Chapter 3 Introduction

Chapter 1: introduction

our goal:

- get "feel" and terminology
- more depth, detail *later* in course
- approach:
  - use Internet as example

overview:

- what's the Internet?
- what's a protocol?
- network edge; hosts, access net, physical media
- network core: packet/circuit switching, Internet structure
- performance: loss, delay, throughput
- protocol layers, service models

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

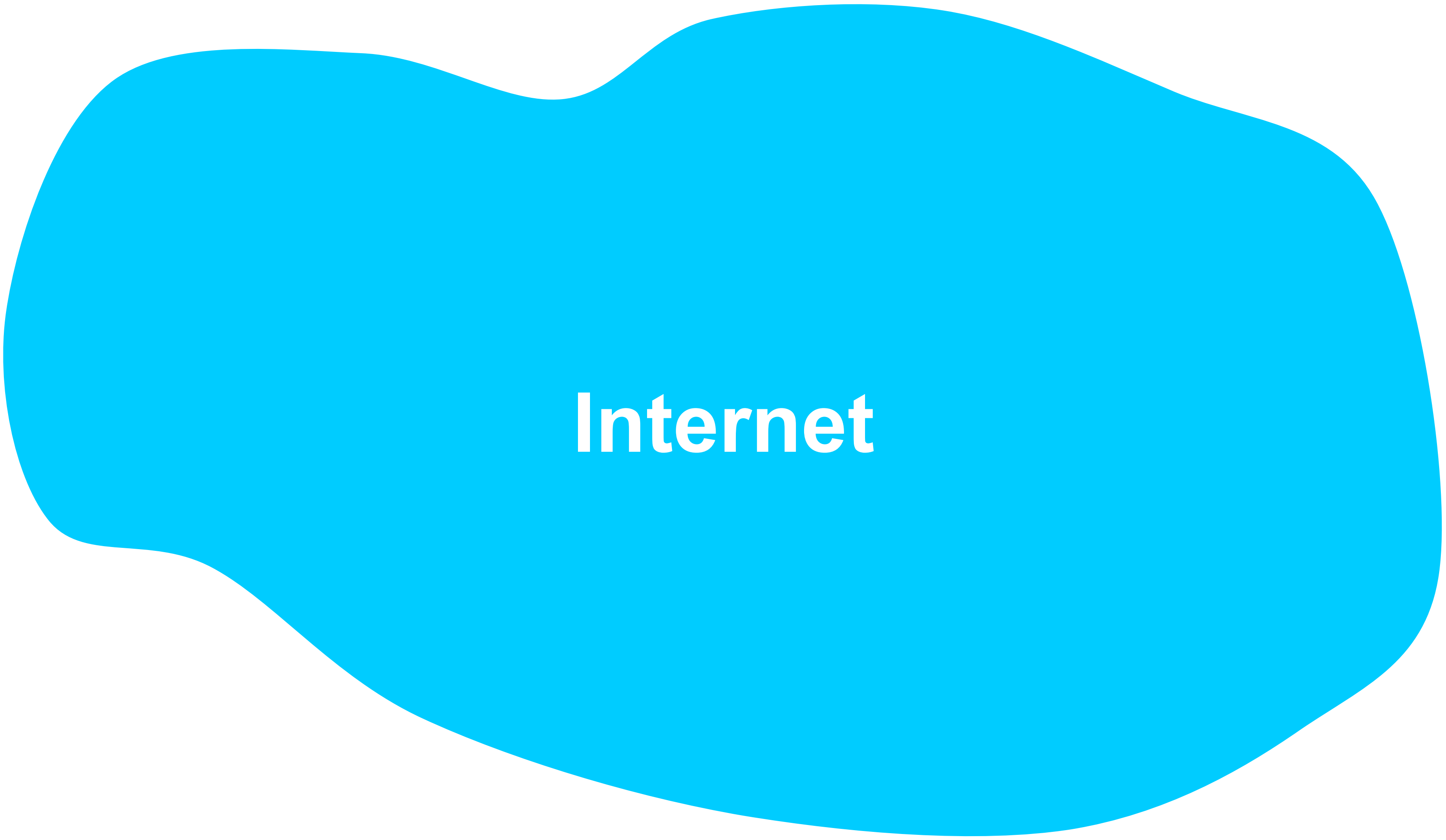
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

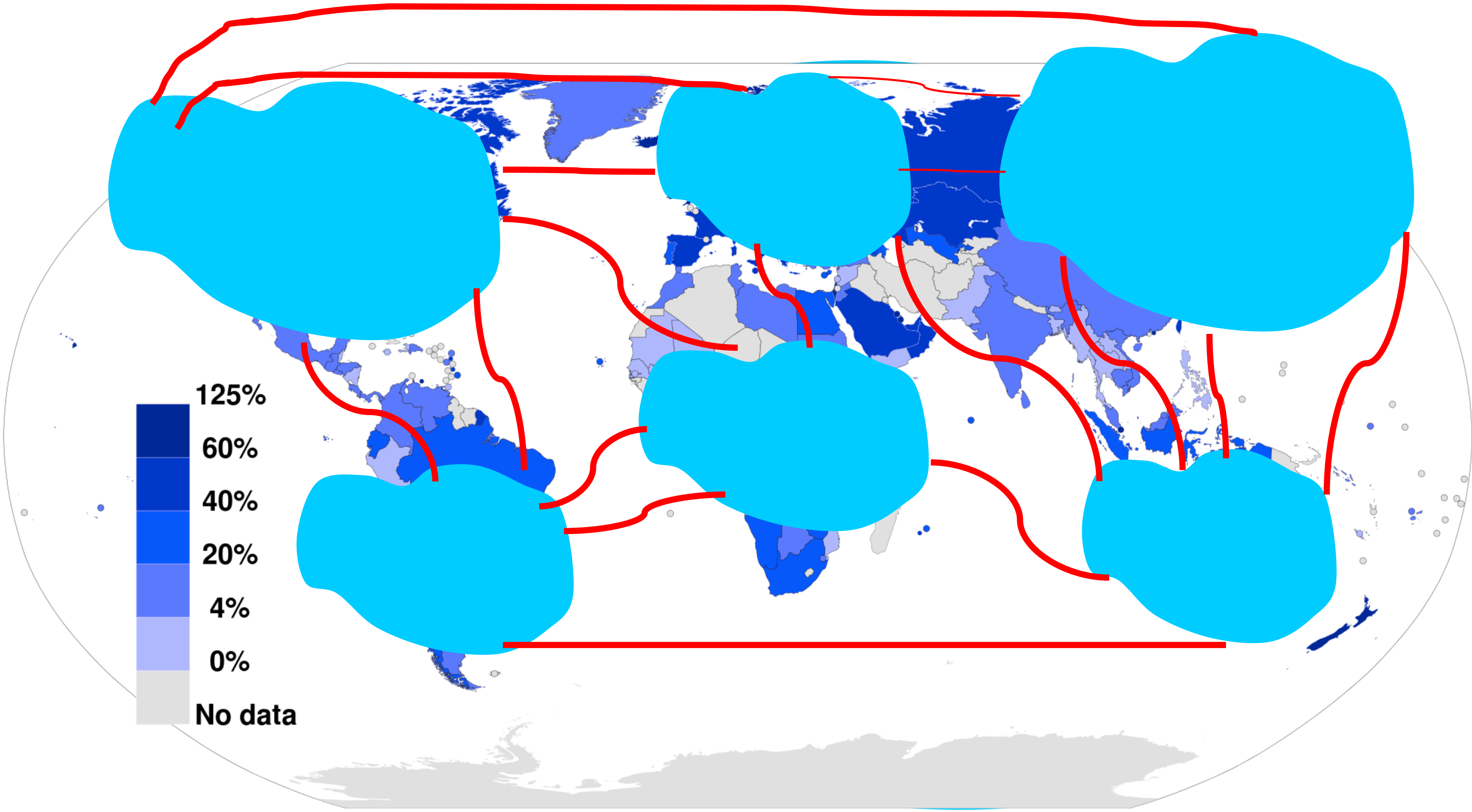
~~1.6 networks under attack: security~~

~~1.7 history~~

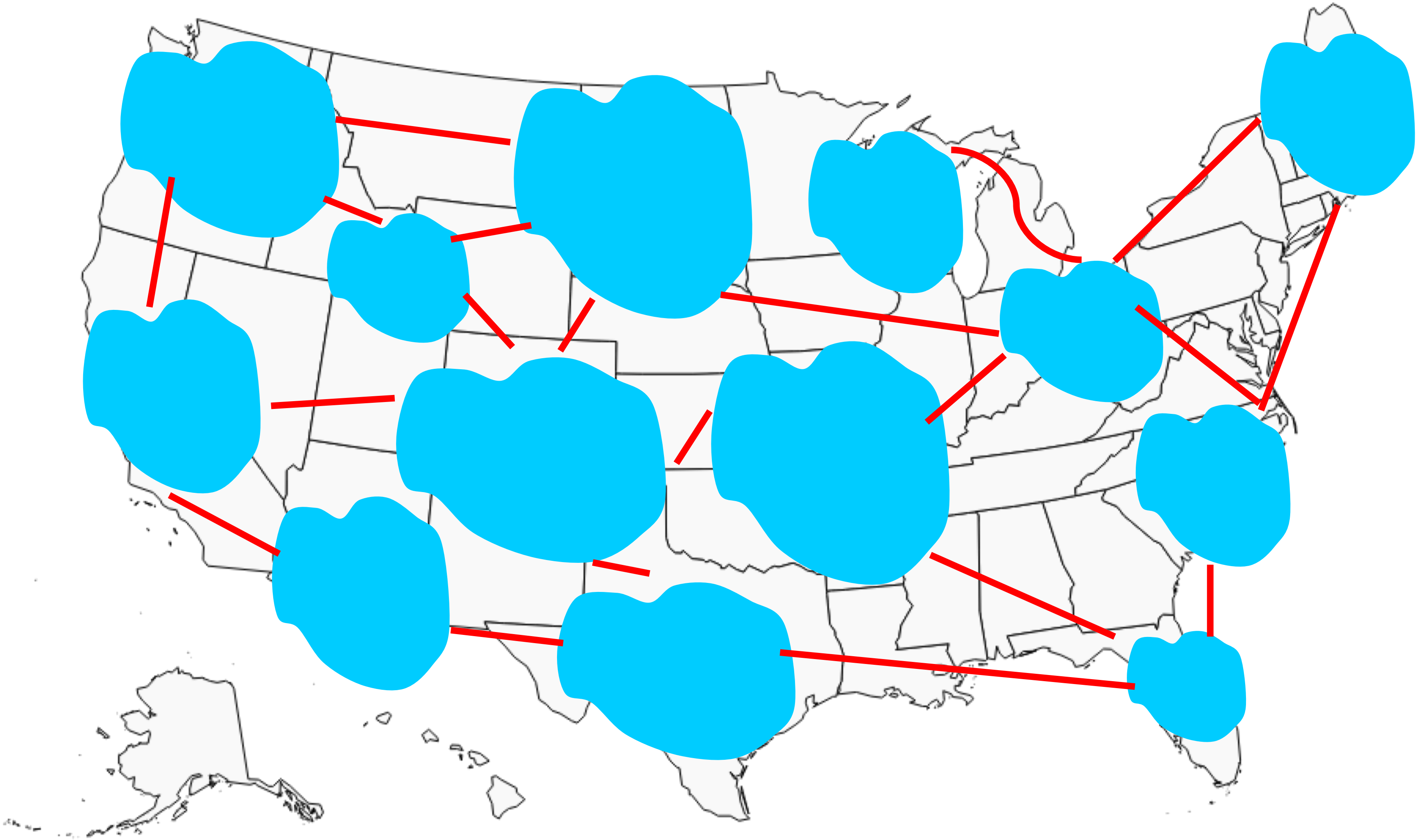
1.1 What is the Internet?



1.1 What is the Internet?

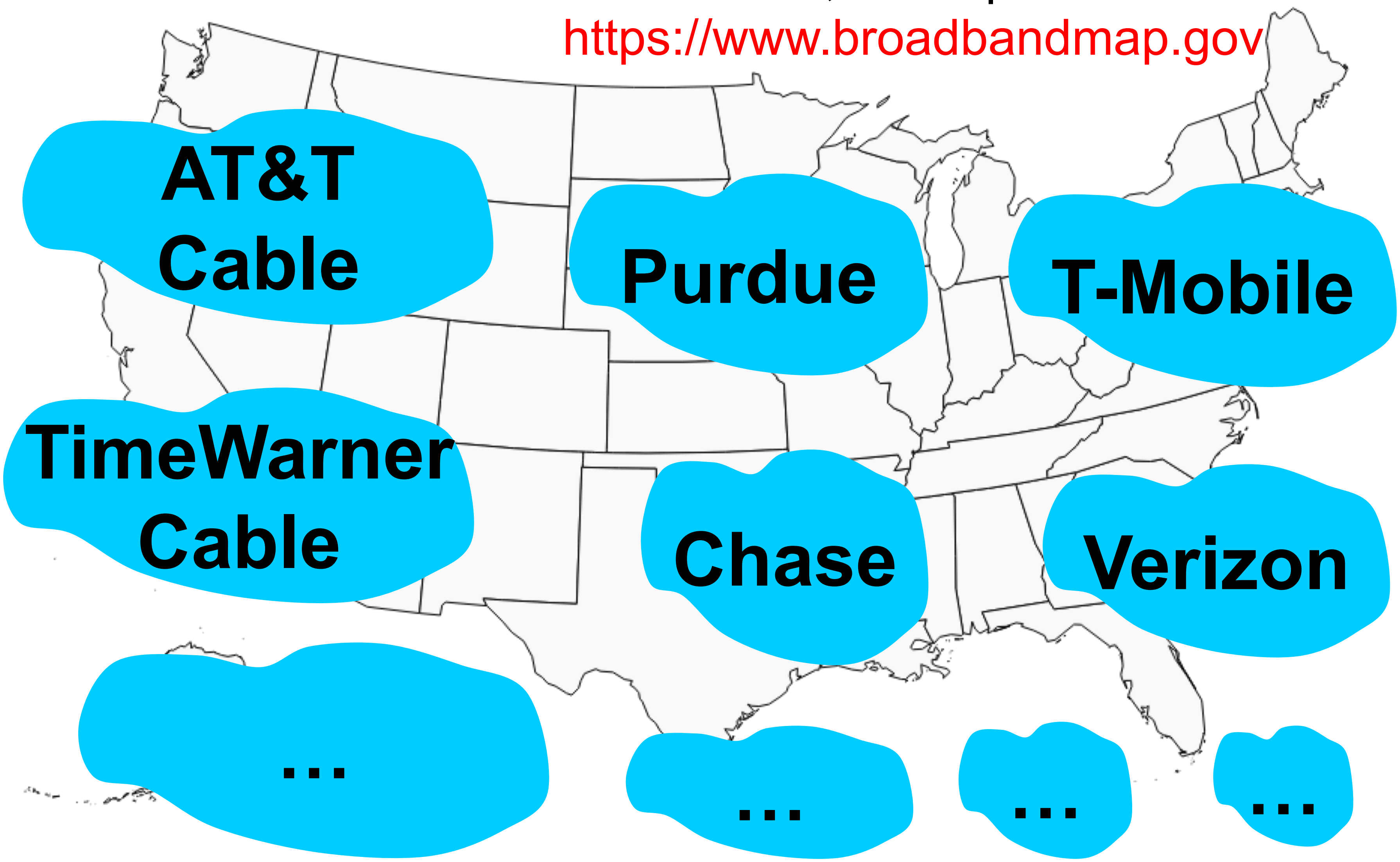


1.1 What is the Internet?



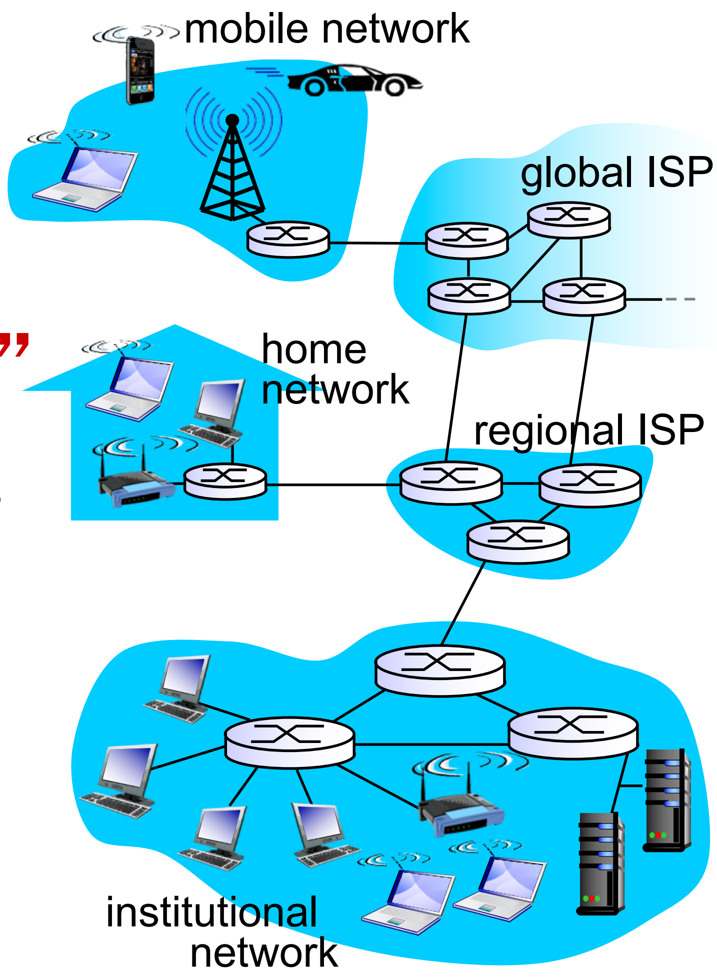
1.1 What is the Internet?

