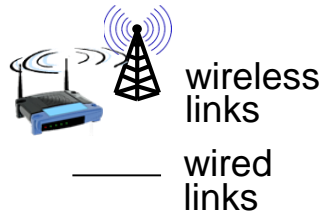


# Chapter I Introduction

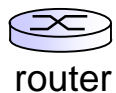
# What's the Internet: “nuts and bolts” view



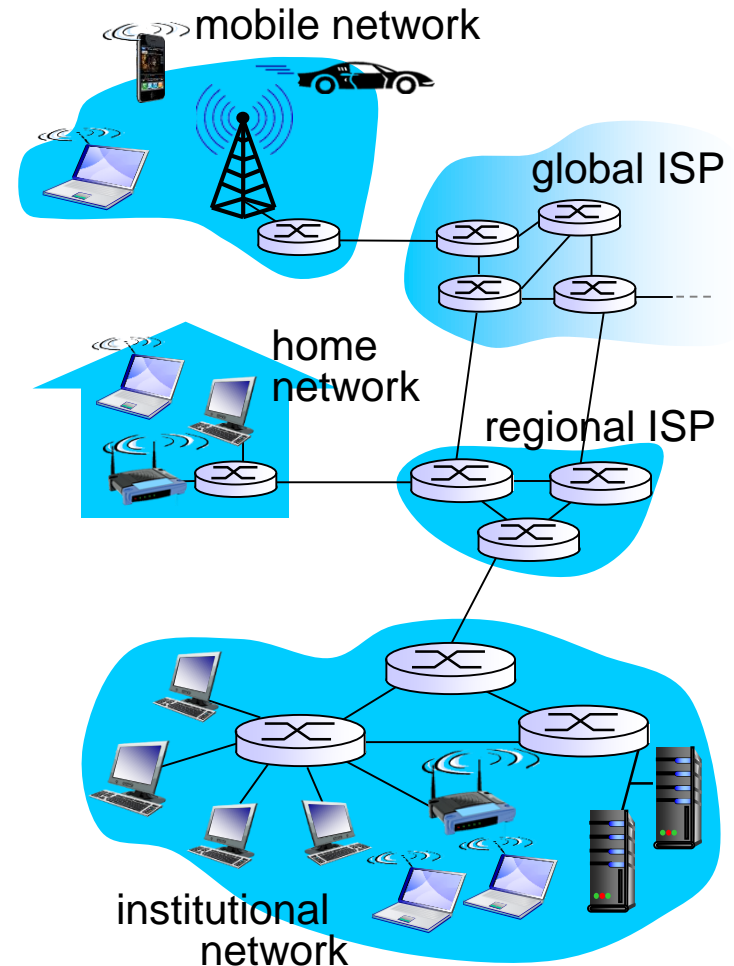
- ❖ billions of connected computing devices:
  - *hosts* = *end systems*
  - running *network apps*



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



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



# “Fun” internet appliances



IP picture frame  
<http://www.ceiva.com/>



Web-enabled toaster +  
weather forecaster



Tweet-a-watt:  
monitor energy use



Internet  
refrigerator



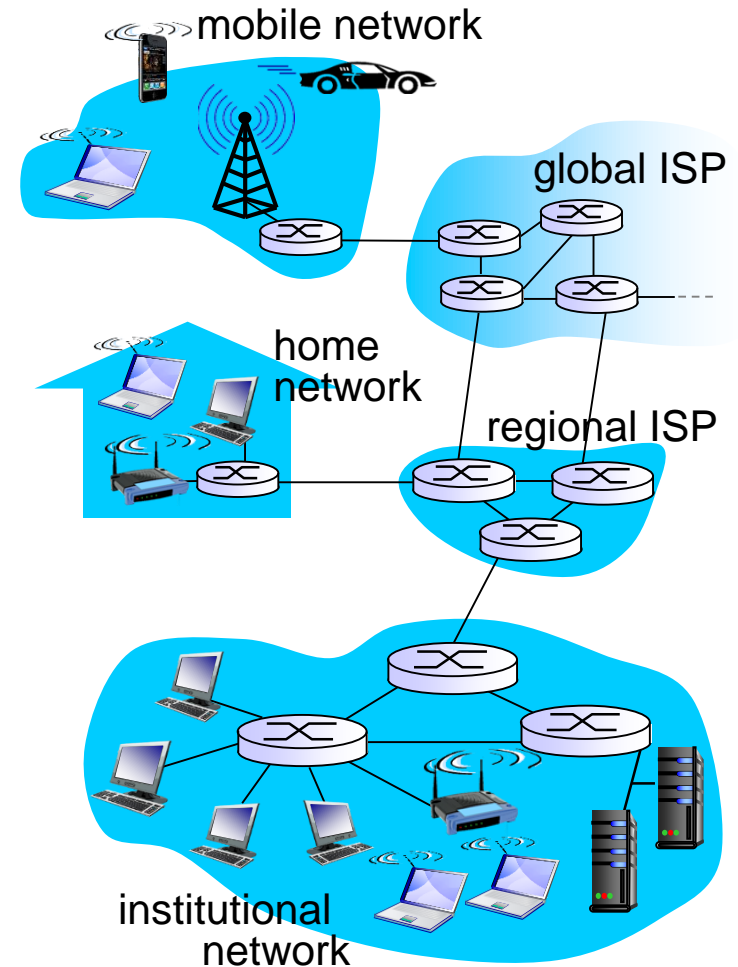
Slingbox: watch,  
control cable TV remotely



Internet phones

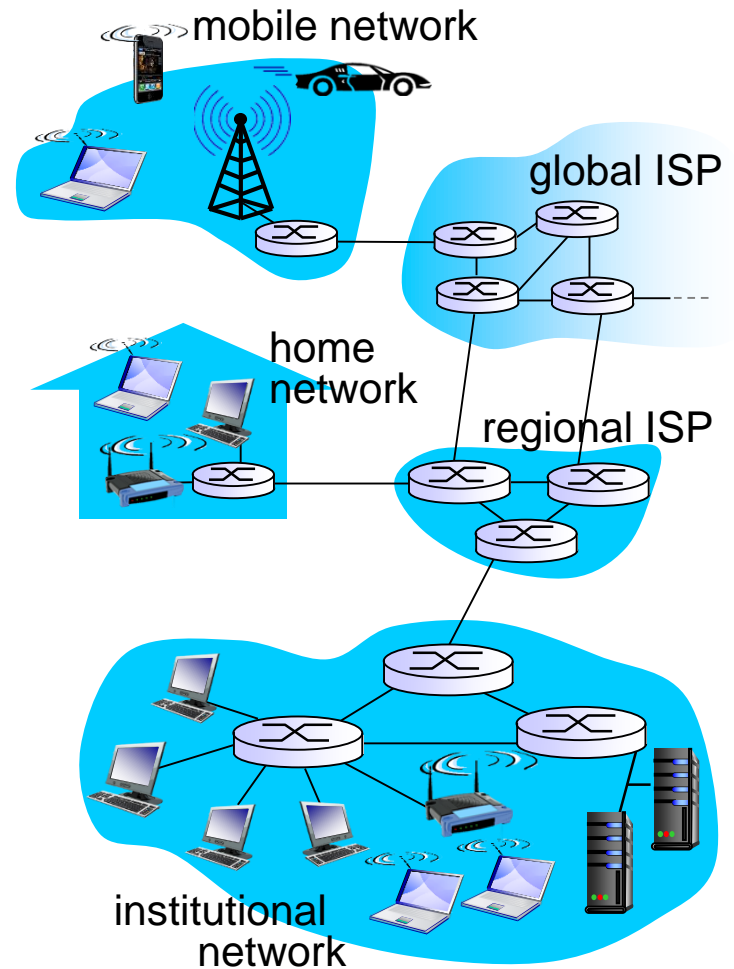
# What's the Internet: “nuts and bolts” view

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



# What's the Internet: a service view

- ❖ *Infrastructure that provides services to applications:*
  - Web, VoIP, email, games, e-commerce, social nets, ...
- ❖ *provides programming interface to apps*
  - hooks that allow sending and receiving app programs to “connect” to Internet
  - provides service options, analogous to postal service



# What's a protocol?

## *human protocols:*

- ❖ “what's the time?”
  - ❖ “I have a question”
  - ❖ introductions
- ... specific msgs sent
- ... specific actions taken  
when msgs received, or  
other events

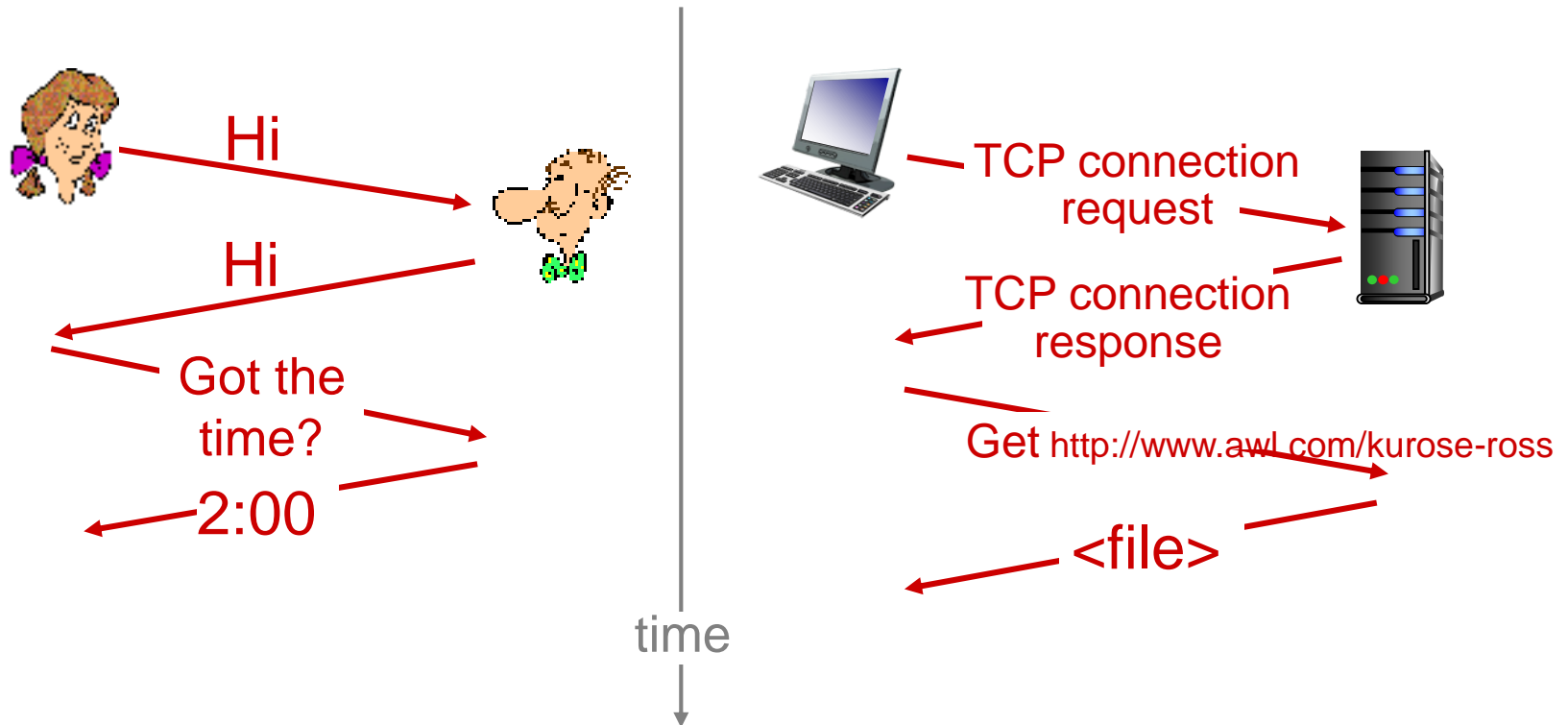
## *network protocols:*

- ❖ 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*

# What's a protocol?

a human protocol and a computer network protocol:



**Q:** other human protocols?

# A closer look at network structure:

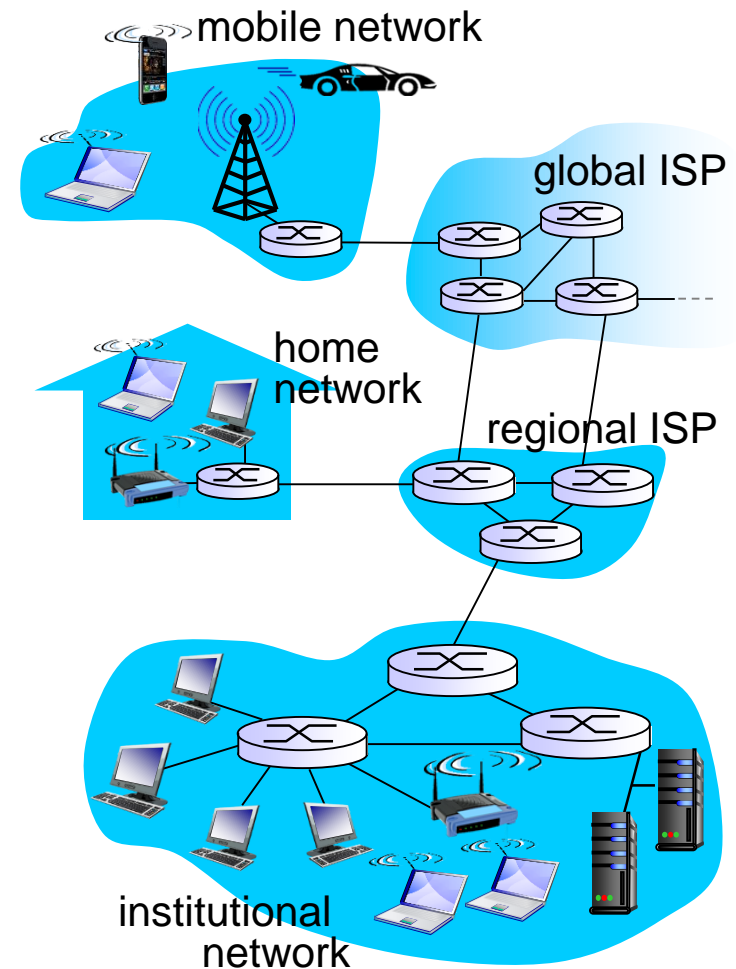
## ❖ *network edge:*

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

## ❖ *access networks, physical media:* wired, wireless communication links

## ❖ *network core:*

- interconnected routers
- network of networks





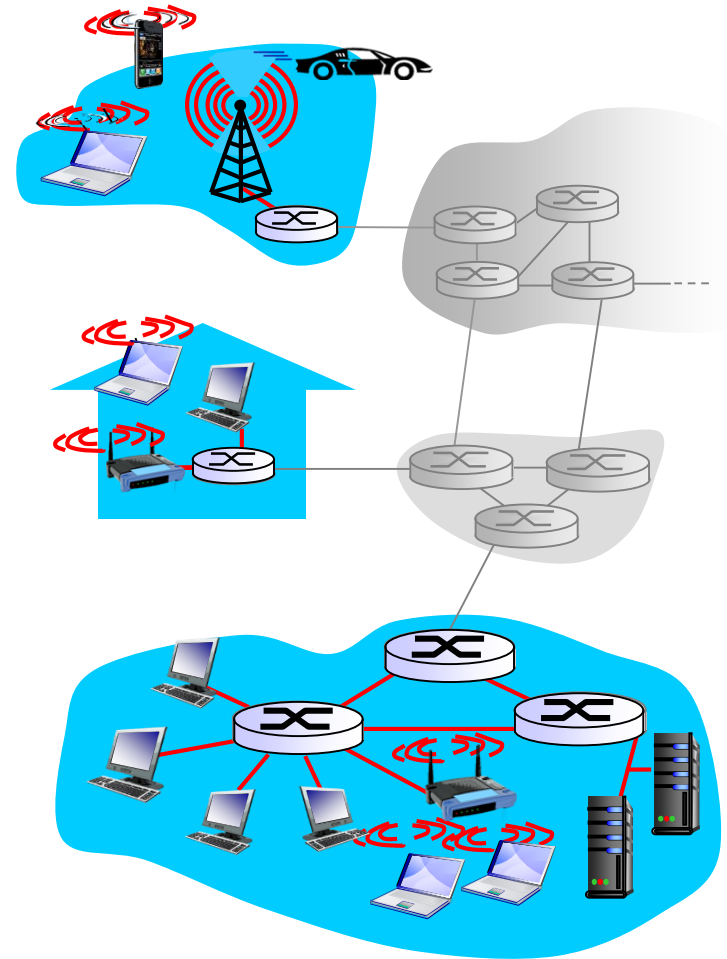
# Access networks and physical media

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

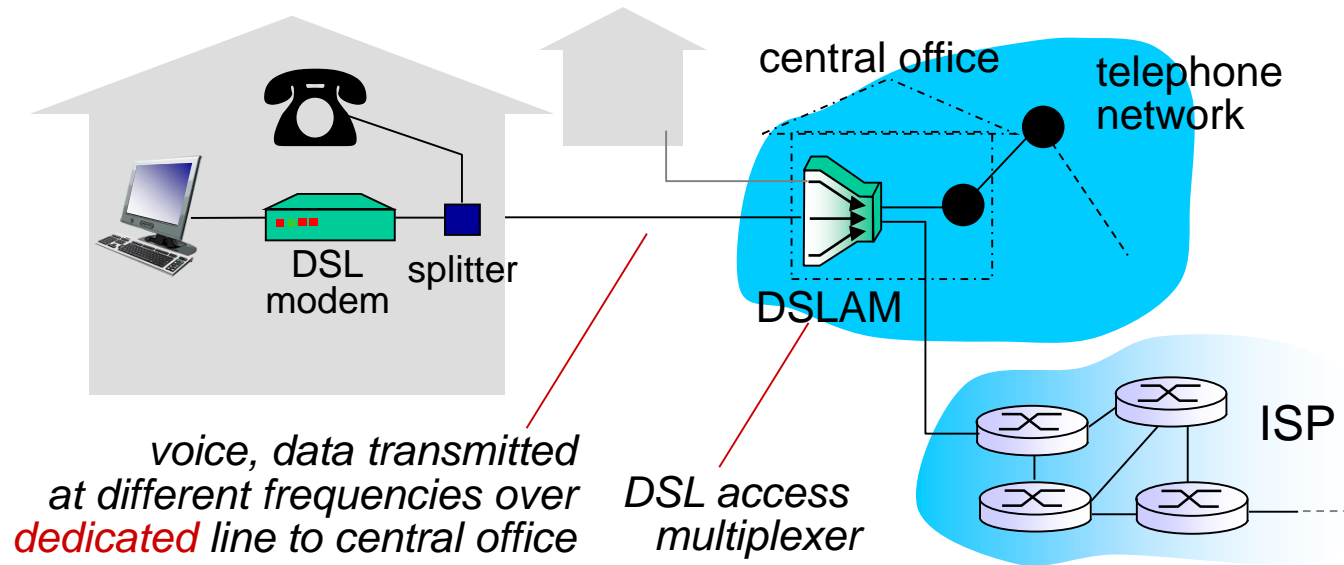
- ❖ residential access nets
- ❖ institutional access networks (school, company)
- ❖ mobile access networks

*keep in mind:*

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

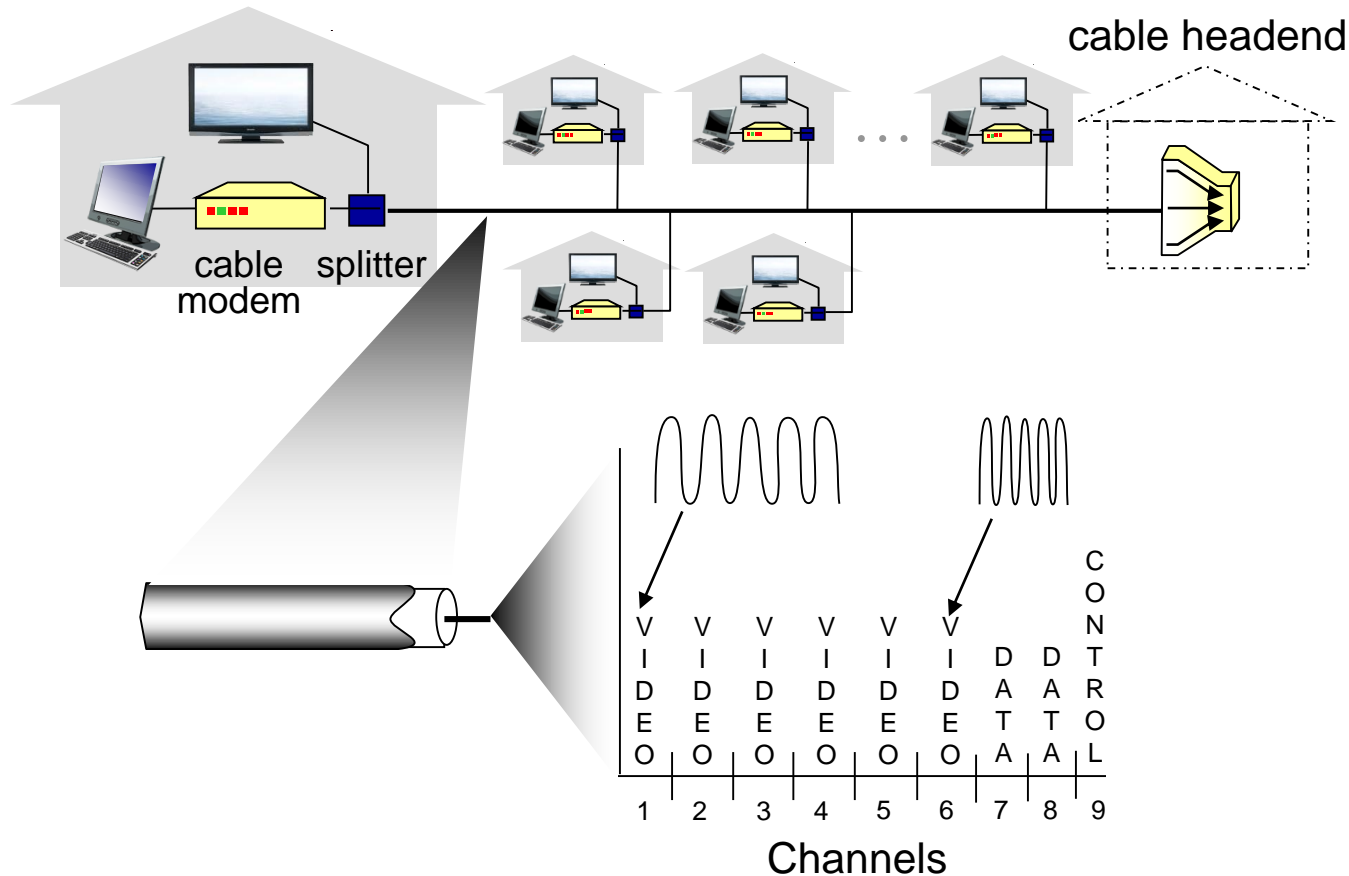


# Access net: digital subscriber line (DSL)



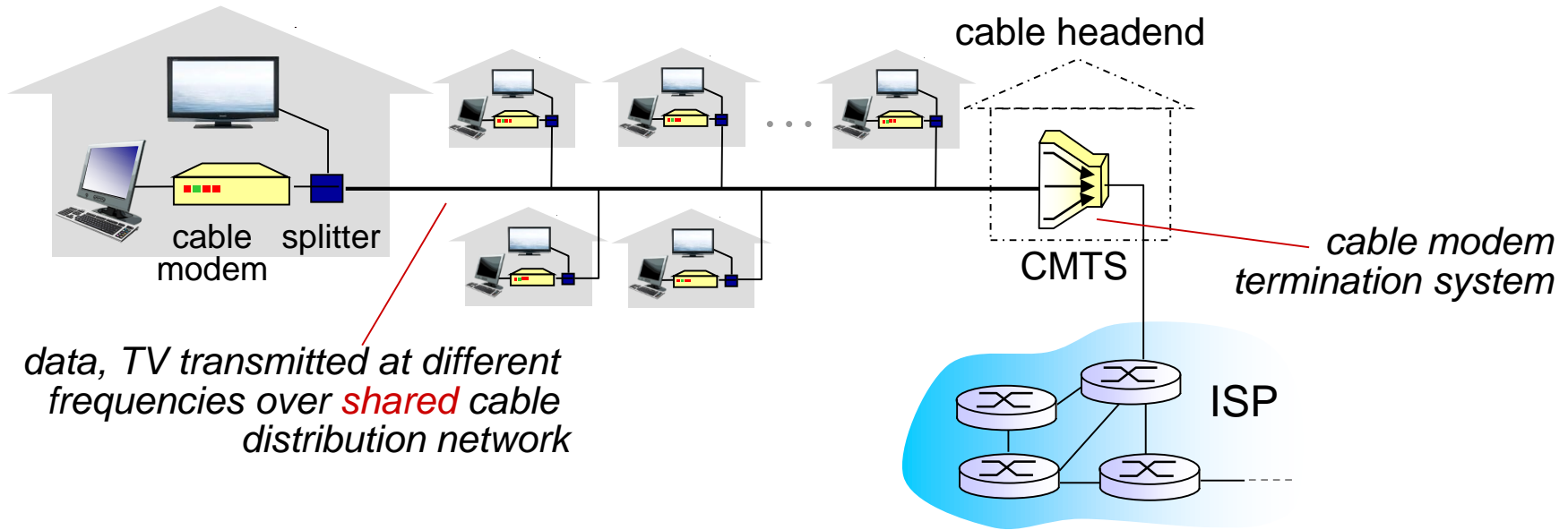
- ❖ use **existing** telephone line to central office DSLAM
  - data over DSL phone line goes to Internet
  - voice over DSL phone line goes to telephone net
- ❖ < 2.5 Mbps upstream transmission rate (typically < 1 Mbps)
- ❖ < 24 Mbps downstream transmission rate (typically < 10 Mbps)

