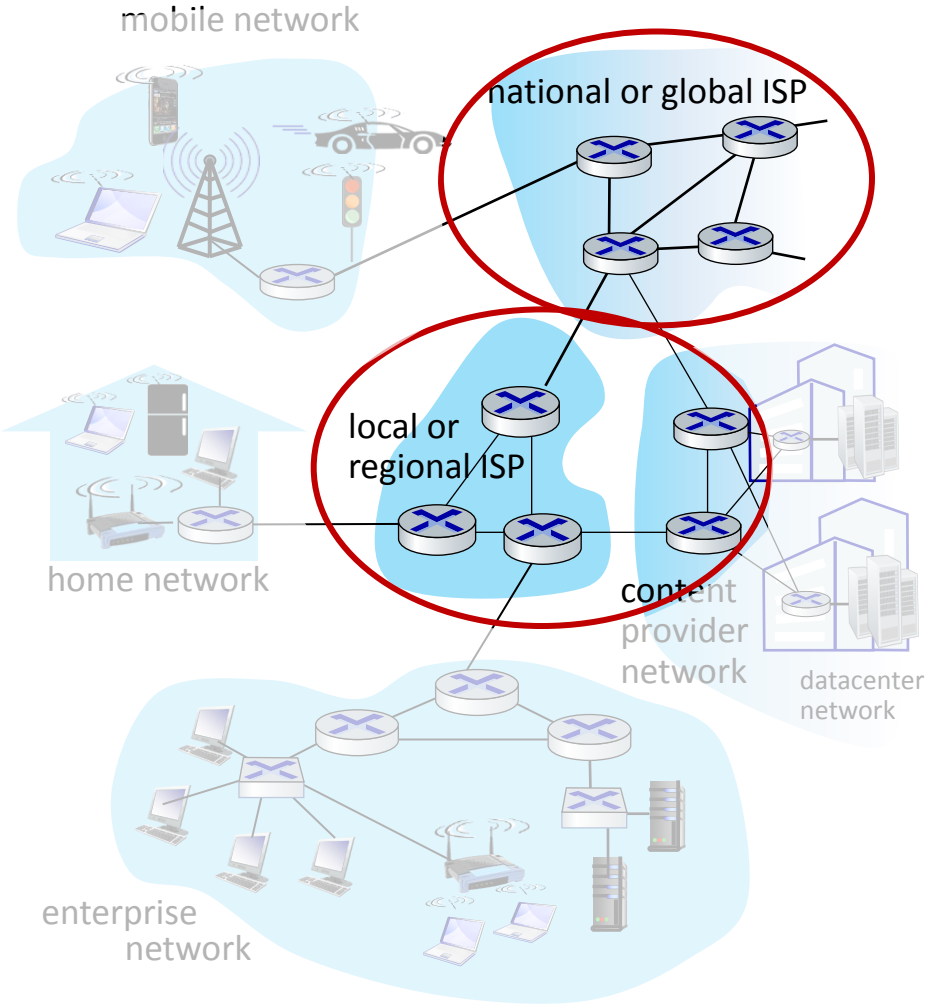
- Wireless LAN (WiFi)
  - 10-100's Mbps; 10's of meters
- wide-area (e.g., 4G cellular)
  - 10's Mbps over ~10 Km
- Bluetooth: cable replacement
  - short distances, limited rates
- terrestrial microwave
  - point-to-point; 45 Mbps channels
- satellite
  - up to 45 Mbps per channel
  - 270 msec end-end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

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

- A network at core: mesh of interconnected routers/switches
- Operating based on **packet switching** or **circuit switching**



How networks Work …

Packet switching

Circuit switching

How networks Work …

Packet switching

Circuit switching

Packet-switching

packet-switching: hosts break application-layer messages into *packets*

Packets = payload+header

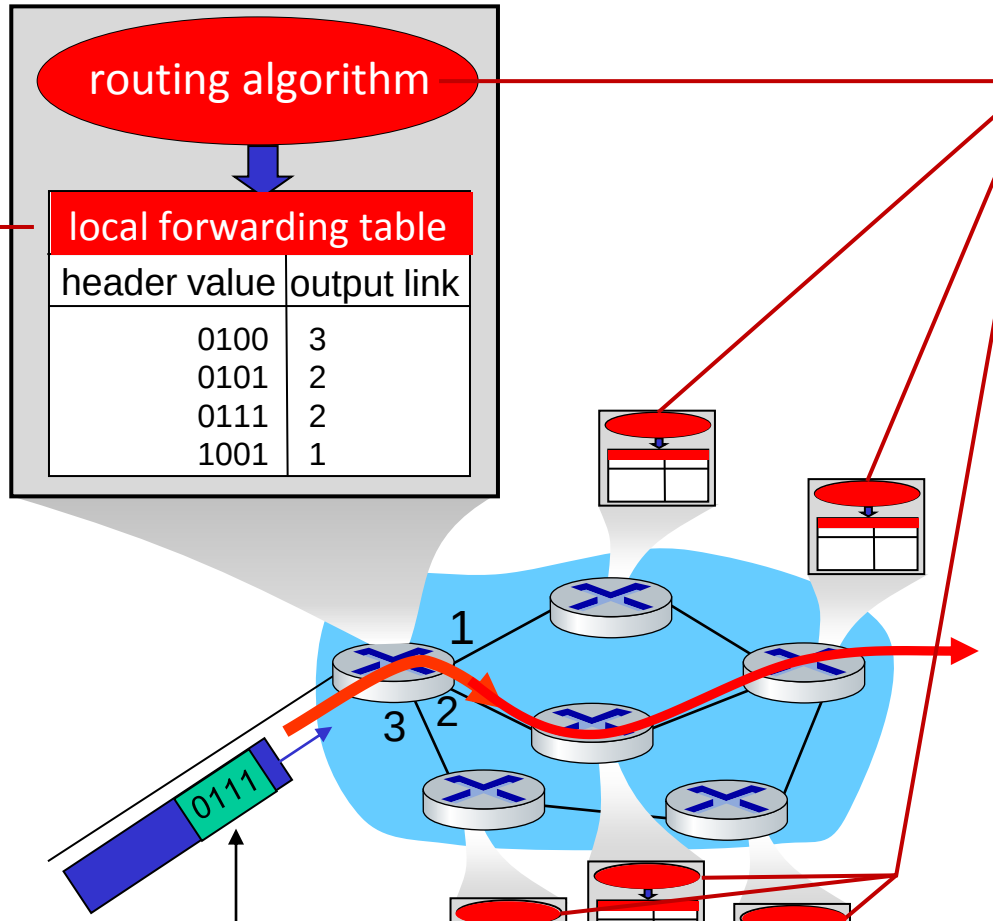
Packets have source/destination addresses inside their headers

network forwards packets from one router to the next, across links on path from source to destination

Packet-switching network: functionalities

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


45 h
3,053 miles
48 h
3,309 miles
routing
San Jose
Northampton
United States
Victoria
Seattle
Portland
Minneapolis
Quebec City
Montreal
Ottawa
Toronto
Detroit
Chicago
New York
Philadelphia
Kansas City
Indianapolis
Nashville
Charlotte
Atlanta
Dallas
Houston
Austin
San Antonio
El Paso
Tucson
Phoenix
San Diego
Los Angeles
Las Vegas
Sacramento
San Francisco
New Orleans
Jacksonville
Orlando
Tampa
WASHINGTON
OREGON
CALIFORNIA
NEVADA
IDAHO
MONTANA
WYOMING
UTAH
ARIZONA
COLORADO
NEW MEXICO
NORTH DAKOTA
SOUTH DAKOTA
NEBRASKA
KANSAS
OKLAHOMA
TEXAS
MINNESOTA
IOWA
MISSOURI
ARKANSAS
LOUISIANA
WISCONSIN
ILLINOIS
MICHIGAN
INDIANA
OHIO
KENTUCKY
TENNESSEE
MISSISSIPPI
ALABAMA
GEORGIA
FLORIDA
SOUTH CAROLINA
NORTH CAROLINA
VIRGINIA
WEST VIRGINIA
PENNSYLVANIA
MARYLAND
DE
NJ
CT
RI
VERMONT
NEW HAMPSHIRE
MAINE
BAJA CALIFORNIA
SONORA
CHIHUAHUA
COAHUILA


forwarding
forwarding
45 h
3,053 miles
48 h
3,309 miles
United States
Victoria
Seattle
Portland
WASHINGTON
NORTH DAKOTA
MINNESOTA
SOUTH DAKOTA
Minneapolis
WISCONSIN
MICHIGAN
Toronto
Quebec City
Montreal
Ottawa
MAINE
VERMONT
NEW HAMPSHIRE
Northampton
WYOMING
IOWA
Chicago
Detroit
ILLINOIS
INDIANA
OHIO
Indianapolis
Kansas City
COLORADO
KANSAS
MISSOURI
KENTUCKY
Nashville
TENNESSEE
San Francisco
San Jose
CALIFORNIA
Las Vegas
OKLAHOMA
ARKANSAS
Atlanta
Los Angeles
ARIZONA
NEW MEXICO
San Diego
Phoenix
Dallas
MISSISSIPPI
ALABAMA
Tucson
El Paso
TEXAS
BAJA CALIFORNIA
SONORA
Austin
Houston
LOUISIANA
New Orleans
CHIHUAHUA
San Antonio
COAHUILA
Tampa
FLORIDA
Cleveland
DOWNTOWN
CENTRAL
OHIO CITY

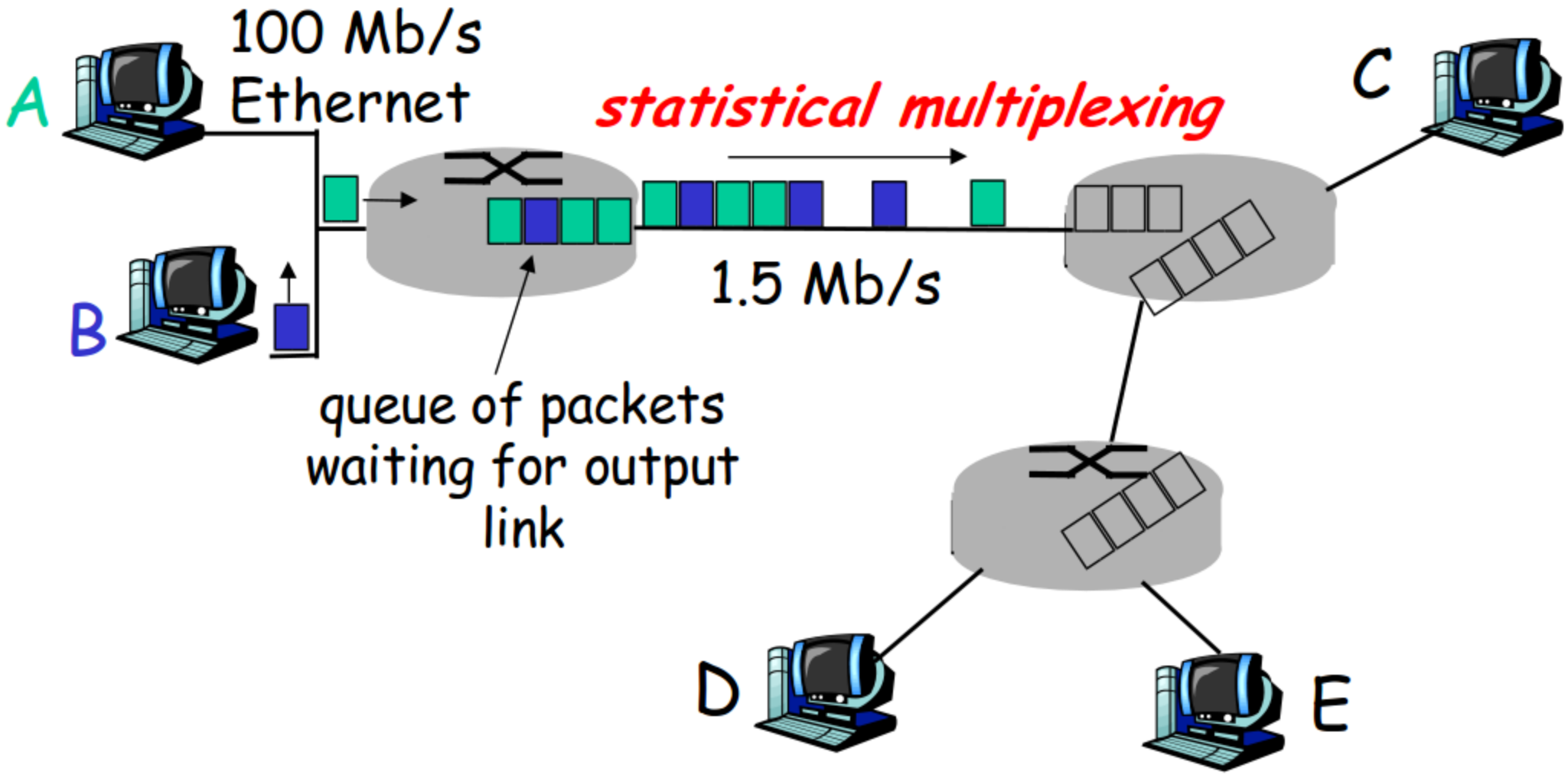
Packet-switching

packet-switching: hosts break application-layer messages into *packets*

>> No resource reservation (no cull setup)

+ Resource sharing (statistical multiplexing)


تلفن همگانی


A
100 Mb/s
Ethernet
statistical multiplexing
C
B
1.5 Mb/s
queue of packets
waiting for output
link
D
E

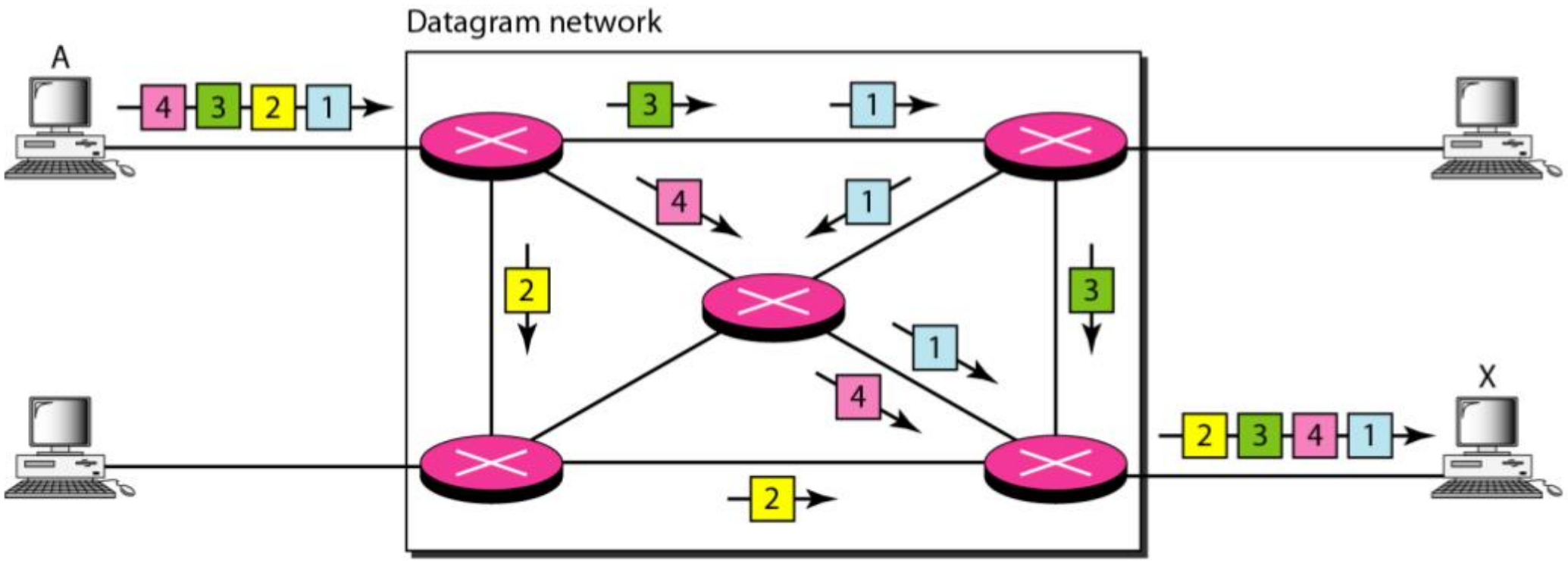
Packet-switching

packet-switching: hosts break application-layer messages into *packets*

>> No resource reservation (no cull setup)

+ Resource sharing (statistical multiplexing)

- Different packets are routed independently, and may take different paths


Datagram network
A
X
4
3
2
1
3
1
4
1
2
3
1
4
2
2
3
4
1

Packet-switching

packet-switching: hosts break application-layer messages into *packets*

>> No resource reservation (no cull setup)

+ Resource sharing (statistical multiplexing)

- Different packets are routed independently, and may

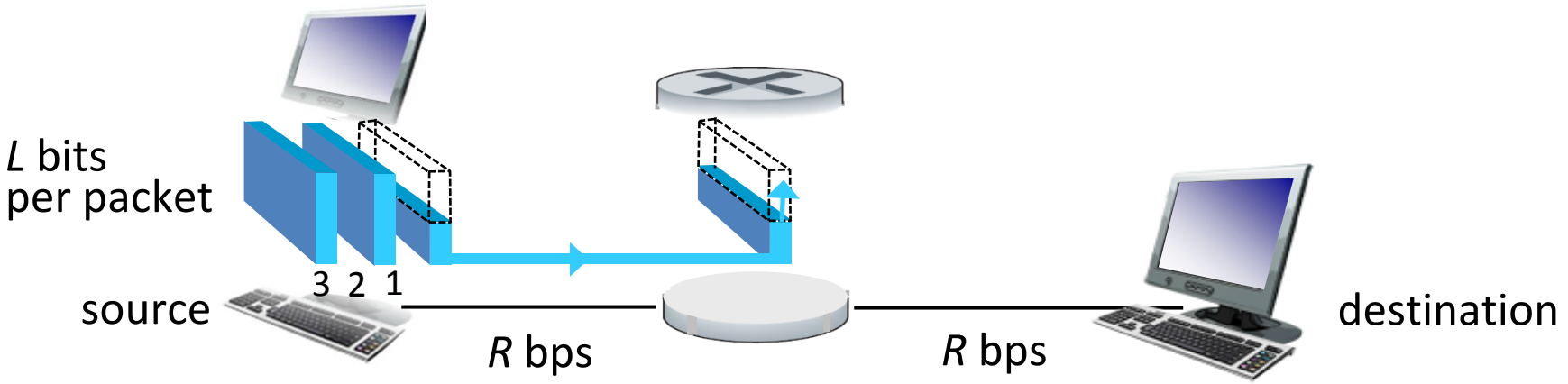
  take different paths

  have different delays (missing orders)

  be dropped (lost)

BEST EFFORT!

Store and forward

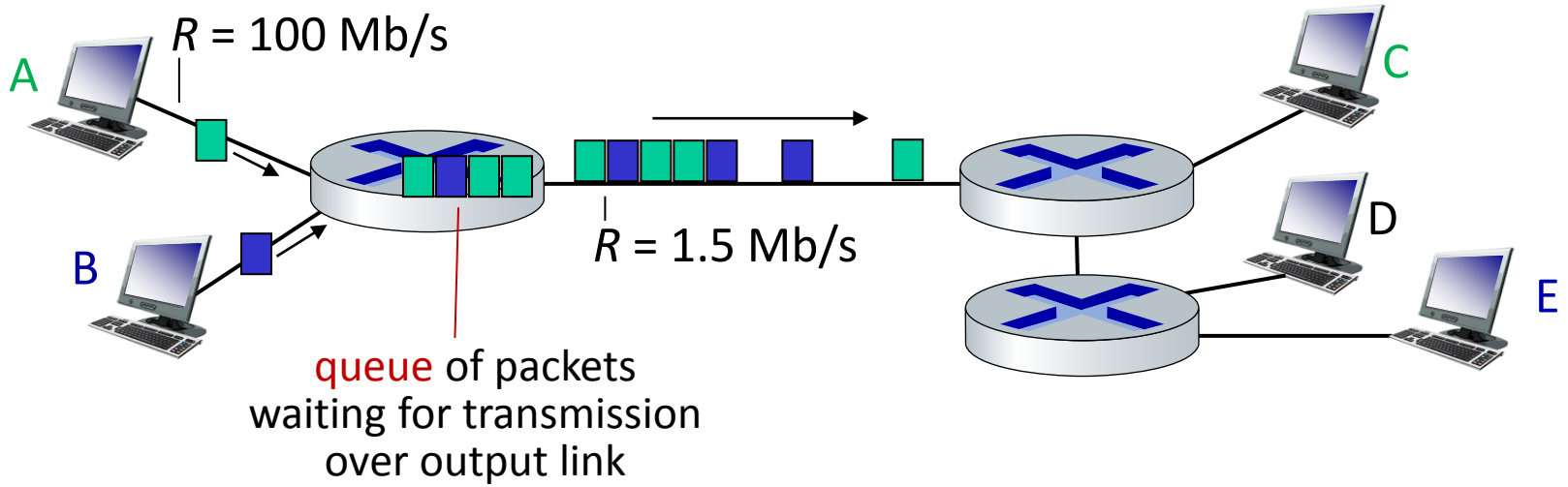


- packet transmission delay: takes L/R seconds to transmit (push out) L-bit packet into link at R bps
- *store and forward: entire* packet must arrive at router before it can be transmitted on next link

One-hop numerical example:

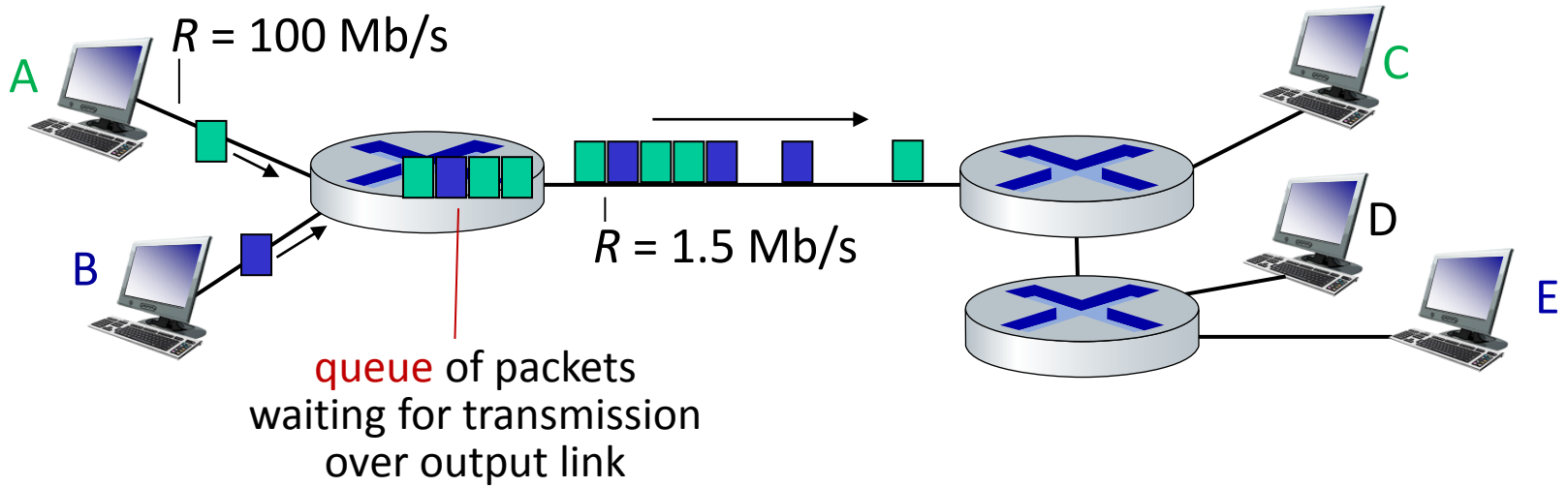
- L = 10 Kbits
- R = 100 Mbps
- one-hop transmission delay = 0.1 msec