If interested, look up our networks at
https://www.broadbandmap.gov

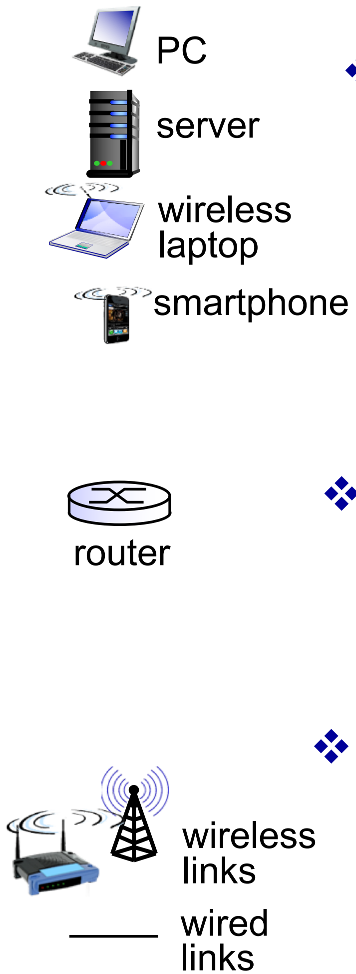


What's the Internet: "nuts and bolts" view

- *Internet* =
  "network of networks"
  - **Interconnected** networks



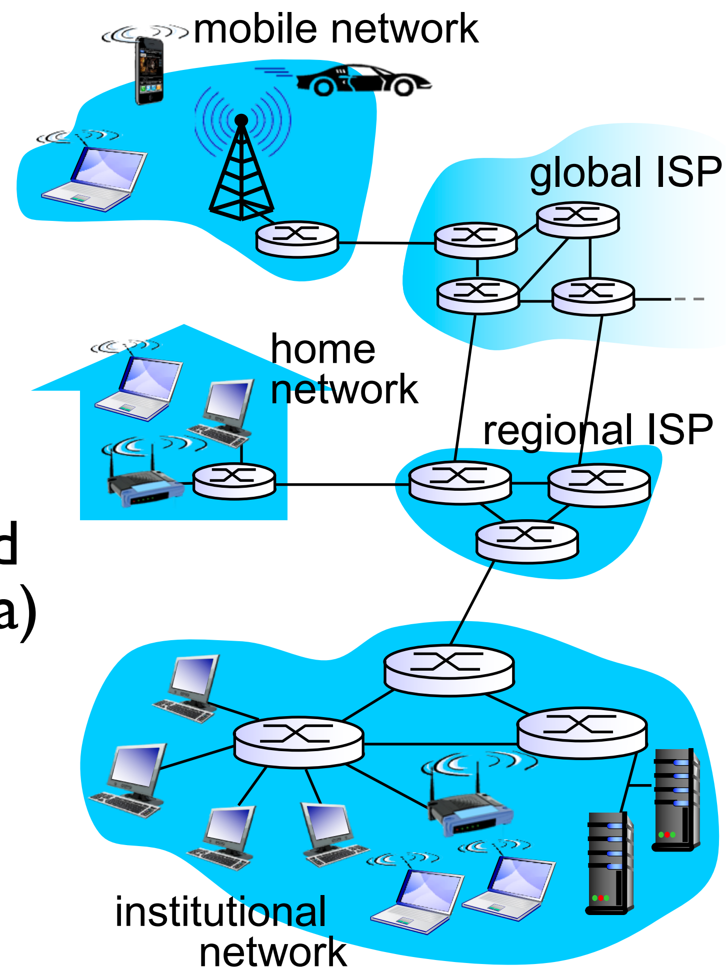
What's the Internet: "nuts and bolts" view



- *Hosts* = *end systems*
  - billions of connected computing devices
  - running *network apps*

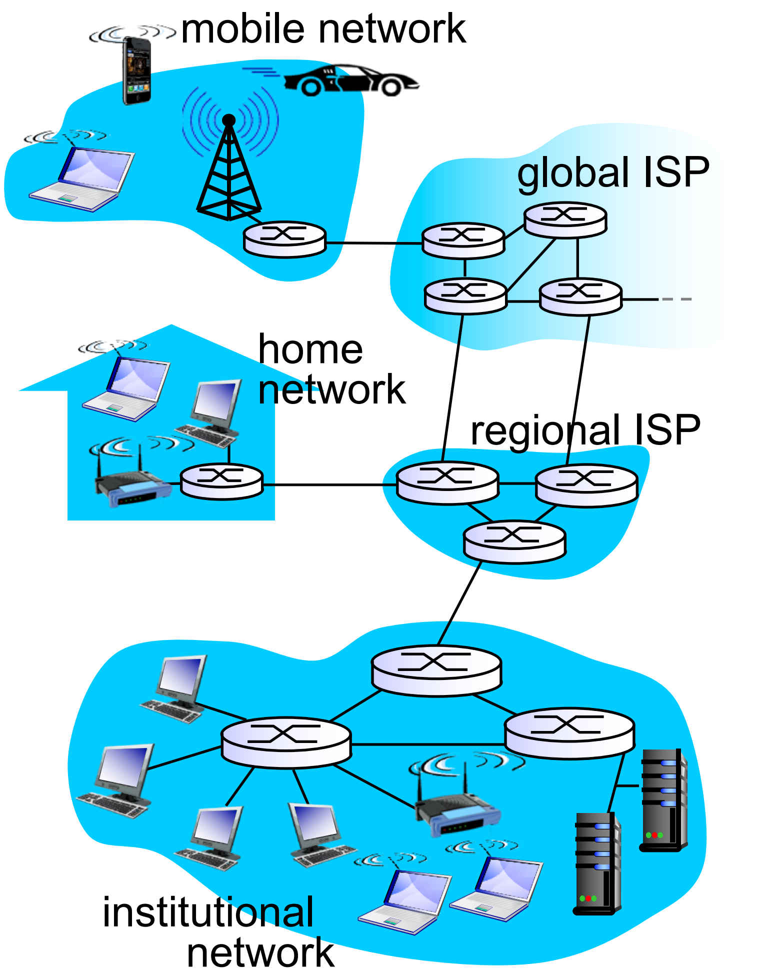
- *Routers and switches*
  - *Packet switches:* forward packets (chunks of data)

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



A closer look at network structure:

- *network edge:*
  - hosts: clients and servers
  - servers often in data centers
- *access networks, physical media:* wired, wireless communication links
- *network core:*
  - interconnected routers
  - network of networks



Access networks and physical media

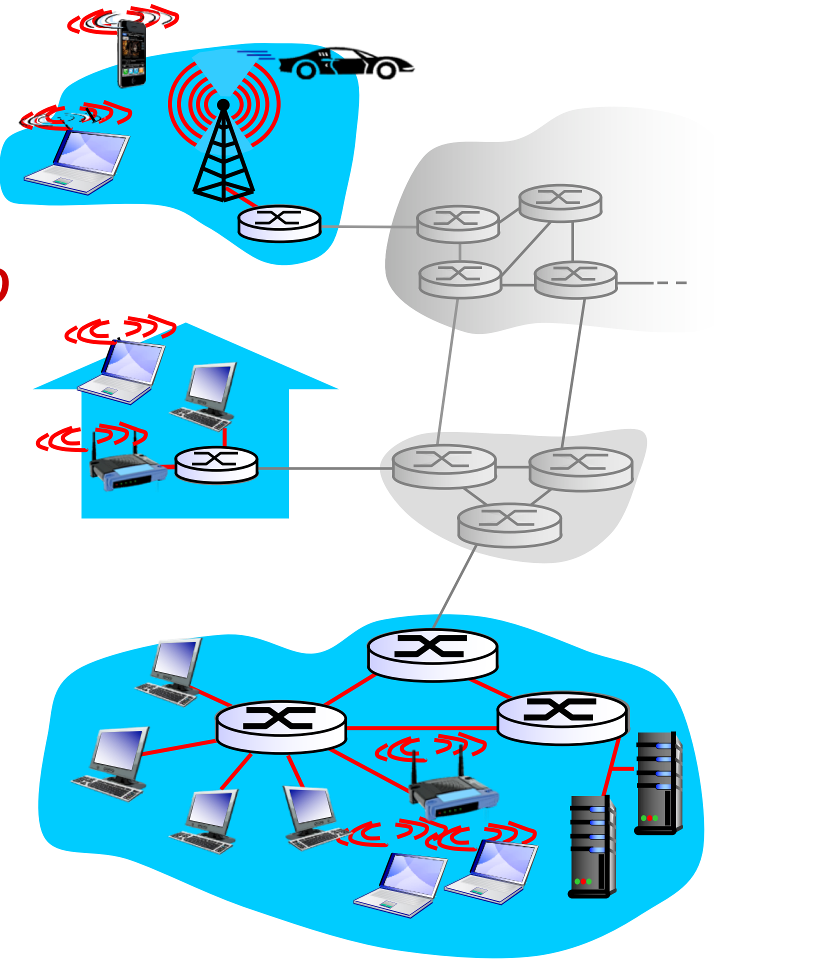
Access network: connect end systems (hosts) to edge routers

Q: How to connect end systems to edge router?

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

keep in mind:

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



Common Access Networks

- Cable network
- Digital subscriber line (DSL)
- Home WiFi Network
- Cellular (mobile) network
- Ethernet
- Fiber (optics network)
- …

- Read Chapter 1.2 for details

Wireless access networks

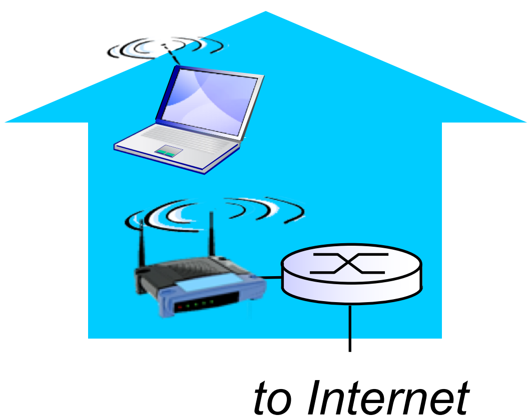
- shared *wireless* access network connects end system to router
  - via base station aka "access point"

wireless LANs:

- within building (100 ft)
- 802.11a/b/g (WiFi): 11, 54 Mbps transmission rate
- 802 11n/ac: hundreds of Mbps

wide-area wireless access

- provided by telco (cellular) operator, 10's km
- between 1 and 10 Mbps
- 3G, 4G LTE
  - LTE: ideally 100-300 Mbps



Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

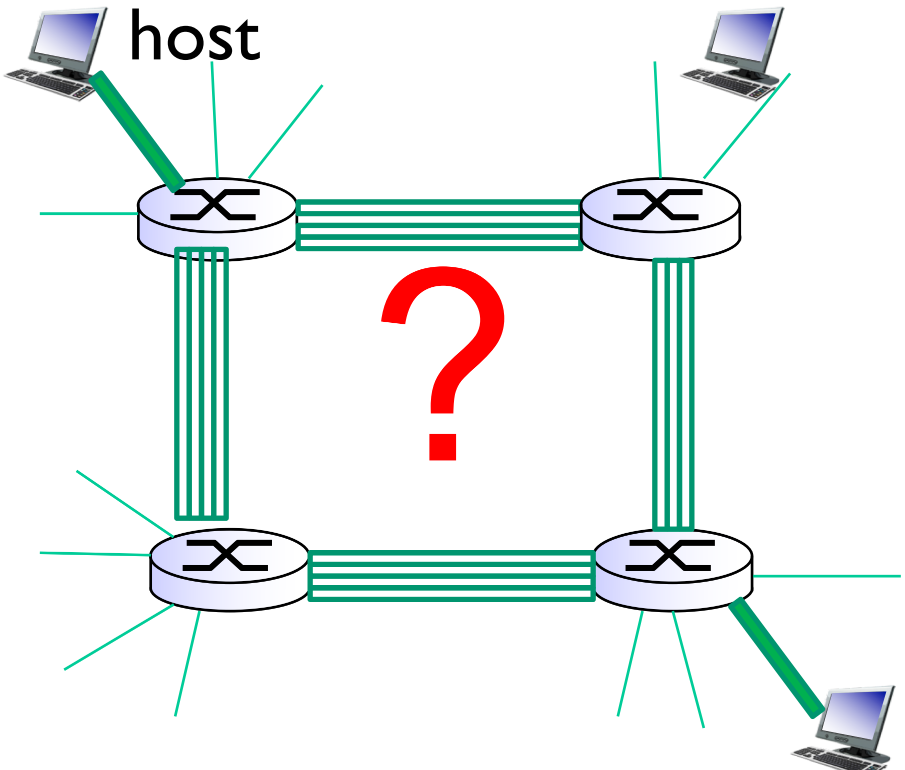
1.5 protocol layers, service models

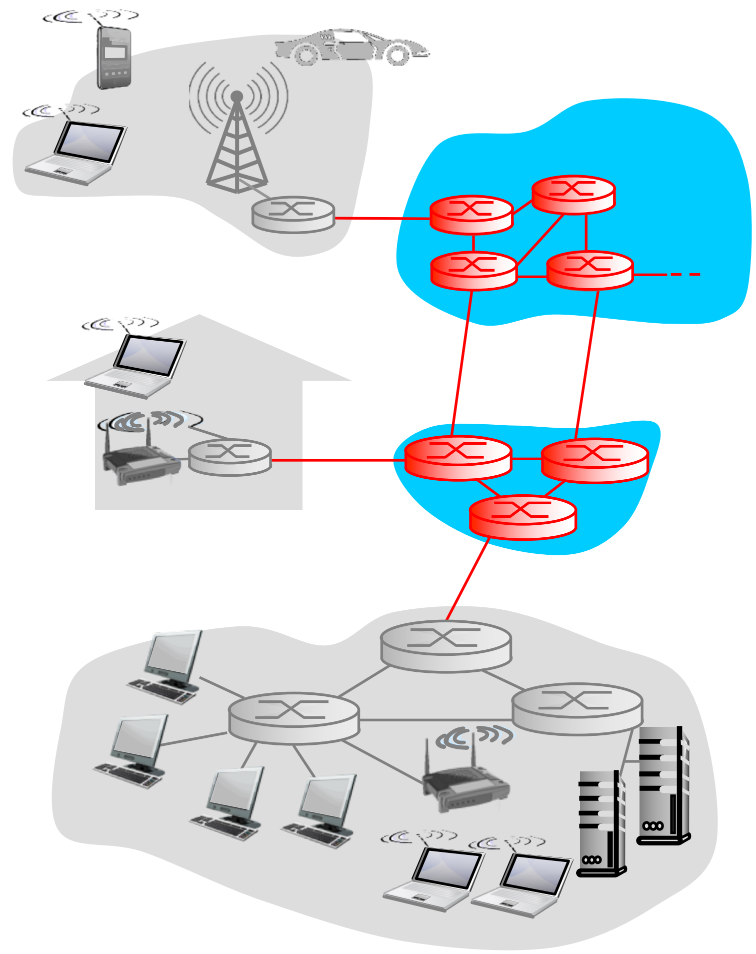
~~1.6 networks under attack: security~~

~~1.7 history~~

Discussion: How to transfer data?