# Access net: cable network



***frequency division multiplexing:*** different channels transmitted in different frequency bands

# Access net: cable network



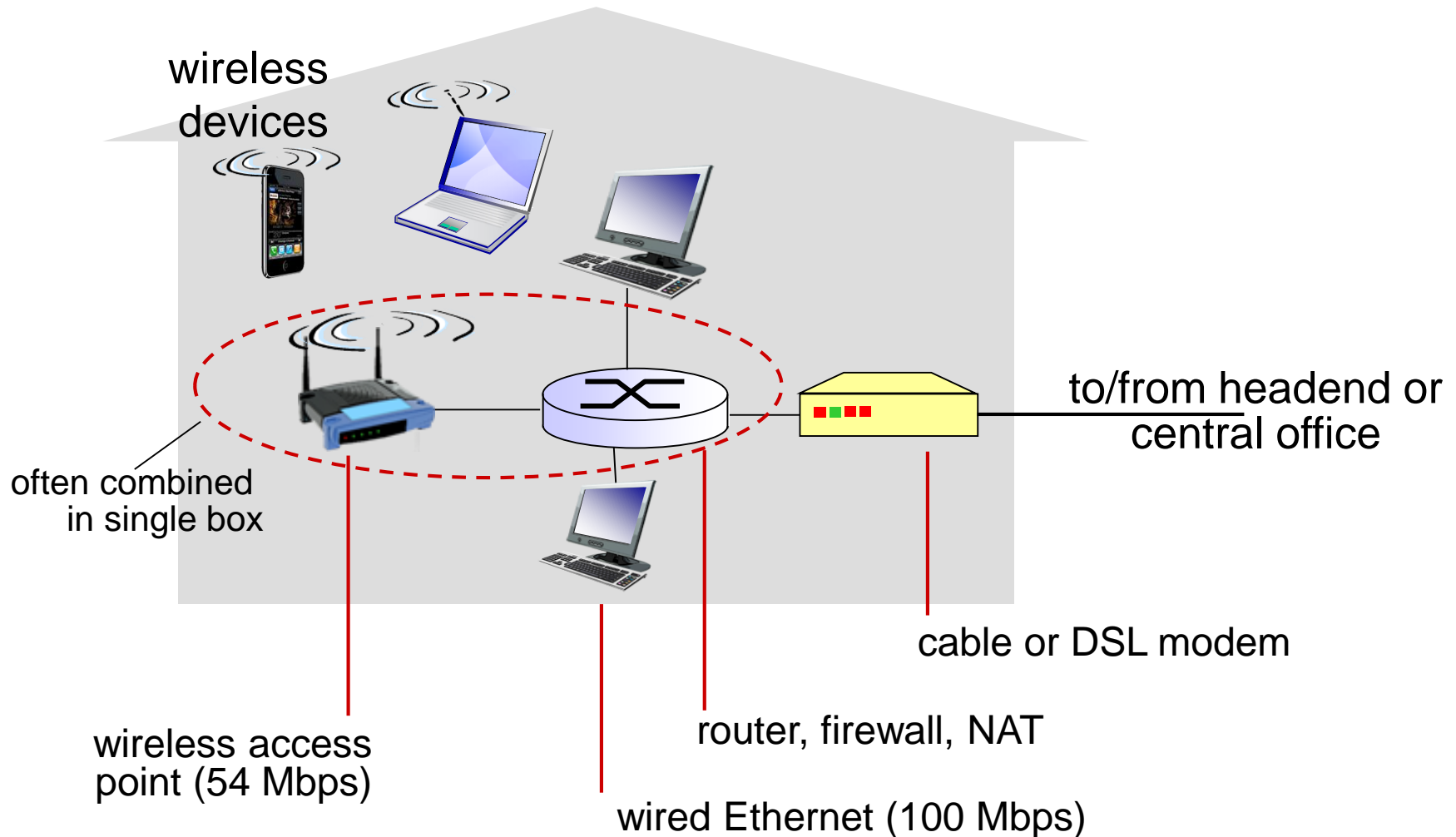
## ❖ HFC: hybrid fiber coax

- asymmetric: up to 30Mbps downstream transmission rate, 2 Mbps upstream transmission rate

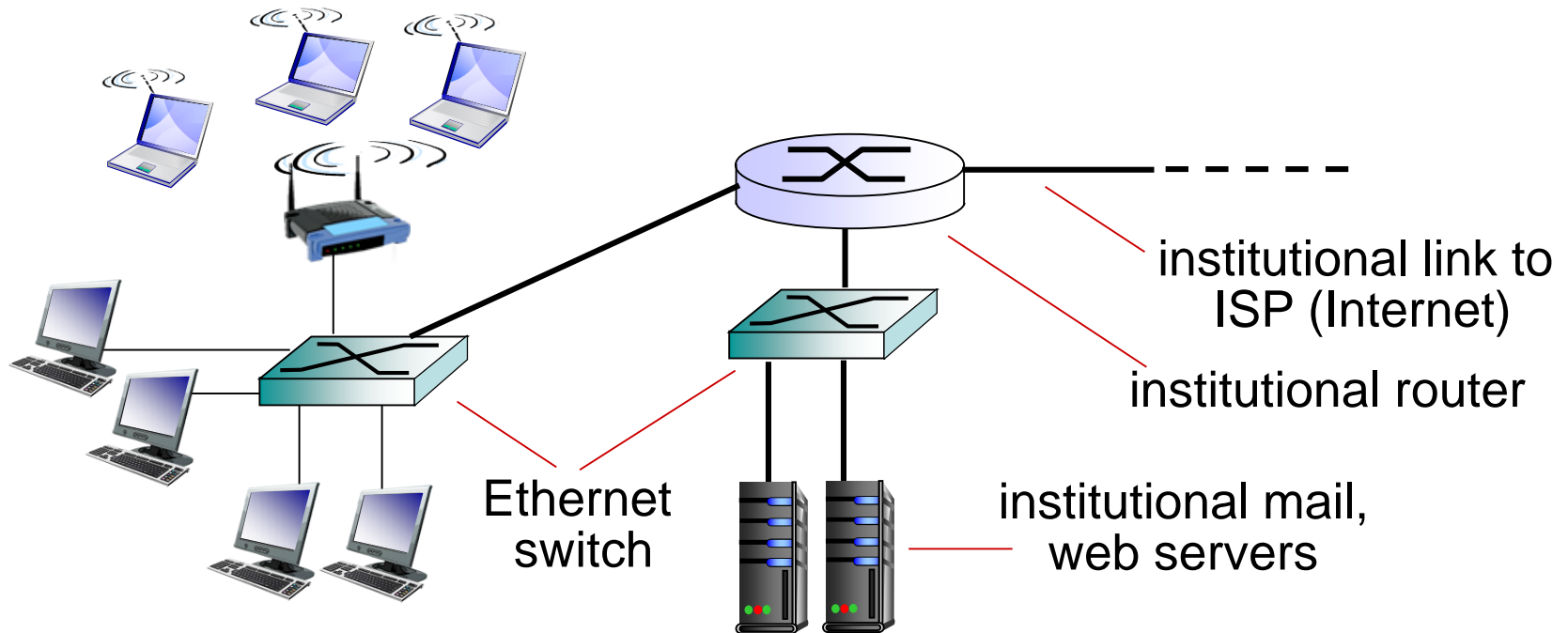
## ❖ network of cable, fiber attaches homes to ISP router

- homes *share access network* to cable headend
- unlike DSL, which has dedicated access to central office

# Access net: home network



# Enterprise access networks (Ethernet)



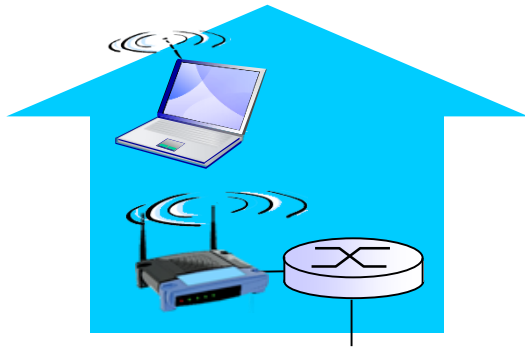
- ❖ typically used in companies, universities, etc
- ❖ 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- ❖ today, end systems typically connect into Ethernet switch

# Wireless access networks

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

## *wireless LANs:*

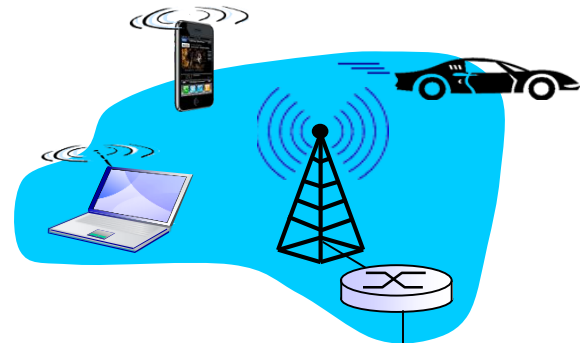
- within building (100 ft)
- 802.11b/g (WiFi): 11, 54 Mbps transmission rate



*to Internet*

## *wide-area wireless access*

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

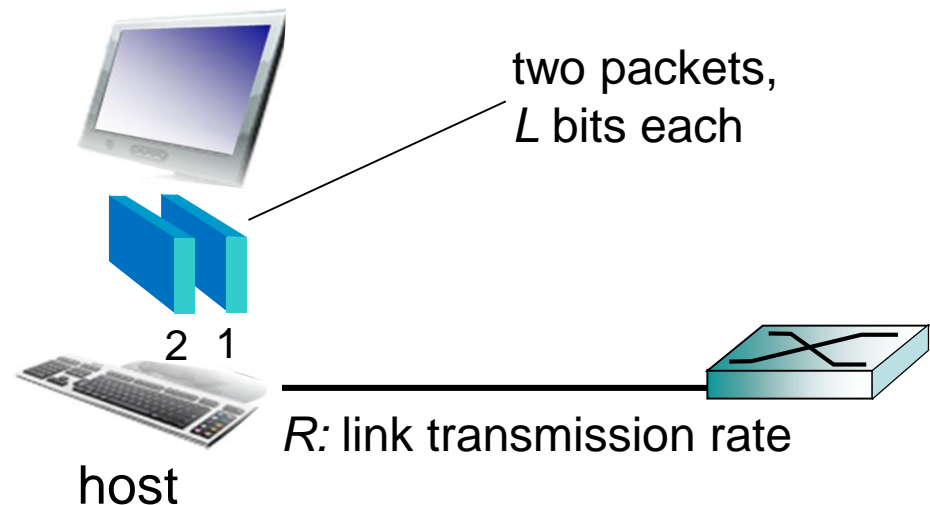


*to Internet*

# Host: sends *packets* of data

host sending function:

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



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



# Physical media

- ❖ **bit:** propagates between transmitter/receiver pairs
- ❖ **physical link:** what lies between transmitter & receiver
- ❖ **guided media:**
  - signals propagate in solid media: copper, fiber, coax
- ❖ **unguided media:**
  - signals propagate freely, e.g., radio

## *twisted pair (TP)*

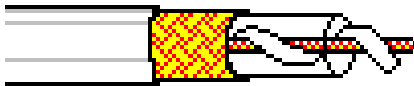
- ❖ two insulated copper wires
  - Category 5: 100 Mbps, 1 Gbps Ethernet
  - Category 6: 10Gbps