Queueing



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

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

How networks Work …

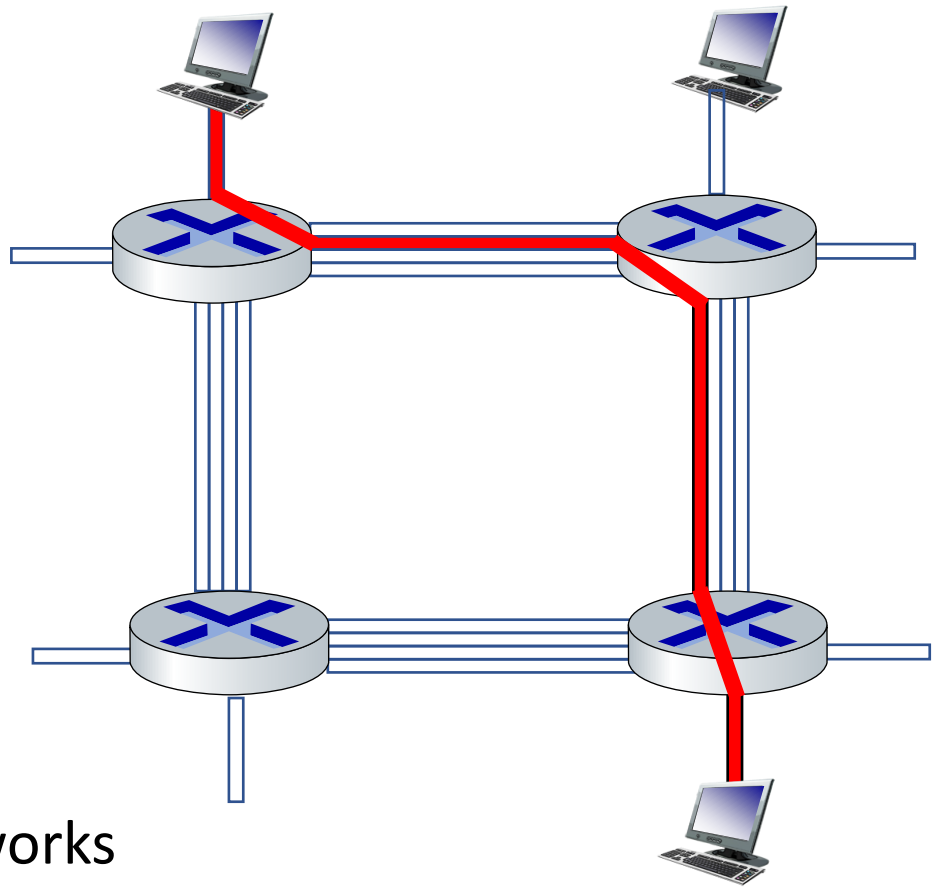
Packet switching

Circuit 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

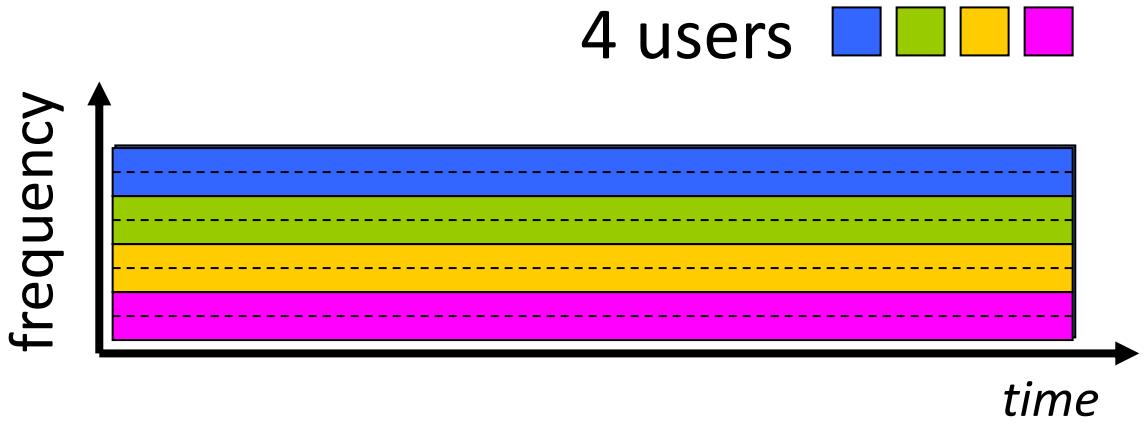


* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive

FDM and TDM

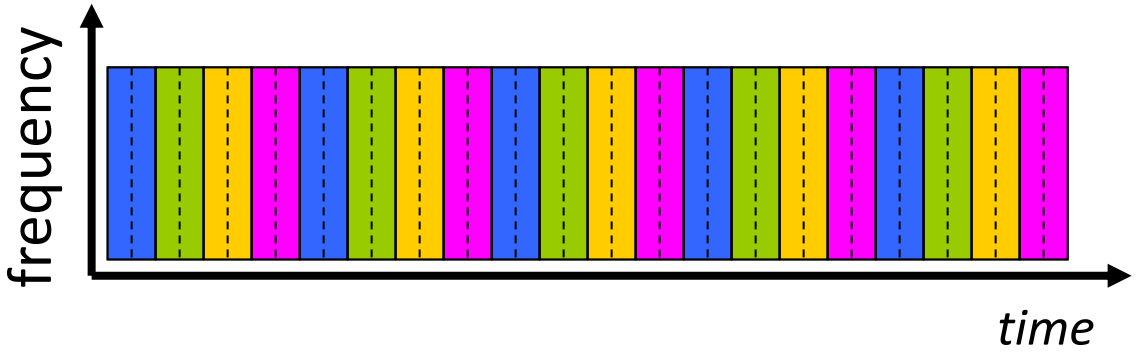
Frequency Division Multiplexing (FDM)

- 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)

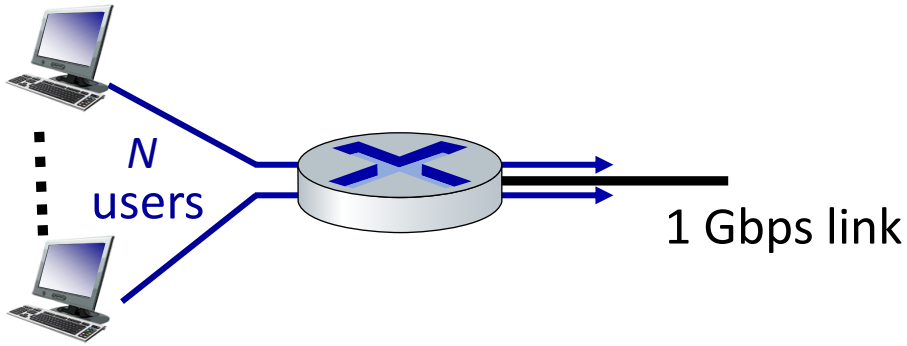
- time divided into slots
- each call allocated periodic slot(s), can transmit at maximum rate of (wider) frequency band (only) during its time slot(s)



Example: Packet switching vs. 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 vs. circuit switching

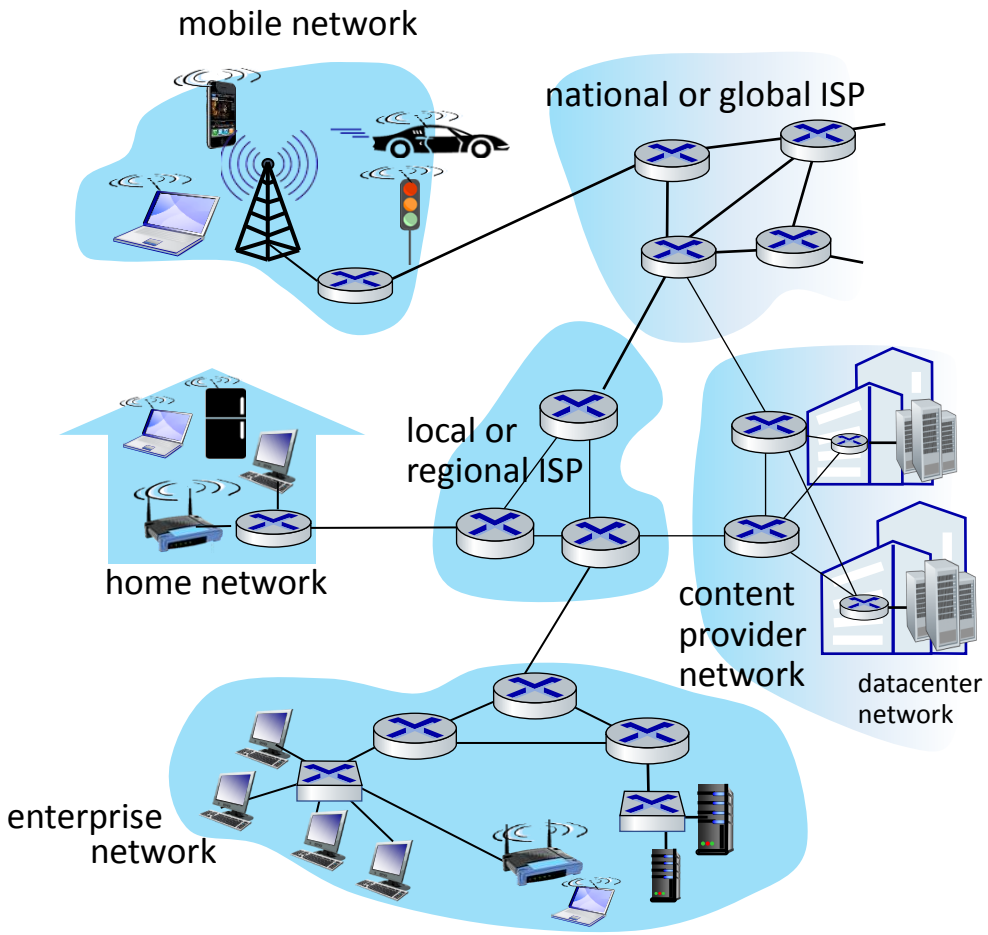
Is packet switching a “slam dunk winner”?

- 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.

Internet Structure

Internet structure: a "network of networks"

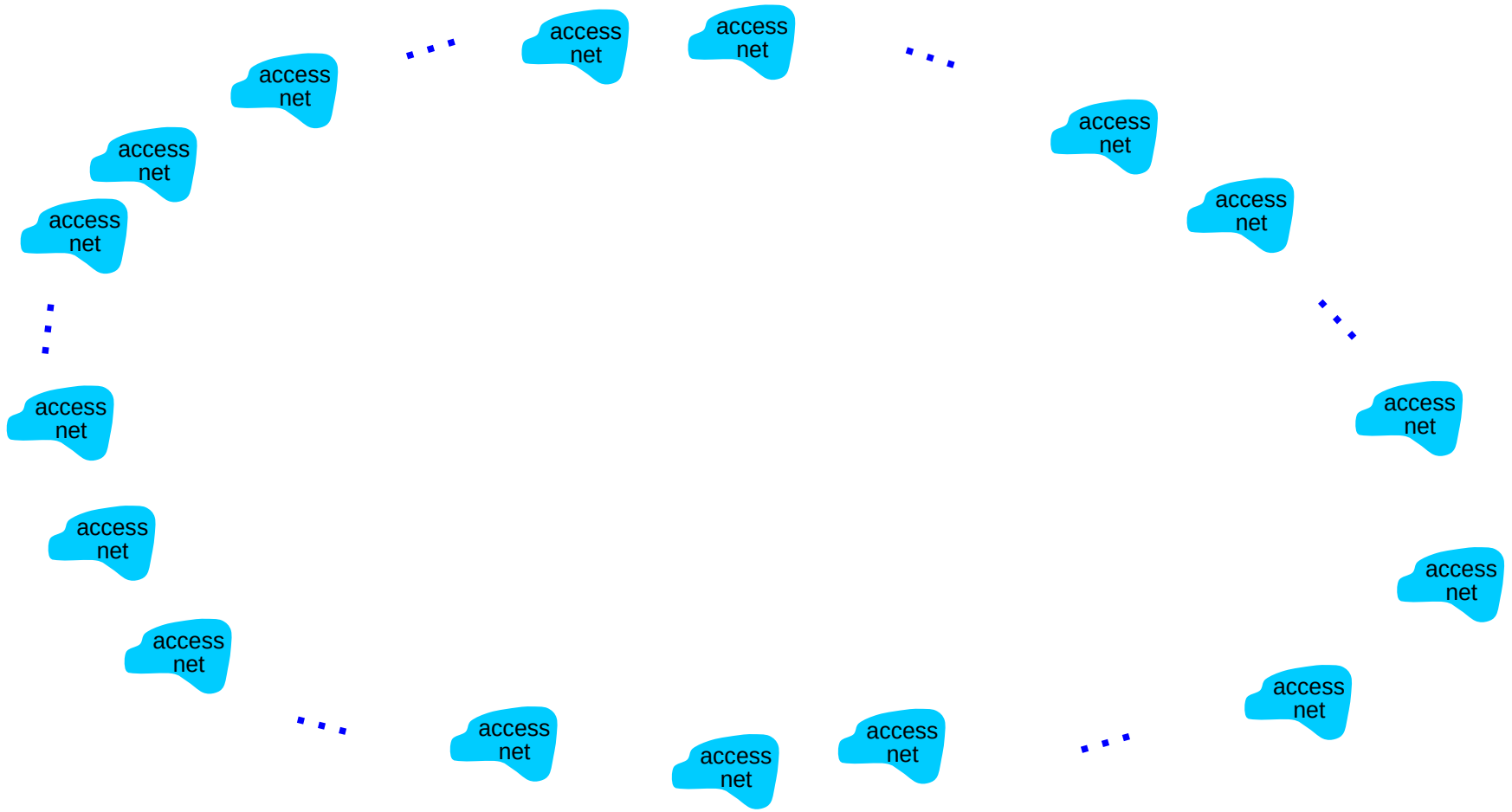
- hosts connect to Internet via access Internet Service Providers (ISPs)
- access ISPs in turn must be interconnected
  - so that *any* two hosts *(anywhere!)* can send packets to each other
- resulting network of networks is very complex
  - evolution driven by economics, national policies



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

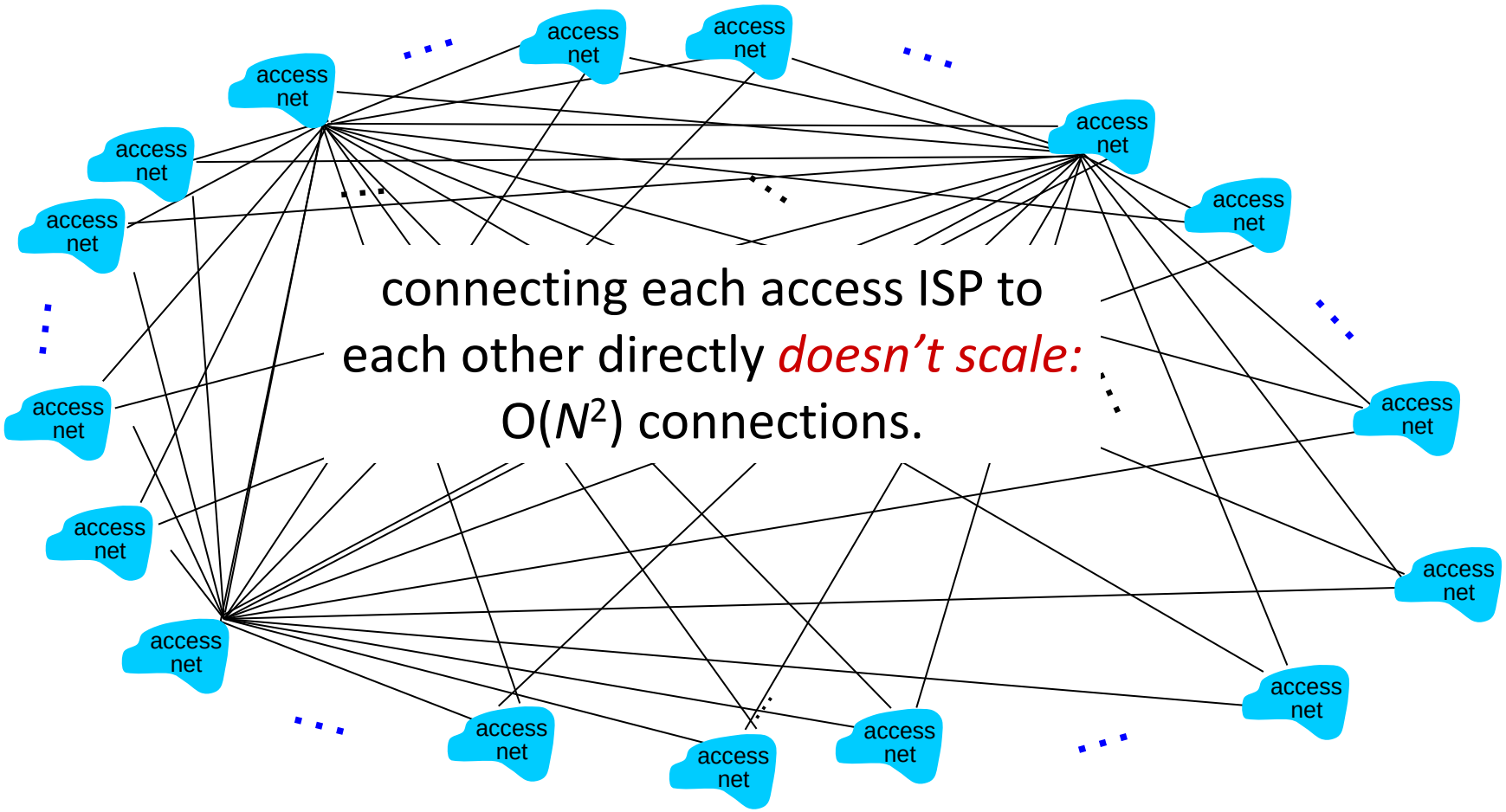
Internet structure: a "network of networks"

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



Internet structure: a "network of networks"

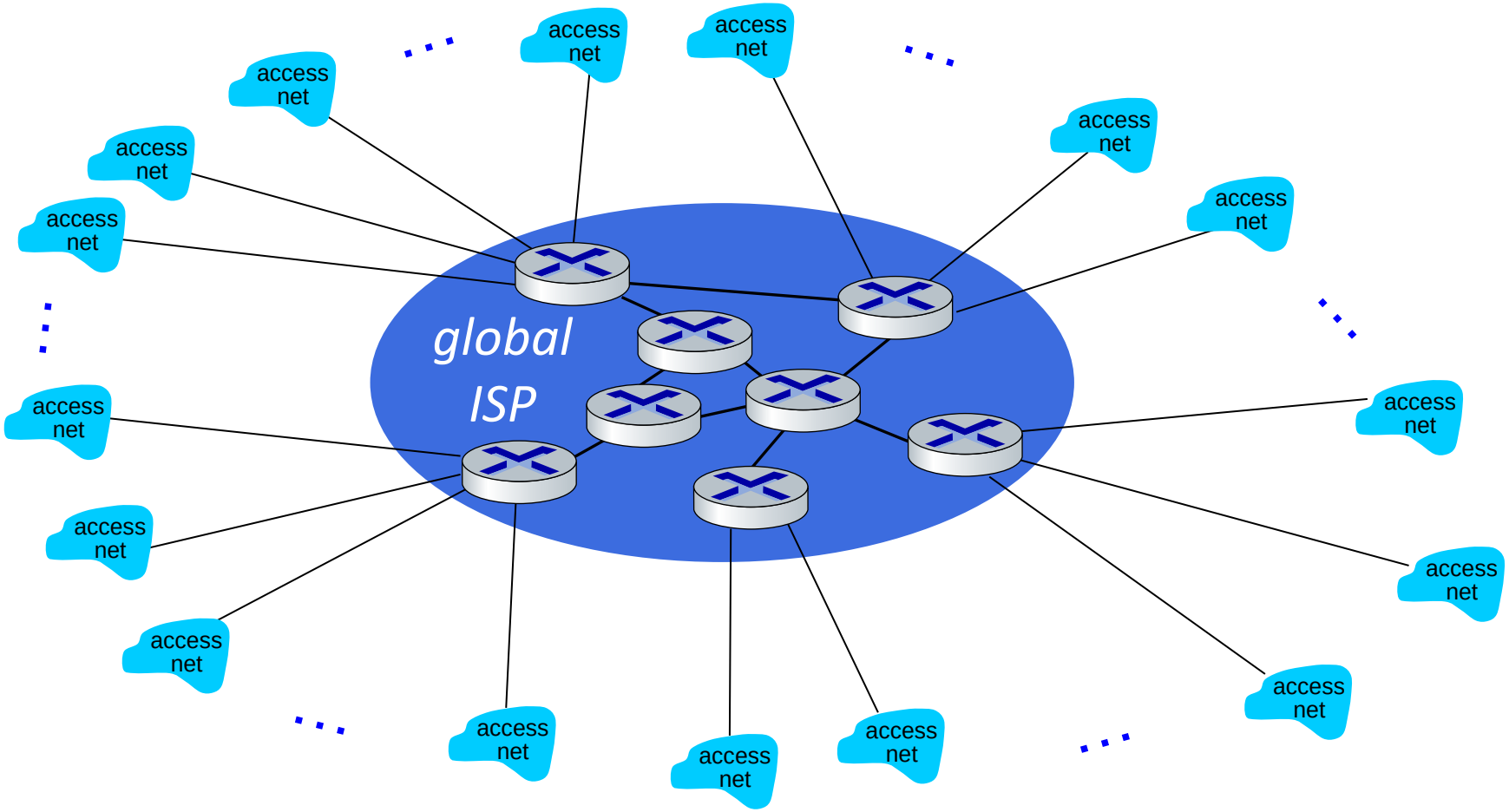
Question: given *millions* of access ISPs, how to connect them together?