- mesh of interconnected routers
- Role: send chunks of data from one host to another host



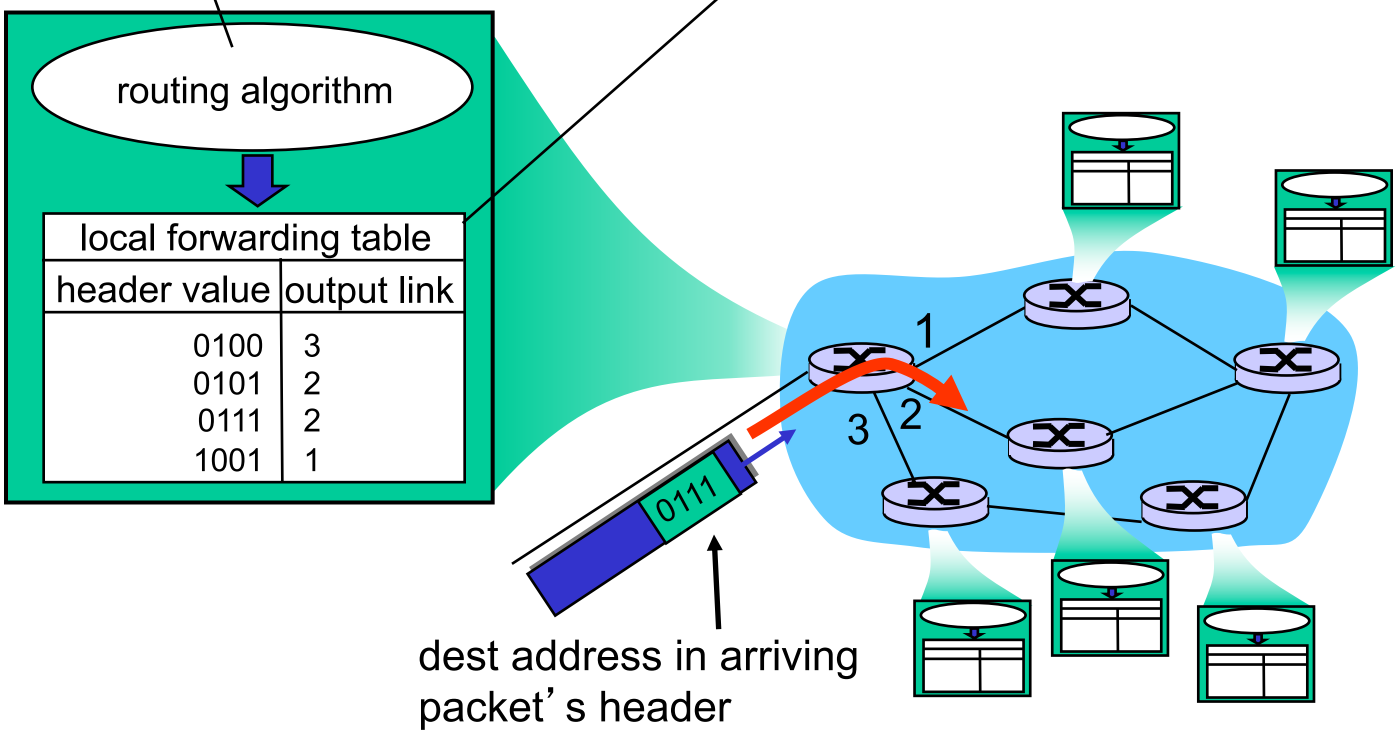


Two key network-core functions

routing: determines source-destination route taken by packets

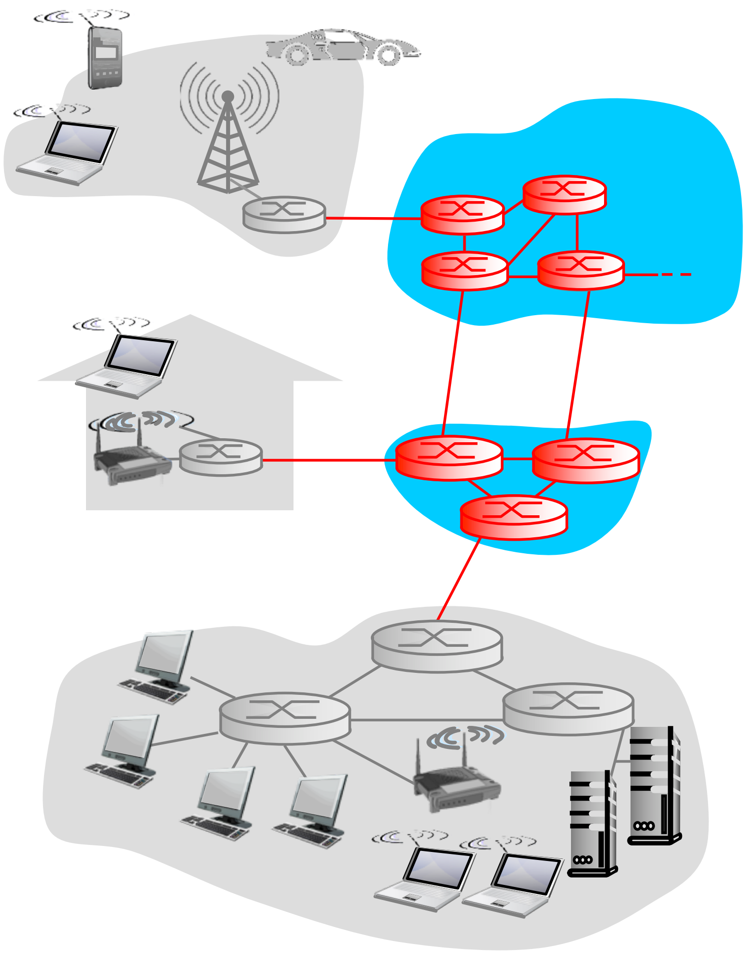
- *routing algorithms*

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



Network core: Packet-switching

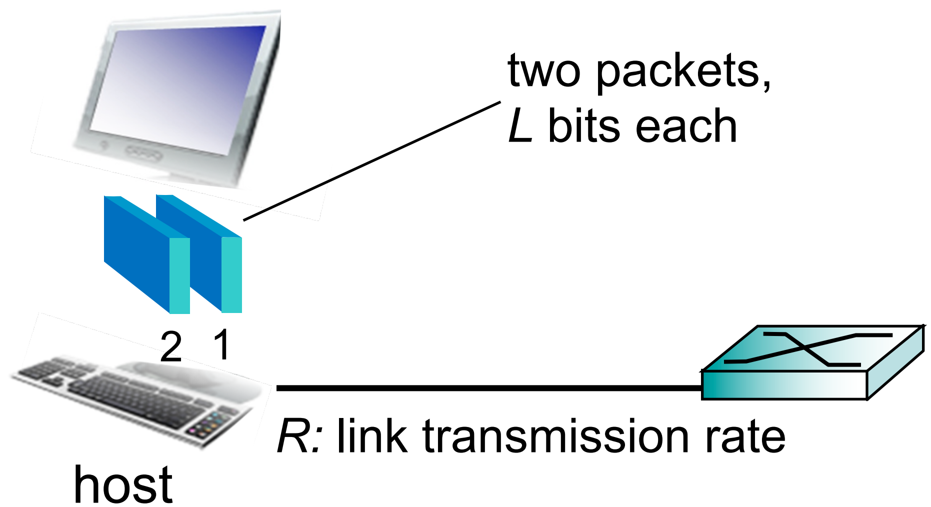
- packet-switching: hosts break application-layer messages into *packets*
  - forward packets from one router to the next, across links on path from source to destination
  - each packet transmitted at full link capacity



Host: sends *packets* of data

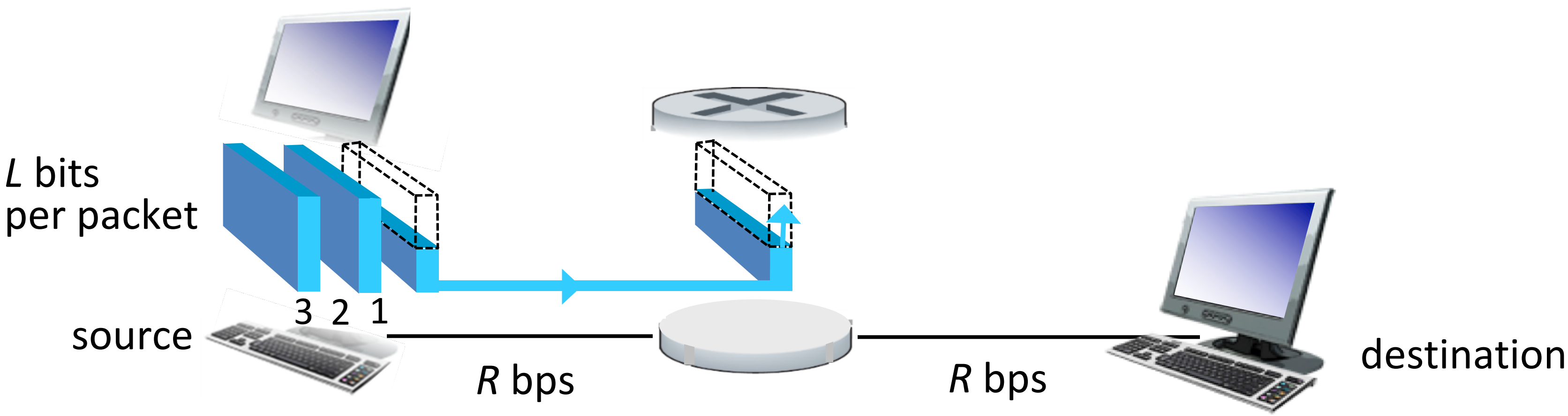
host sending function:

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



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

Packet-switching: store-and-forward



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

more on delay shortly …

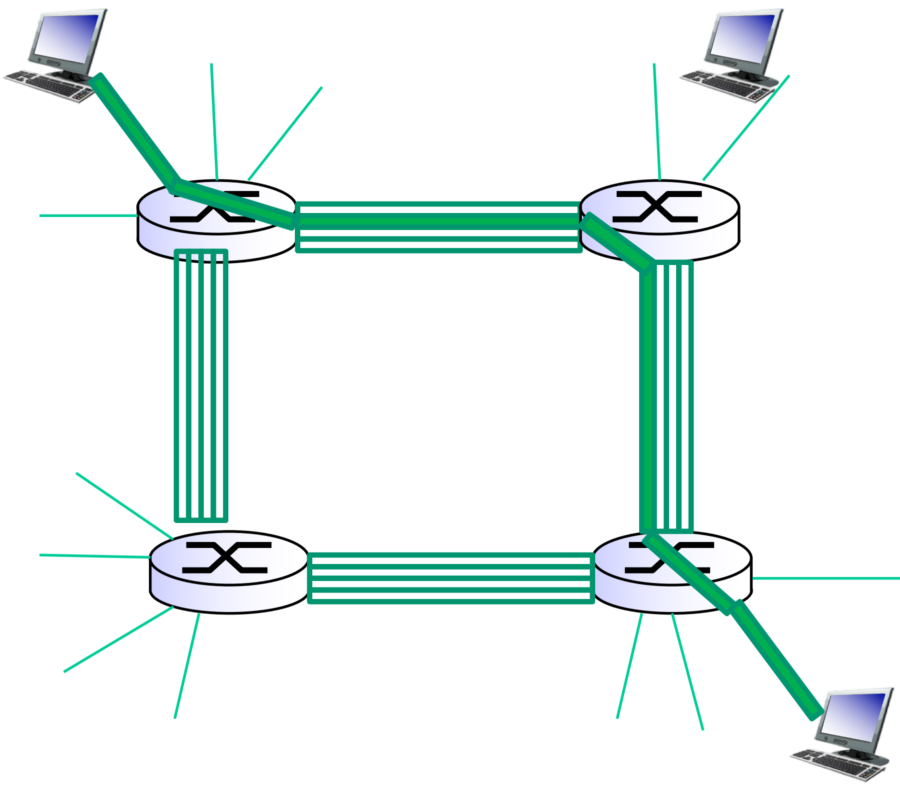
one-hop numerical example:

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

Alternative core: circuit switching

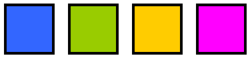
end-end resources allocated to, reserved for "call" between source & dest:

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

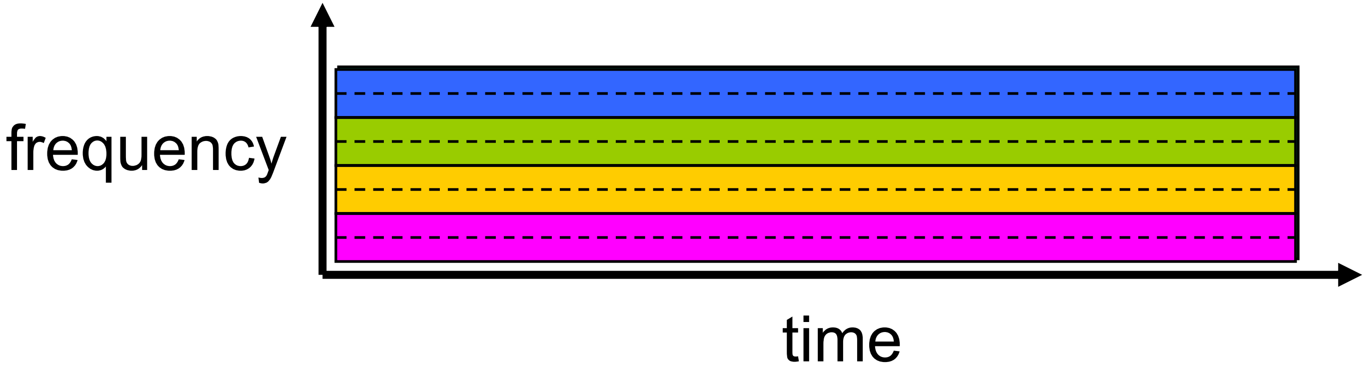


Circuit switching: FDM versus TDM

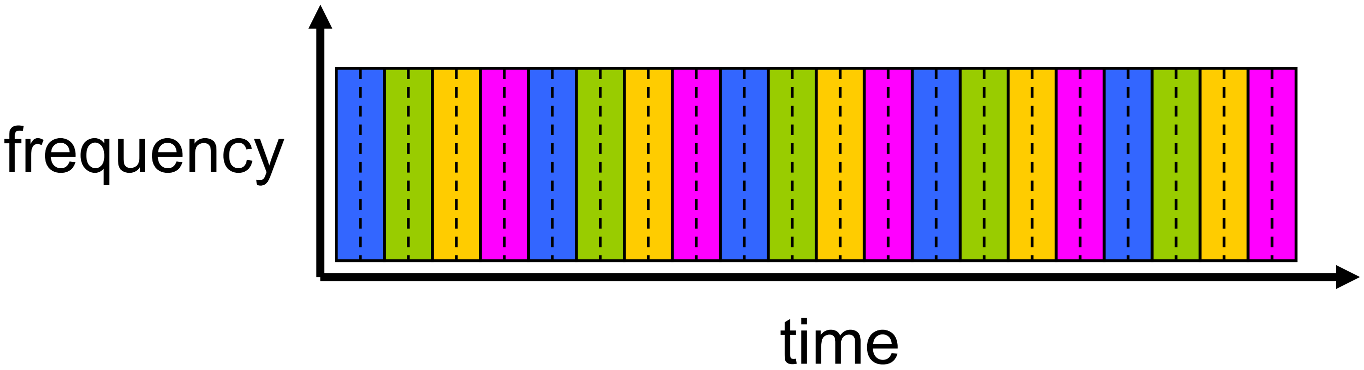
Example:

4 users 

FDM



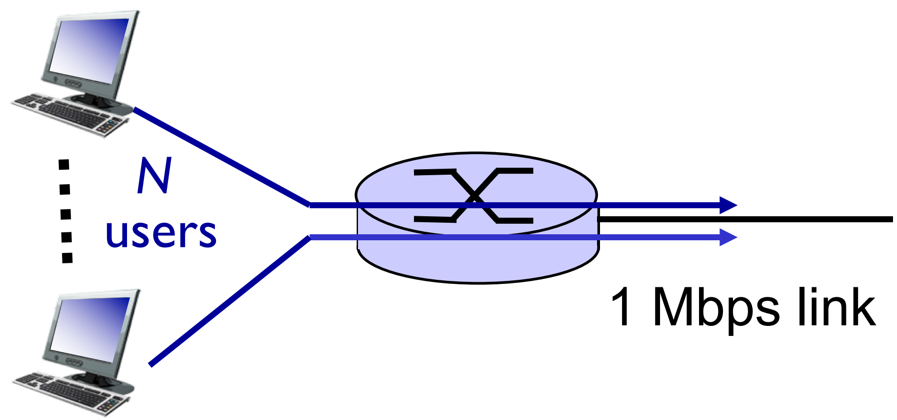
TDM



Packet switching versus circuit switching

packet switching allows more users to use network!

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



- *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?

Q: what happens if > 35 users ?

* Check out the online interactive exercises for more examples

More on “how to calculate N?”