# Physical media: coax, fiber

## *coaxial cable:*

- ❖ two concentric copper conductors
- ❖ bidirectional
- ❖ broadband:
  - multiple channels on cable
  - HFC



## *fiber optic cable:*

- ❖ glass fiber carrying light pulses, each pulse a bit
- ❖ high-speed operation:
  - high-speed point-to-point transmission (e.g., 10s-100s Gpbs transmission rate)
- ❖ low error rate:
  - repeaters spaced far apart
  - immune to electromagnetic noise



# Physical media: radio

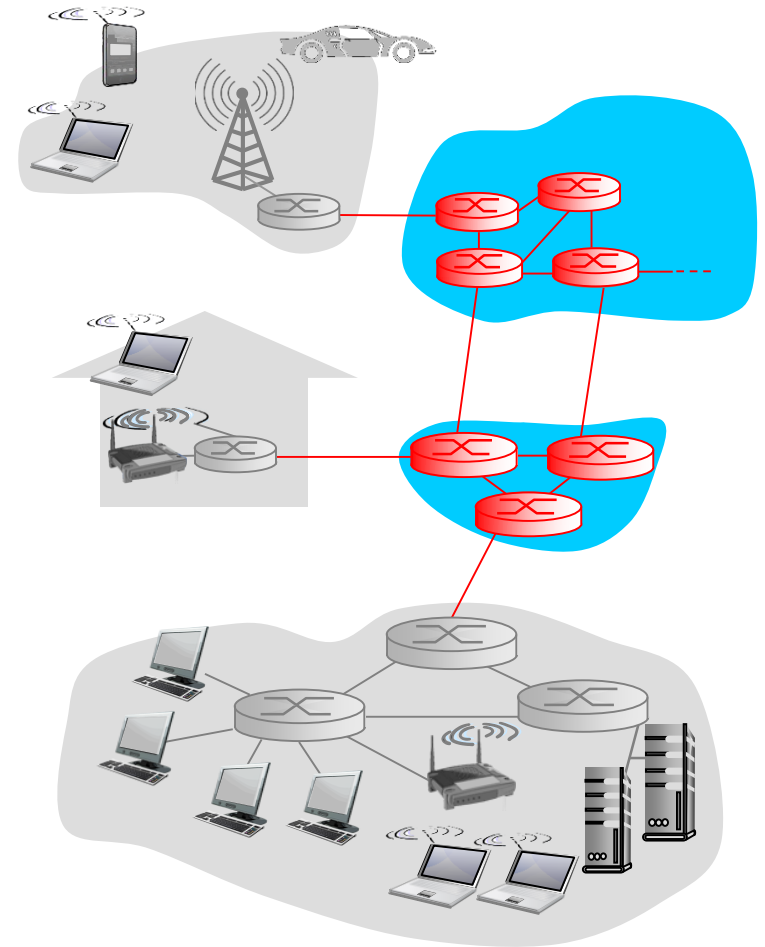
- ❖ signal carried in electromagnetic spectrum
- ❖ no physical “wire”
- ❖ bidirectional
- ❖ propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

## *radio link types:*

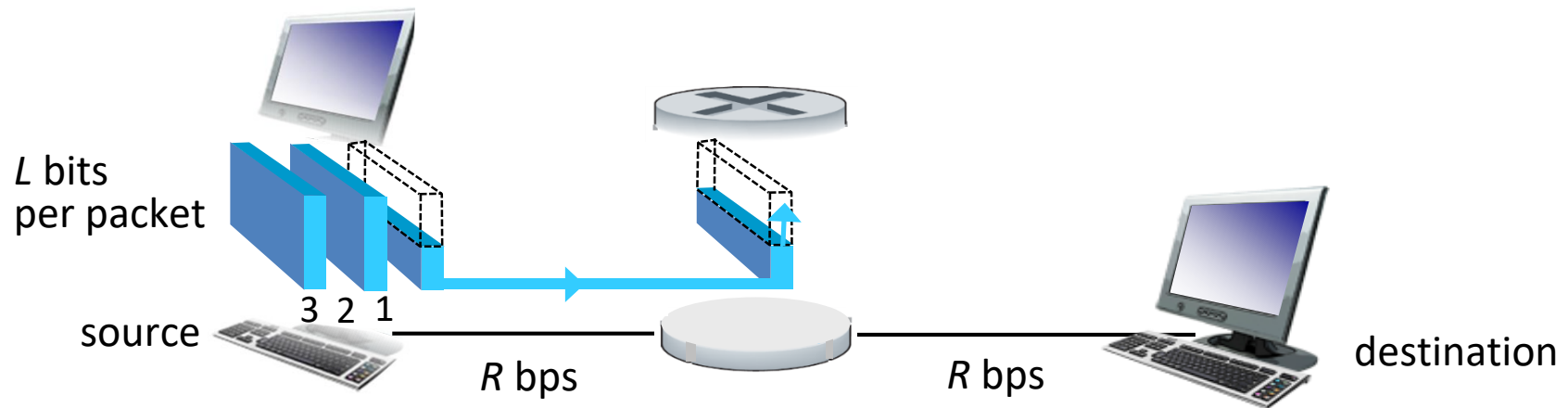
- ❖ **terrestrial microwave**
  - e.g. up to 45 Mbps channels
- ❖ **LAN** (e.g., WiFi)
  - 11 Mbps, 54 Mbps
- ❖ **wide-area** (e.g., cellular)
  - 3G cellular: ~ few Mbps
- ❖ **satellite**
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 280 msec end-end delay
  - geosynchronous versus low altitude

# The network core

- ❖ mesh of interconnected routers
- ❖ 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



# Packet-switching: store-and-forward



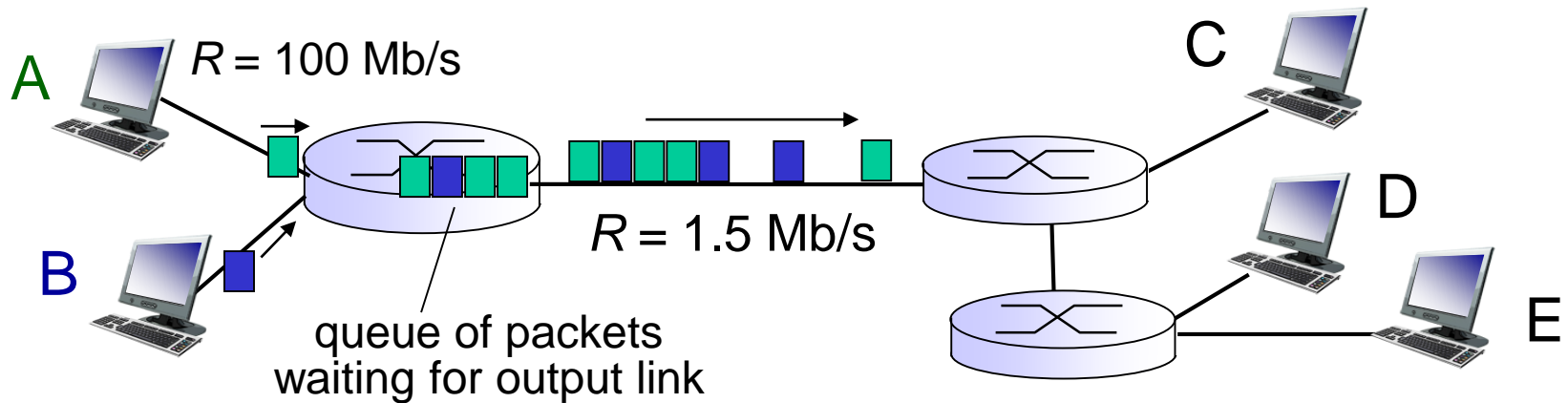
- ❖ takes  $L/R$  seconds to transmit (push out)  $L$ -bit packet into link at  $R$  bps
- ❖ *store and forward*: entire packet must arrive at router before it can be transmitted on next link
- ❖ end-end delay =  $2L/R$  (assuming zero propagation delay)

} more on delay shortly ...

*one-hop numerical example:*

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

# Packet Switching: queueing delay, loss



## queuing and loss:

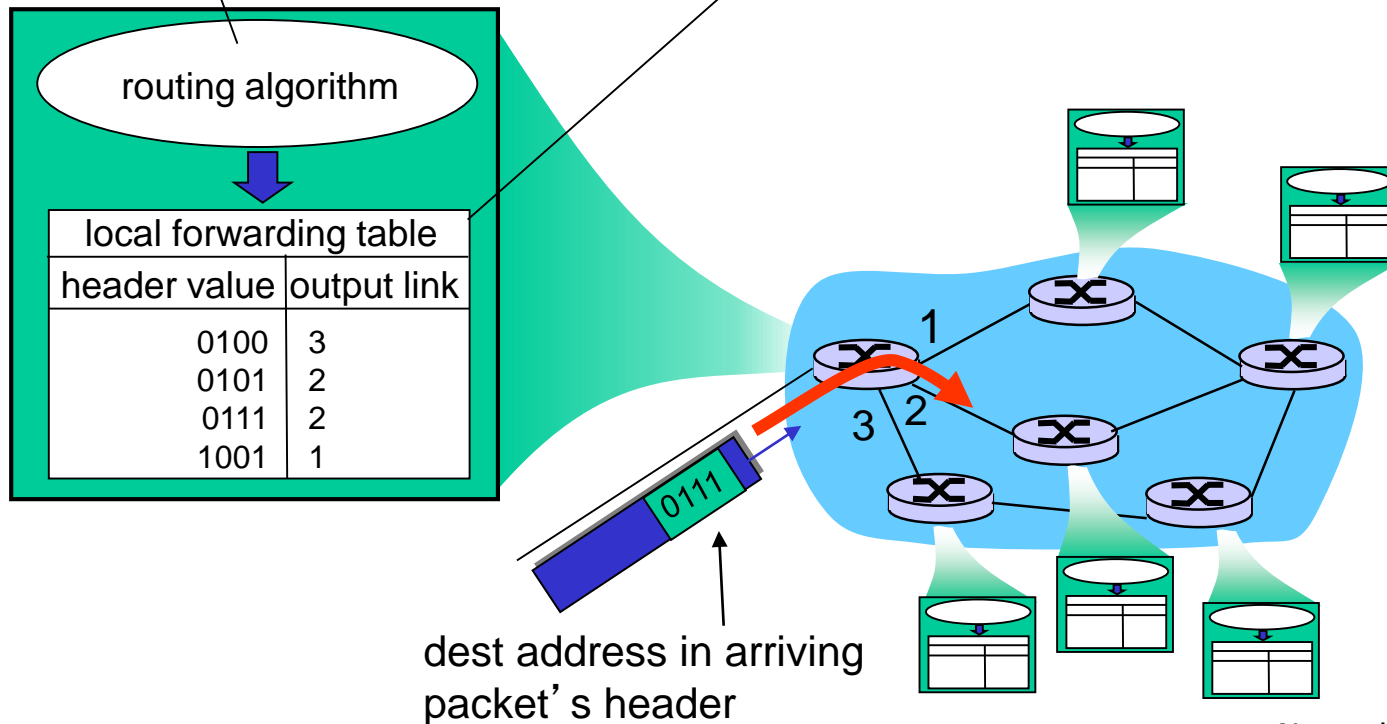
- ❖ If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  - packets will queue, wait to be transmitted on link
  - packets can be dropped (lost) if memory (buffer) fills up