Internet structure: a "network of networks"

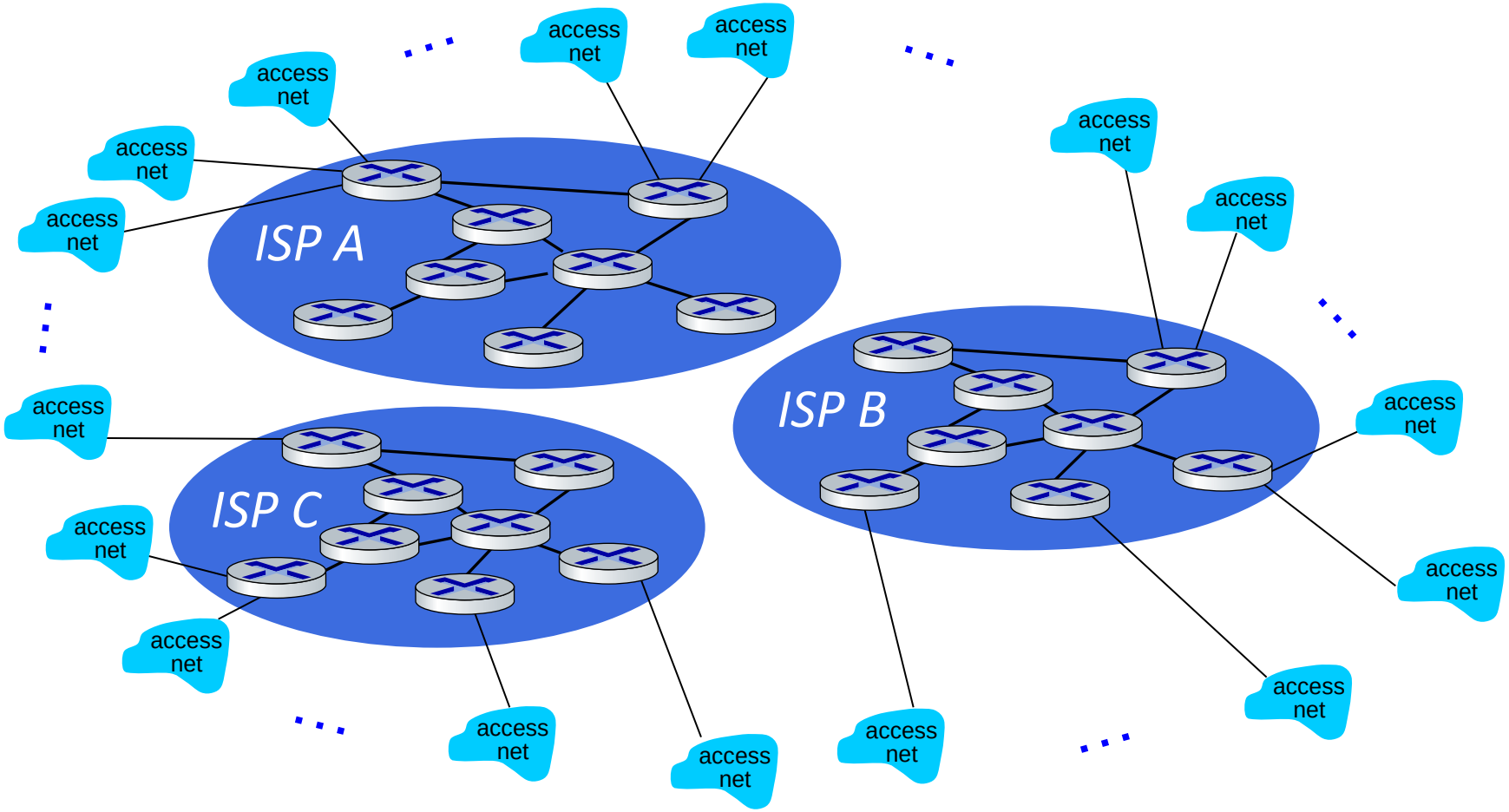
Option: connect each access ISP to one global transit ISP?

Customer and provider ISPs have economic agreement.



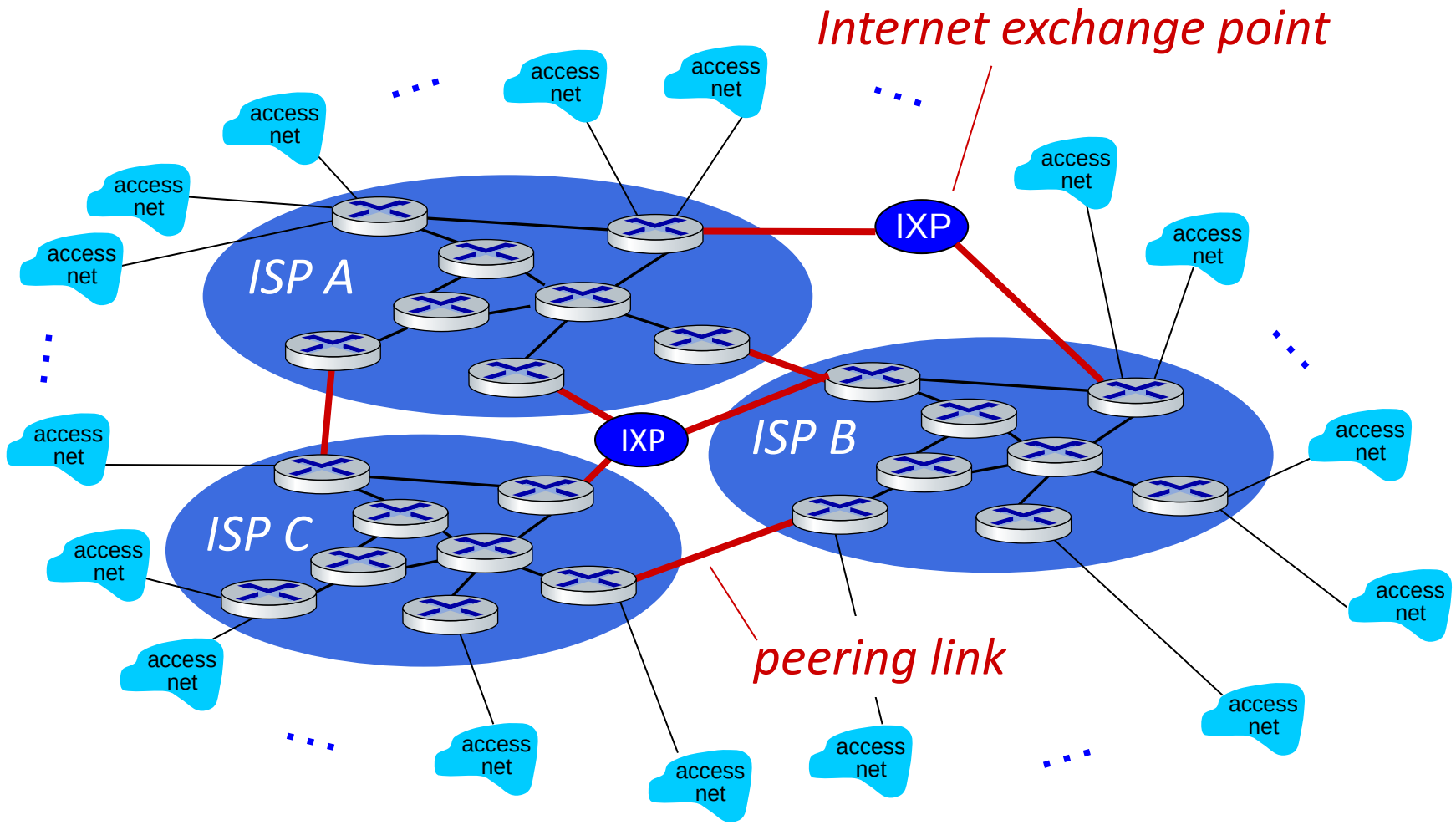
Internet structure: a "network of networks"

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



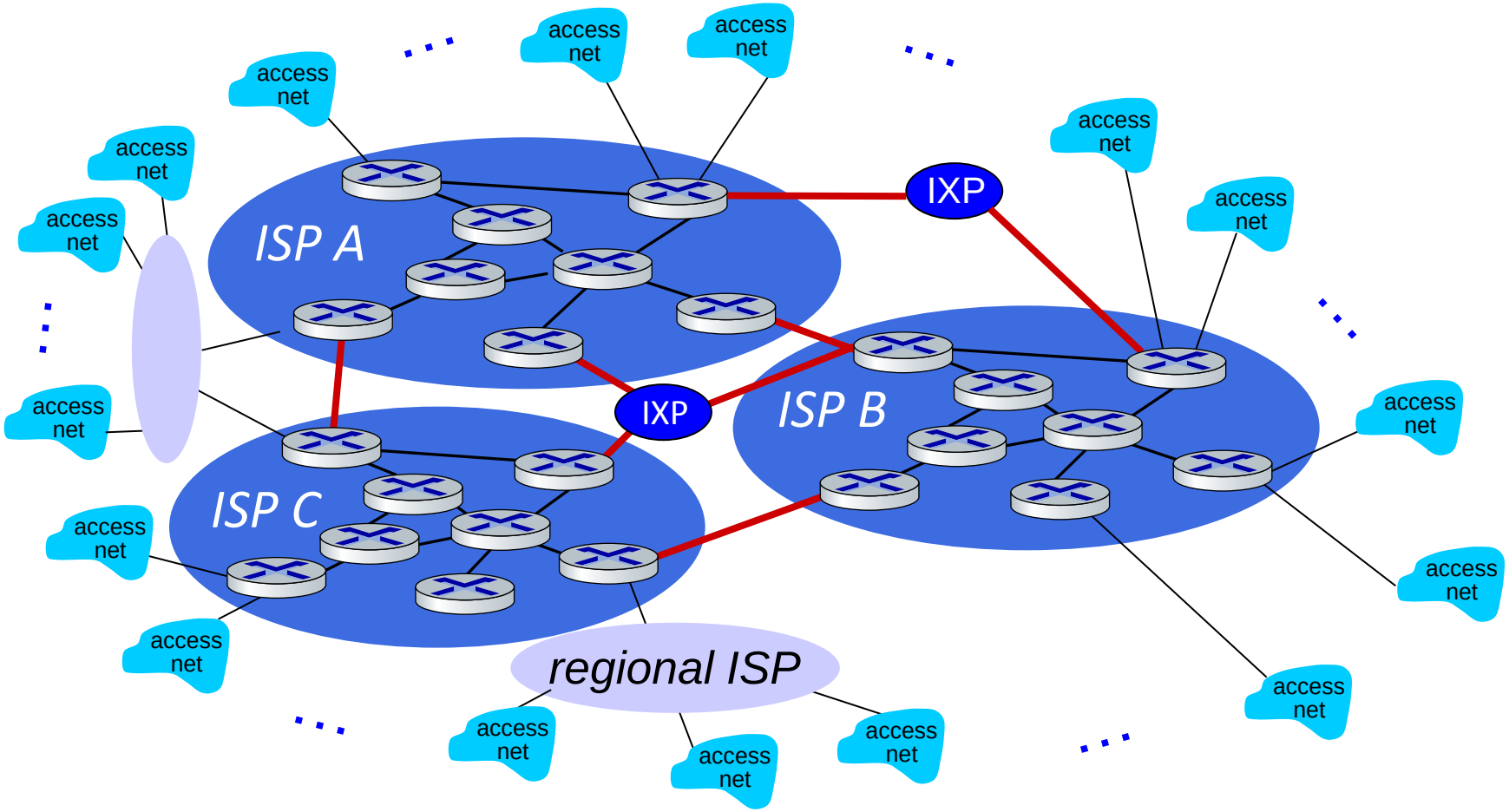
Internet structure: a "network of networks"

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



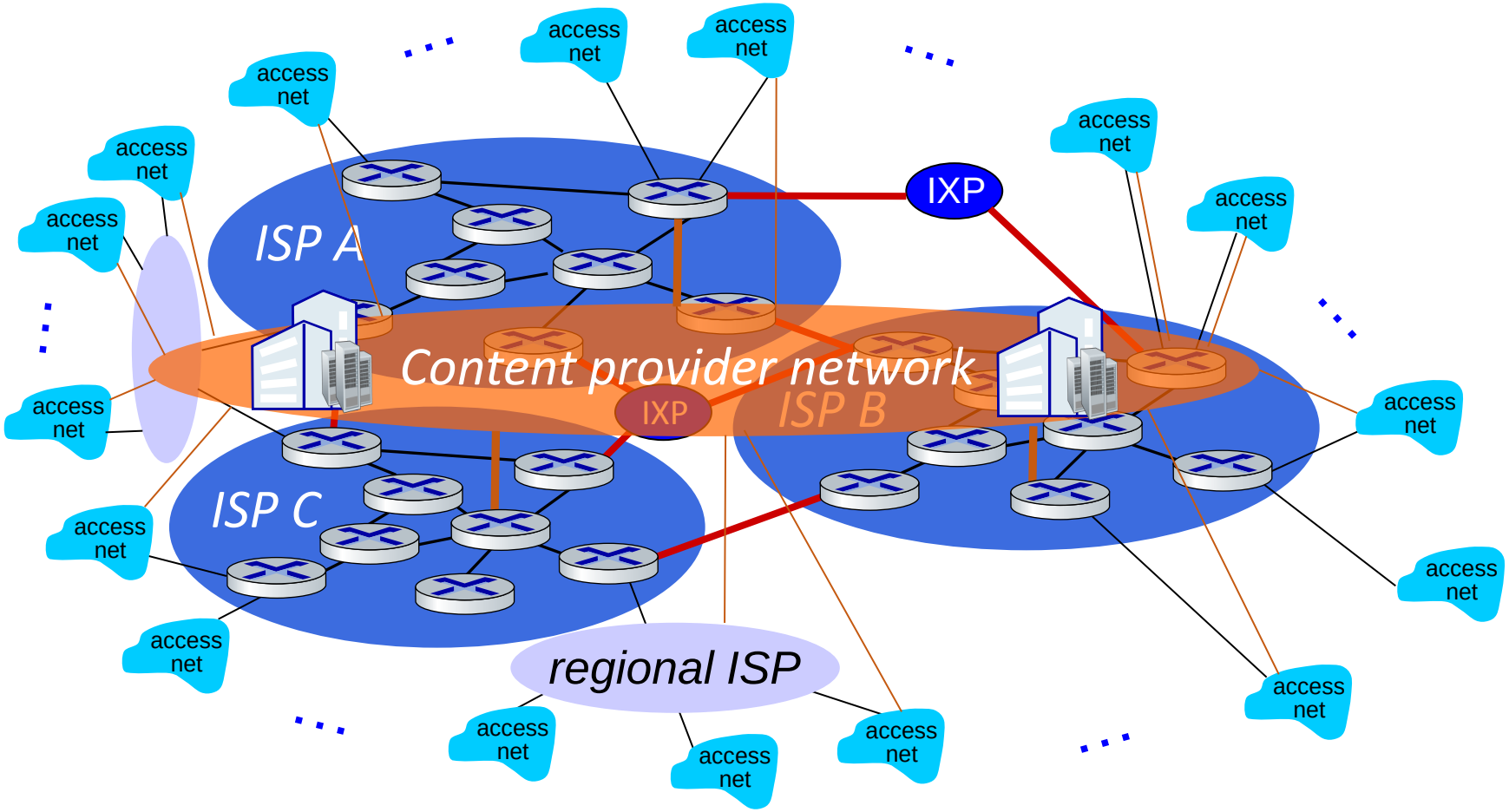
Internet structure: a "network of networks"

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

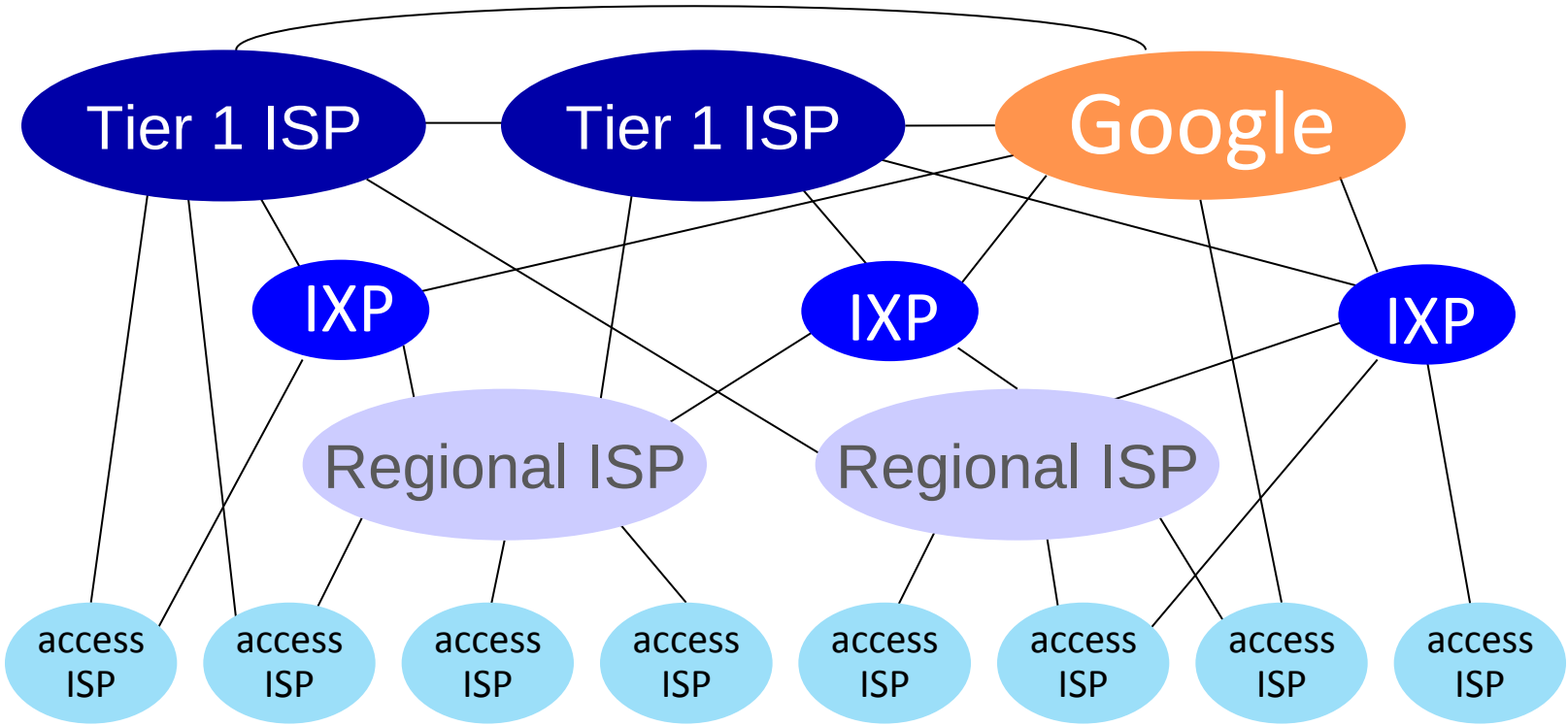


Internet structure: a "network of networks"

… 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

Performance

Delay

Throughput

Loss

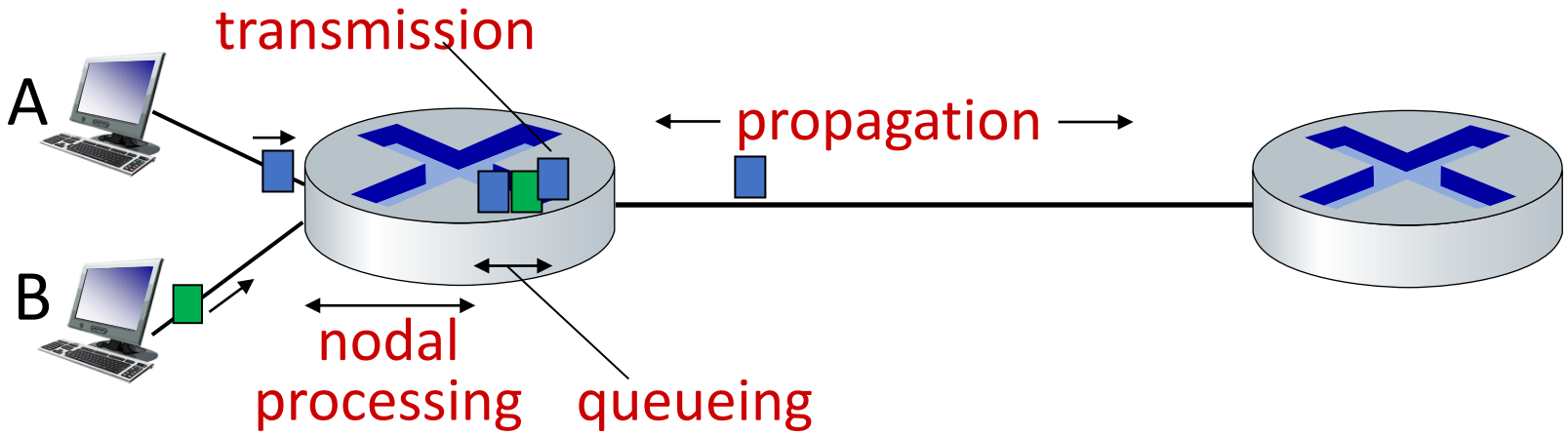
Performance

Delay

Throughput

Loss

Packet delay: four sources



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

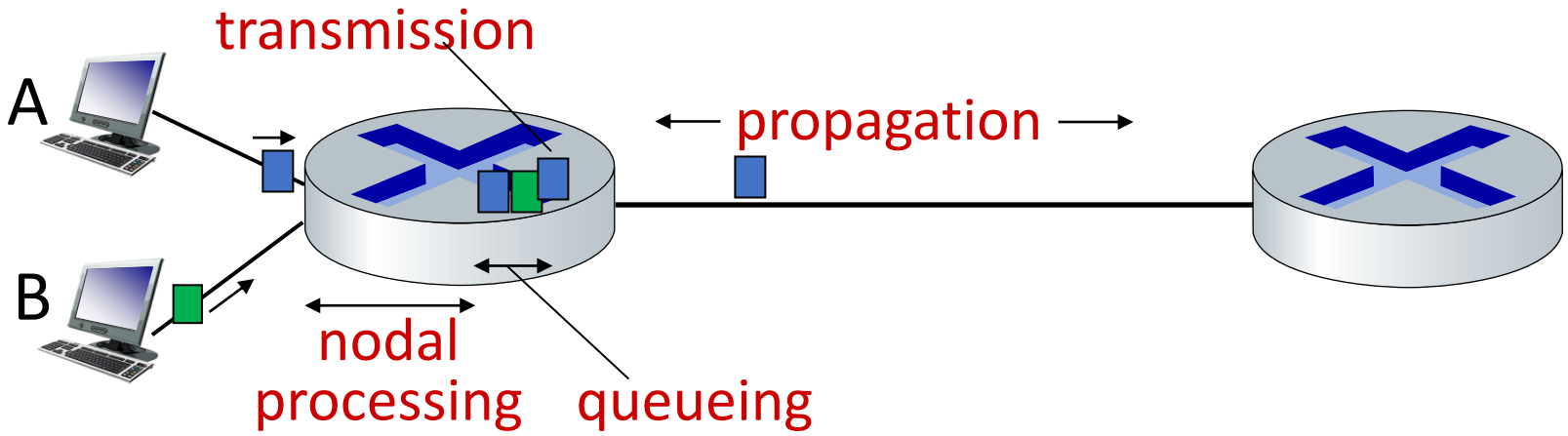
d_{proc}: nodal processing

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

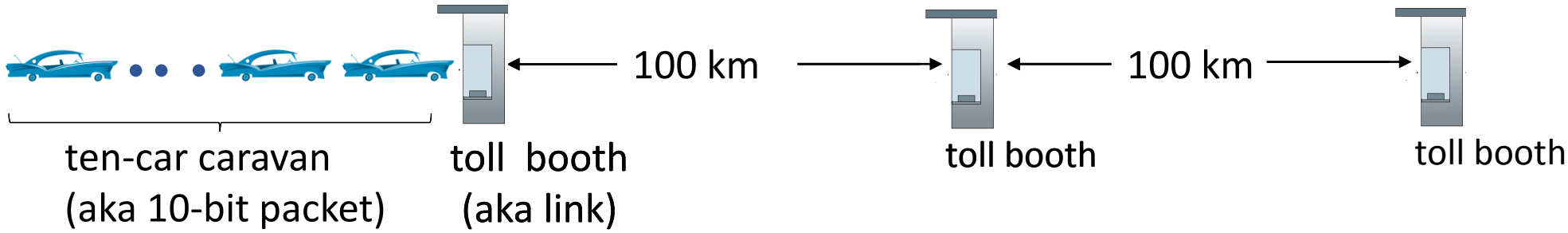
- L: packet length (bits)
- R: link *transmission rate (bps)*
- $d_{trans} = L/R$

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



- 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?