- Given N users, the probability that x users are active is P(N,x)

$$P(N,x) = \binom{N}{x} p^x (1-p)^{N-x}$$

- In order to afford N users, the probability that more than 11 (including 11) users are active at the same should be extremely small (e.g., < 0.1%)
- Therefore,

$$\sum_{x=0}^{10} P(N,x) \geq (1-0.1\%) = 0.999$$

More on "how to calculate N?"

- Given N, we can obtain the probability of no more than x active users at the same time
  - Calculator
  - Program (sum them up)

- Given the maximal probability of no more than x active users at the same time, how to calculate N?
  - Approach #1: program (for loop, and increment N by 1 until it meets the threshold)

  - Approach #2: Approximation via central limit theorem (optional)

More on "how to calculate N?"

- Step 0: Xi: random variable, status of user i
  - Xi = 1: user i is active
  - Xi = 0: user i is not active

$$P(X_i = 1) = p = 0.1, P(X_i = 0) = 0.9$$

$$\mu = E(X_i) = 1 * P(X_i = 1) + 0 * P(X_i = 0) = p$$

$$\sigma^2 = Var(X_i) = E[(X_i - E(X_i))^2]$$

$$= p(1-p)^2 + (1-p)p^2 = p(1-p)$$

- Step 1: Z = sum(Xi), another random variable
  - The number of active users, given N users in total

Step II: Z conforms to normal distribution

- Xi is iid (Independent and identically distributed)
- Central limit theorem

$$\mu_z = E[Z] = Np$$

$$\sigma_z^{\ 2} = Var(Z) = Np(1-p)$$

- Z conforms to normal distribution $(Np, \sqrt{Np(1-p)})$

Step III: look up the cumulative distribution function (CDF) for a normal distribution

- Given P(Z<= 10) > 99.9%
- Let us look up the standard normal distribution, we have (for simplicity)

$$P(Z <= \mu_z + 3\sigma_z) > 99.9\%$$

- Then, we have $\mu_z + 3\sigma_z \leq 10$ and then calculate N

https://www.stat.tamu.edu/~lzhou/stat302/standardnormaltable.pdf

Packet switching versus circuit switching

Q: is packet switching a "slam dunk winner?"

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

Q: How to provide circuit-like behavior?

- bandwidth guarantees needed for audio/video apps
- still an unsolved problem (chapter 7)

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

Packet switching versus circuit switching

Q: is packet switching a "slam dunk winner?"

- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss (see the following slides)
  - protocols needed for reliable data transfer, congestion control

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

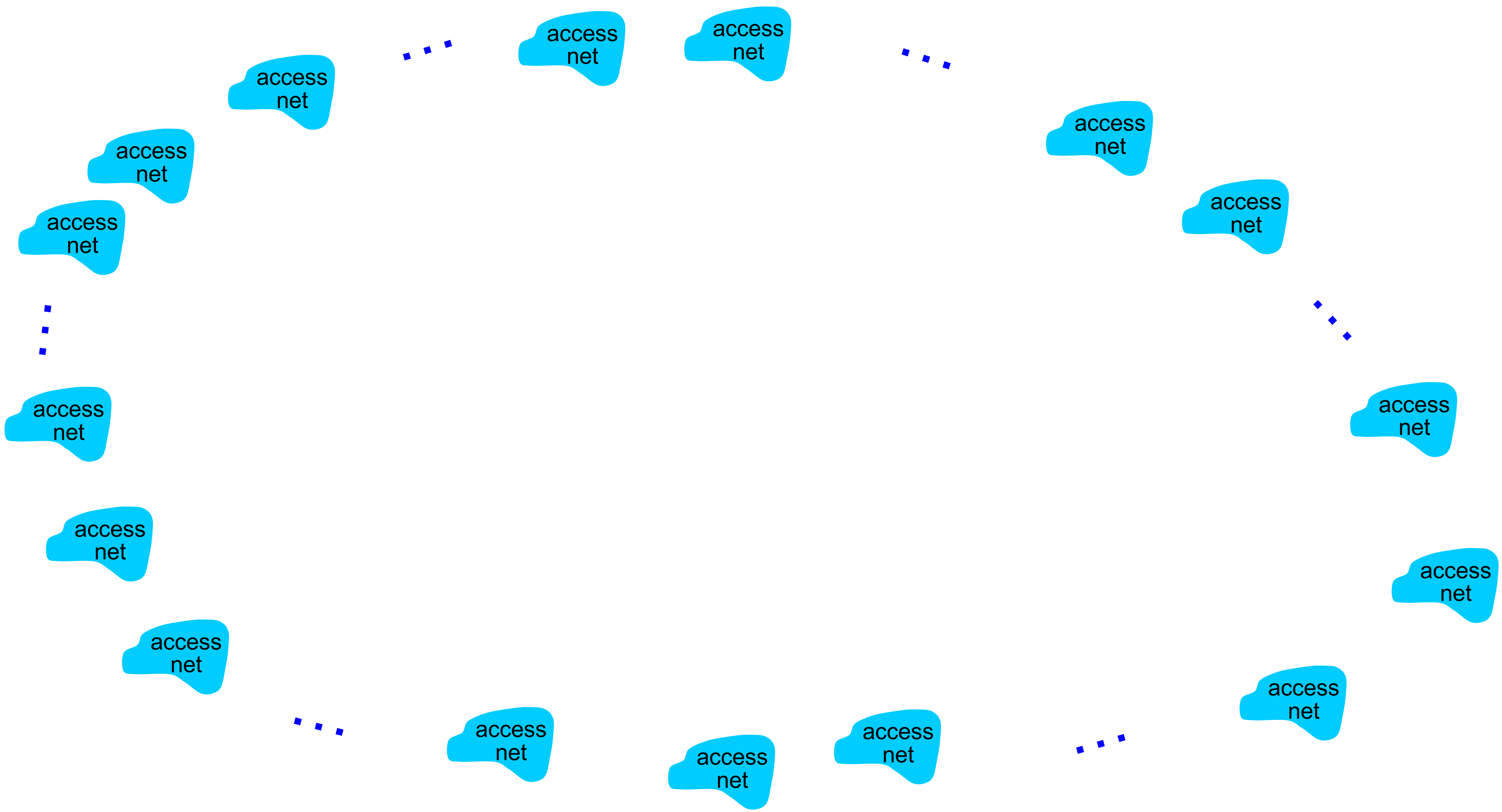
Finally,

- Let us see how to build the Internet: network of networks.

- Optional. Read Chapter 1.3.3

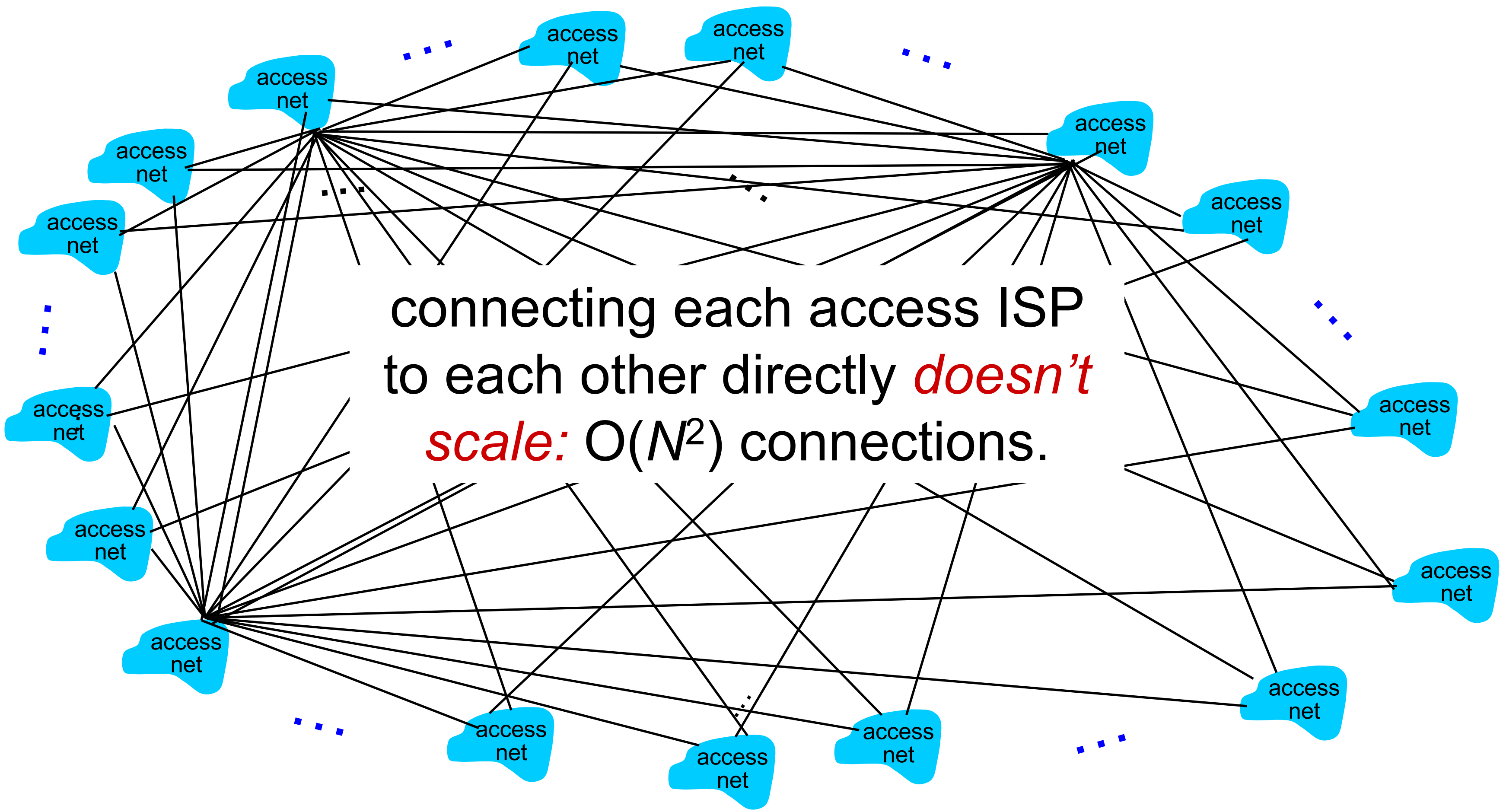
Internet structure (Optional)

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



Option 1: all inter-connected?

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

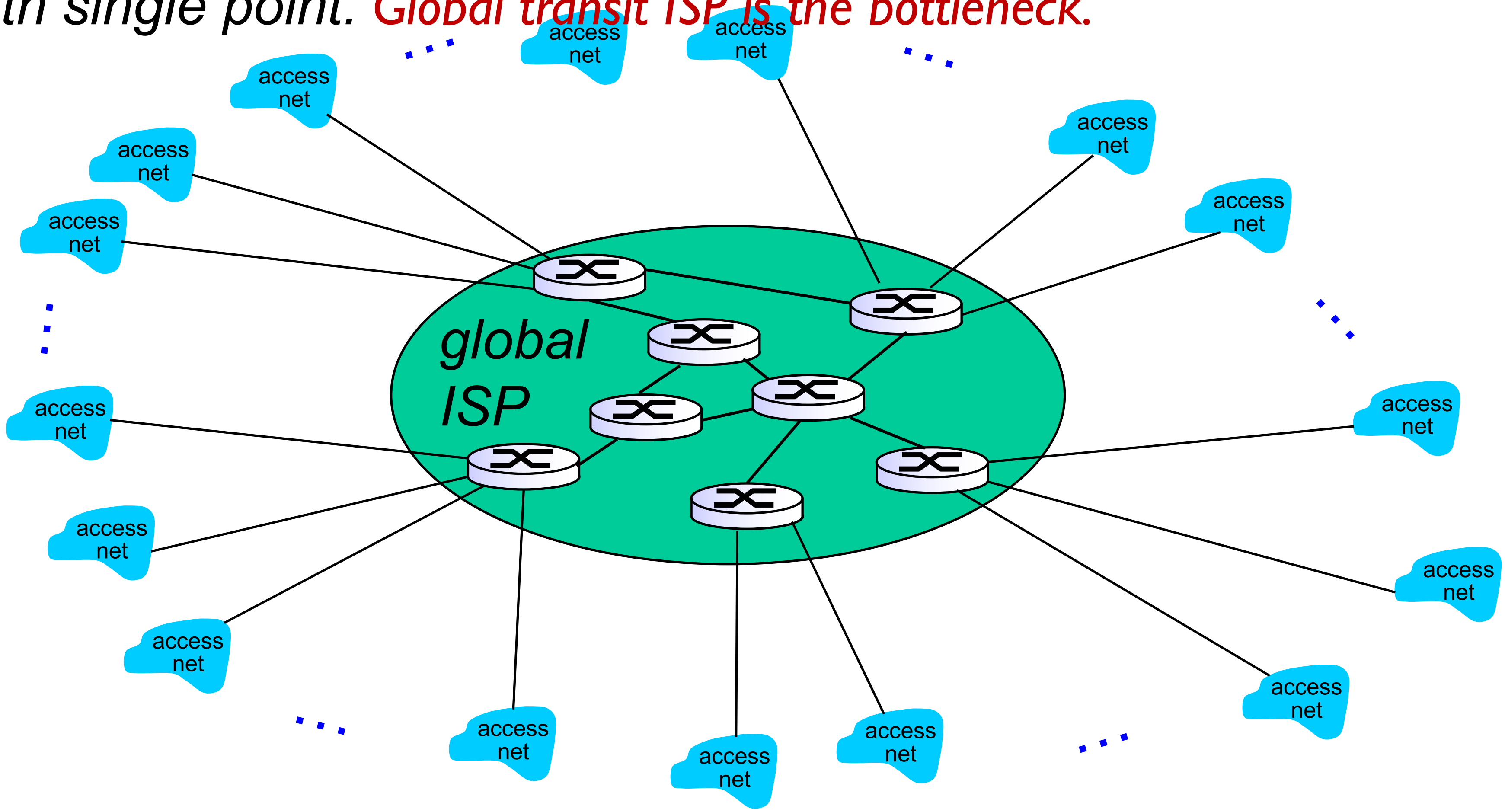


connecting each access ISP to each other directly *doesn't scale:* $O(N^2)$ connections.

Option 2: one Global ISP?

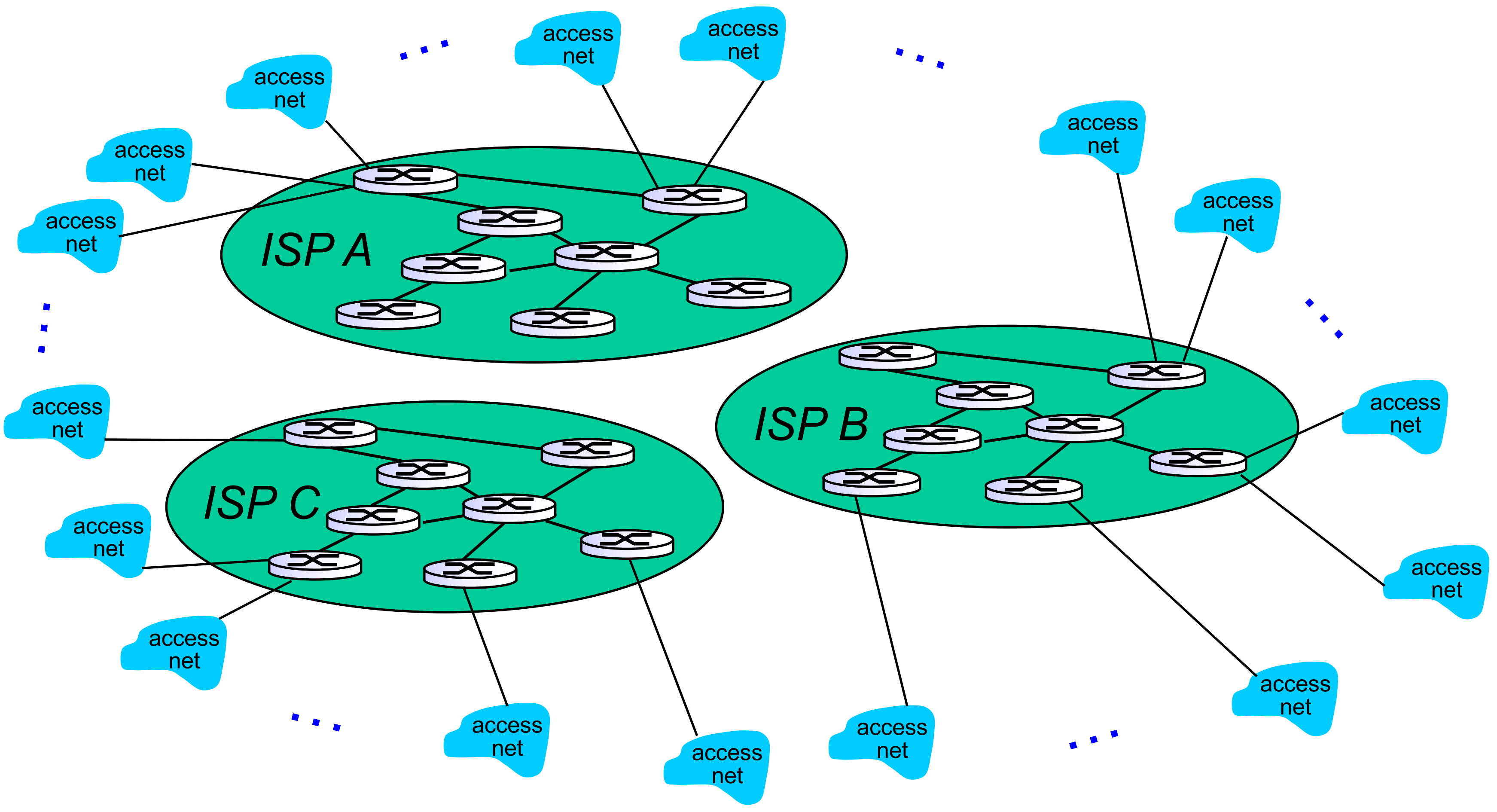
Option: connect each access ISP to a global transit ISP?