# Two key network-core functions

**routing:** determines source-destination route taken by packets

- *routing algorithms*

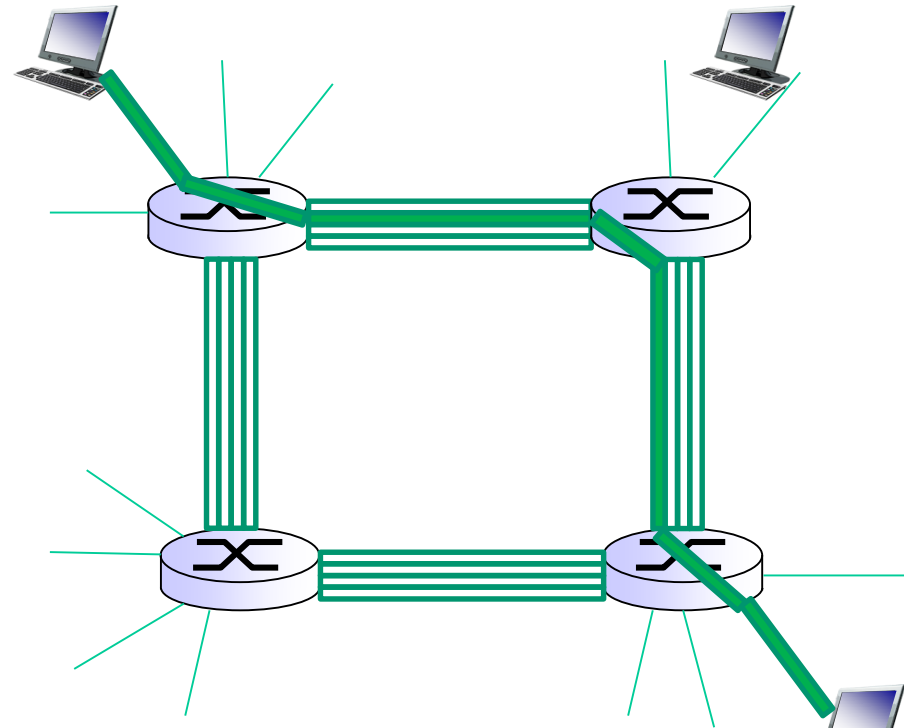
**forwarding:** move packets from router's input to appropriate router output



# Alternative core: circuit switching

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

- ❖ In diagram, each link has four circuits.
  - call gets 2<sup>nd</sup> circuit in top link and 1<sup>st</sup> 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



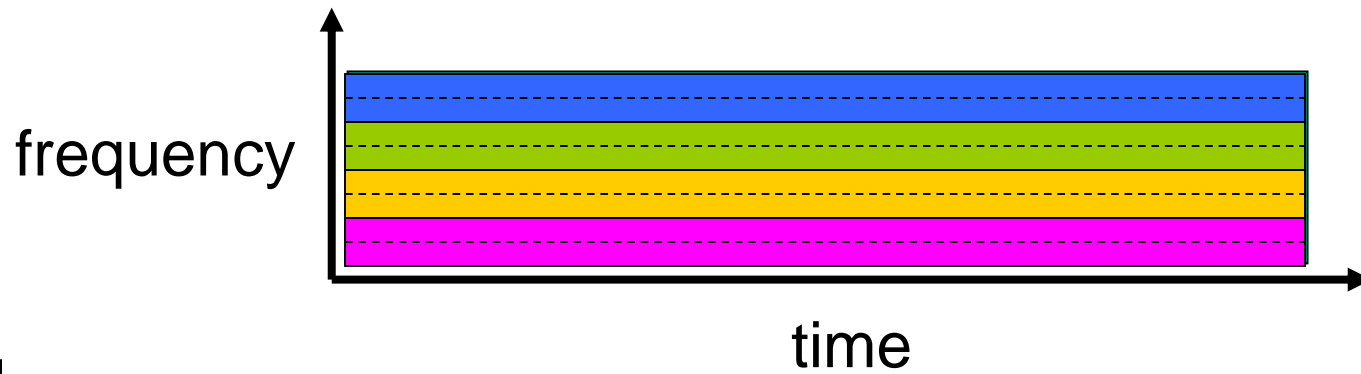


# Circuit switching: FDM versus TDM

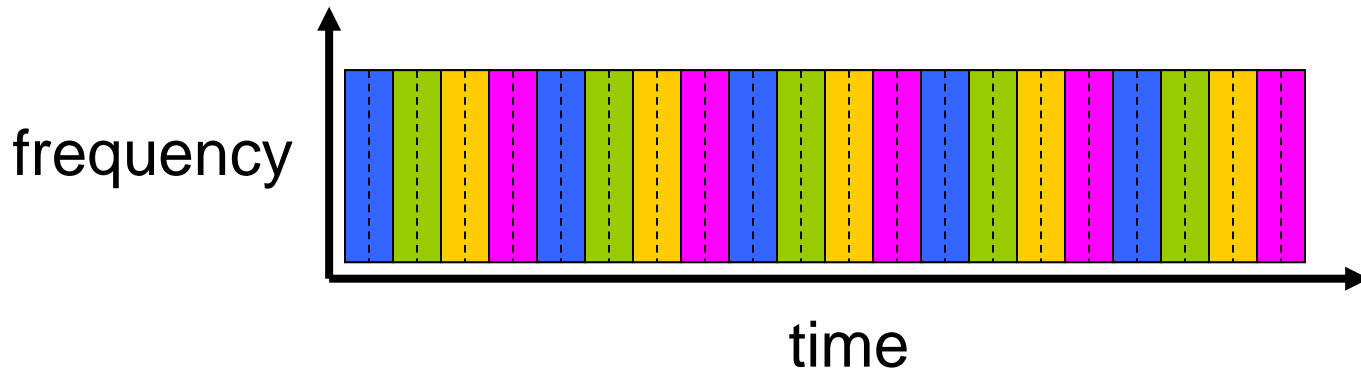
FDM

Example:

4 users



TDM

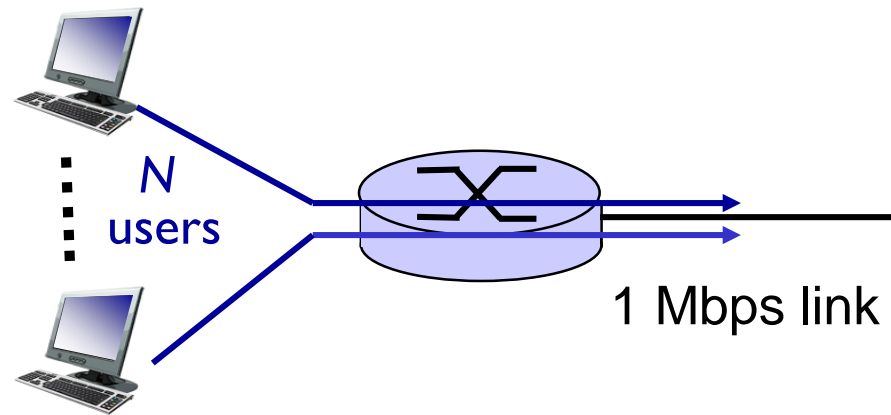


# Packet switching versus circuit switching

*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 \*

# Packet switching versus circuit switching

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 9)

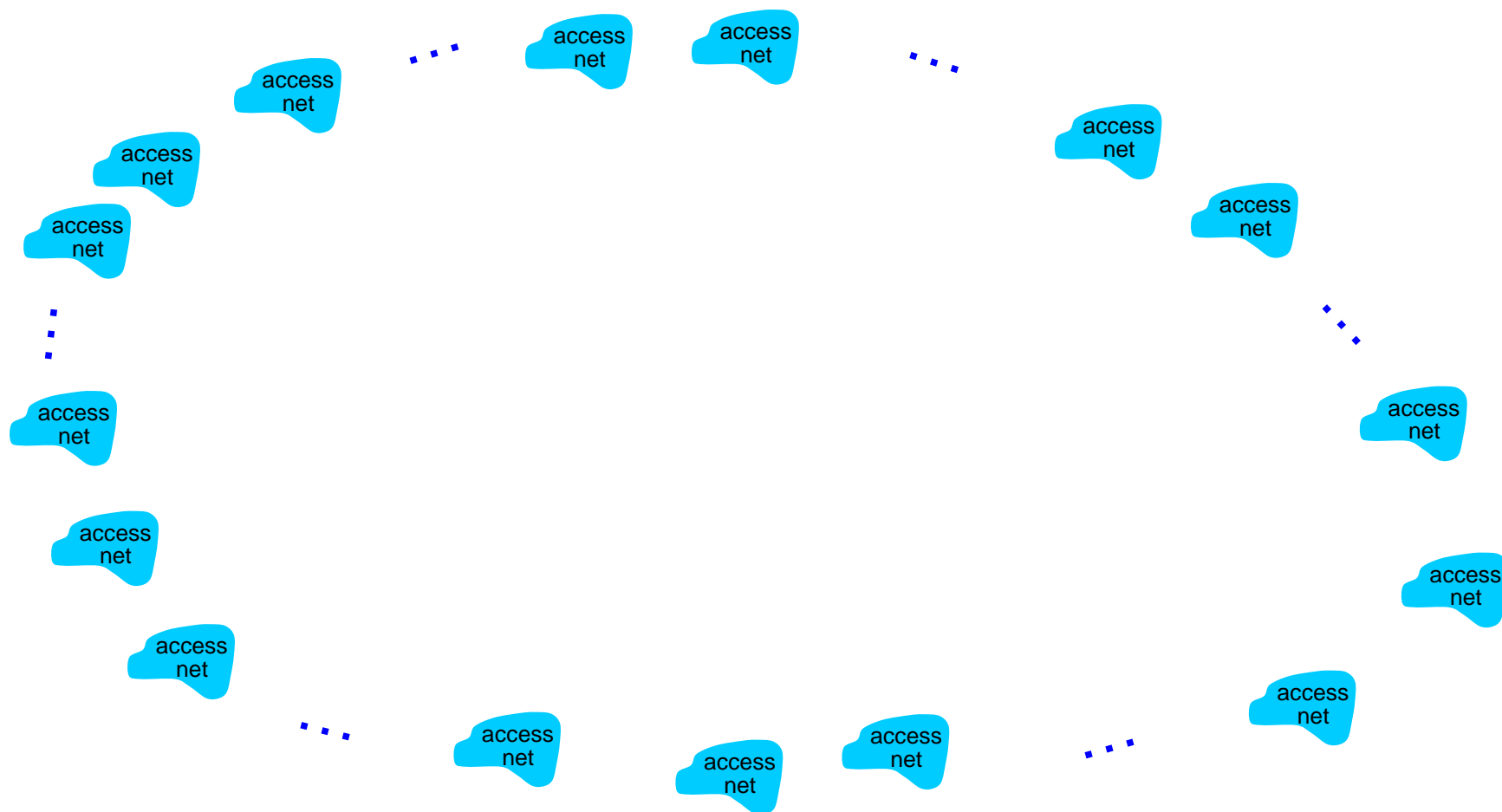
**Q:** human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

# Internet structure: network of networks

- ❖ End systems connect to Internet via **access ISPs** (Internet Service Providers)
  - Residential, company and university ISPs
- ❖ Access ISPs in turn must be interconnected.
  - ❖ So that any two hosts can send packets to each other
- ❖ Resulting network of networks is very complex
  - ❖ Evolution was driven by **economics** and **national policies**
- ❖ Let's take a stepwise approach to describe current Internet structure

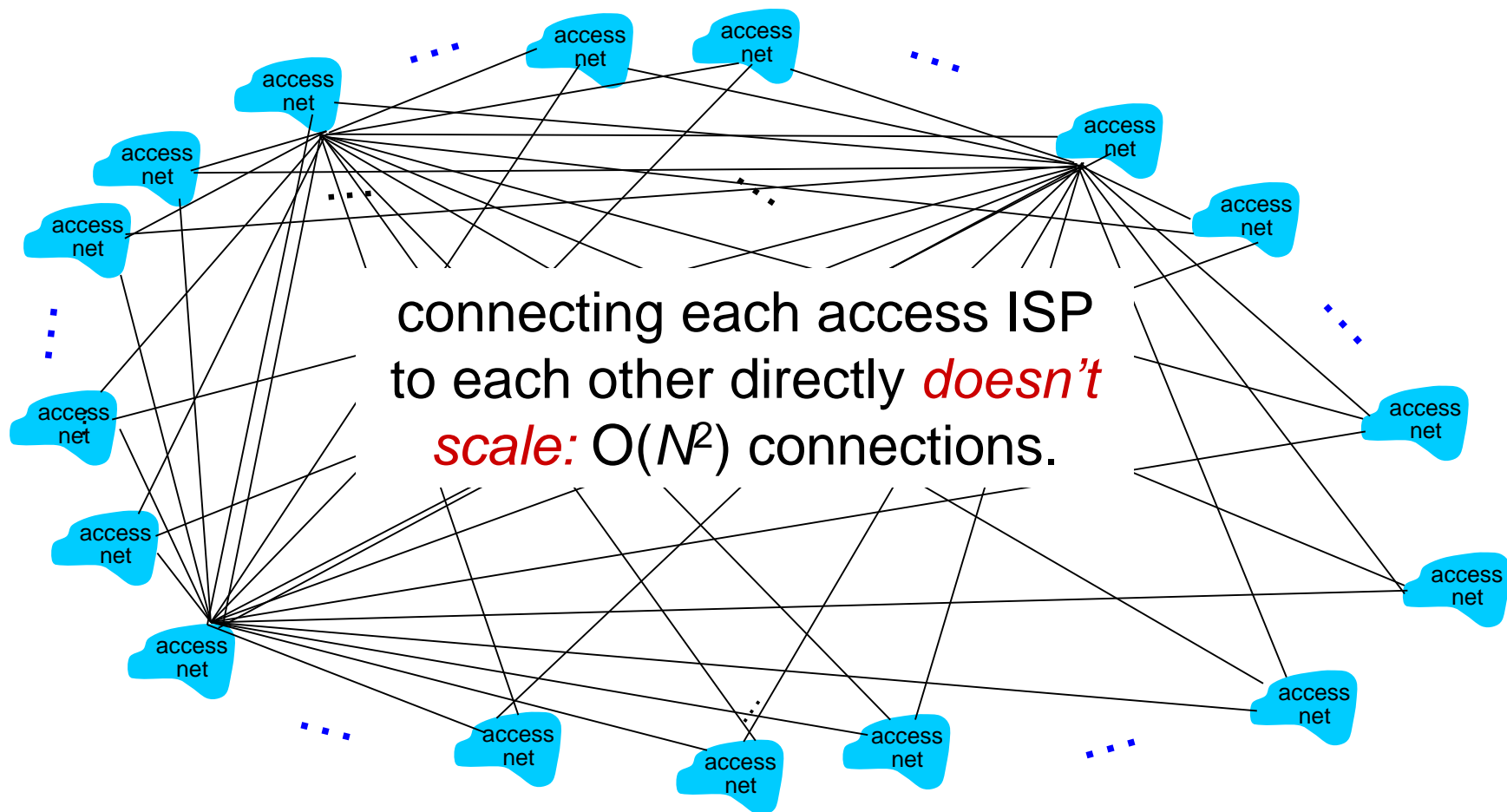
# Internet structure: network of networks