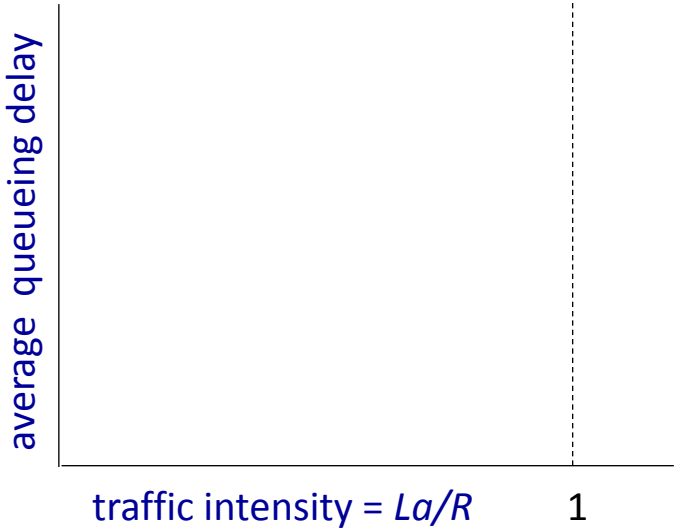
- time to "push" entire caravan through toll booth onto highway = 12*10 = 120 sec
- time for last car to propagate from 1st to 2nd toll both: 100km/(100km/hr) = 1 hr
- *A:* 62 minutes

Packet queueing delay (revisited)

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

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

"traffic intensity"



- *La/R* ~ 0: avg. queueing delay small

- *La/R* > 1: more "work" arriving is more than can be serviced - average delay infinite!

La/R ~ 0

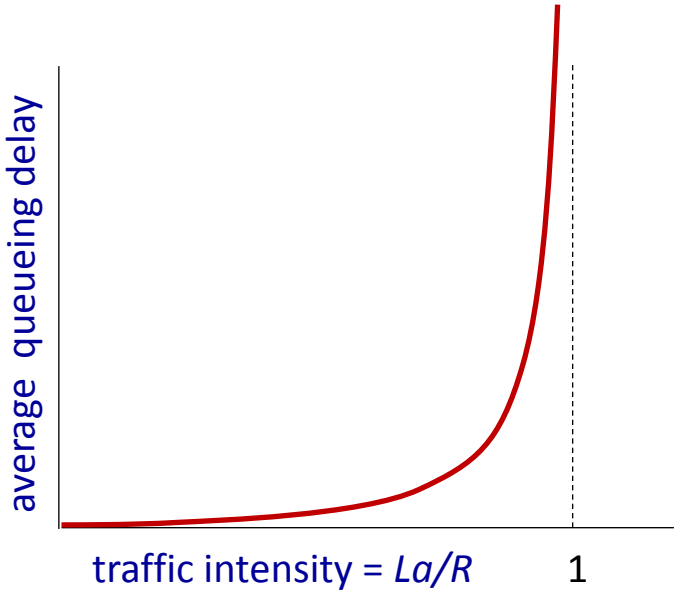
La/R > 1

Packet queueing delay (revisited)

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

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

"traffic intensity"



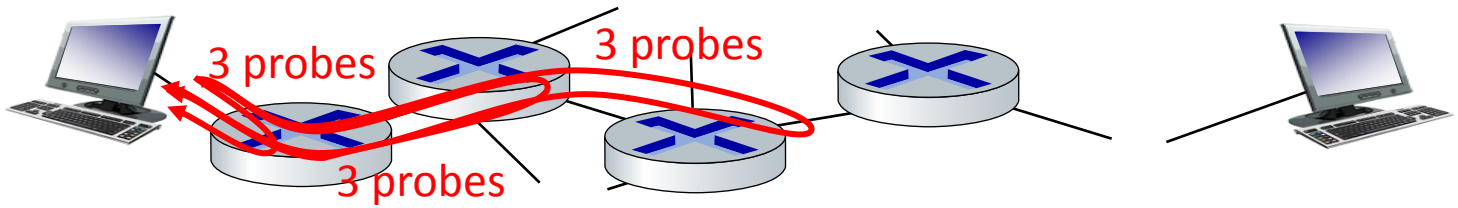
- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving is more than can be serviced - average delay infinite!

La/R ~ 0

La/R -> 1

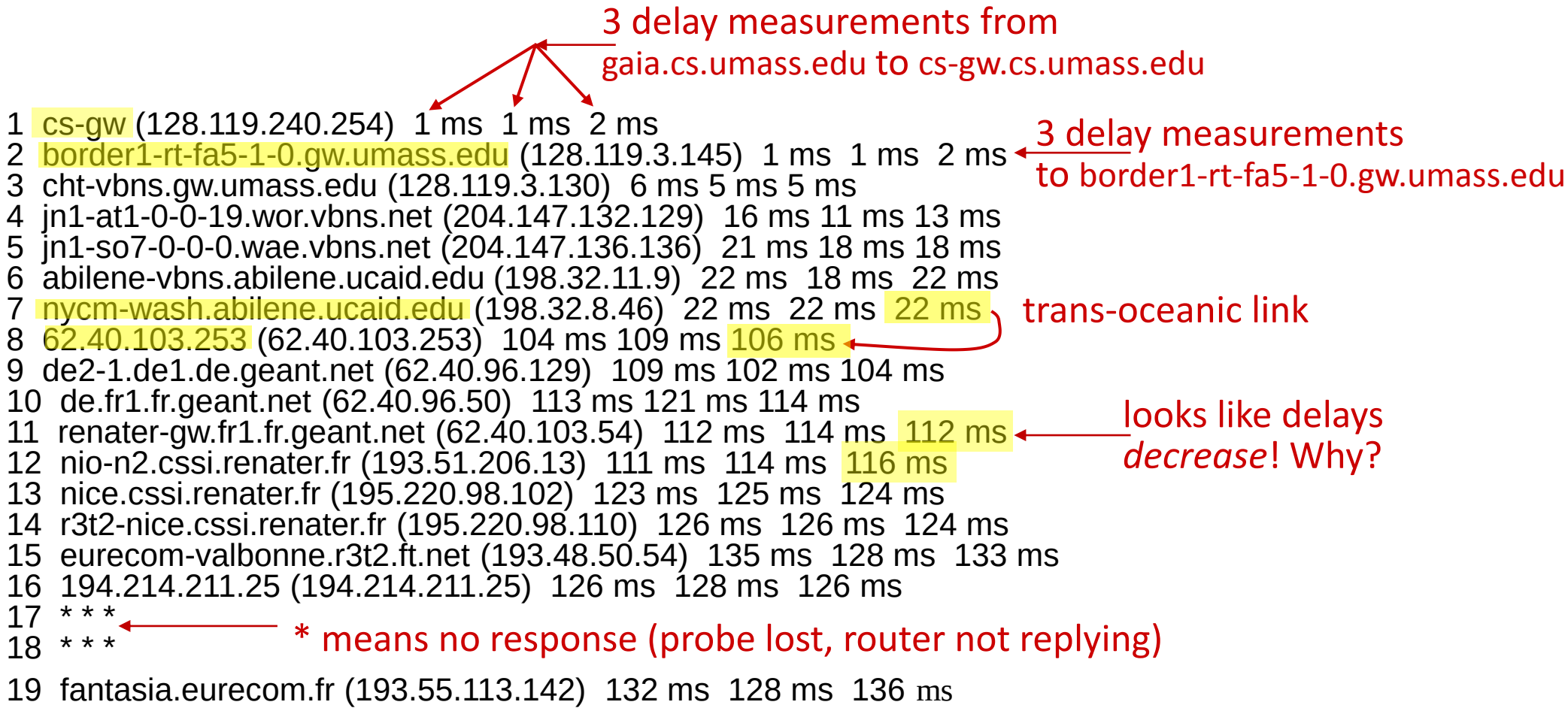
“Real” Internet delays and routes

- 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

Performance

Delay

Throughput

Loss

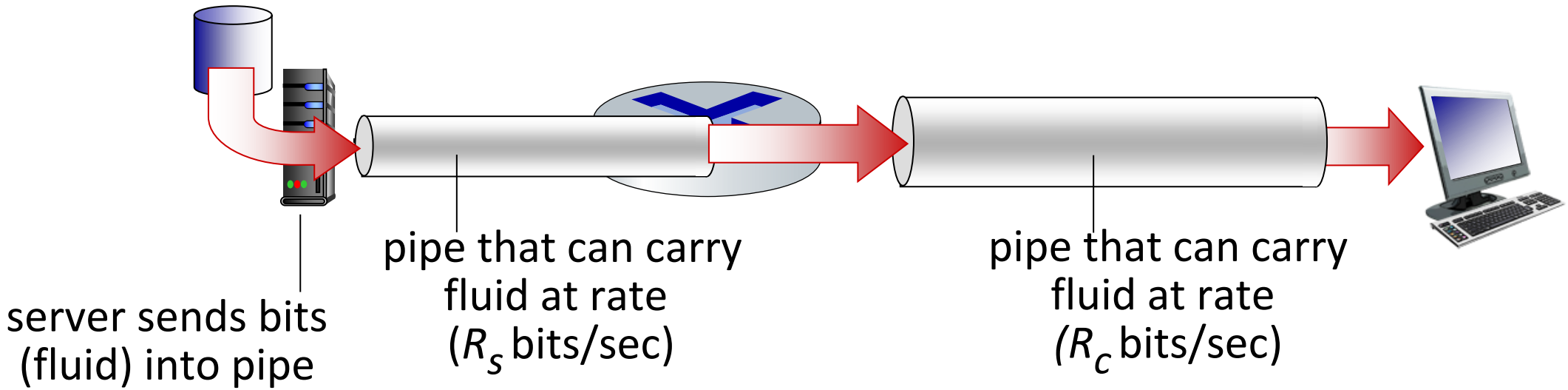
Throughput

- *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

$$Throughput = \frac{Packet\ size\ (bit)}{delay\ (s)}$$

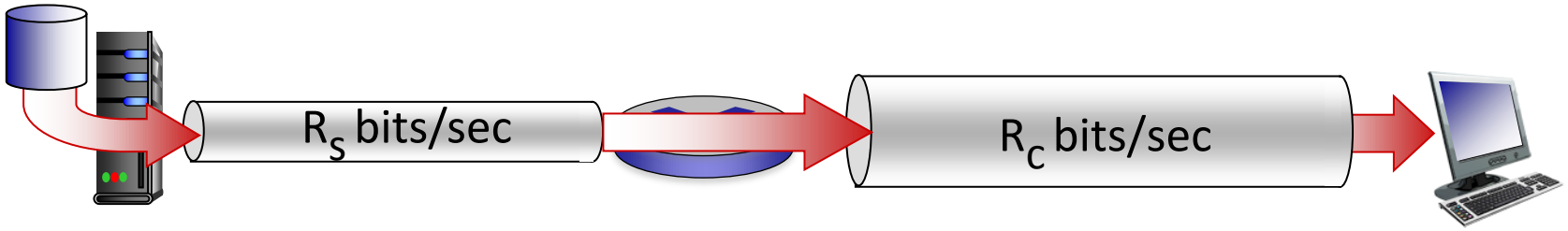
Throughput calculation: Approximate, accurate

Links as pipes

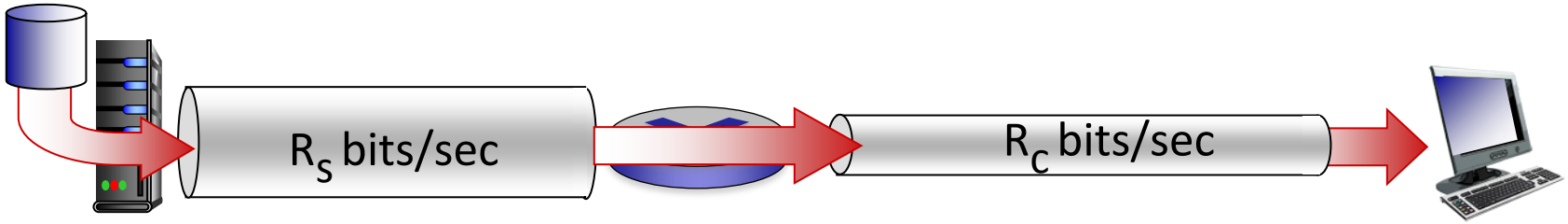


Throughput calculation: Approximate, accurate

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



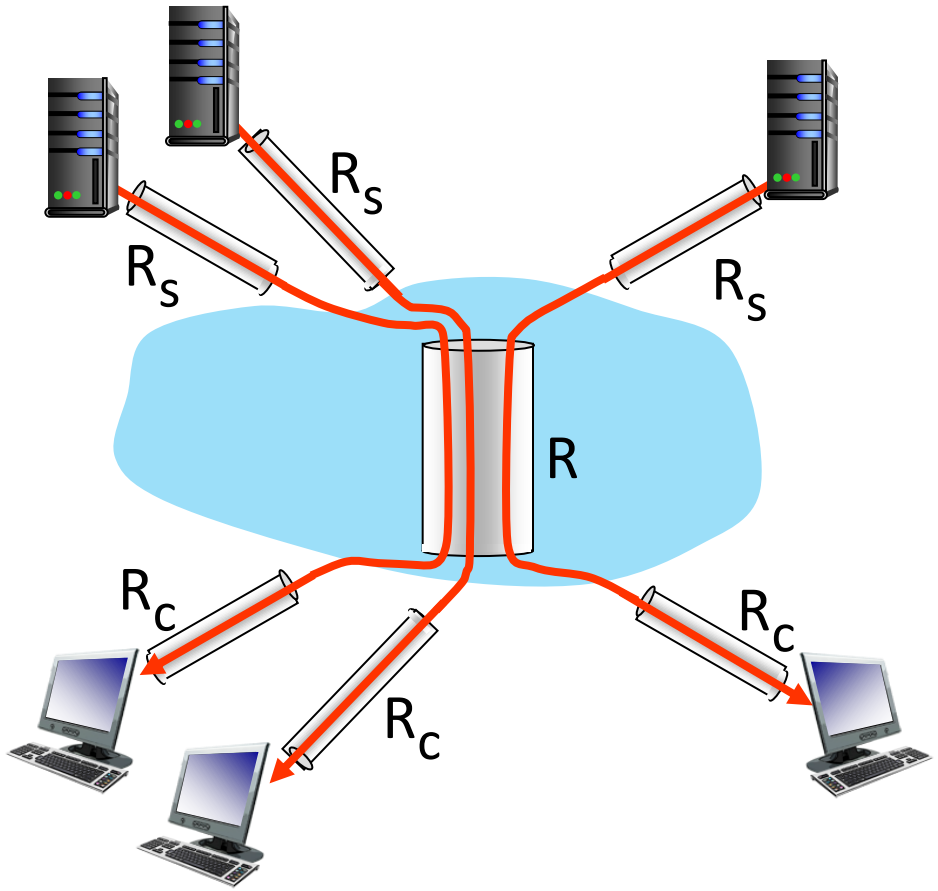
$R_s > R_c$ What is average end-end throughput?



bottleneck link

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

Throughput calculation: Approximate, accurate

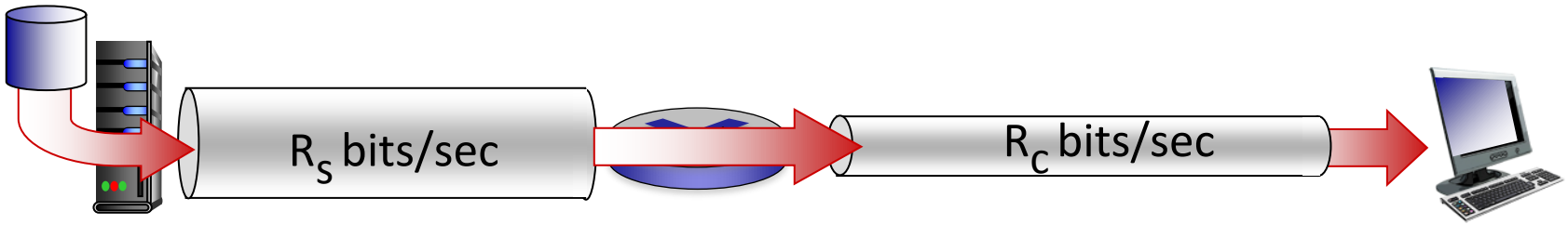


10 connections (fairly) share backbone bottleneck link *R* bits/sec

- 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/

Throughput calculation: Approximate, accurate



$$delay = \frac{L}{R_S} + \frac{d_1}{S_1} + d_{proc} + d_{quque} + \frac{L}{R_C} + \frac{d_2}{S_2} \rightarrow Throughput = \frac{L}{\frac{L}{R_S} + \frac{d_1}{S_1} + d_{proc} + d_{quque} + \frac{L}{R_C} + \frac{d_2}{S_2}}$$

Just considering transmission delay

$$\Rightarrow Throughput \simeq \frac{L}{\frac{L}{R_S} + \frac{L}{R_C}} = \frac{R_S R_C}{R_S + R_C} \simeq \min(R_S, R_C)$$

Performance

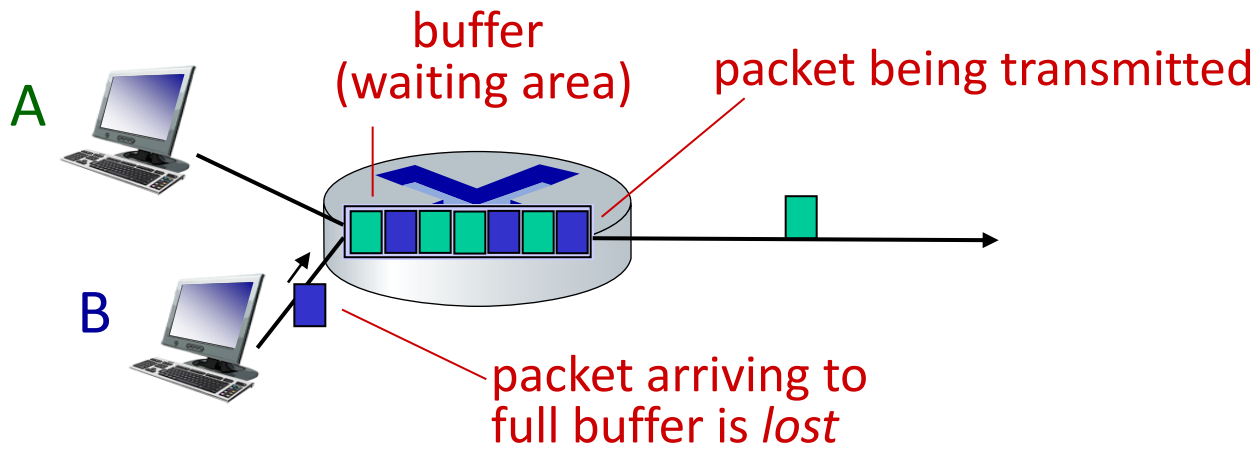
Delay

Throughput

Loss

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

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

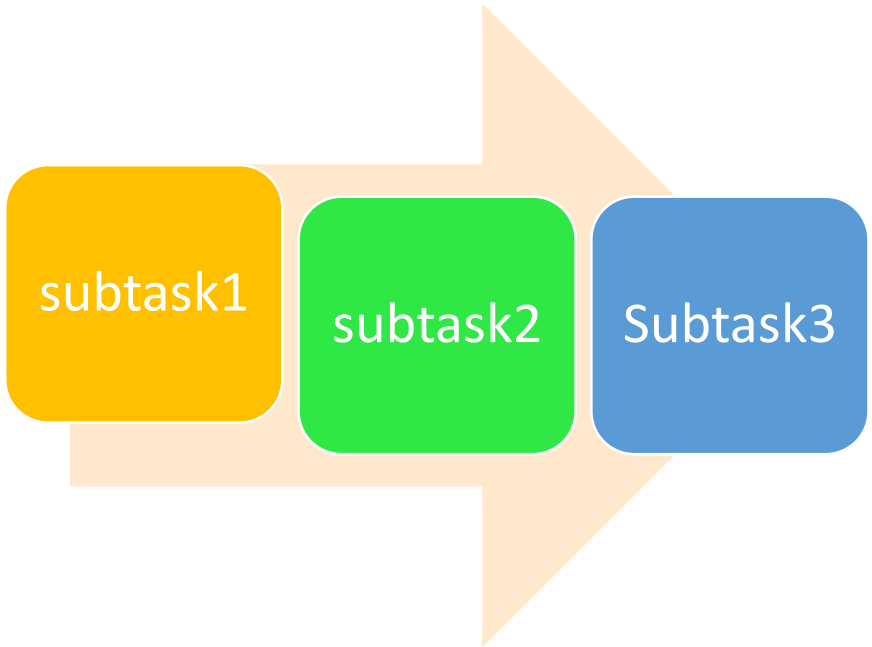
Question: is there any hope of *organizing* structure of network?

- and/or our *discussion* of networks?