But, Customer and provider ISPs have economic agreement with single point. Global transit ISP is the bottleneck.

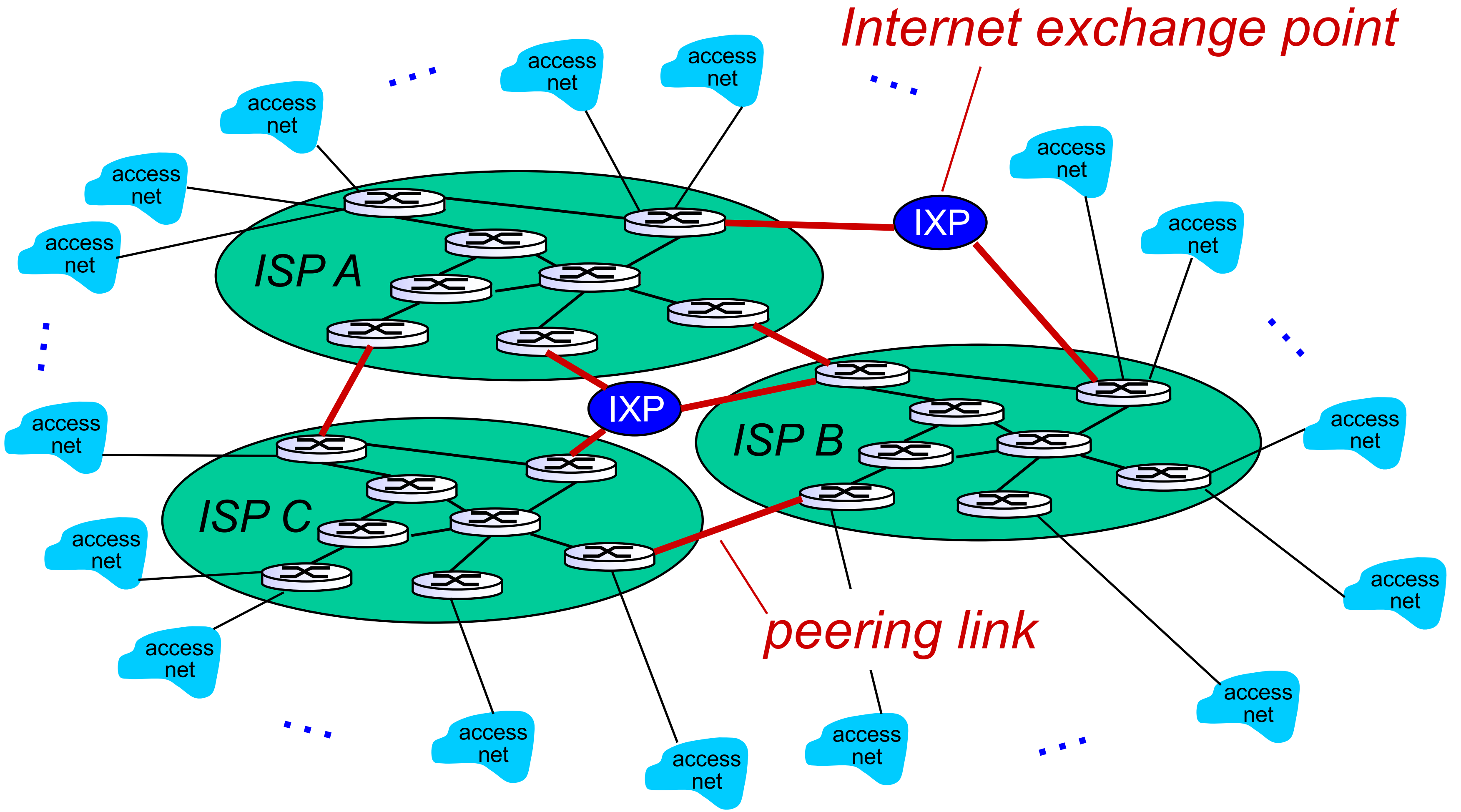


Option 2*: multiple Global ISPs?

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

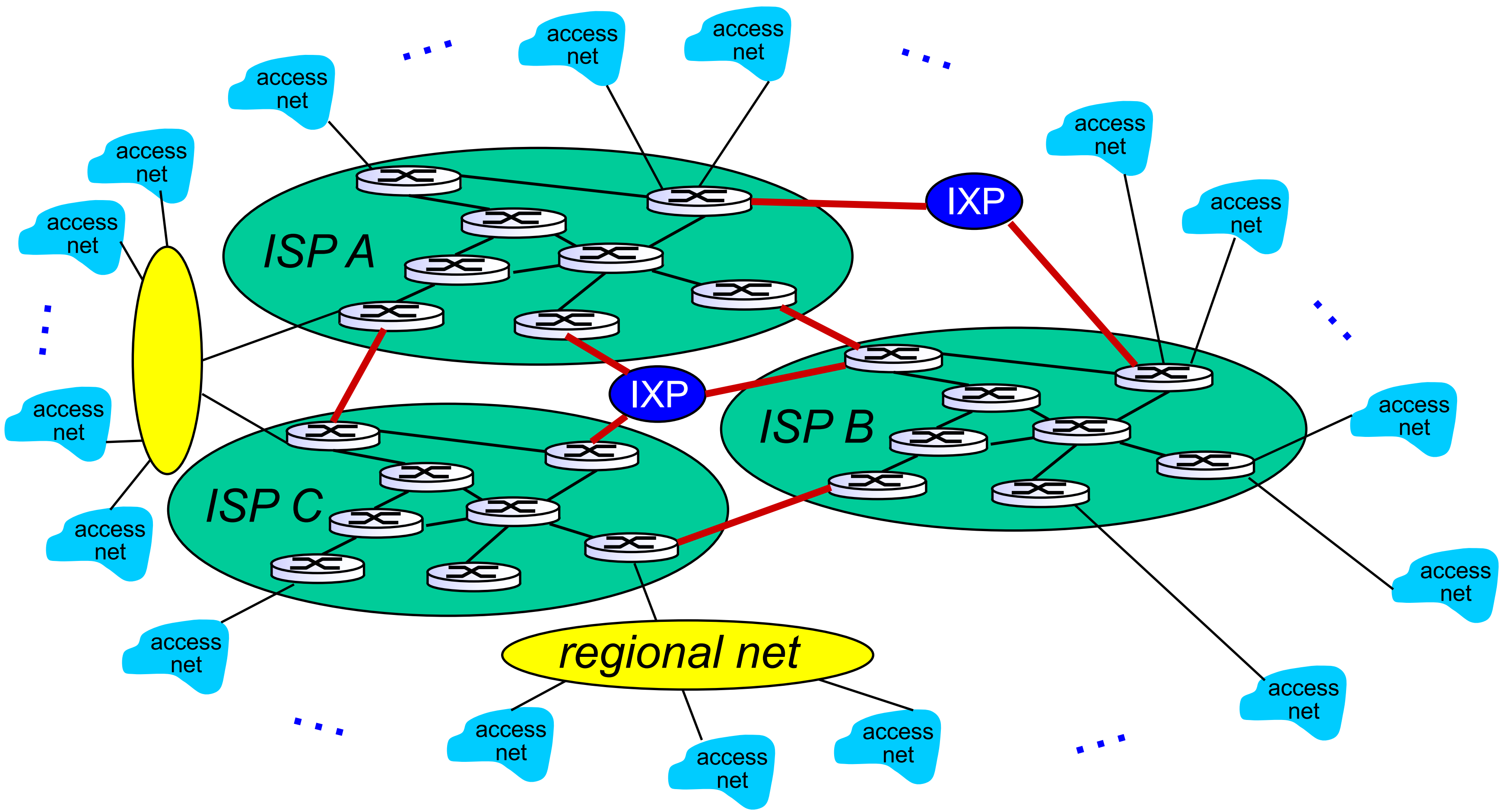


But if one global ISP is viable business, there will be competitors …. which must be interconnected



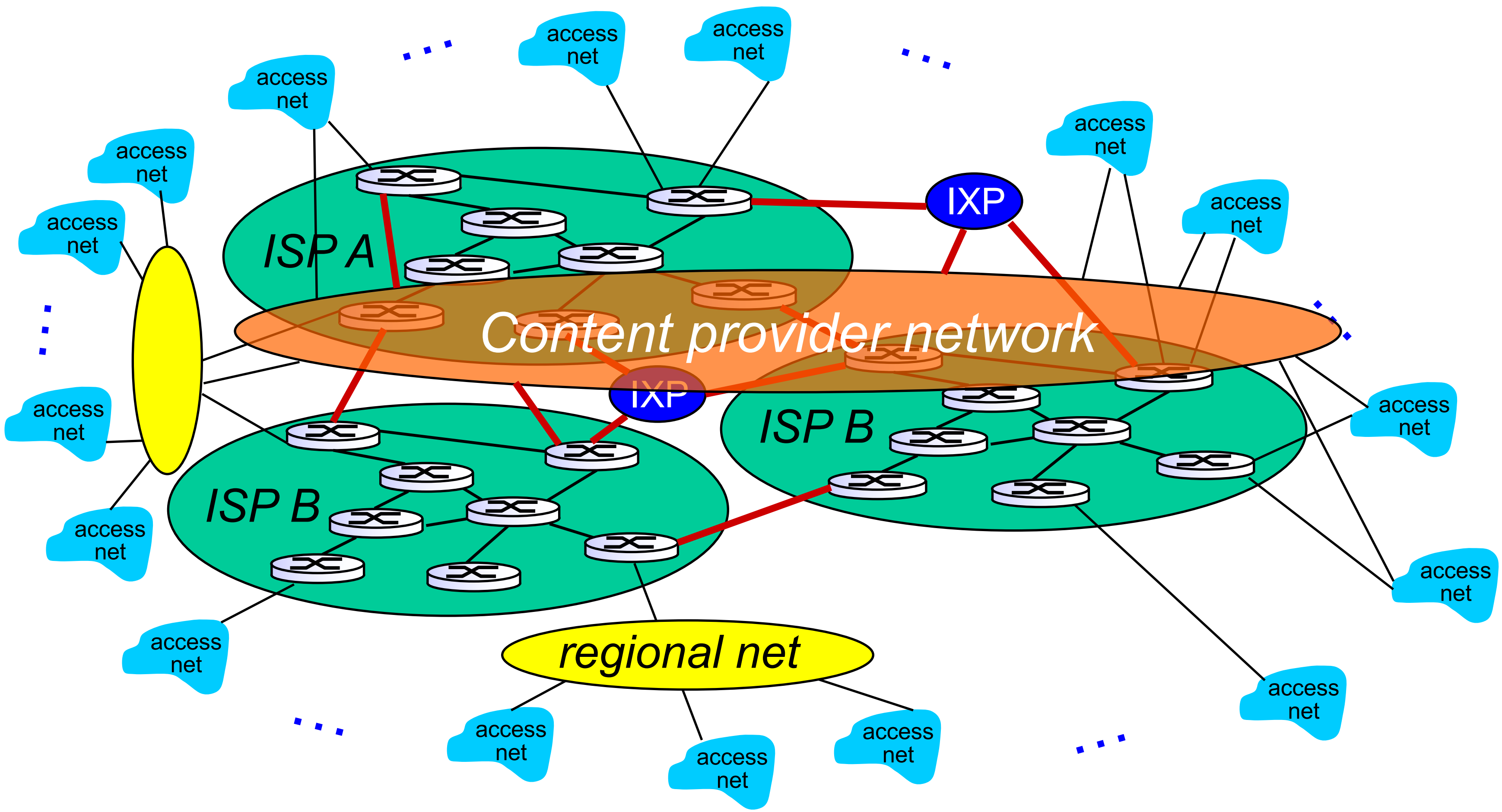
Option 3: Hierarchical structure

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

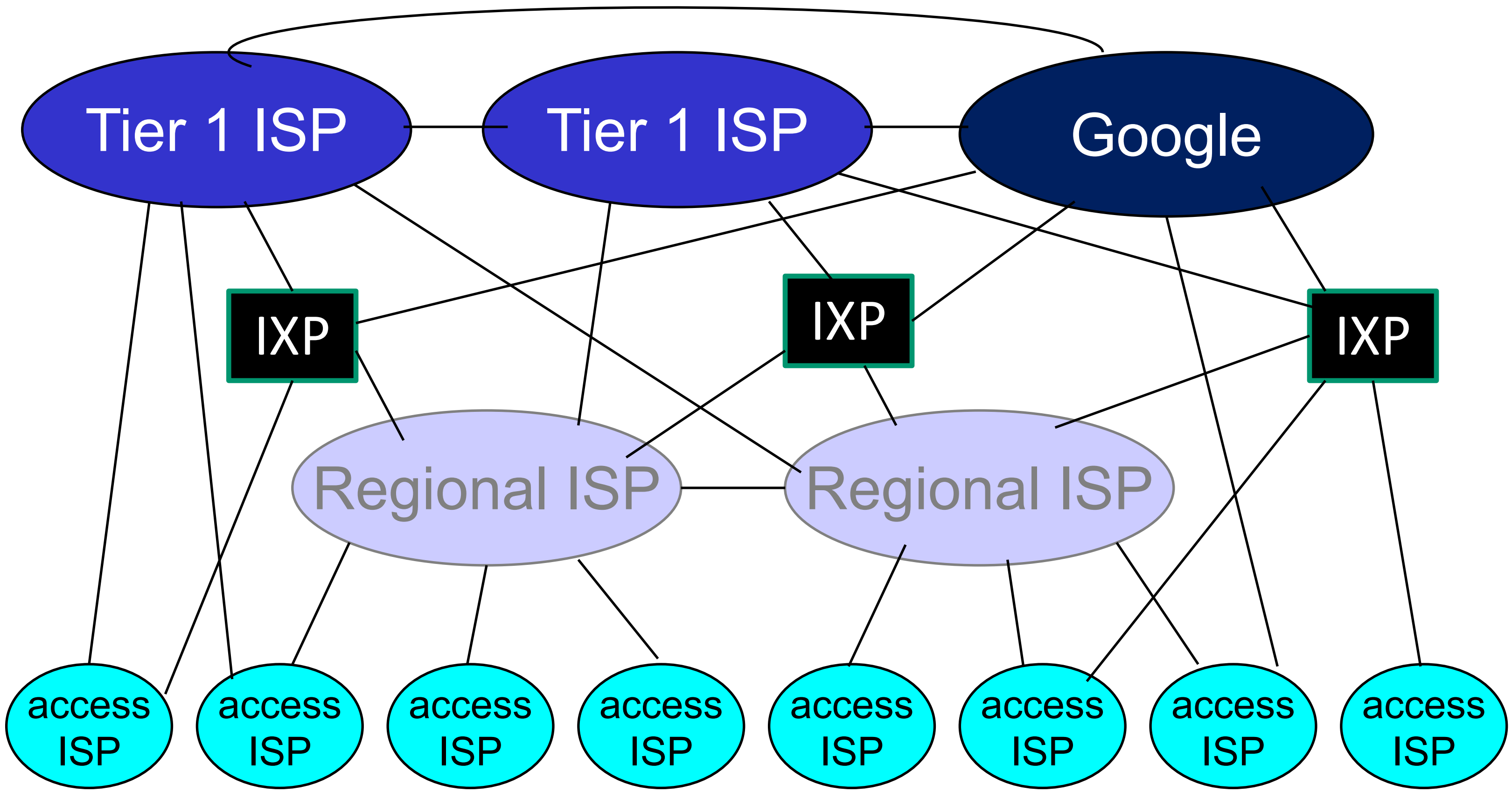


Internet structure: network of networks

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



Internet structure: 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 network (e.g, Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