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



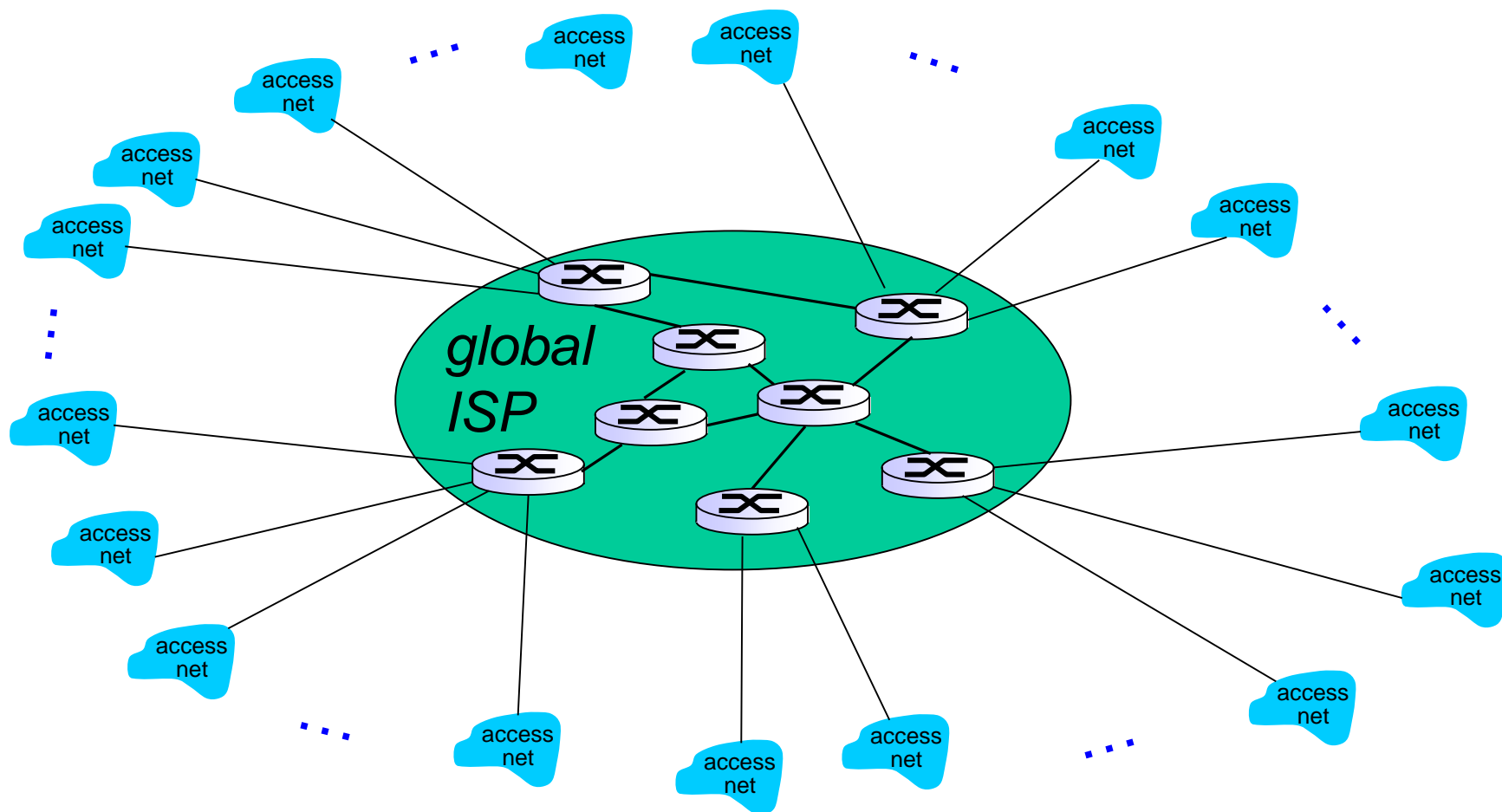
# Internet structure: network of networks

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



# Internet structure: network of networks

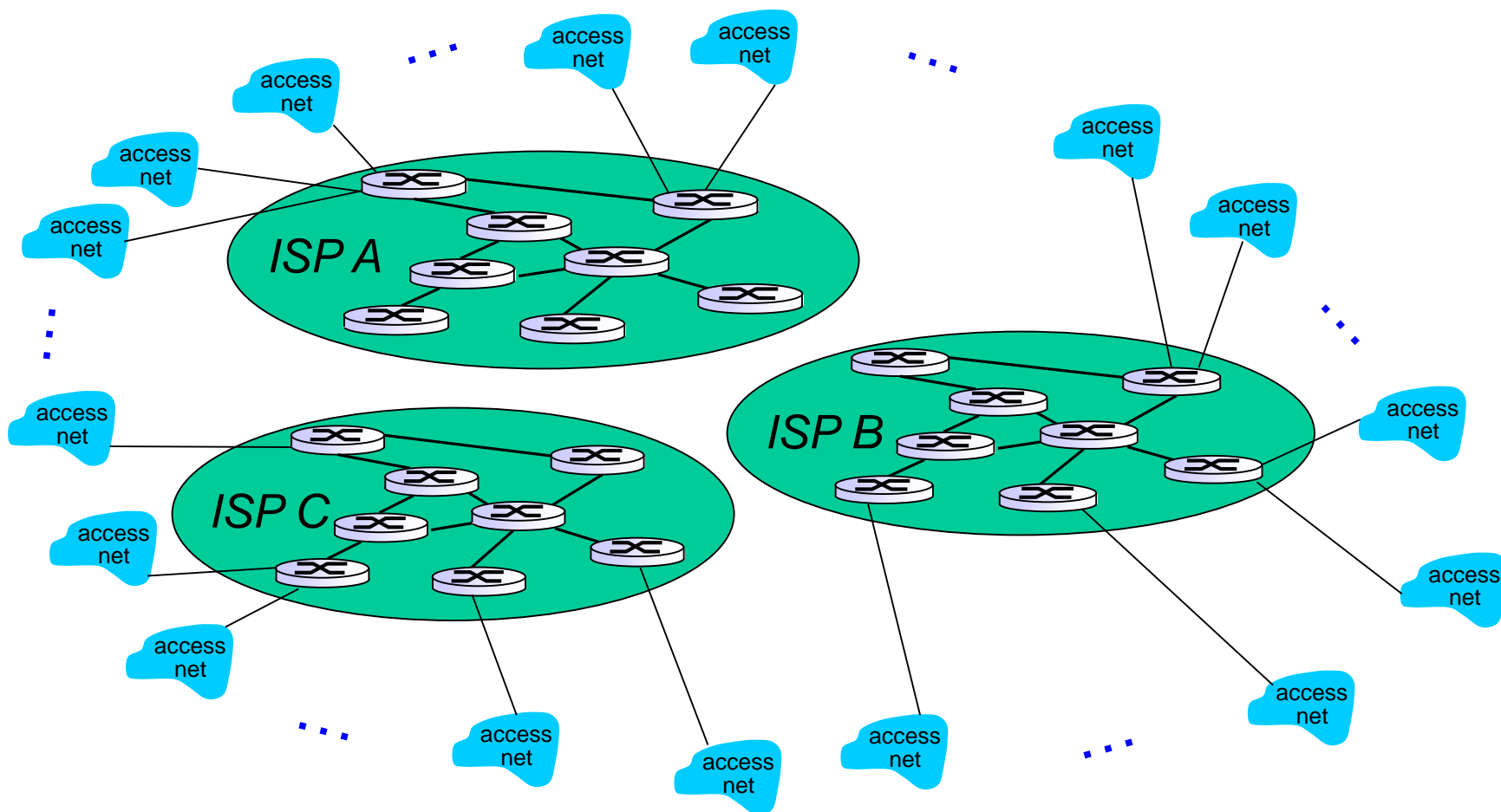
*Option: connect each access ISP to a global transit ISP? **Customer** and **provider** ISPs have economic agreement.*



# Internet structure: network of networks

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

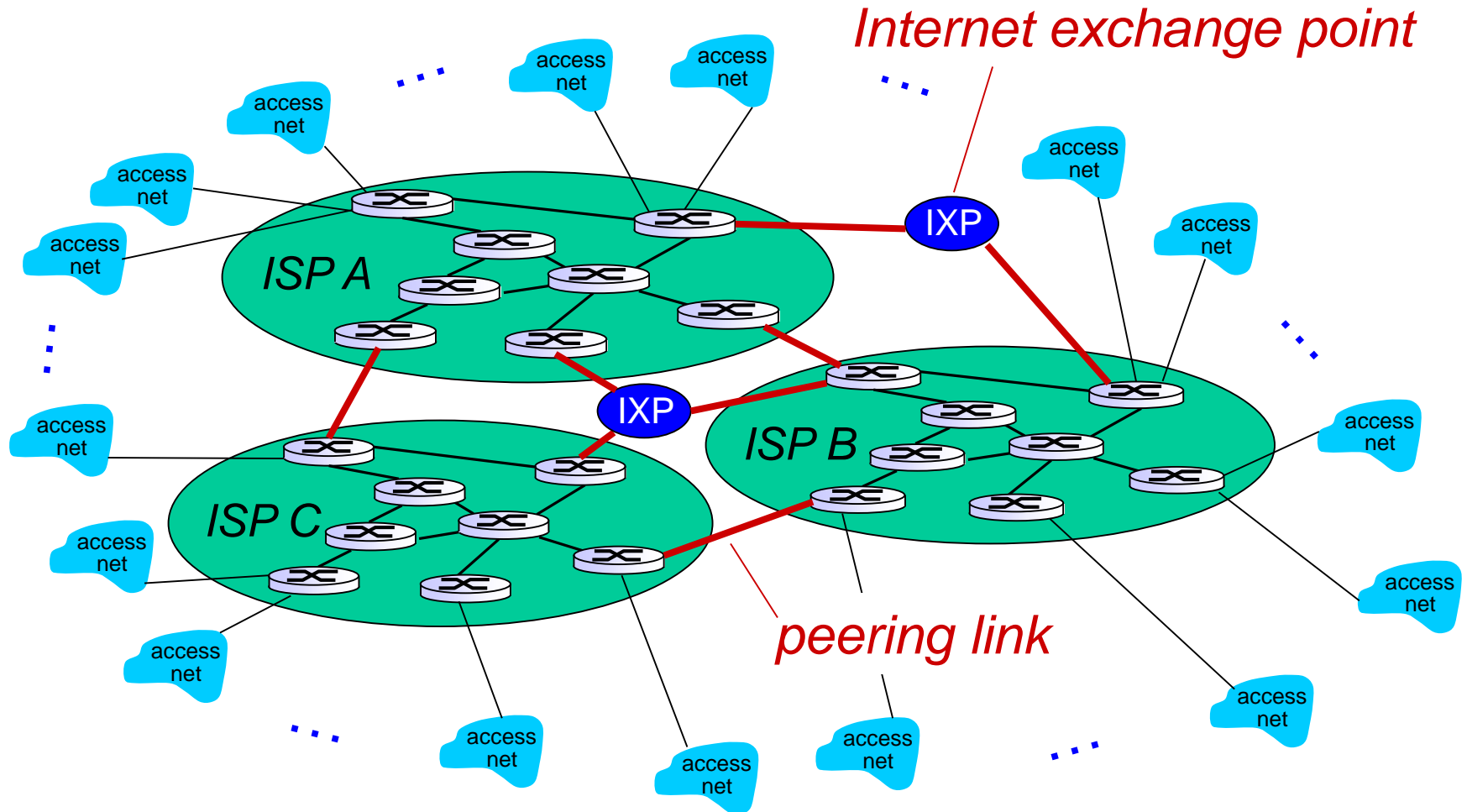
....