Layering is every where

Protocol layering (stack)

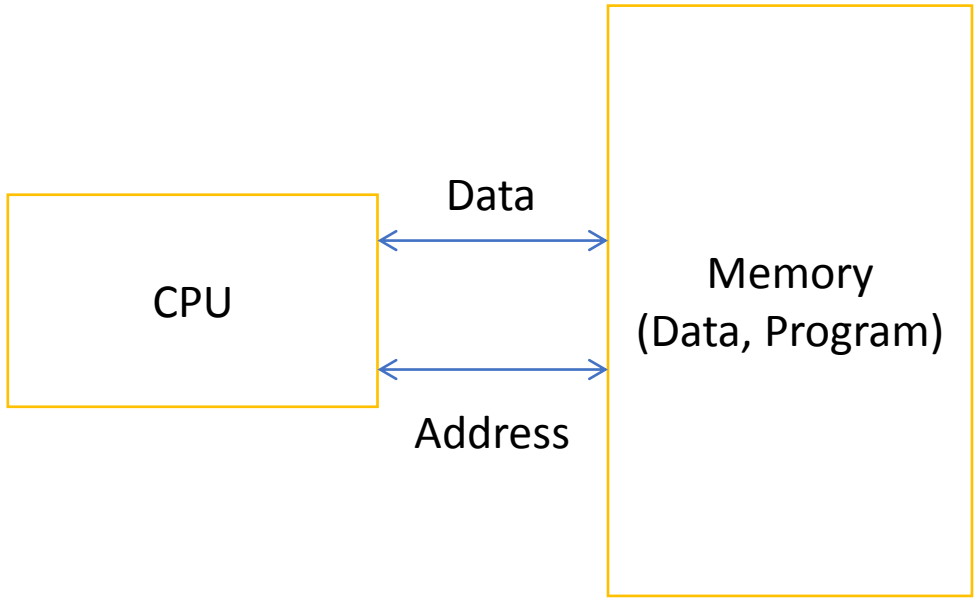
We usually organize our tasks into sub-tasks and define the interfaces



Example 1

We usually organize our tasks into sub-tasks and define the interfaces

- Task: Computation
- Sub-tasks: Hardware design & software design
- Interfaces: ISA



Example 2

We usually organize our tasks into sub-tasks and define the interfaces

- Task: Programming
- Sub-tasks: Procedures
- Interfaces: Arguments/return values

```
#include <stdio.h>
int main(){
      printf("hello world"):
}
```

Example 3

We usually organize our tasks into sub-tasks and define the interfaces

- Task: travel
- Sub-tasks: agency, check-in, gates, …
- Interfaces: Passport, ticket, serial number, …

ticket (purchase)

baggage (check)

gates (load)

runway takeoff

airplane routing

Example 3

We usually organize our tasks into sub-tasks and define the interfaces

- Task: travel
- Sub-tasks: agency, check-in, gates, …
- Interfaces: Passport, ticket, serial number, …

ticket (purchase)

baggage (check)

gates (load)

runway takeoff

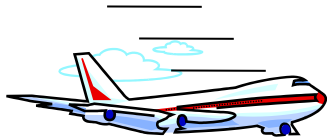
airplane routing

The Computer network's goal is similar to human's travel:

>> Transporting packets from source application to the destination application

>> Transporting humans from home airport to destination airport

So let's look deeper



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

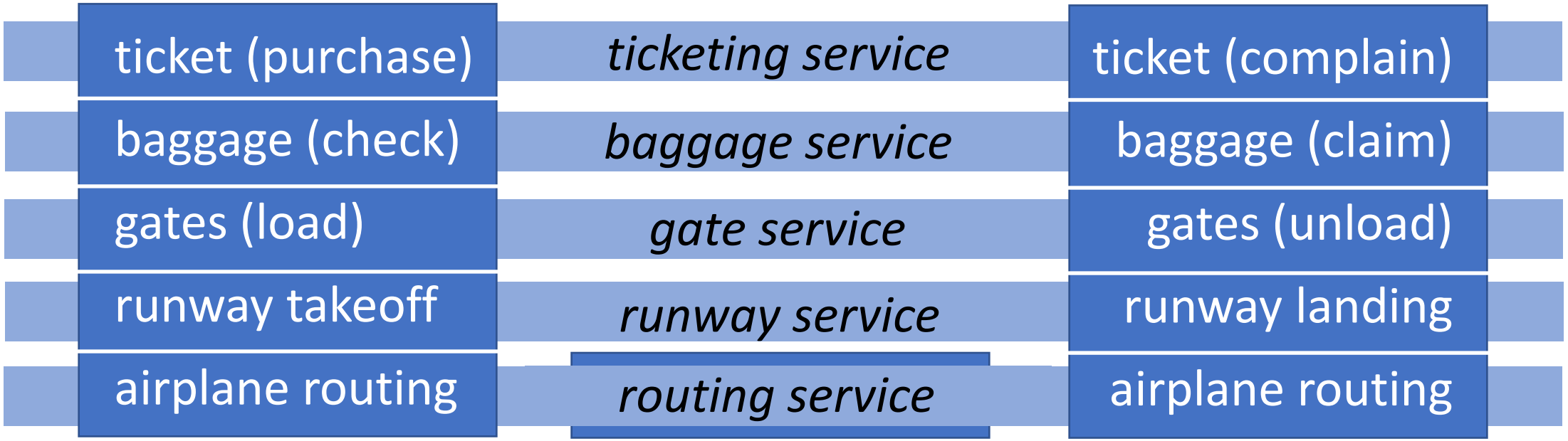
airplane routing

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?

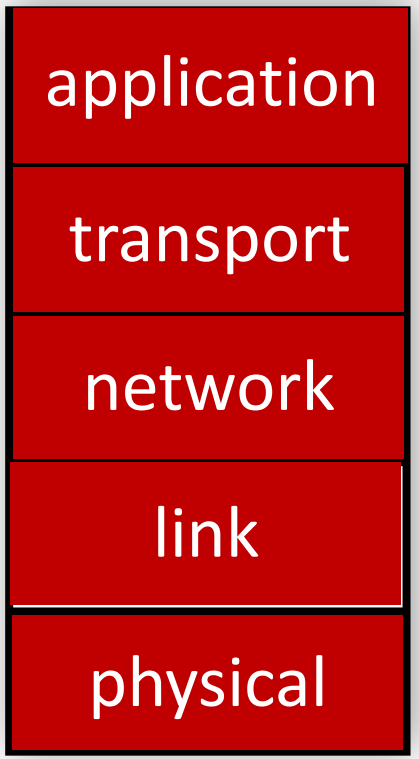
Approach to designing/discussing complex systems:

- 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

Layering is every where

Protocol layering (stack)

Layered Internet protocol stack

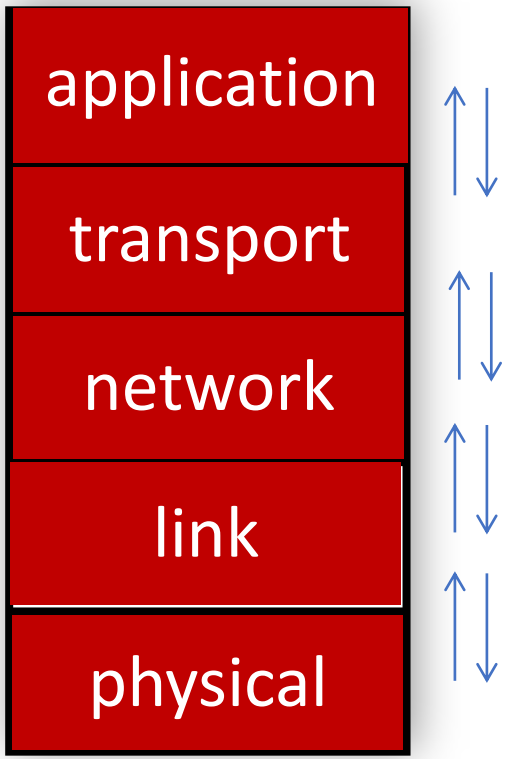


Layered Internet protocol stack

Stack:

Each layer connects to Just above and below layers

Gets service from the below
Gives service to the above

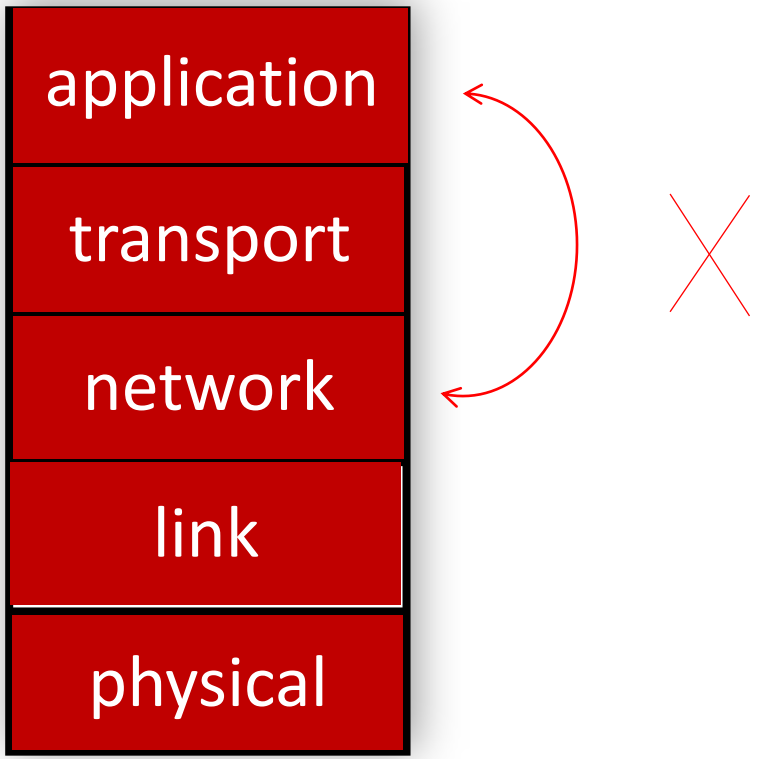


Layered Internet protocol stack

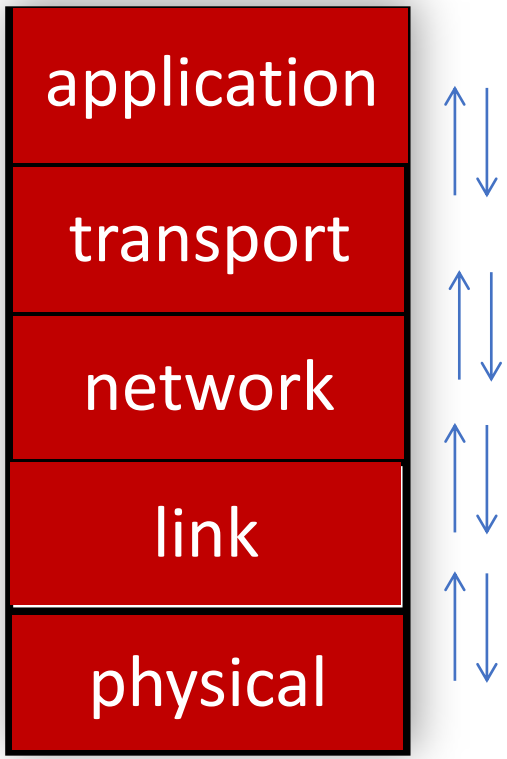
Stack:

Each layer connects to Just above and below layers

Gets service from the below
Gives service to the above



Layered Internet protocol stack



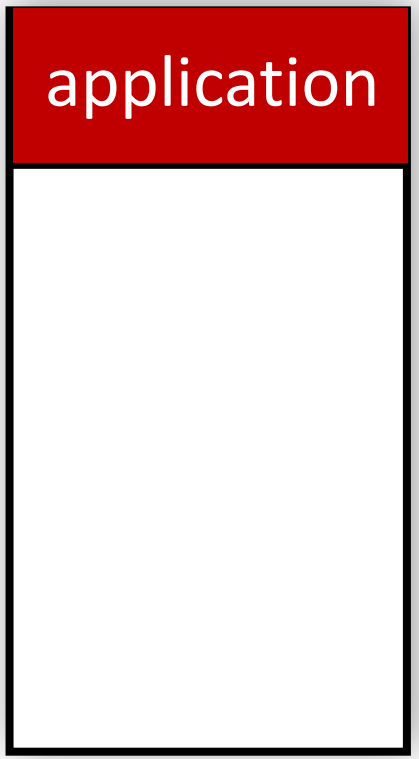
Each layer does his jobs by adding its header to the packet

(Encapsulation)

Layered Internet protocol stack

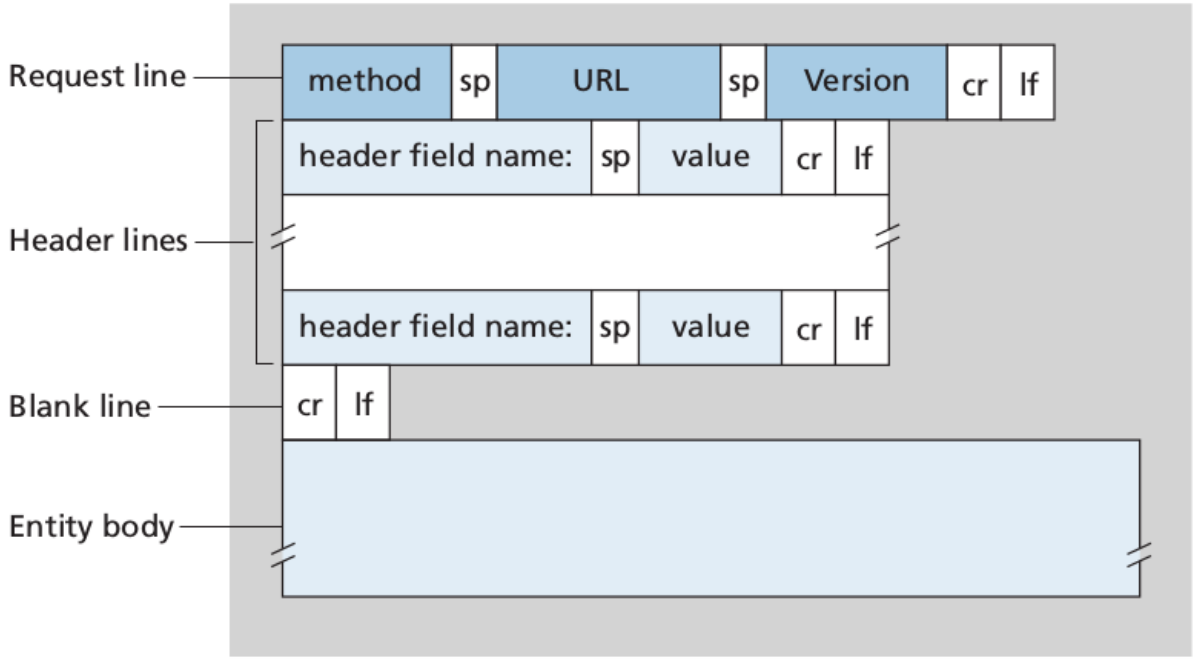
- *application:* supporting network applications
  - HTTP, IMAP, SMTP, DNS

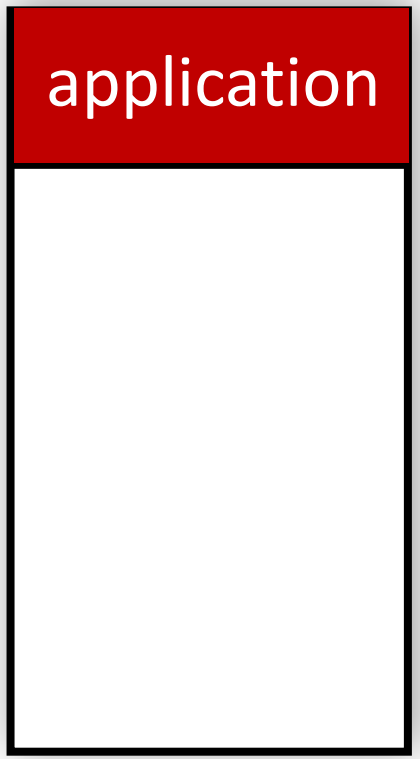
HTTP Demo using telnet



Layered Internet protocol stack

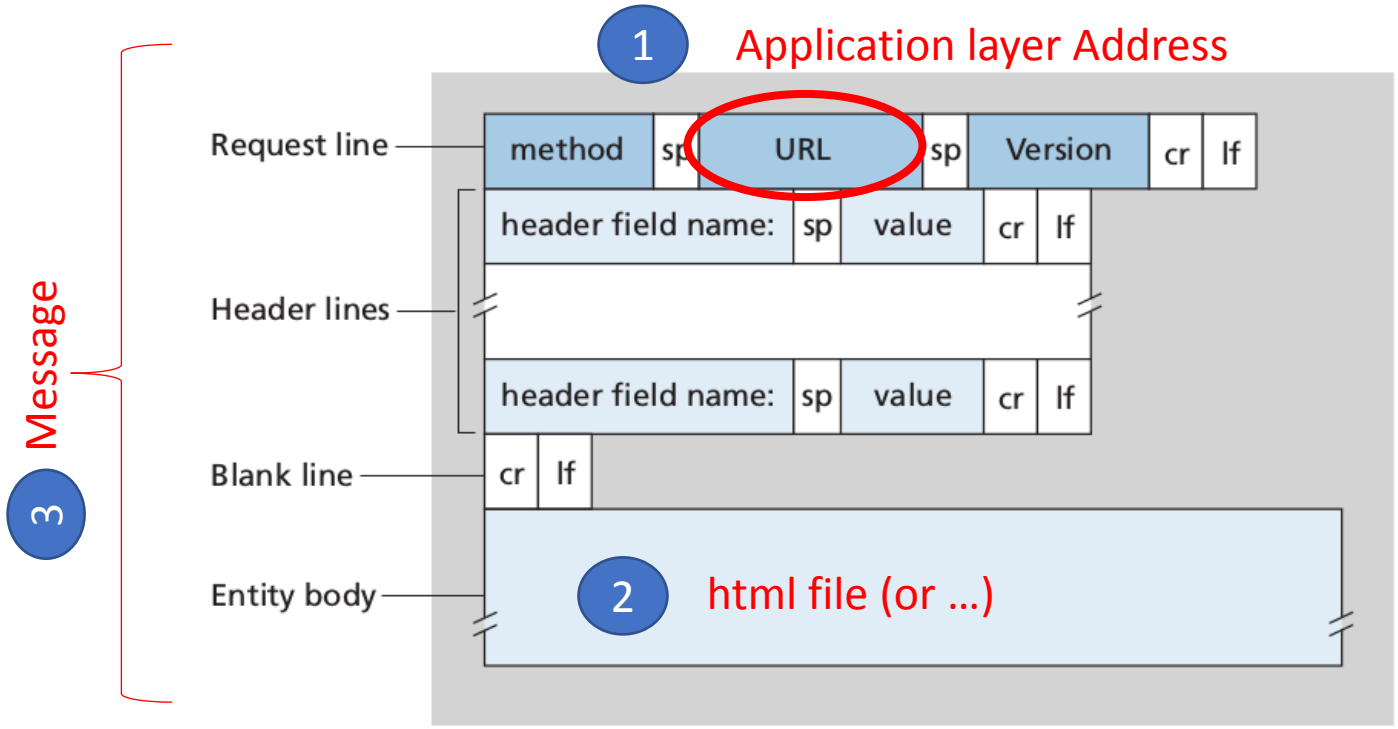
- *application:* supporting network applications
  - HTTP, IMAP, SMTP, DNS

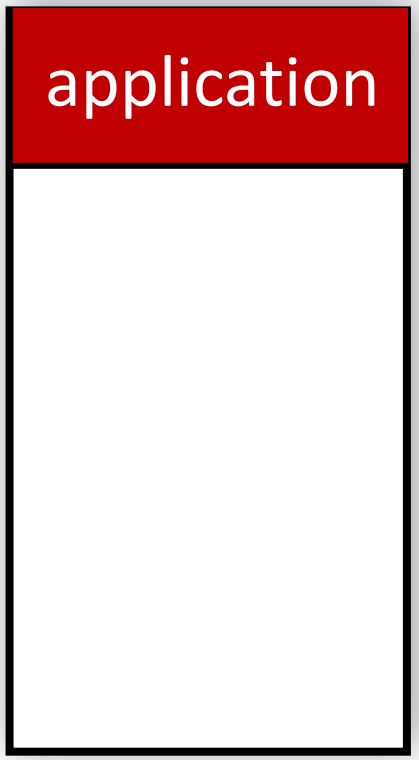




Layered Internet protocol stack

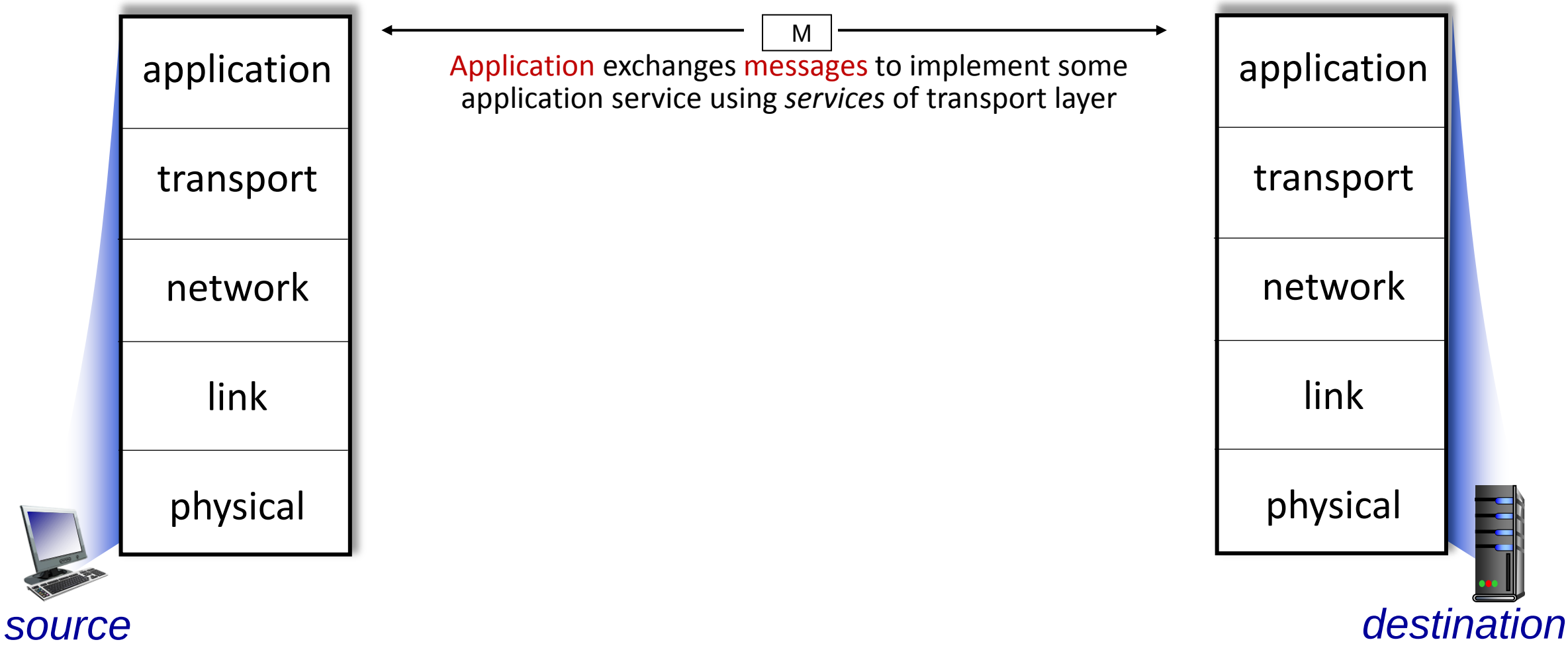
- *application:* supporting network applications
  - HTTP, IMAP, SMTP, DNS