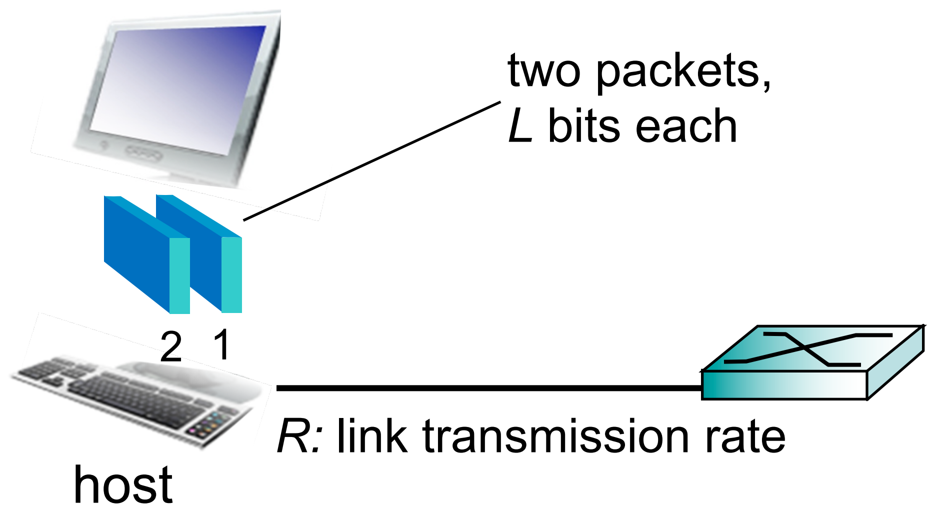
~~1.6 networks under attack: security~~

~~1.7 history~~

Host: sends *packets* of data (Recap)

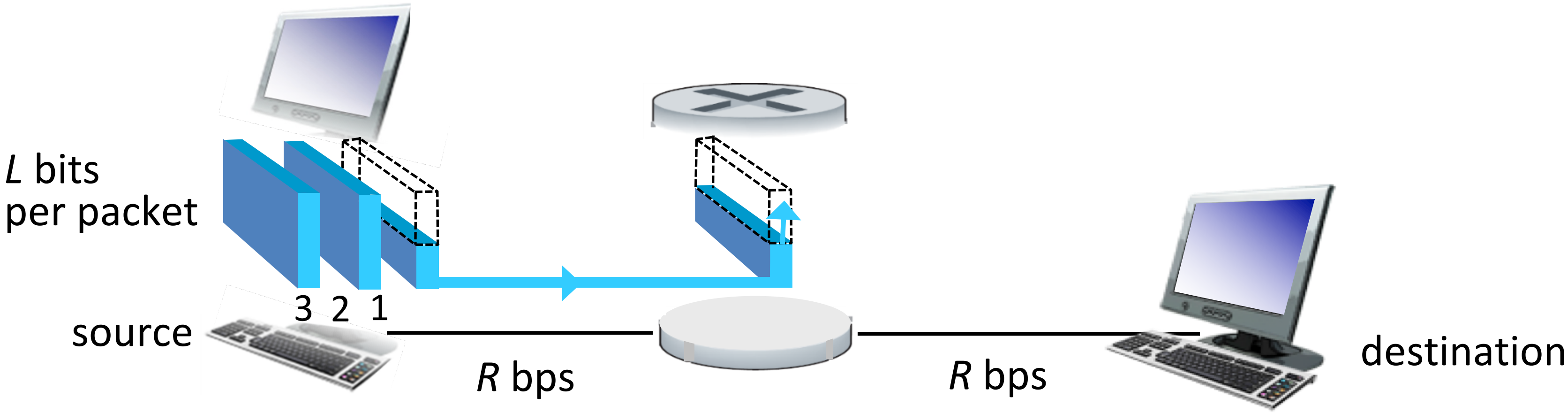
host sending function:

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



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

Store-and-forward: transmission delay



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

more on delay shortly …

one-hop numerical example:

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

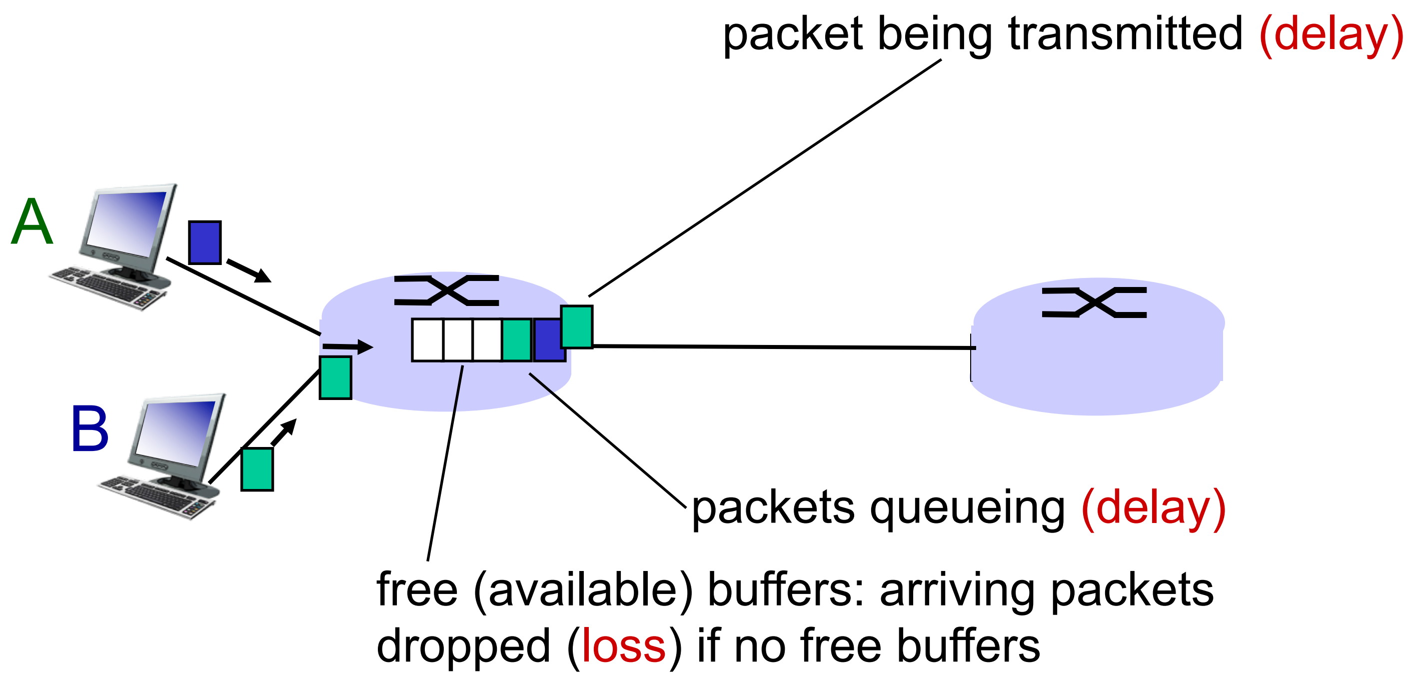
Follow-up questions

- What if we have N hops?
- What if we have P packets?
    - Back-to-back

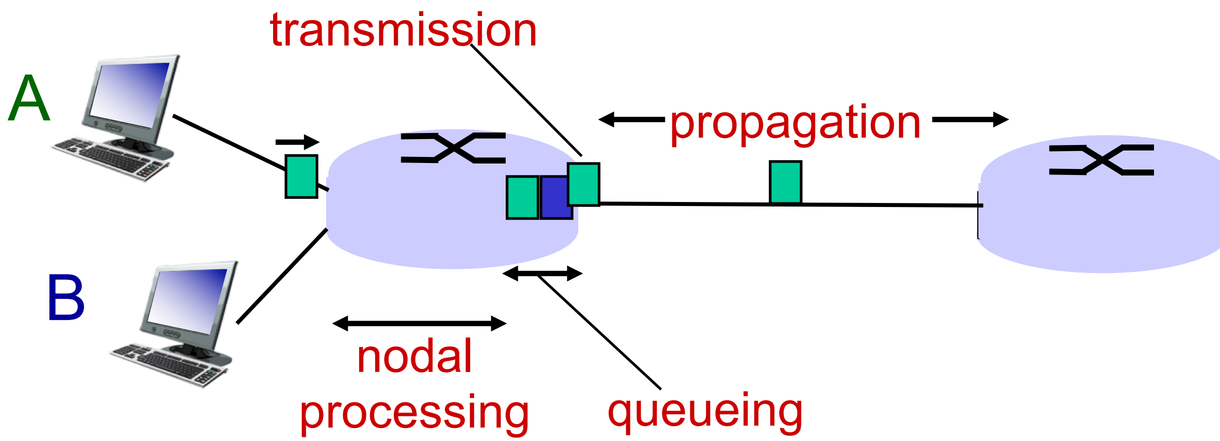
How do loss and delay occur?

packets *queue* in router buffers

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



Four sources of packet delay



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

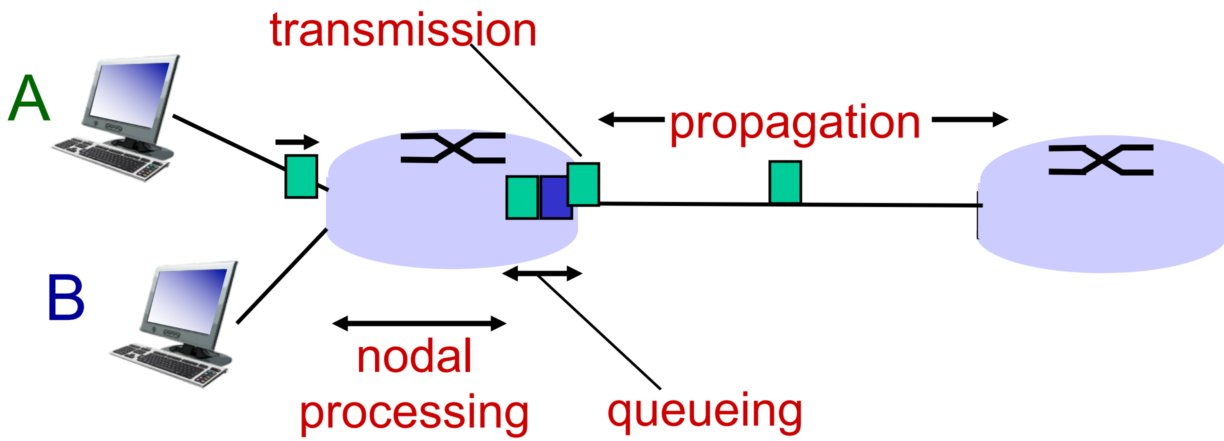
d_{proc}: nodal processing

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

d_{queue}: queueing delay

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

Four sources of packet delay



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

d_{trans}: transmission delay:

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

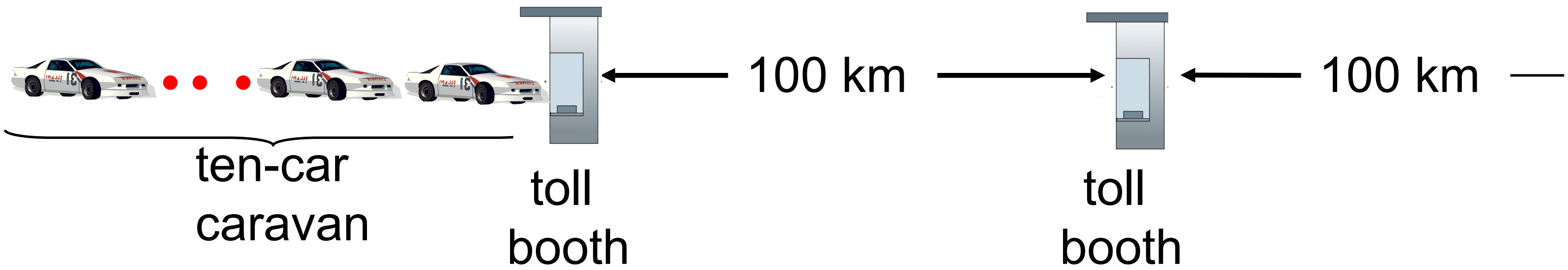
d_{prop}: propagation delay:

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

d_{trans} and d_{prop}
very different

* Check out the Java applet for an interactive animation on trans vs. prop delay

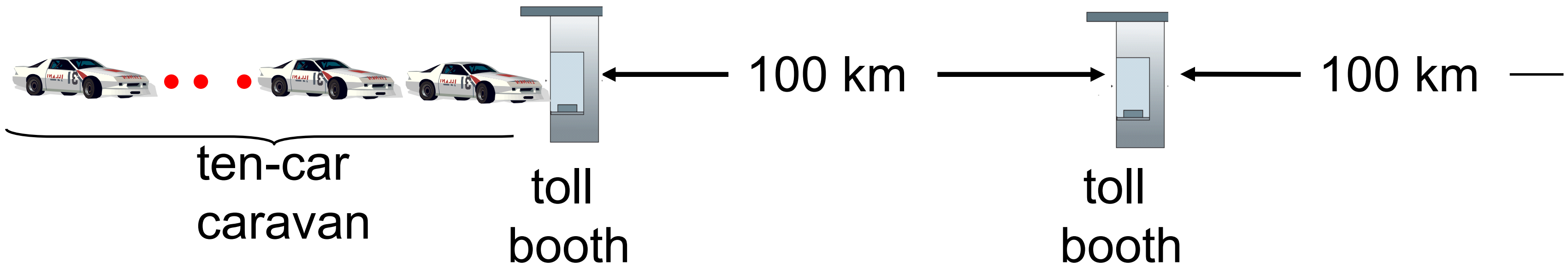
Caravan analogy



- cars "propagate" at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- car~bit; caravan ~ packet
- *Q: How long until caravan is lined up before 2nd toll booth?*

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

Caravan analogy (more)

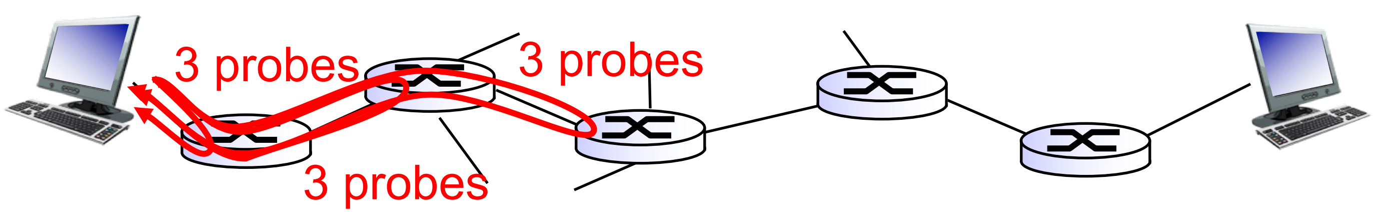


- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?

  - *A: Yes!* after 7 min, 1st car arrives at second booth; three cars still at 1st 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
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.



"Real" Internet delays, routes

traceroute: www.google.com

3 delay measurements from home (ATT)

1 192.168.0.1 (192.168.0.1) 2.079 ms 1.292 ms 1.299 ms
2 96.120.112.229 (96.120.112.229) 10.446 ms 10.082 ms 9.130 ms
3 96.110.168.189 (96.110.168.189) 9.258 ms 8.858 ms 9.012 ms
4 be-22-ar01.indianapolis.in.indiana.comcast.net (68.86.188.97) 17.090 ms 16.266 ms 16.521 ms
5 be-3-ar01.area4.il.chicago.comcast.net (68.86.188.181) 29.714 ms 28.072 ms 28.933 ms
6 be-33491-cr02.350ecermak.il.ibone.comcast.net (68.86.91.165) 30.148 ms 28.974 ms 29.939 ms
7 be-10577-pe03.350ecermak.il.ibone.comcast.net (68.86.86.2) 28.834 ms 28.609 ms 30.490 ms
8 173.167.56.22 (173.167.56.22) 27.929 ms 29.434 ms 28.429 ms
9 * *

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

Demo in Class

1. traceroute www.google.com

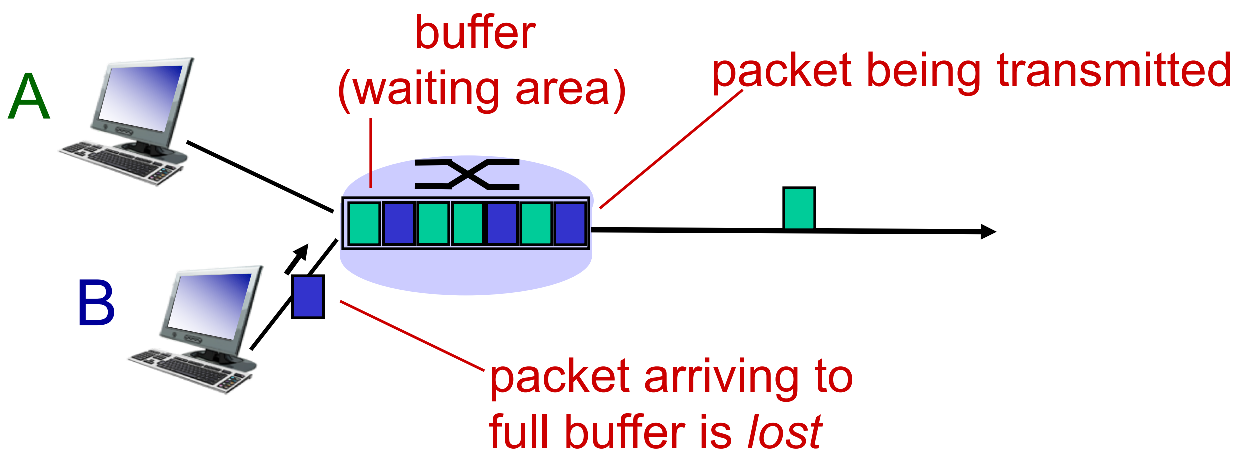
2. traceroute europa.eu

- What are differences you can see?
- Can you see the link across the ocean?
- Why?