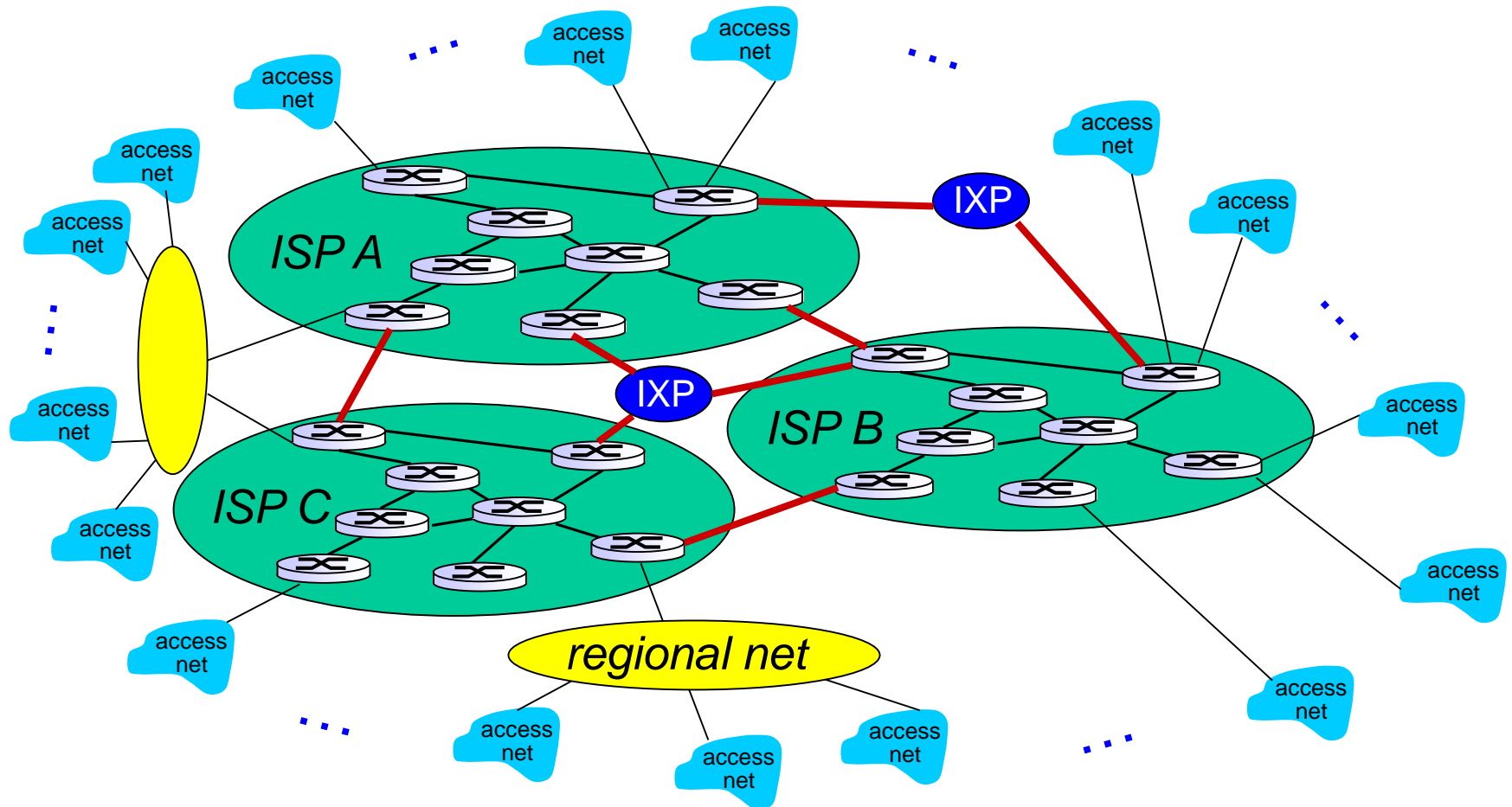
# Internet structure: network of networks

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



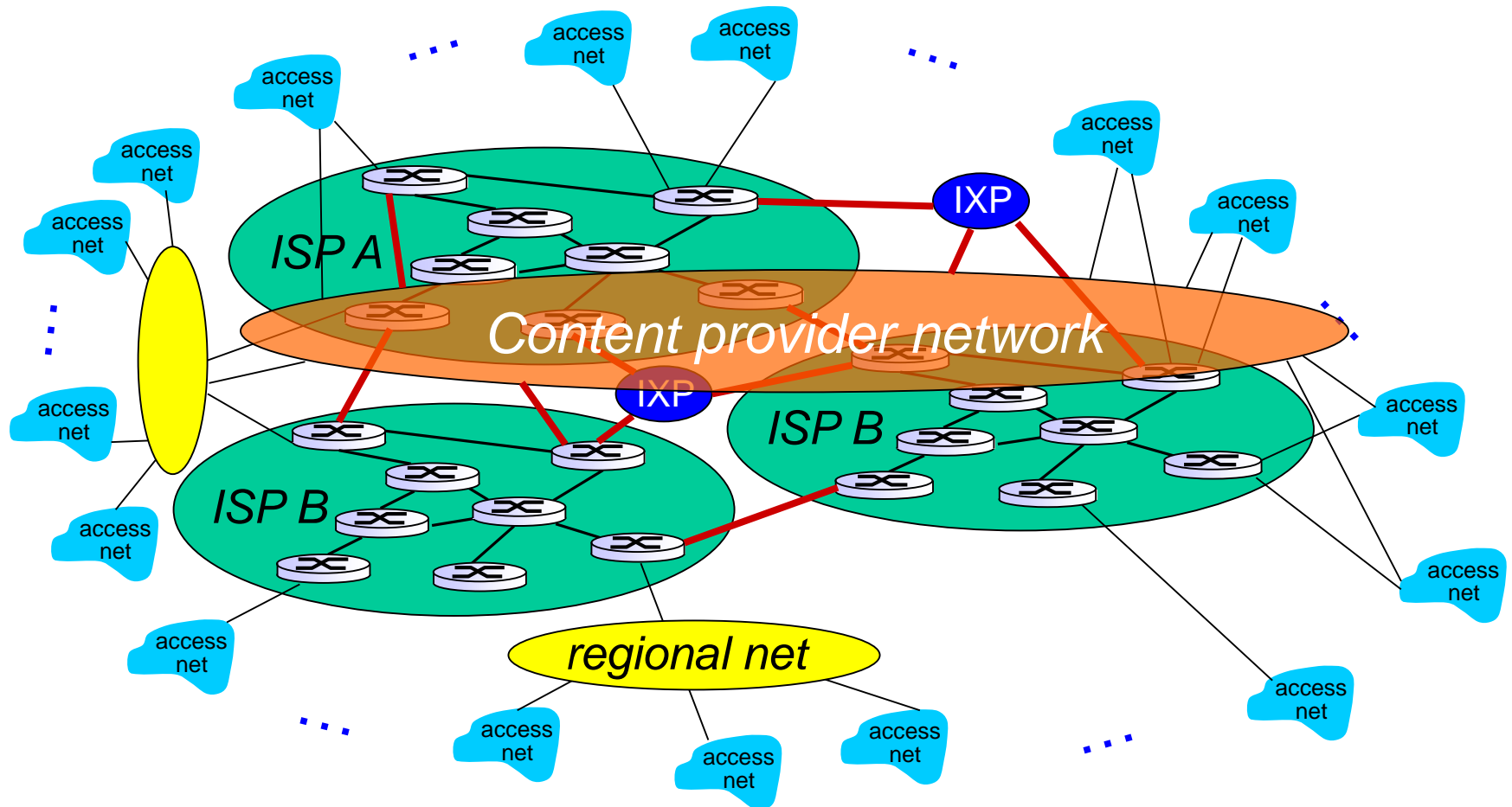
# Internet structure: network of networks