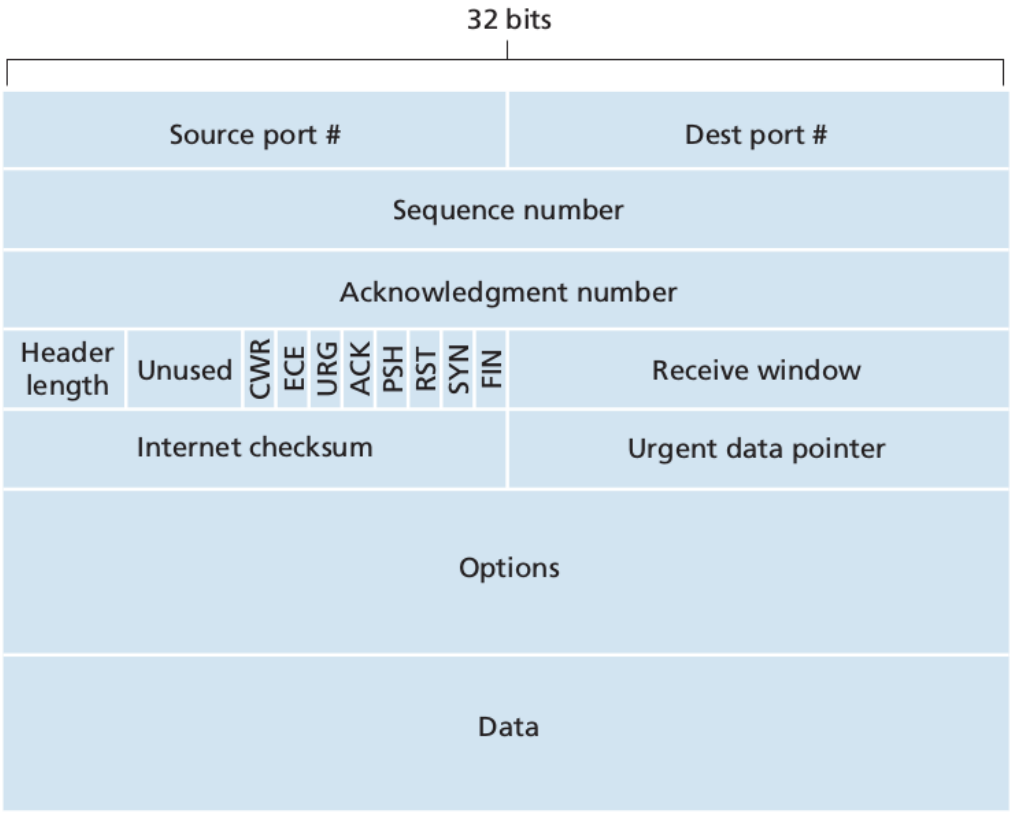
4 Only in end-systems

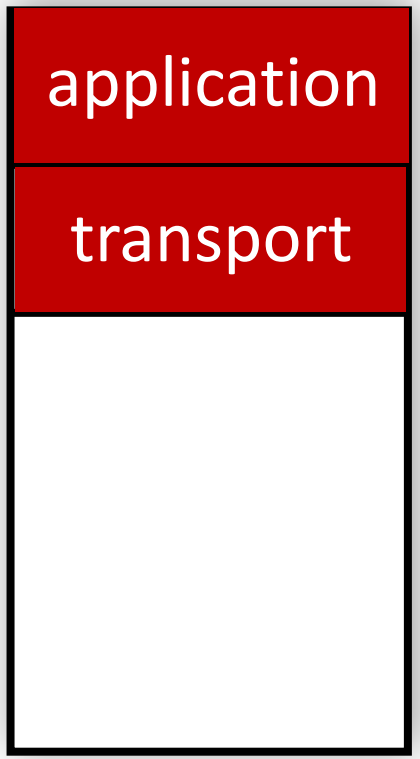
Encapsulation



Layered Internet protocol stack

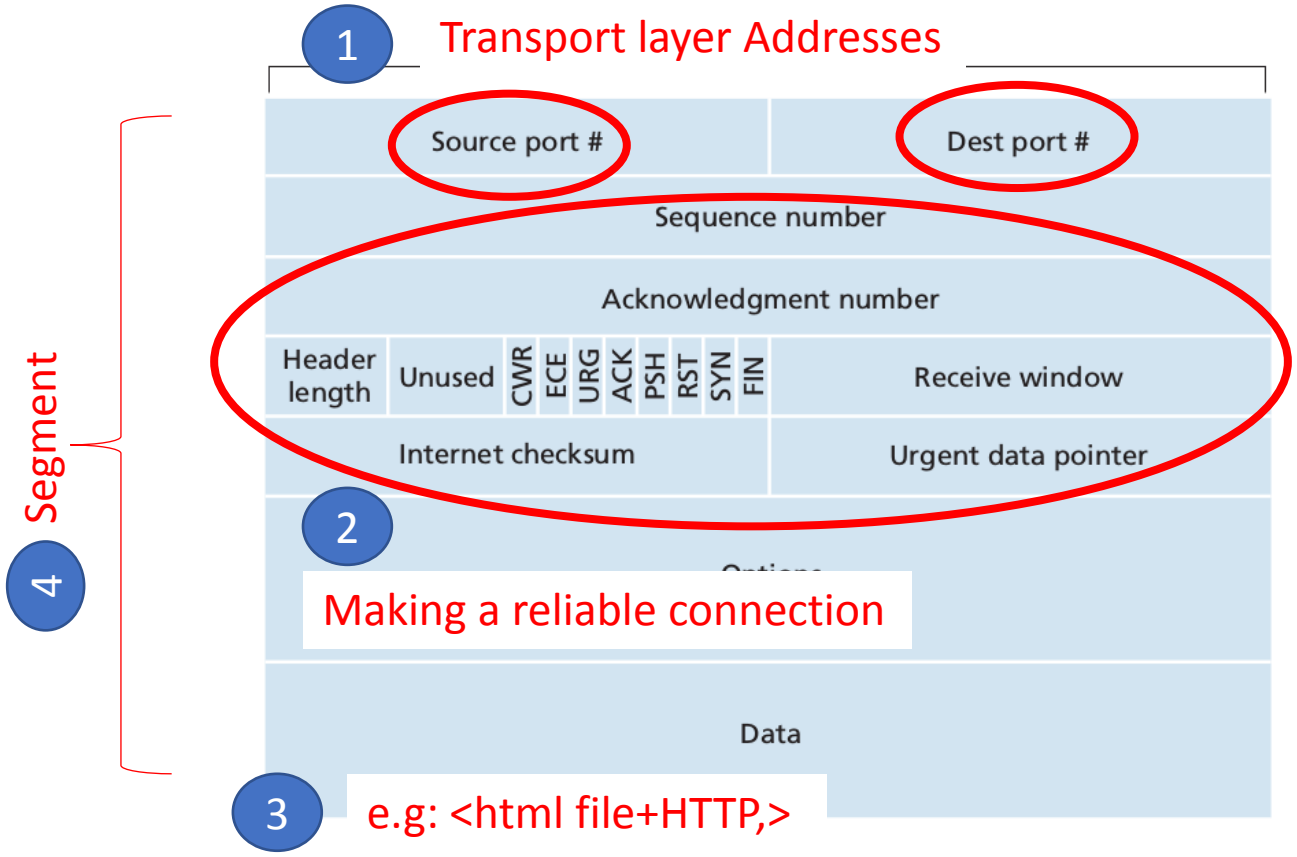
- *transport:* process-process data transfer
  - TCP, UDP

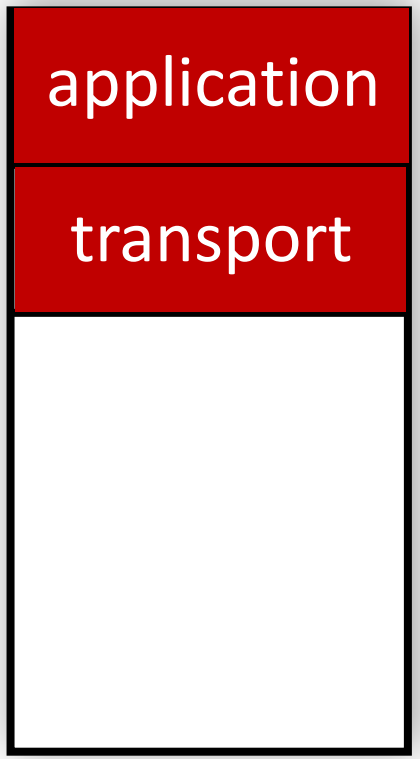




Layered Internet protocol stack

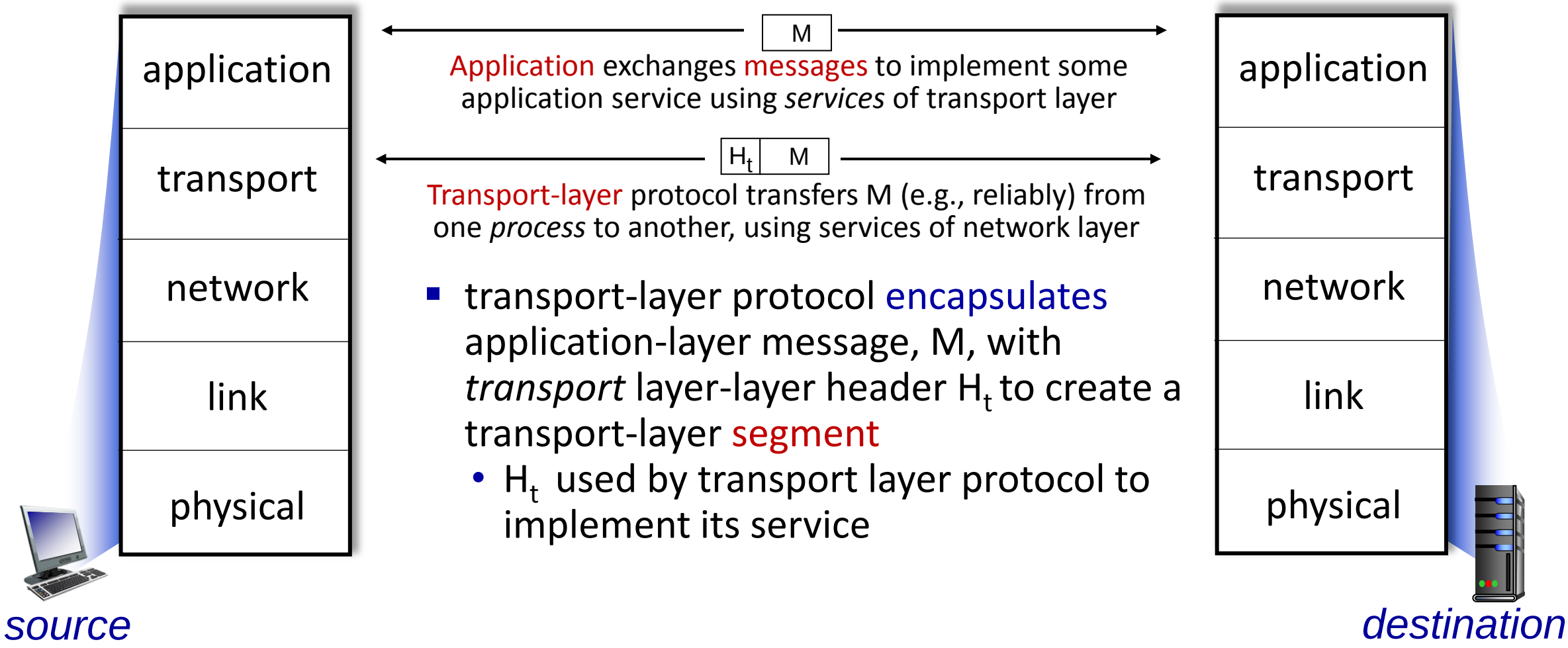
- *transport:* process-process data transfer
  - TCP, UDP





5 Only in end-systems

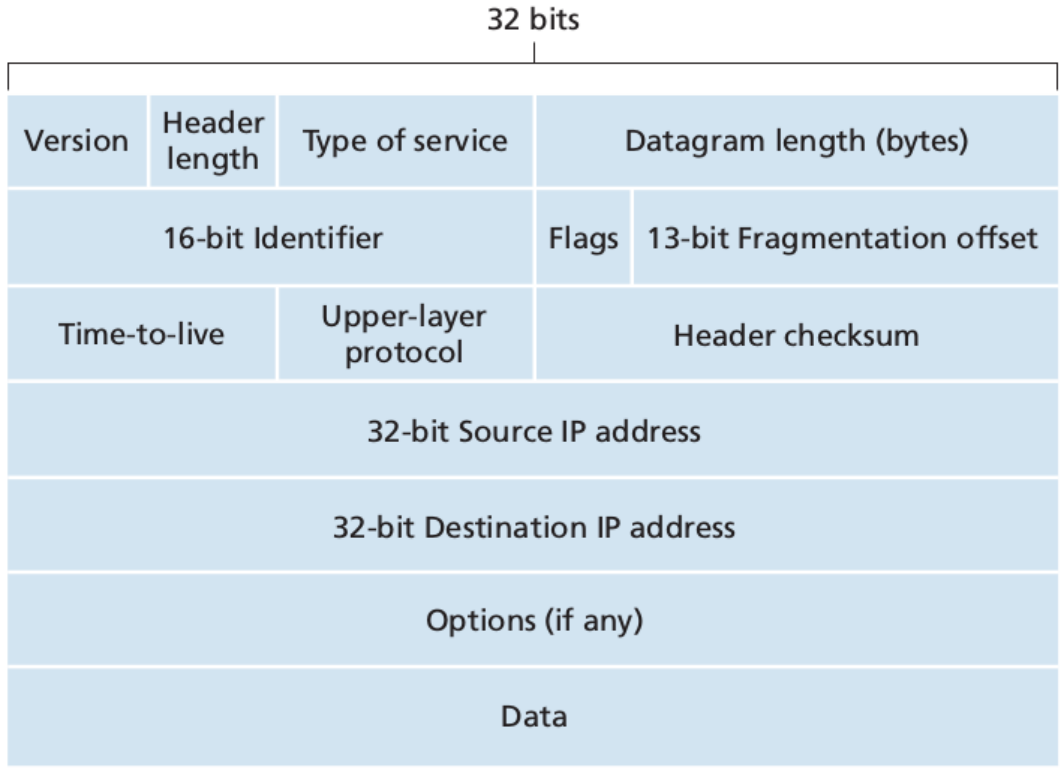
Encapsulation

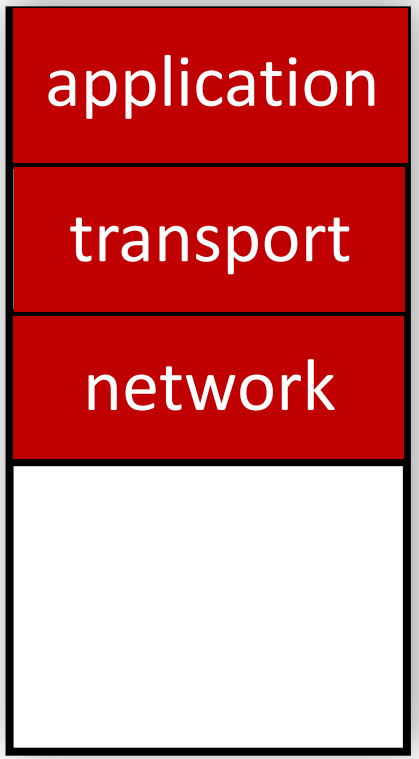


- transport-layer protocol encapsulates application-layer message, M, with *transport* layer-layer header H_t to create a transport-layer segment
  - H_t used by transport layer protocol to implement its service

Layered Internet protocol stack

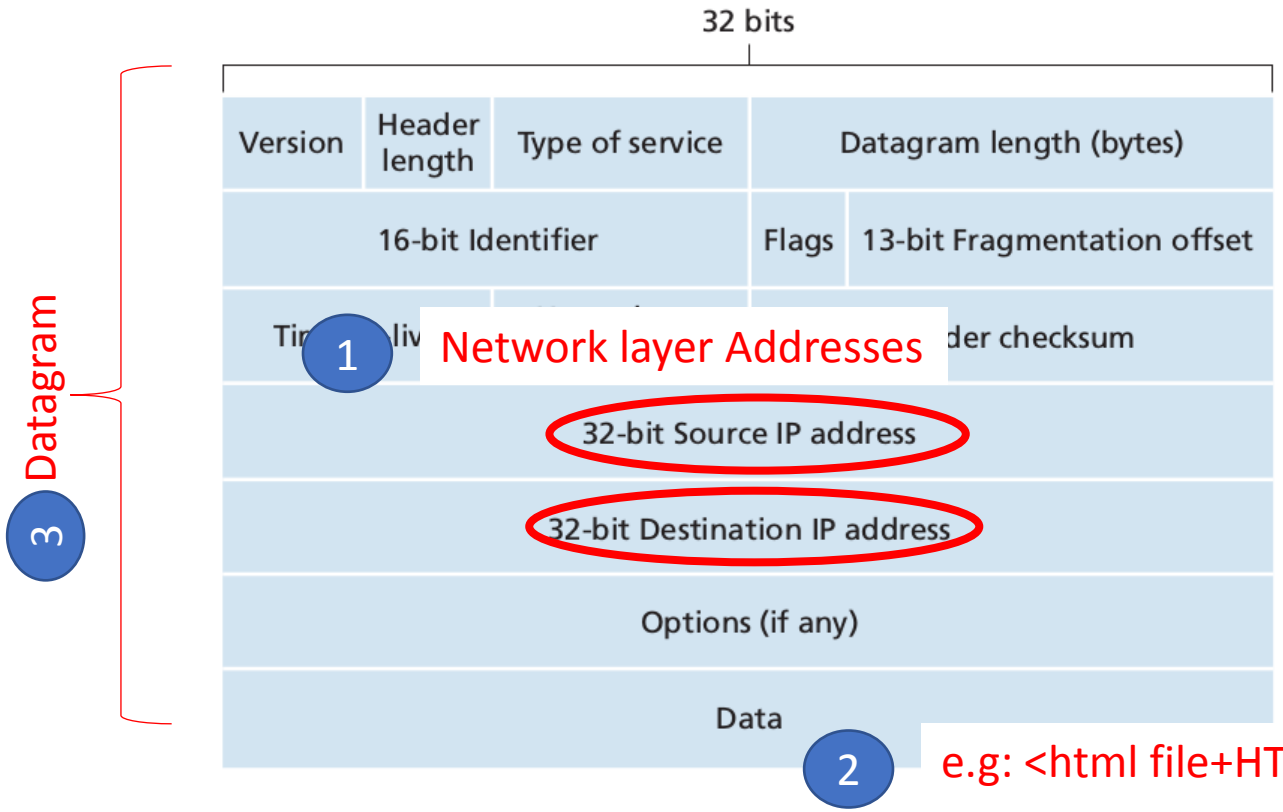
- *network:* routing of datagrams from source to destination
  - IP, routing protocols

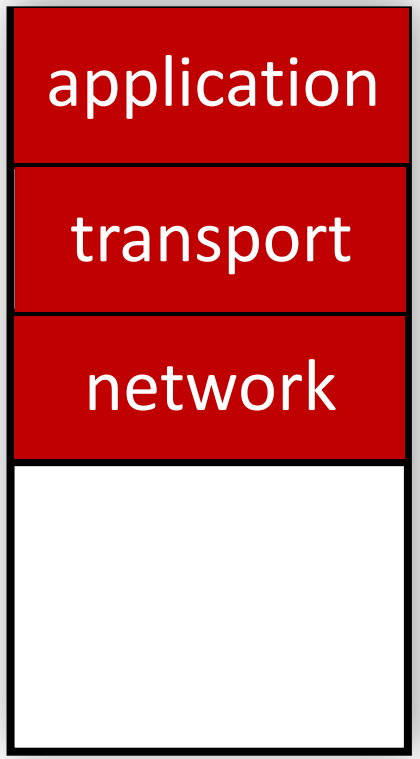




Layered Internet protocol stack

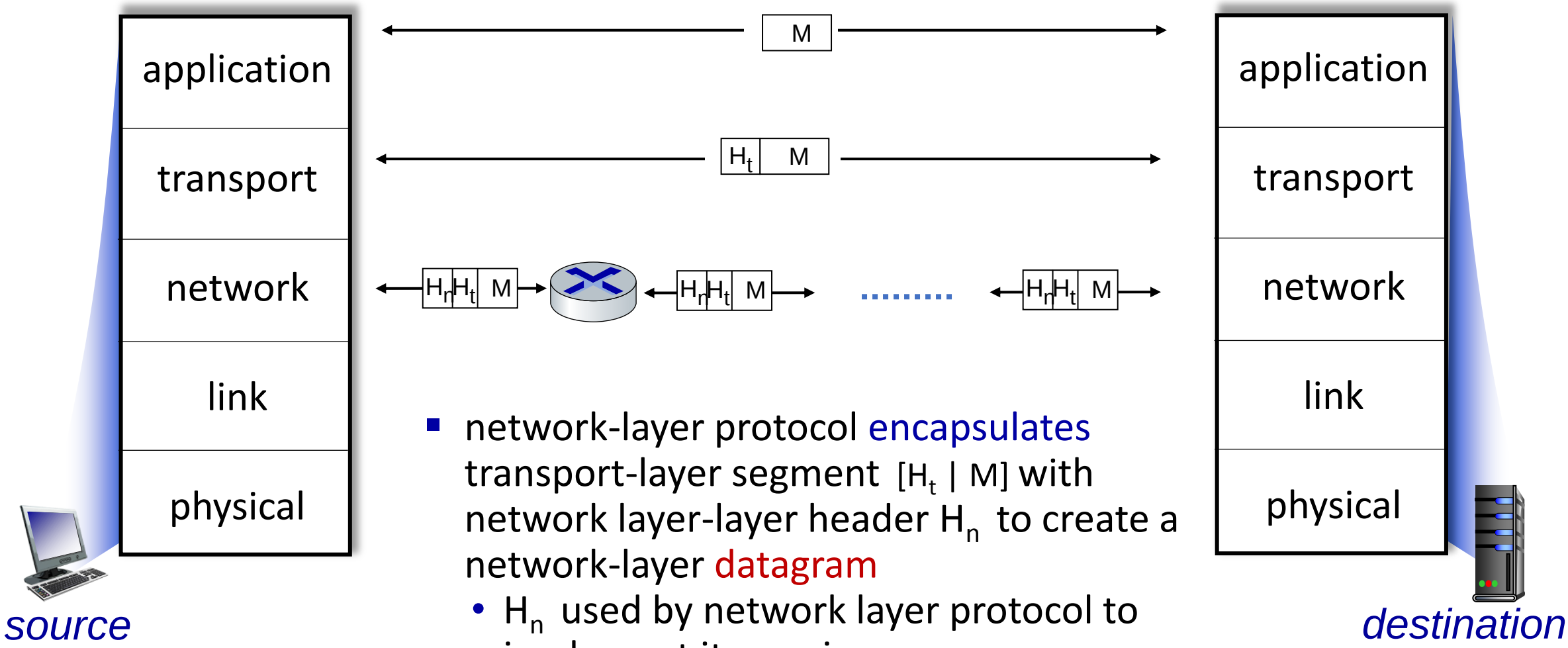
- *network:* routing of datagrams from source to destination
  - IP, routing protocols