Packet loss

- Queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)

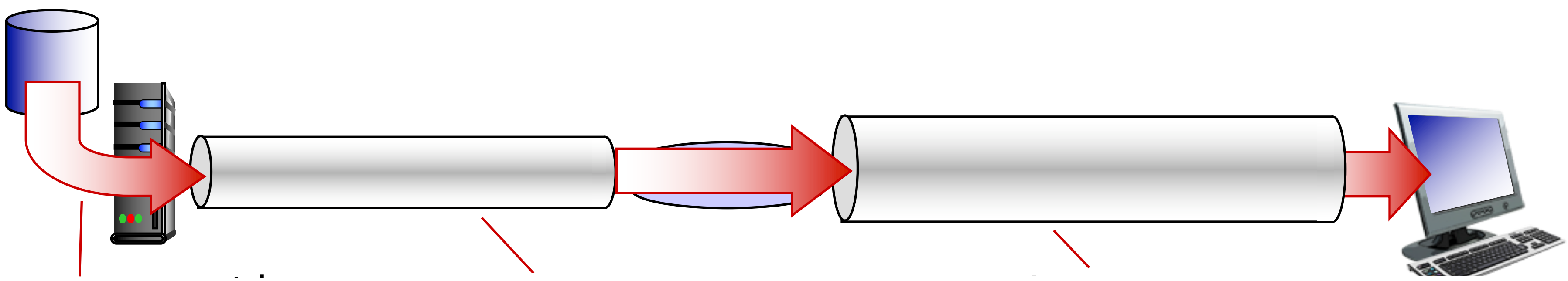
- lost packet may be retransmitted by previous node, by source end system, or not at all



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

Throughput

- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time



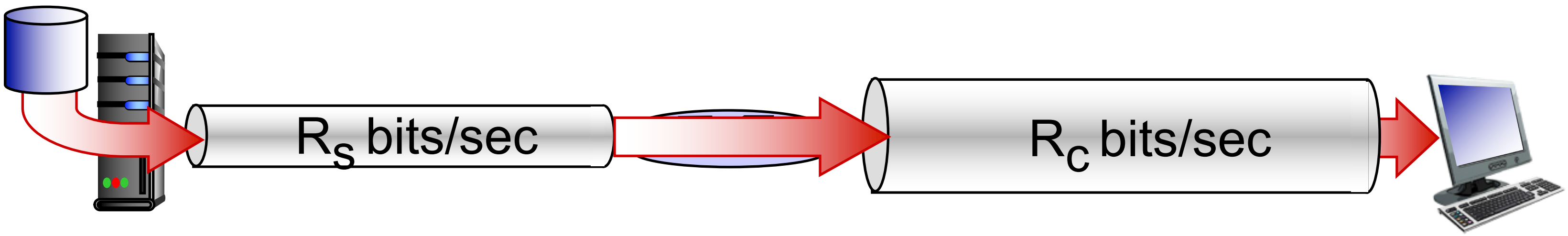
server sends bits (fluid) into pipe

pipe that can carry fluid at rate R_s bits/sec)

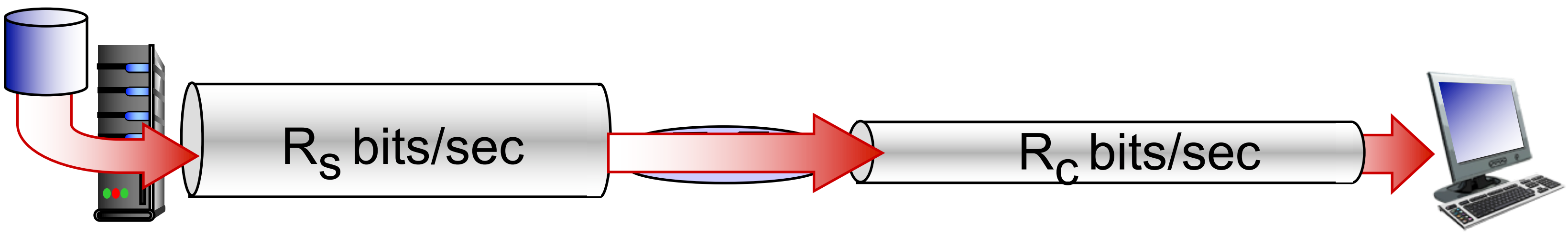
pipe that can carry fluid at rate R_c bits/sec)

Throughput (more)

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



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

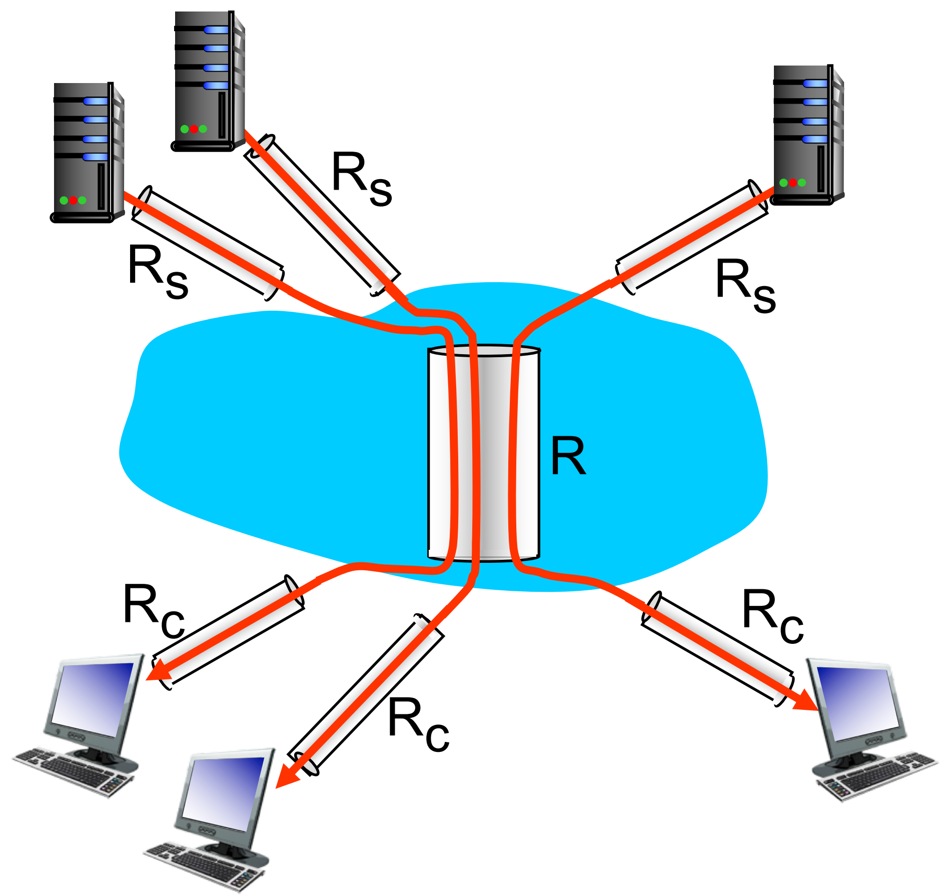


bottleneck link

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

Throughput: Internet scenario

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



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

Chapter 1: roadmap

1.1 what *is* the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

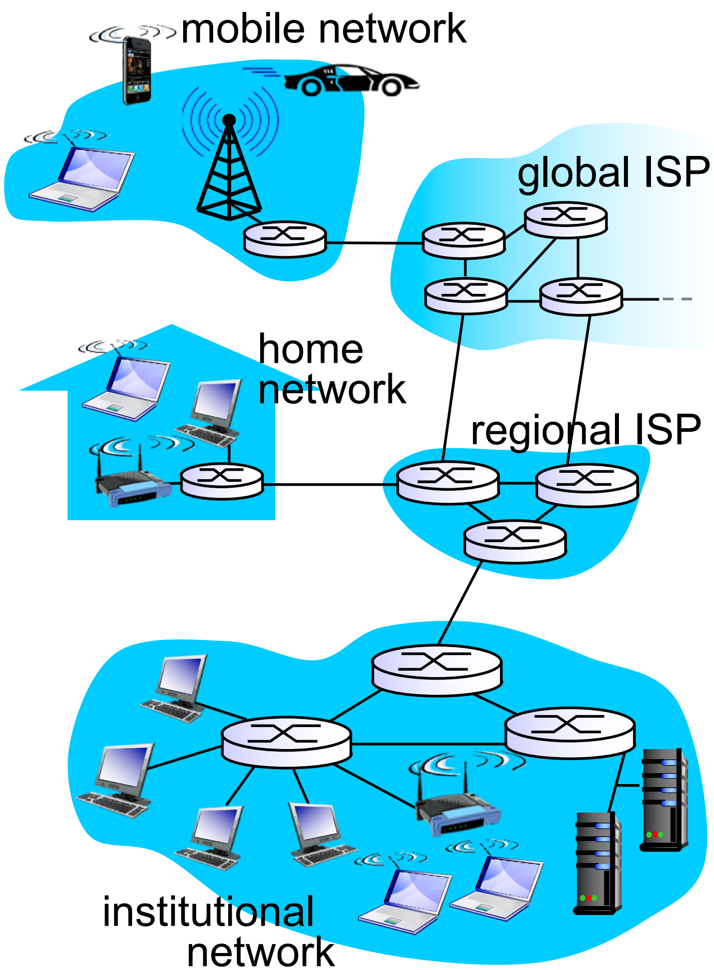
1.5 protocol layers, service models

~~1.6 networks under attack: security~~

~~1.7 history~~

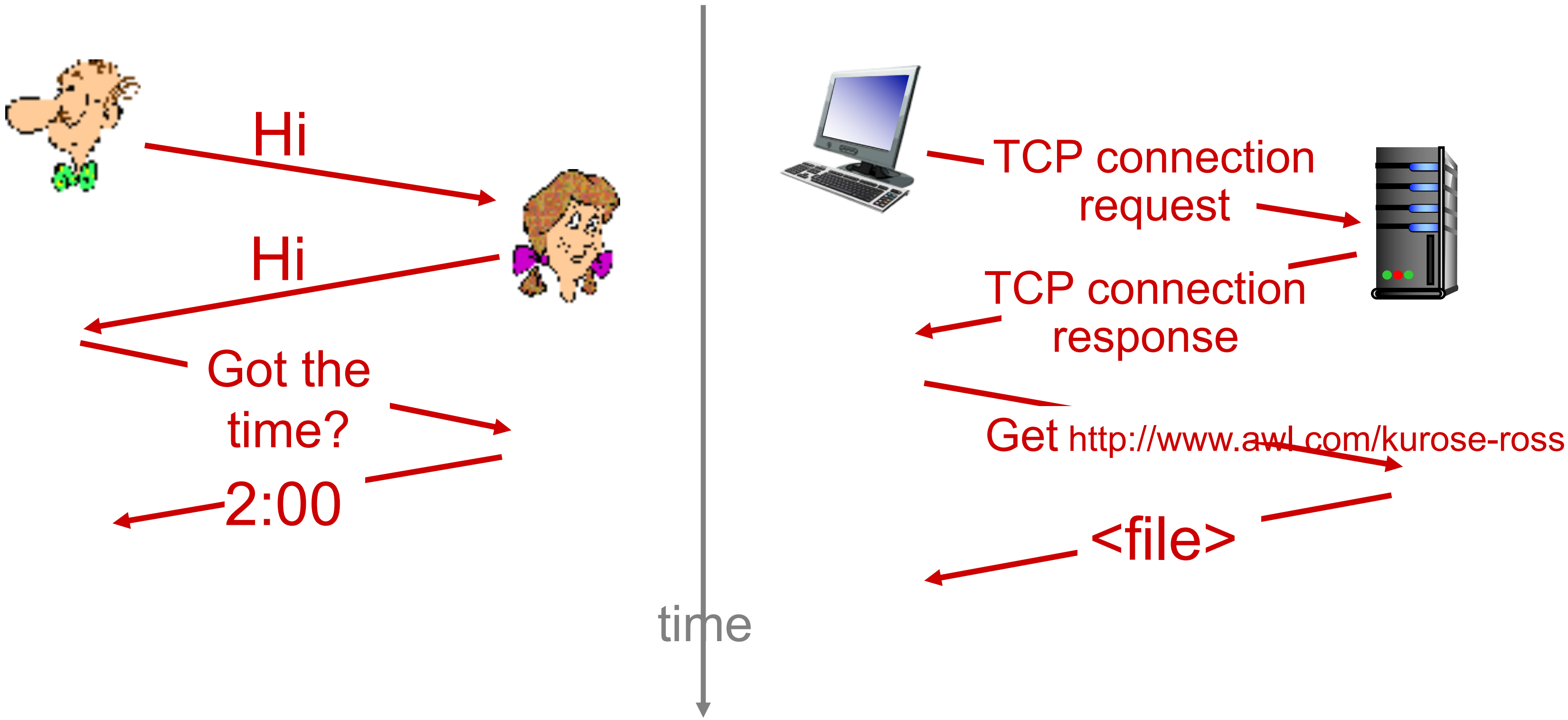
What's the Internet: a software view

- *protocols* control sending, receiving of msgs
  - e.g., HTTP, Skype, TCP, IP, 802.11
- *Internet standards*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



Recap: What's a protocol?

a human protocol and a computer network protocol:



3 elements: format, order of msgs, and *actions*

Recap: What's a protocol?

human protocols:

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

… specific msgs sent

… in a specific order

… specific actions taken when msgs received, or other events

network protocols:

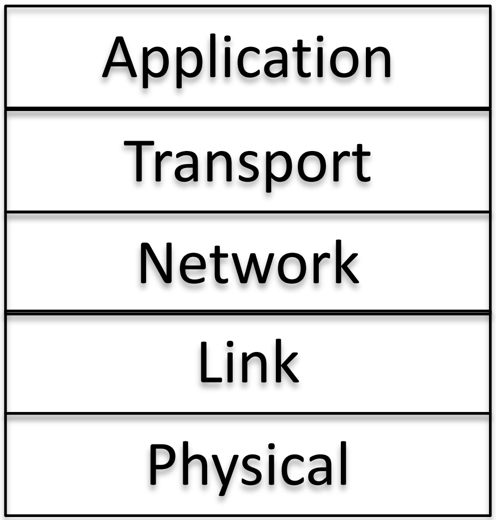
- machines rather than humans
- all communication activity in Internet governed by protocols

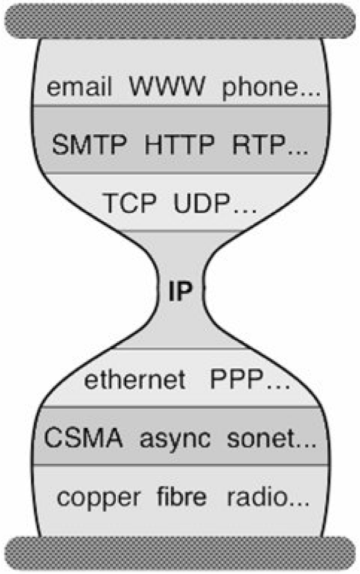
protocols define format, order of msgs sent and received among network entities, and actions taken on msg transmission, receipt

Internet Protocol "layers"

Networks are complex, with many "pieces":