4 In end-systems & routers

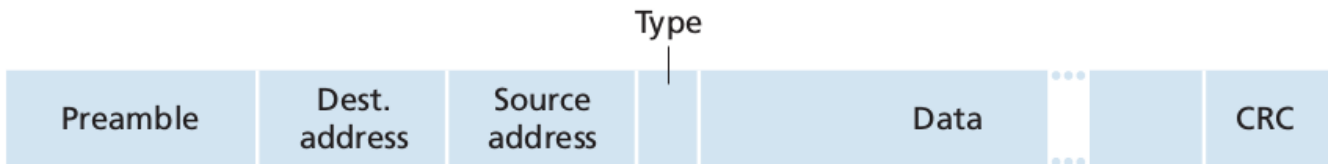
Encapsulation

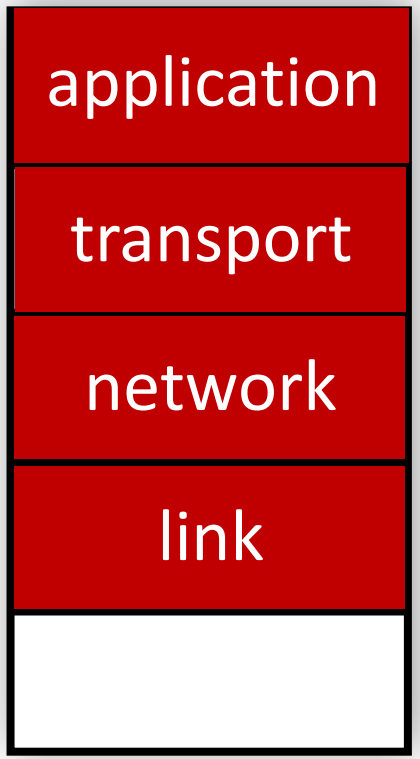


- network-layer protocol encapsulates transport-layer segment [H_t | M] with network layer-layer header H_n to create a network-layer datagram
  - H_n used by network layer protocol to implement its service

Layered Internet protocol stack

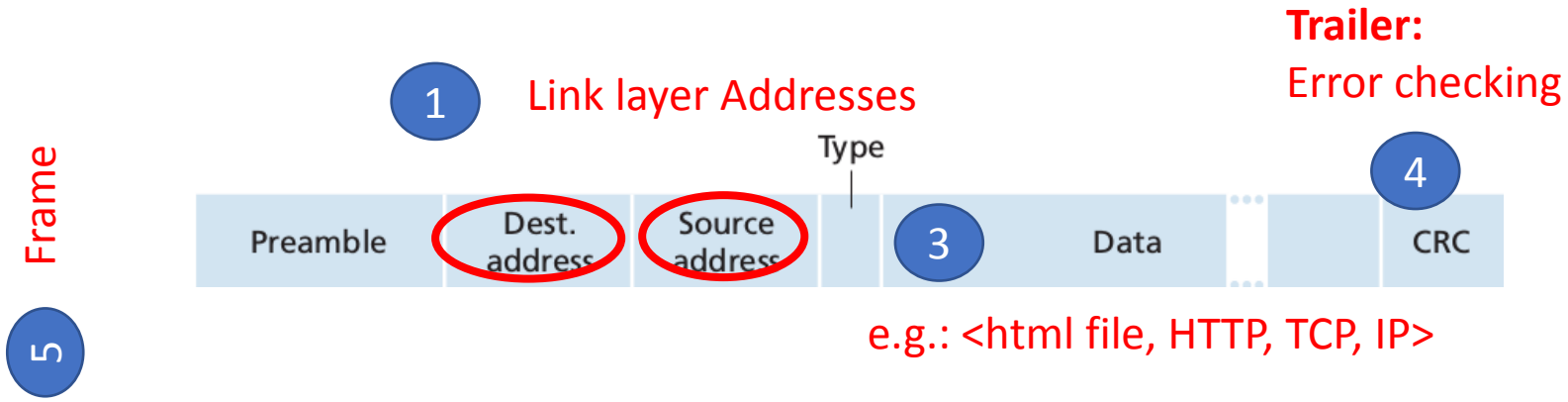
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP

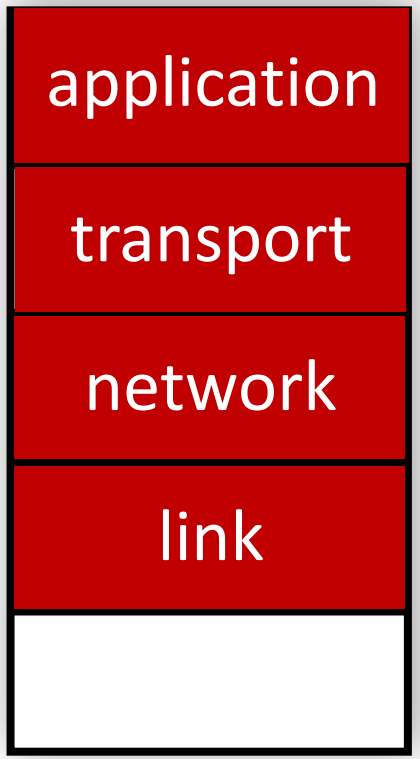




Layered Internet protocol stack

- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP

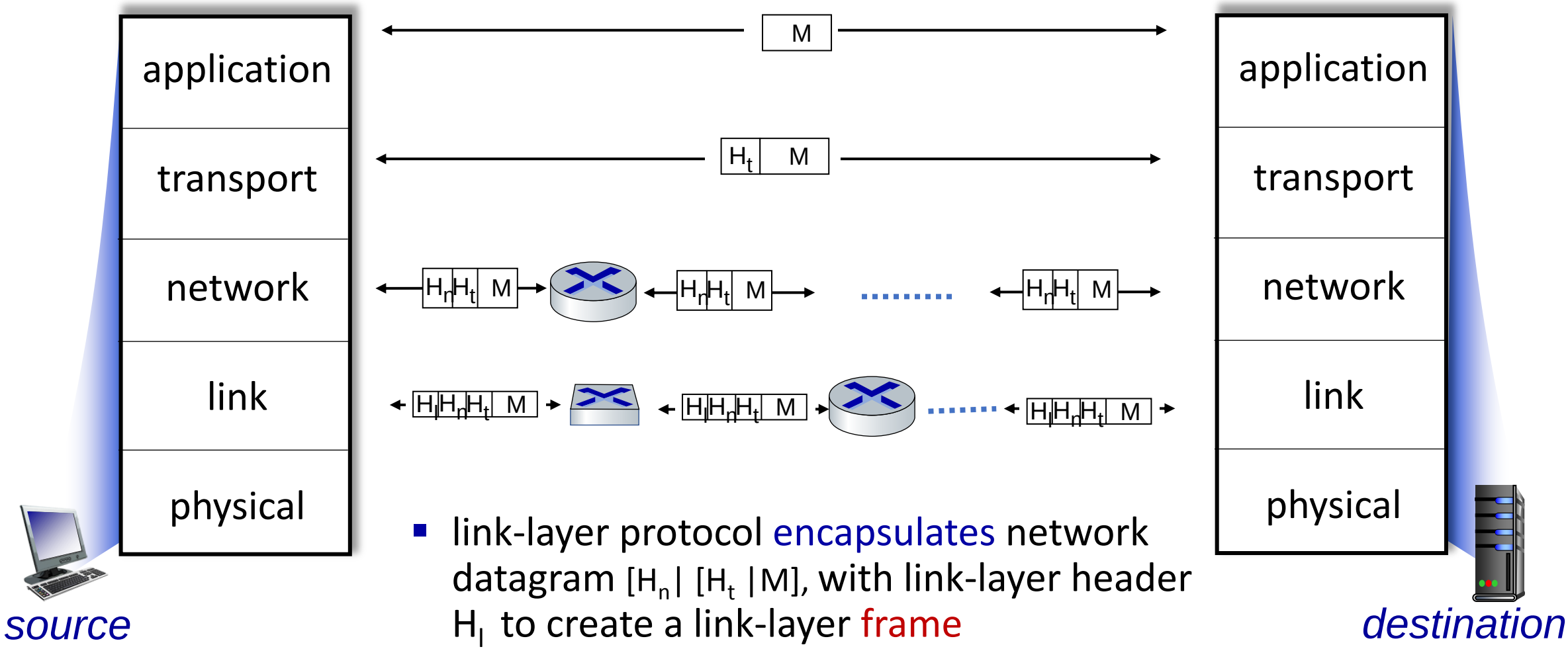




2 Link addresses (e.g. MAC) are different from IP address
>> IP addresses are fixed from source to destination
>>Link addresses change node-to-node (exception: switch)

6 In all nodes

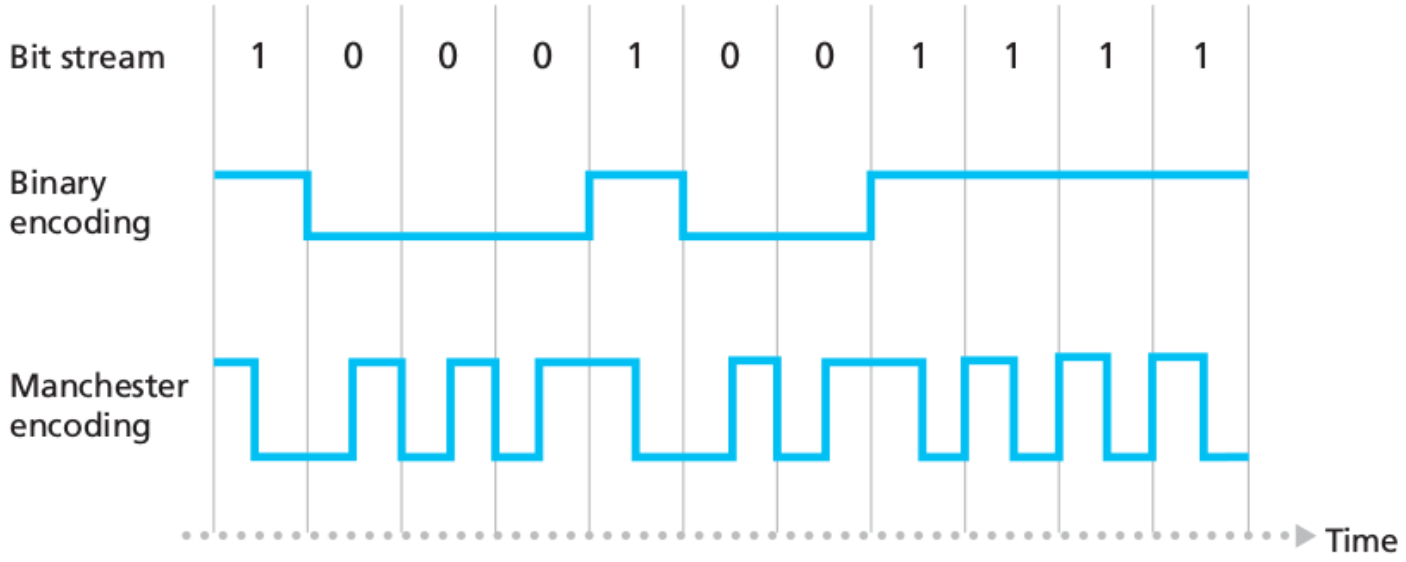
Encapsulation

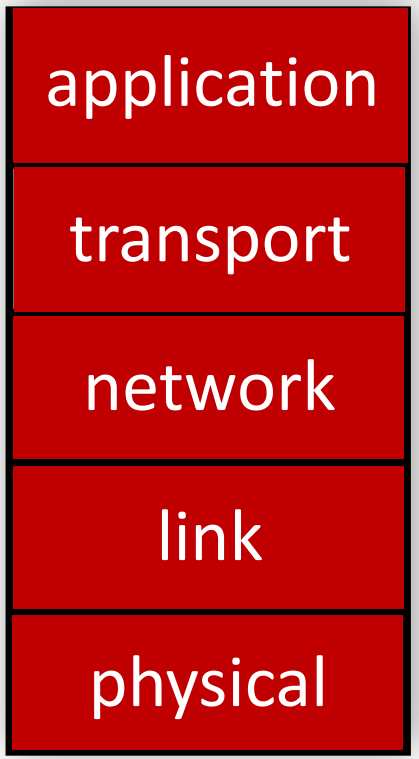


- link-layer protocol encapsulates network datagram $[H_n|\ [H_t\ |M]$, with link-layer header H_l to create a link-layer frame

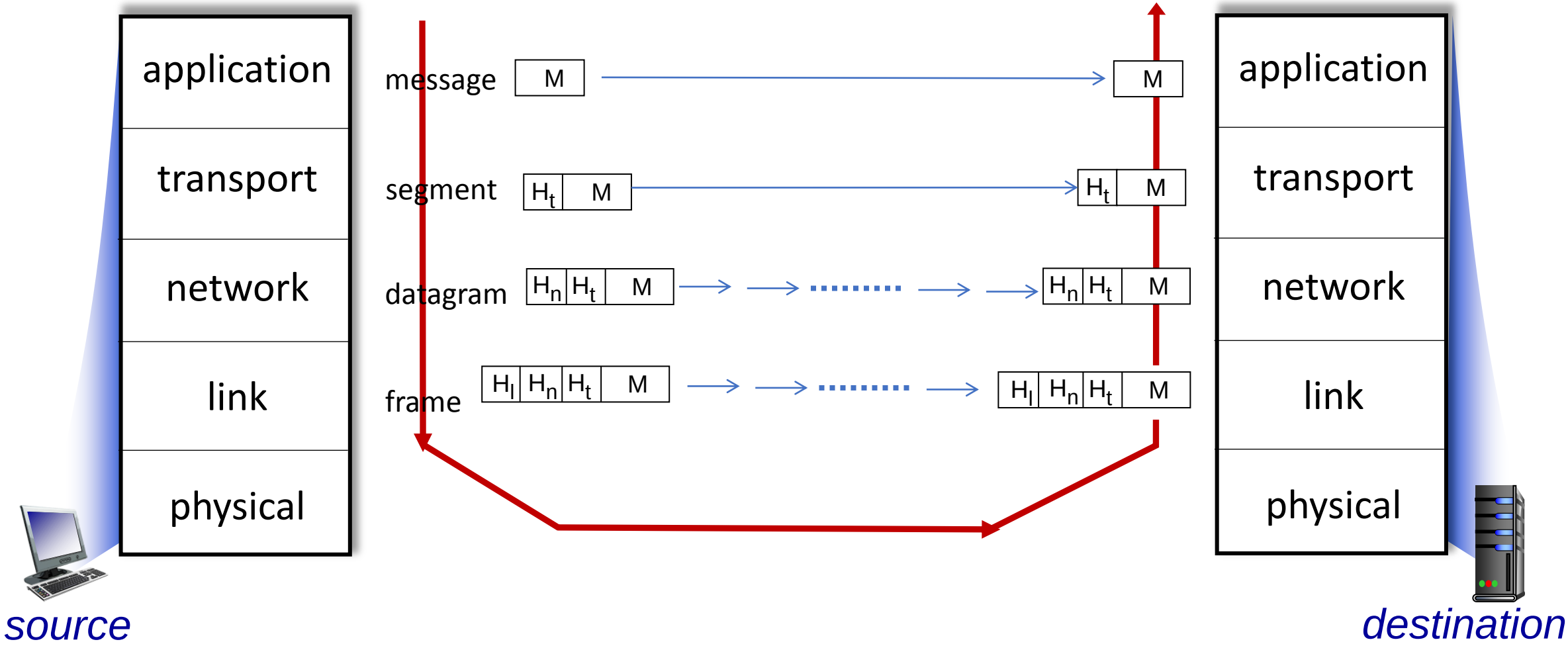
Layered Internet protocol stack

- *physical:* bits “on the wire”

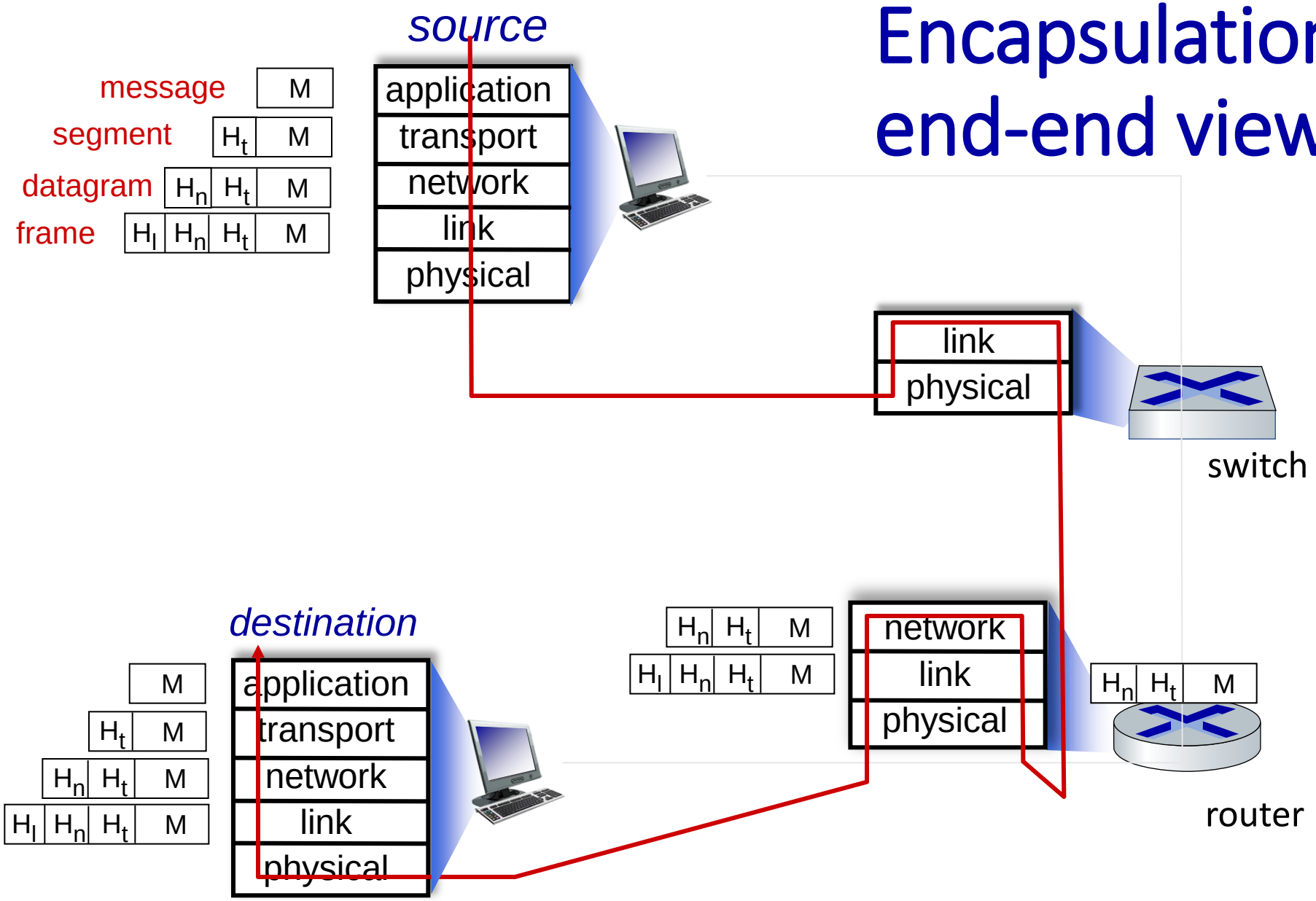




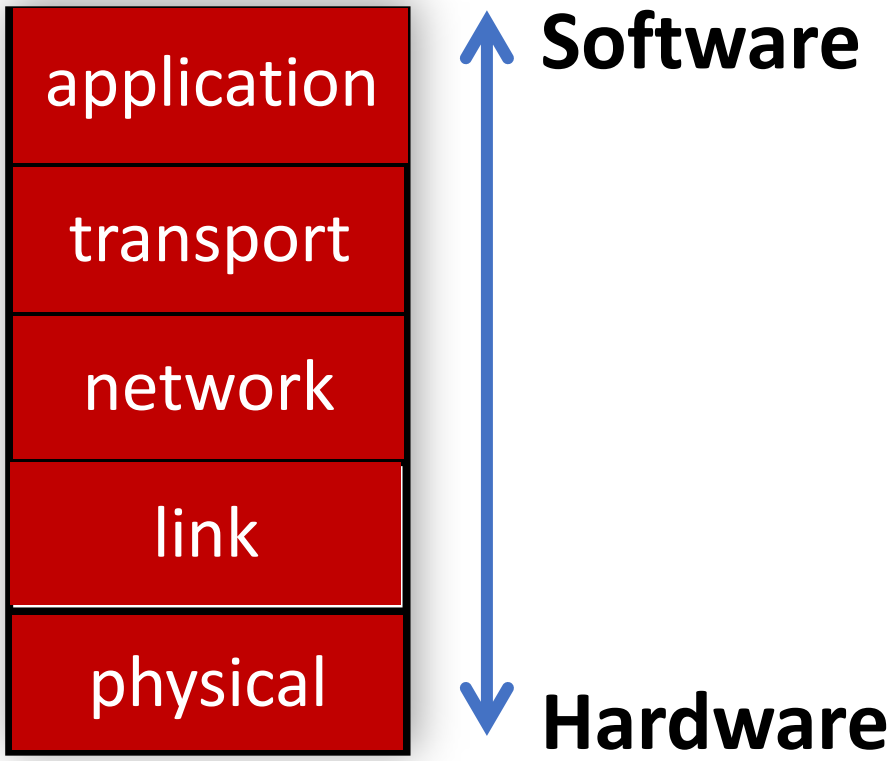
Encapsulation/De-encapsulation



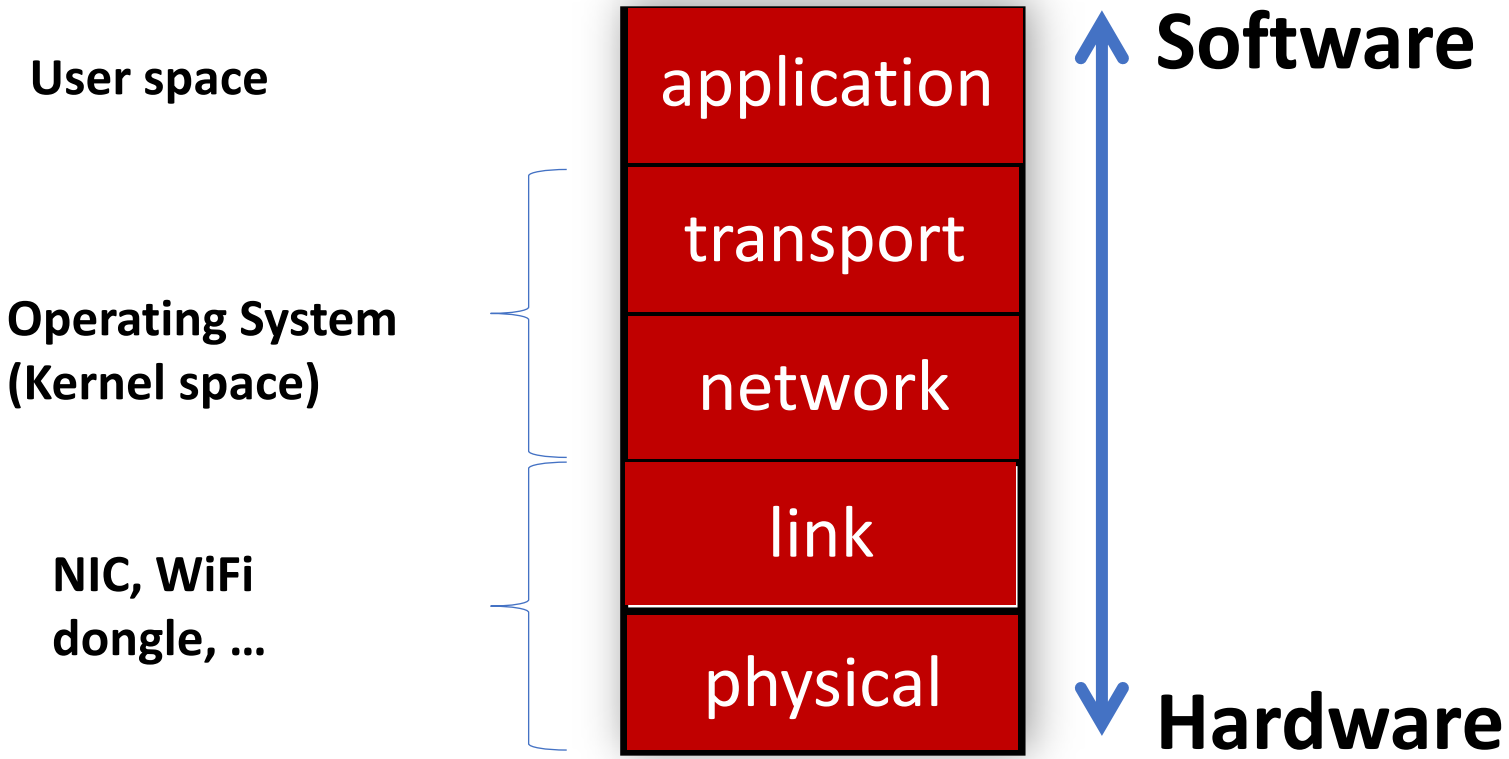
Encapsulation: an end-end view



Layered Internet protocol stack

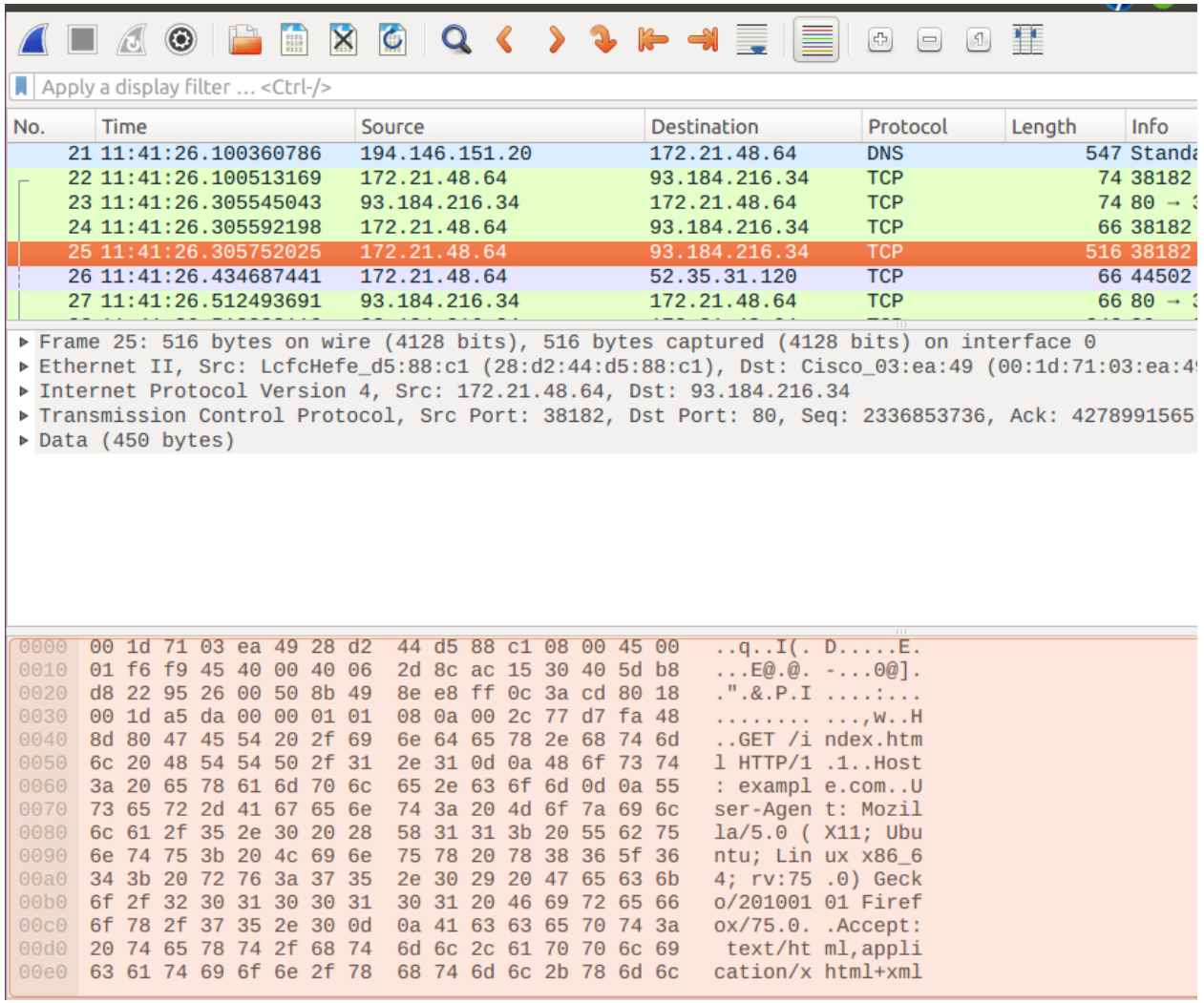


Layered Internet protocol stack



Wireshark


WIRESHARK


Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
21 11:41:26.100360786 194.146.151.20 172.21.48.64 DNS 547 Standa
22 11:41:26.100513169 172.21.48.64 93.184.216.34 TCP 74 38182
23 11:41:26.305545043 93.184.216.34 172.21.48.64 TCP 74 80 →
24 11:41:26.305592198 172.21.48.64 93.184.216.34 TCP 66 38182
25 11:41:26.305752025 172.21.48.64 93.184.216.34 TCP 516 38182
26 11:41:26.434687441 172.21.48.64 52.35.31.120 TCP 66 44502
27 11:41:26.512493691 93.184.216.34 172.21.48.64 TCP 66 80 →
▸ Frame 25: 516 bytes on wire (4128 bits), 516 bytes captured (4128 bits) on interface 0
▸ Ethernet II, Src: LcfcHefe_d5:88:c1 (28:d2:44:d5:88:c1), Dst: Cisco_03:ea:49 (00:1d:71:03:ea:4
▸ Internet Protocol Version 4, Src: 172.21.48.64, Dst: 93.184.216.34
▸ Transmission Control Protocol, Src Port: 38182, Dst Port: 80, Seq: 2336853736, Ack: 4278991565
▸ Data (450 bytes)
0000 00 1d 71 03 ea 49 28 d2 44 d5 88 c1 08 00 45 00 ..q..I(. D.....E.
0010 01 f6 f9 45 40 00 40 06 2d 8c ac 15 30 40 5d b8 ...E@.@. -...0@].
0020 d8 22 95 26 00 50 8b 49 8e e8 ff 0c 3a cd 80 18 ."&.P.I ....:...
0030 00 1d a5 da 00 00 01 01 08 0a 00 2c 77 d7 fa 48 ....,w..H
0040 8d 80 47 45 54 20 2f 69 6e 64 65 78 2e 68 74 6d ..GET /i ndex.htm
0050 6c 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 l HTTP/1 .1..Host
0060 3a 20 65 78 61 6d 70 6c 65 2e 63 6f 6d 0d 0a 55 : exampl e.com..U
0070 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c ser-Agen t: Mozil
0080 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 55 62 75 la/5.0 (X11; Ubu
0090 6e 74 75 3b 20 4c 69 6e 75 78 20 78 38 36 5f 36 ntu; Lin ux x86_6
00a0 34 3b 20 72 76 3a 37 35 2e 30 29 20 47 65 63 6b 4; rv:75 .0) Geck
00b0 6f 2f 32 30 31 30 30 31 30 31 20 46 69 72 65 66 o/201001 01 Firef
00c0 6f 78 2f 37 35 2e 30 0d 0a 41 63 63 65 70 74 3a ox/75.0. .Accept:
00d0 20 74 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 text/ht ml,appli
00e0 63 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c cation/x html+xml

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

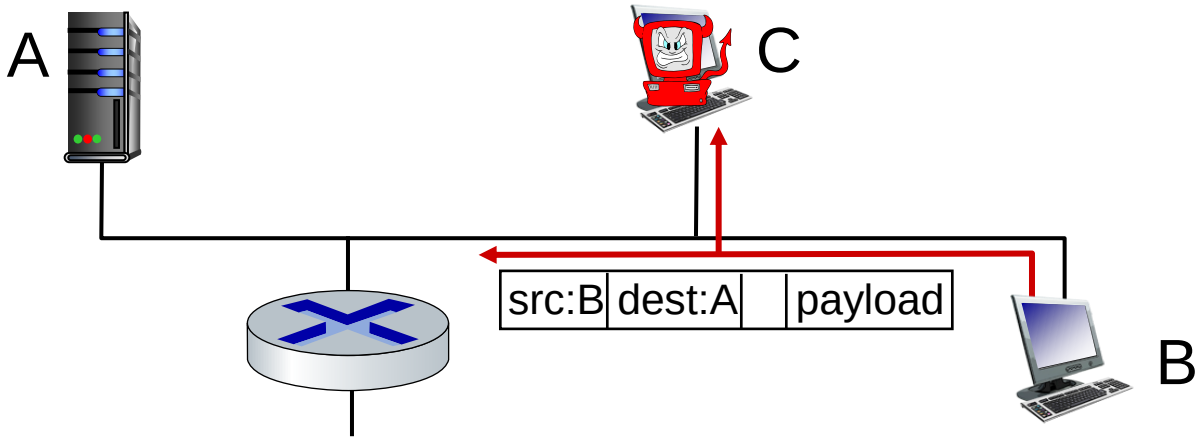
Network security

- 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":

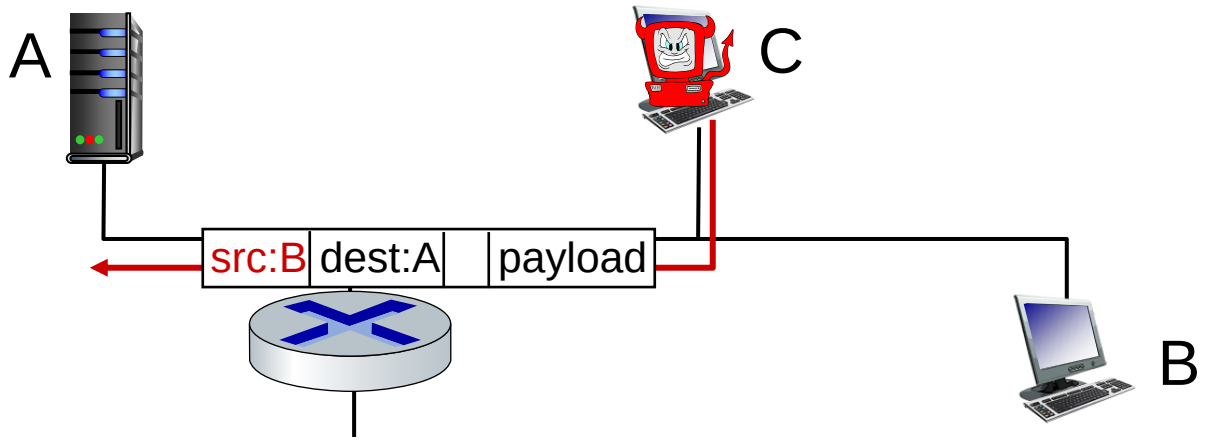
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



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

Bad guys: fake identity

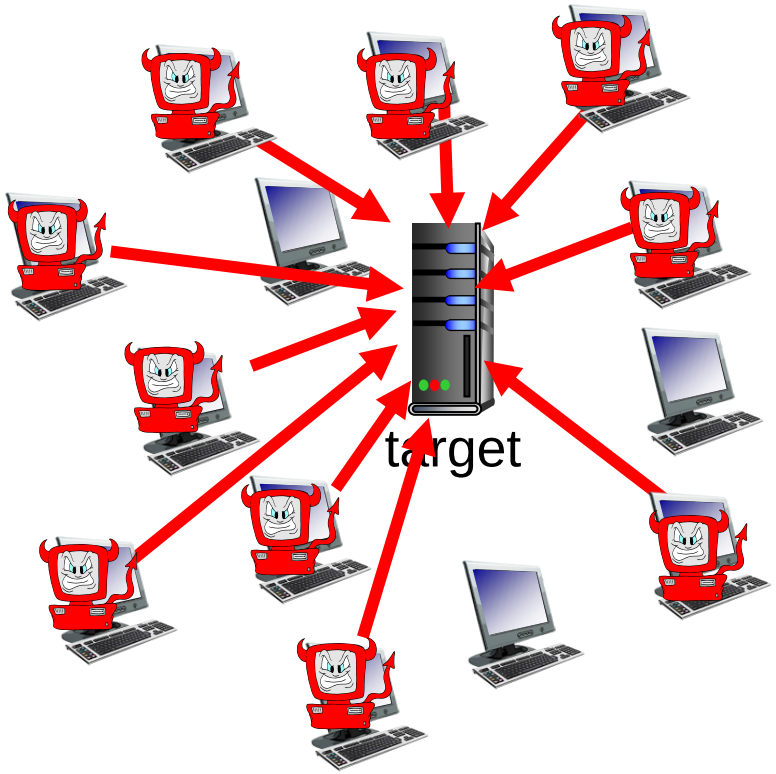
IP spoofing: injection of packet with false source address



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



Chapter 1: roadmap

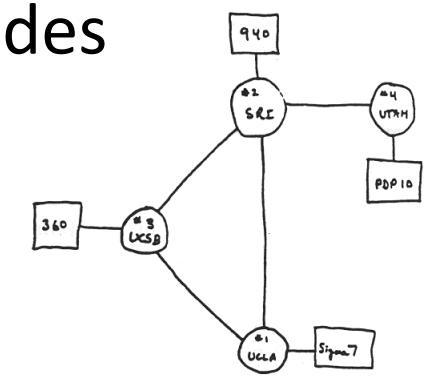
- What *is* the Internet?
- What *is* a protocol?
- Network edge: hosts, access network, physical media
- Network core: packet/circuit switching, internet structure
- Performance: loss, delay, throughput
- Security
- Protocol layers, service models
- History

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



THE ARPA NETWORK

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

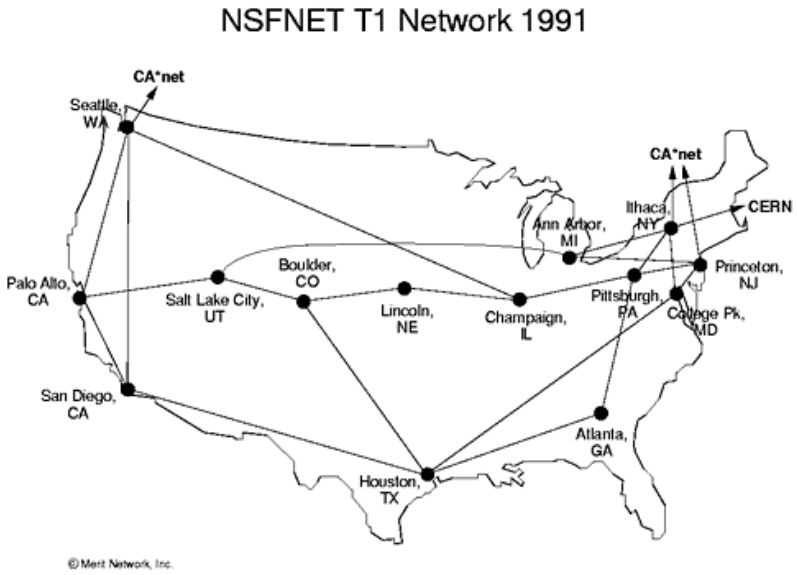
define today's Internet architecture

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)
- ~18B devices attached to Internet (2017)

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:

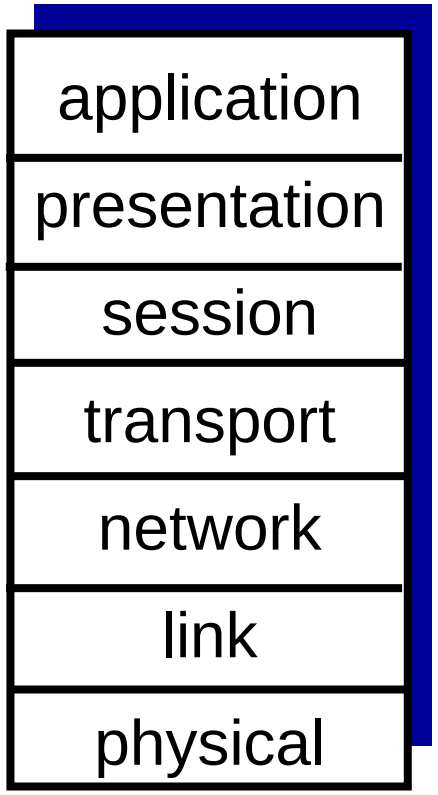
- context, overview, vocabulary, "feel" of networking
- more depth, detail, *and fun* to follow*!*

Additional Chapter 1 slides

ISO/OSI reference model

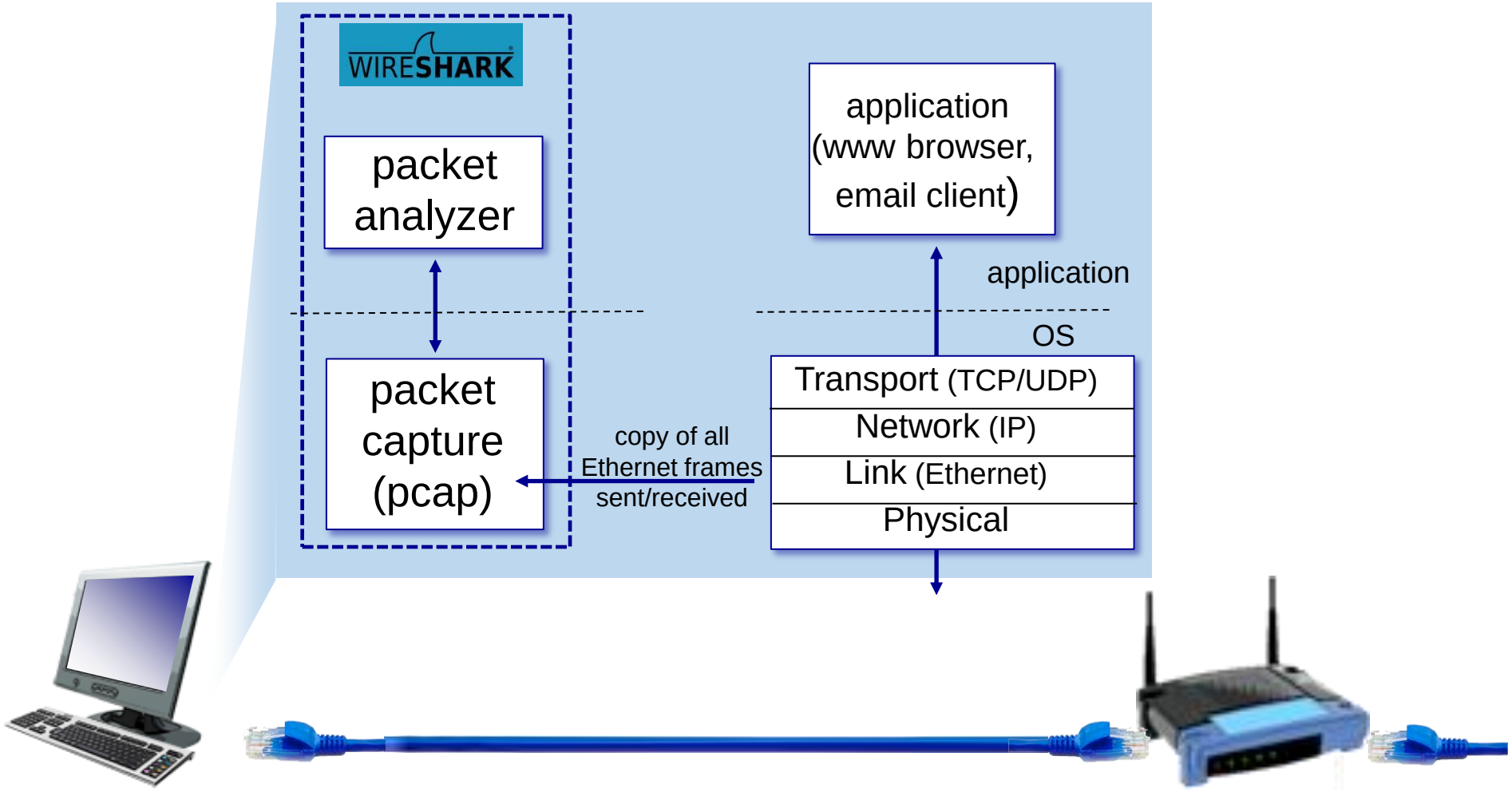
Two layers not found in Internet protocol stack!

- *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

Wireshark


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