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



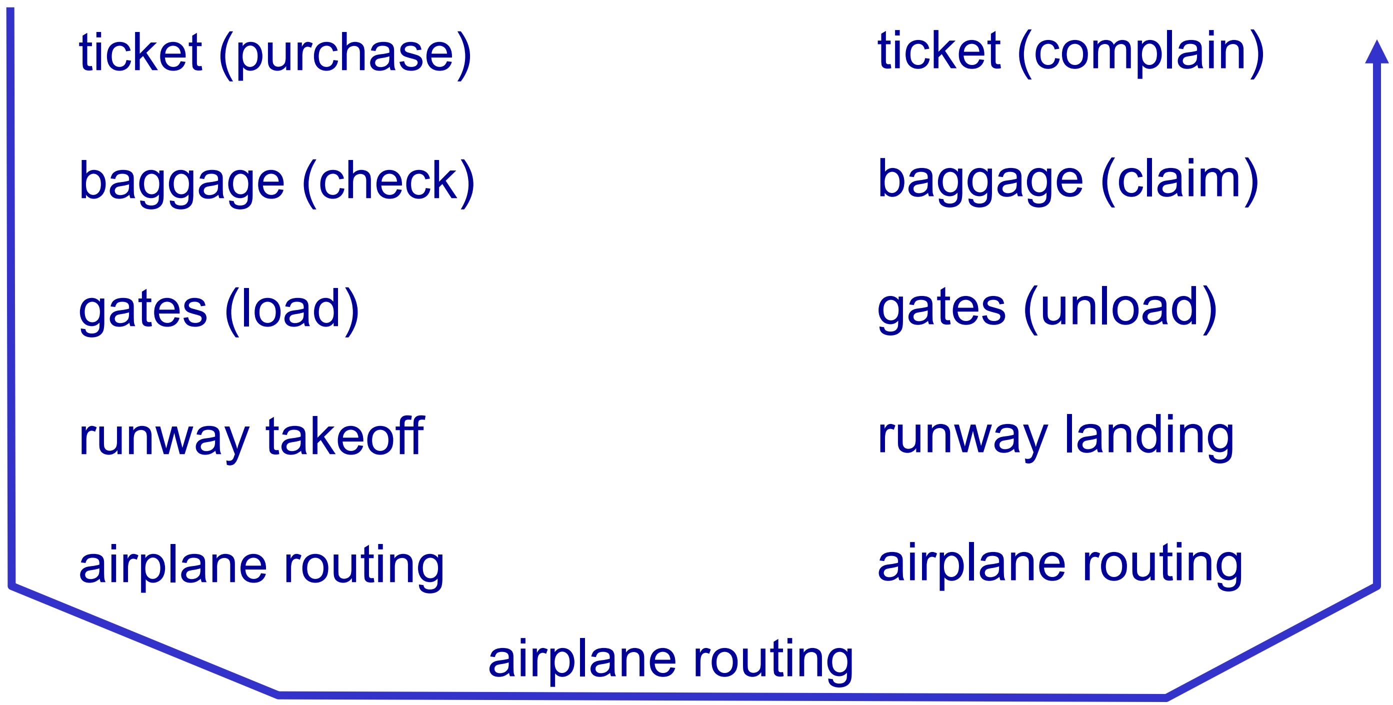


Why Layering?

Decomposed complex delivery into fundamental components

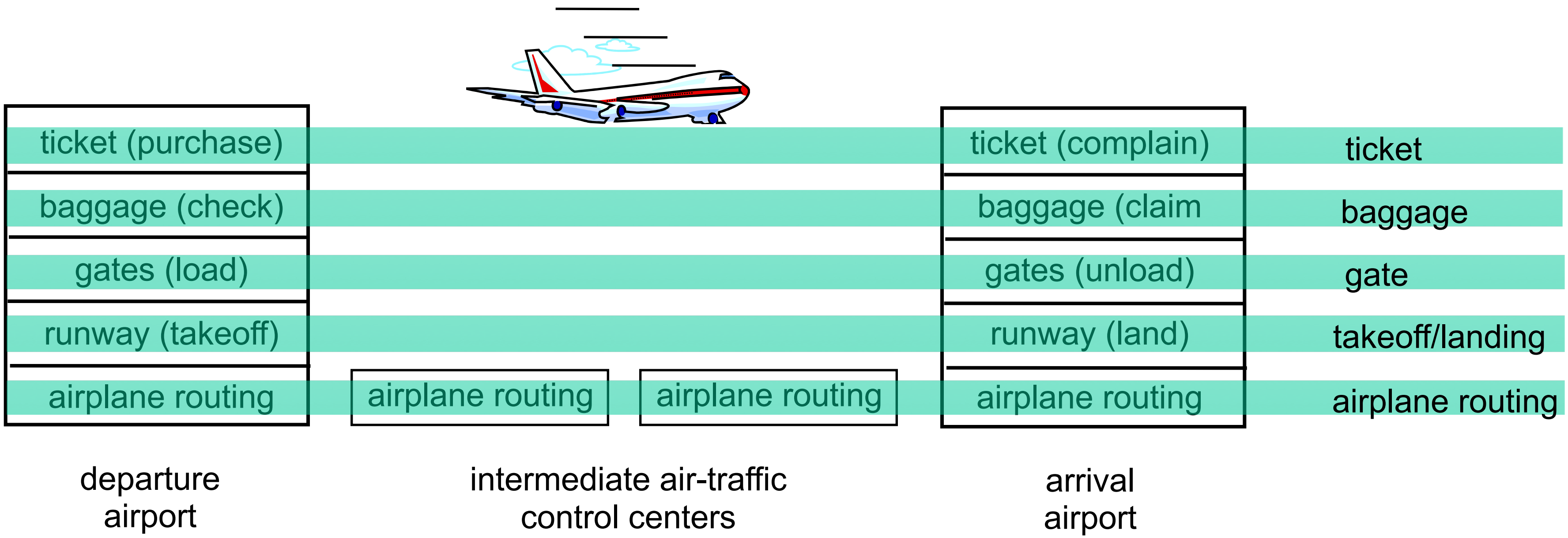
- Explicit structure allows identification, relationship of complex system's pieces
  - layered *reference model* for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system

Analog: air travel



- a series of steps

Layering of airline functionality



layers: each layer implements a service

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

Layering in Internet protocol stack

Applications

… built on …

Reliable (or unreliable) transport

… built on …

Best-effort global packet delivery

… built on …

Best-effort local packet delivery

… built on …

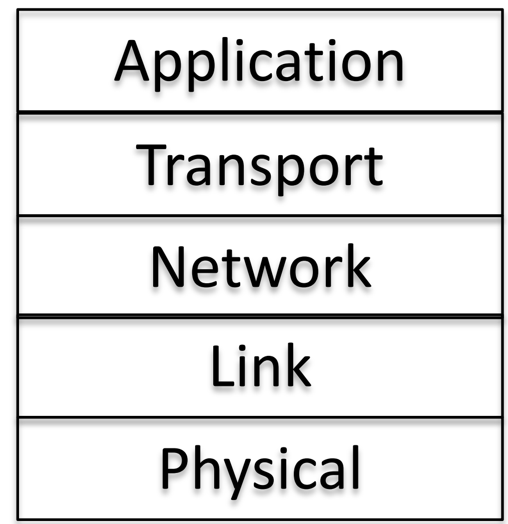
Physical transfer of bits

Application
Transport
Network
Link
Physical

Source: Scott Shenker (UC Berkeley): slide 7 at The Future of Networking, and the Past of Protocols
https://www.youtube.com/watch?v=YHeyuD89n1Y&t=111s

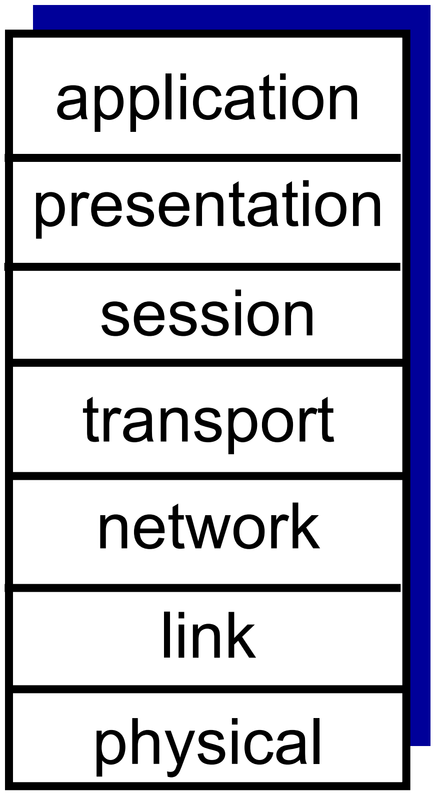
Layering in Internet protocol stack

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- *physical:* bits "on the wire"

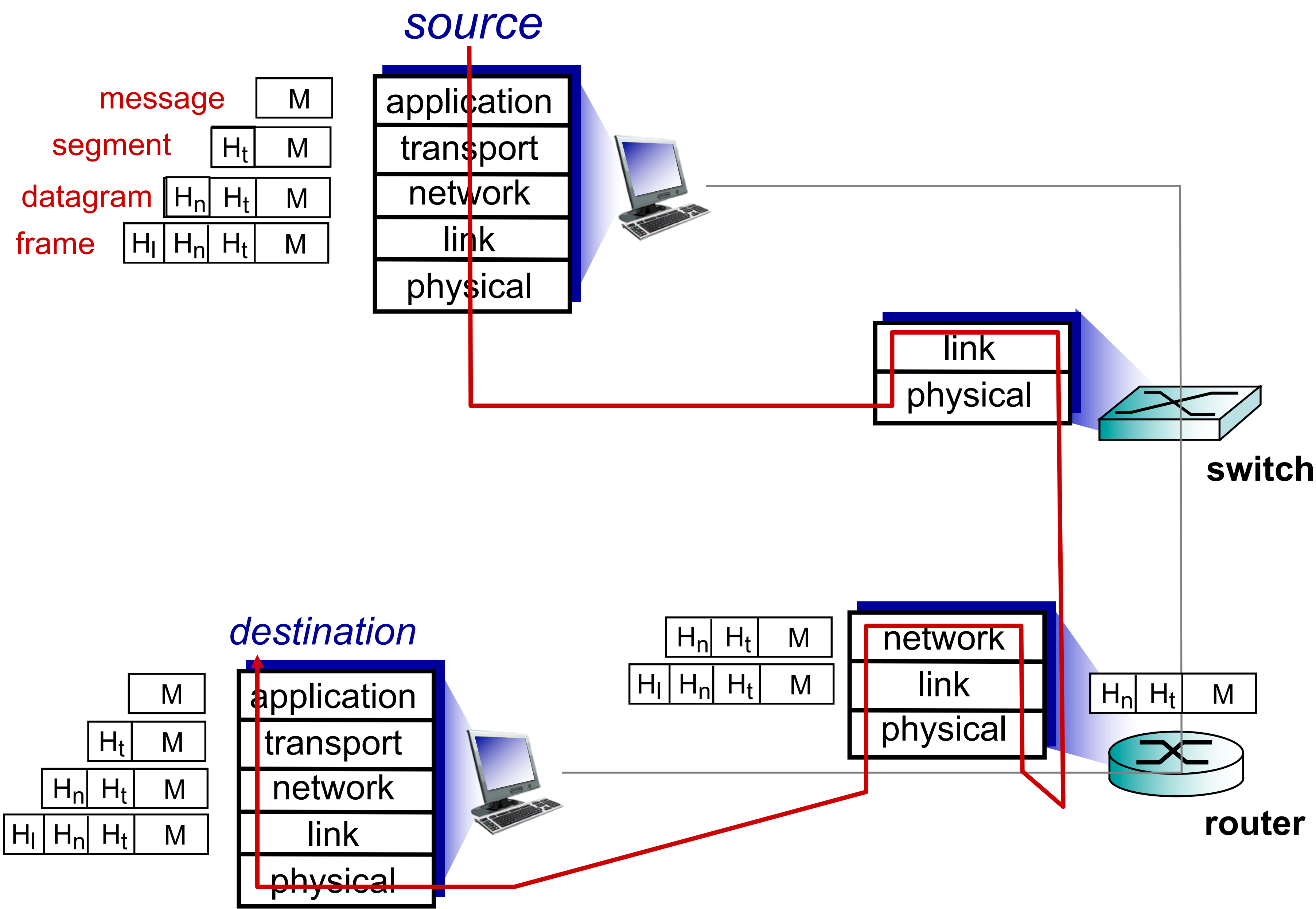


ISO/OSI reference model

- *presentation:* allow applications to interpret meaning of data, e.g., encryption, compression, machine-specific conventions
- *session:* synchronization, checkpointing, recovery of data exchange
- Internet stack "missing" these layers!
  - these services, *if needed*, must be implemented in application
  - needed?



Encapsulation & Decapsulation



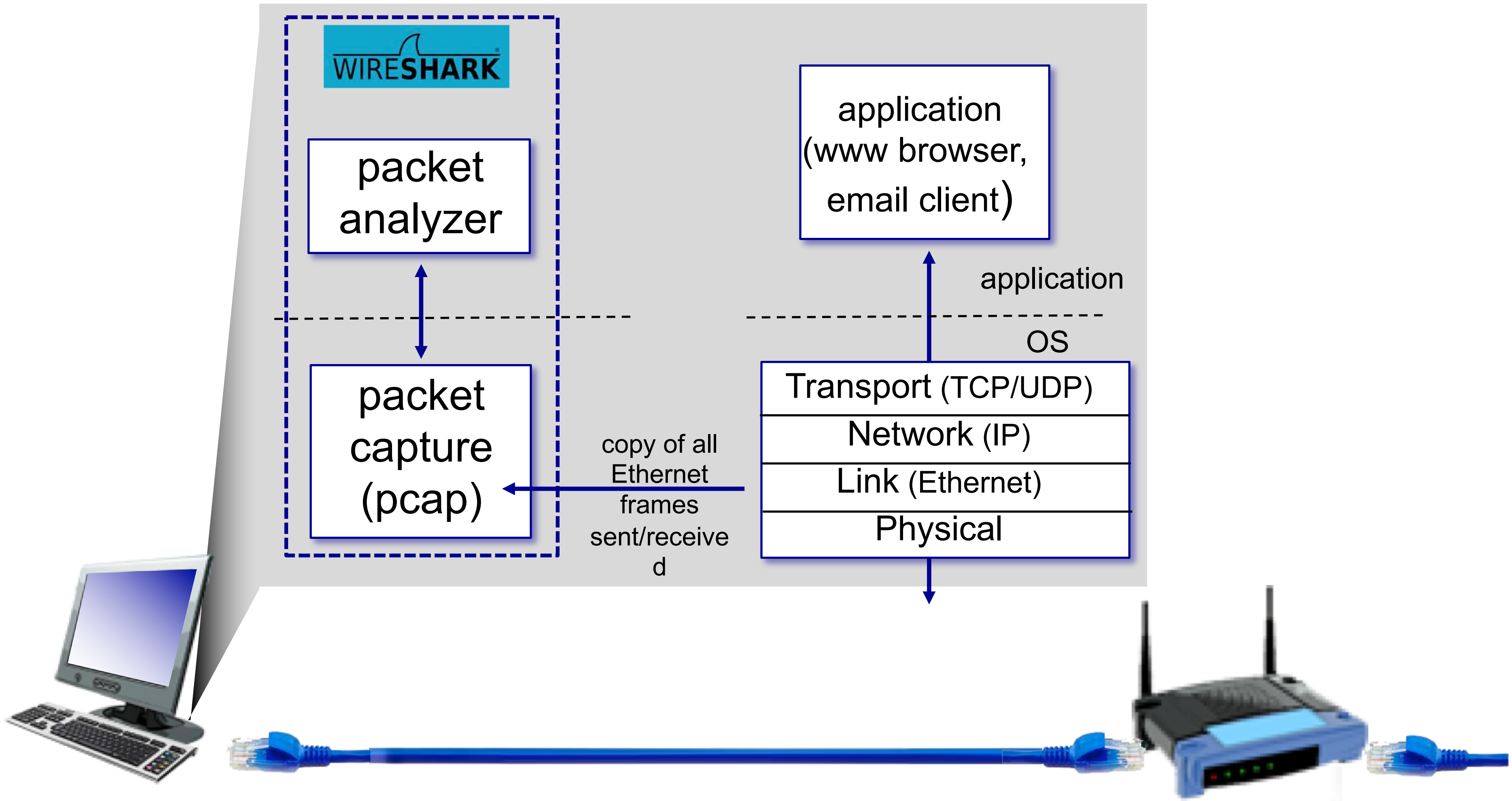
Advanced topics (optional)

- Packet sniffer and analyzer
  - tcpdump (command)
    - > tcpdump –i en0
    - > tcpdump -i en0 -c 10 -w test.cap
    - > tcpdump –r test.cap
    - More usage via Google " tcpdump"
  - wireshark (UI)

Chapter 1: Summary

- what's the Internet?
- network edge
  - hosts, access network
- network core
  - Packet switching versus. circuit switching
- performance: loss, delay, throughput
- what's a protocol?
  - protocol layers, service models

Chapter 1
Additional Slides



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