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

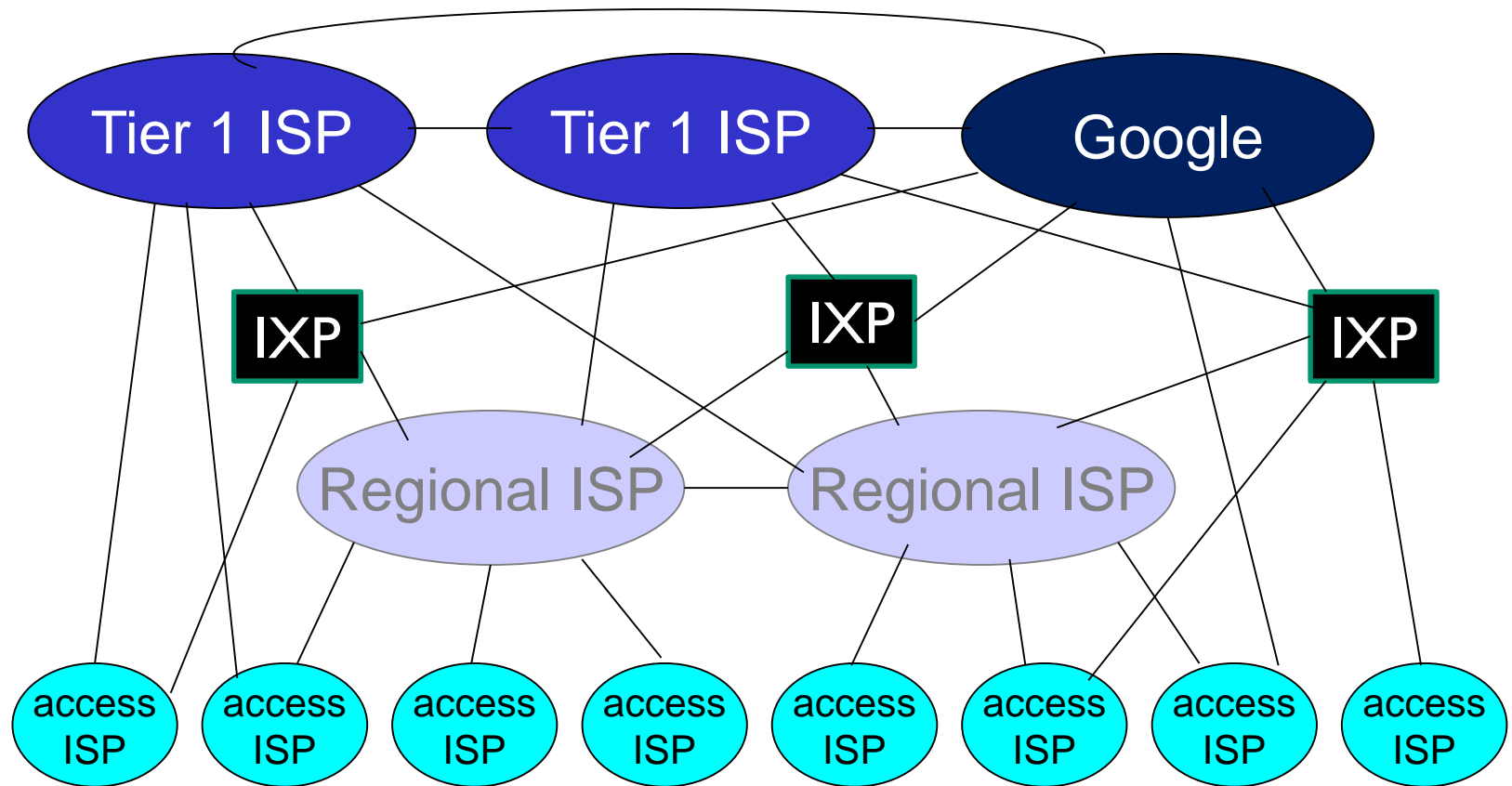


# Internet structure: network of networks

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

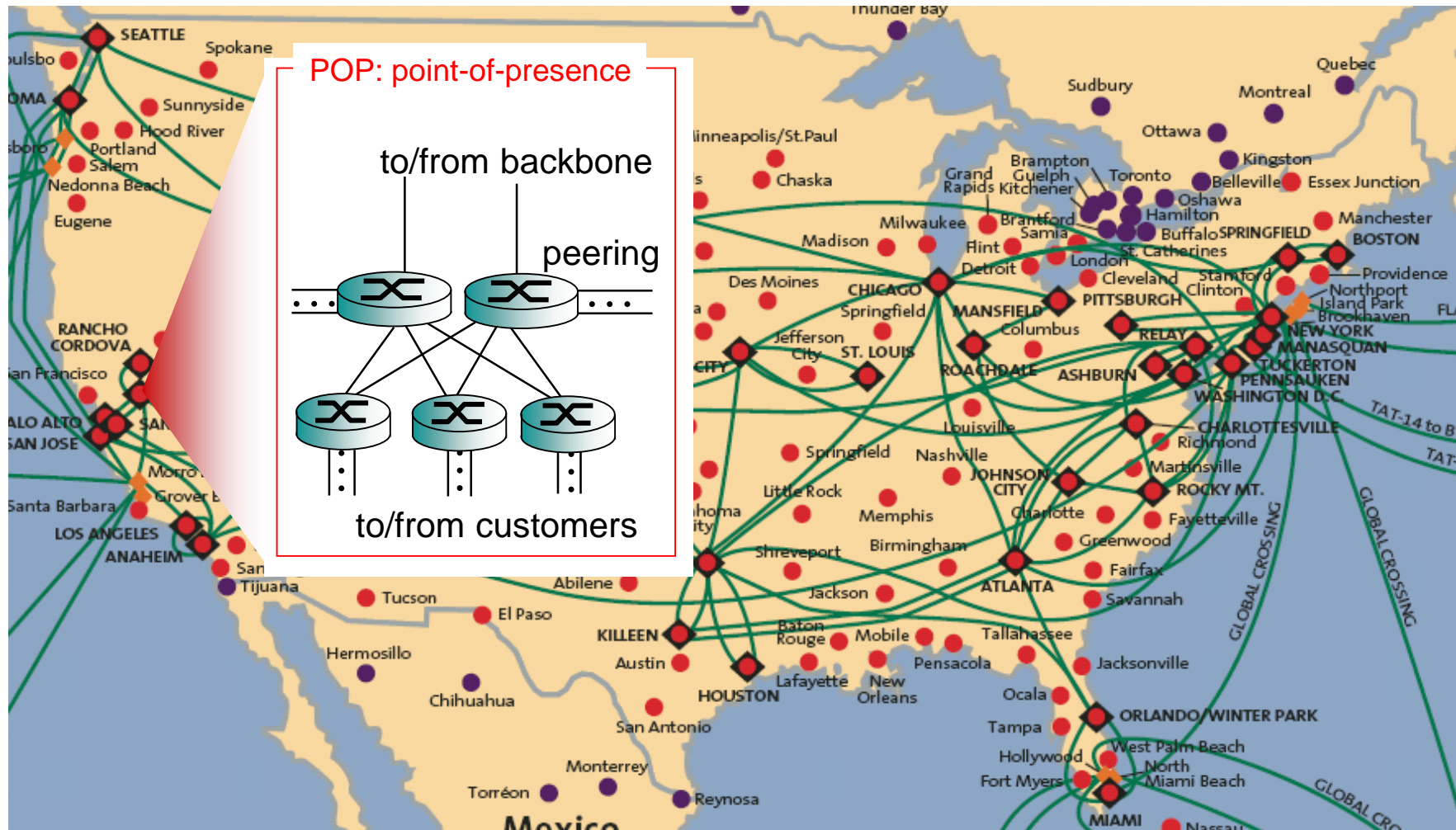


# Internet structure: network of networks



- ❖ 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 its data centers to Internet, often bypassing tier-1, regional ISPs

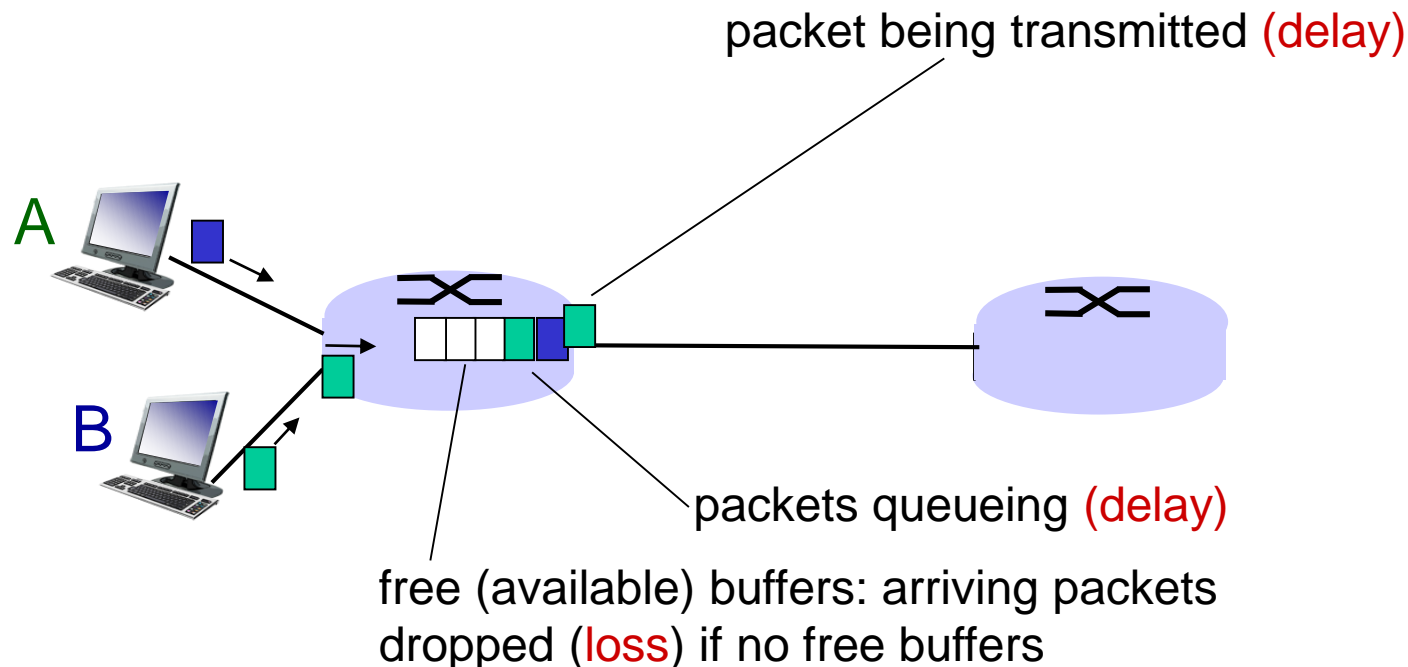
# Tier-I ISP: e.g., Sprint



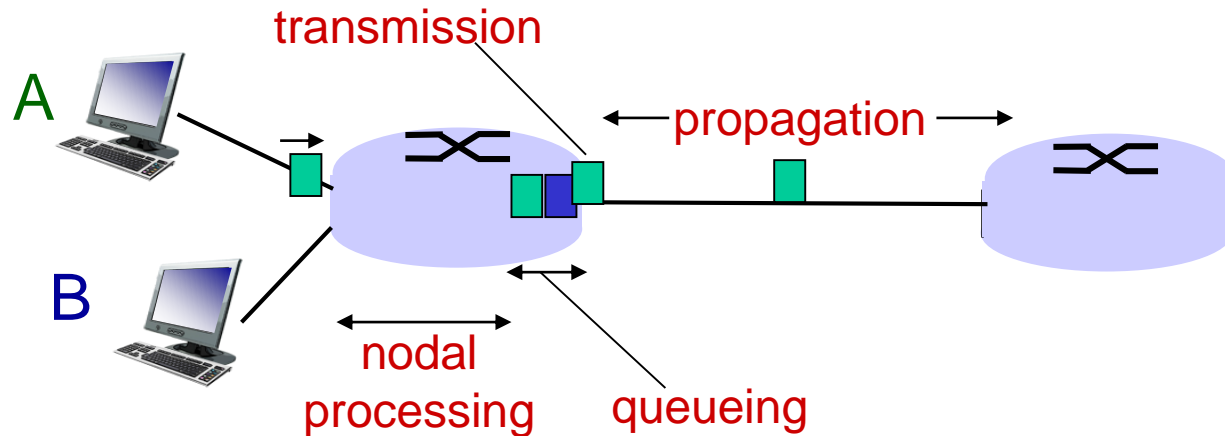
# How do loss and delay occur?

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



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

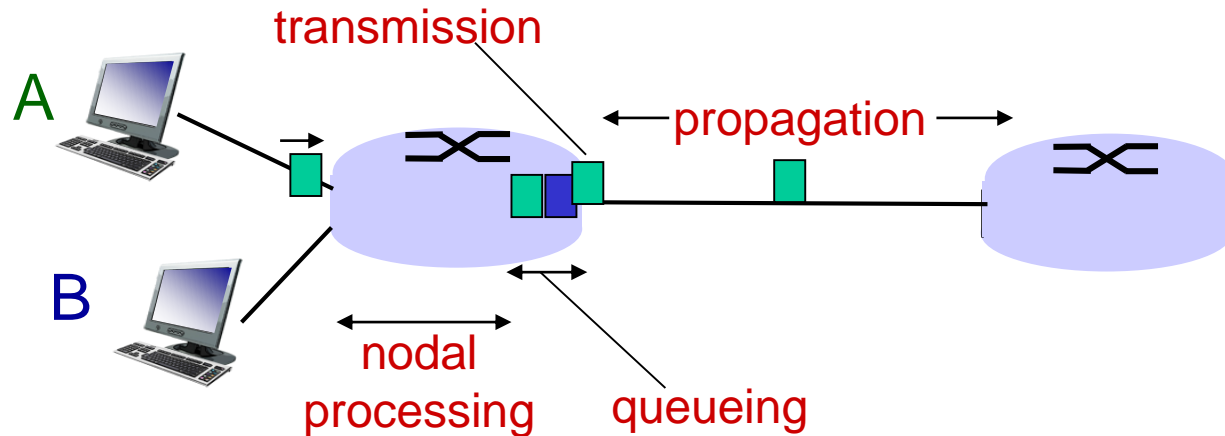
## $d_{\text{proc}}$ : nodal processing

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

## $d_{\text{queue}}$ : queueing delay

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

# Four sources of packet delay



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

**$d_{\text{trans}}$ : transmission delay:**

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

**$d_{\text{prop}}$ : propagation delay:**

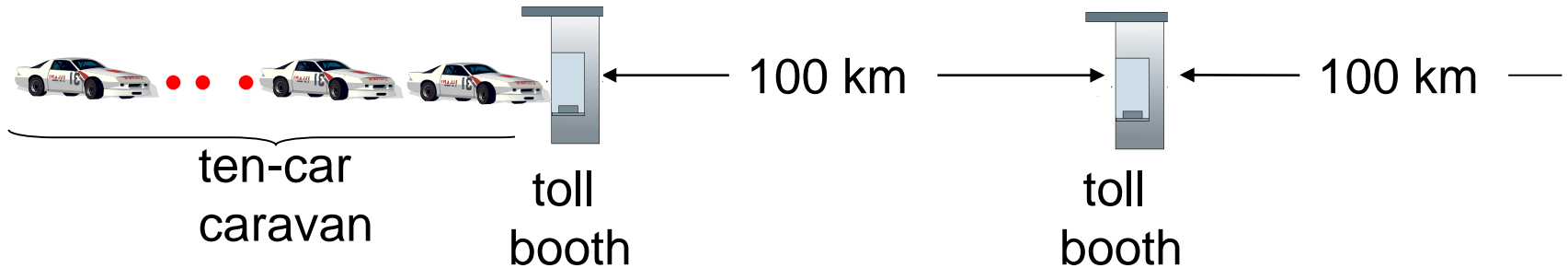
- $d$ : length of physical link
- $s$ : propagation speed in medium ( $\sim 2 \times 10^8$  m/sec)
- $d_{\text{prop}} = d/s$

$d_{\text{trans}}$  and  $d_{\text{prop}}$   
very different

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

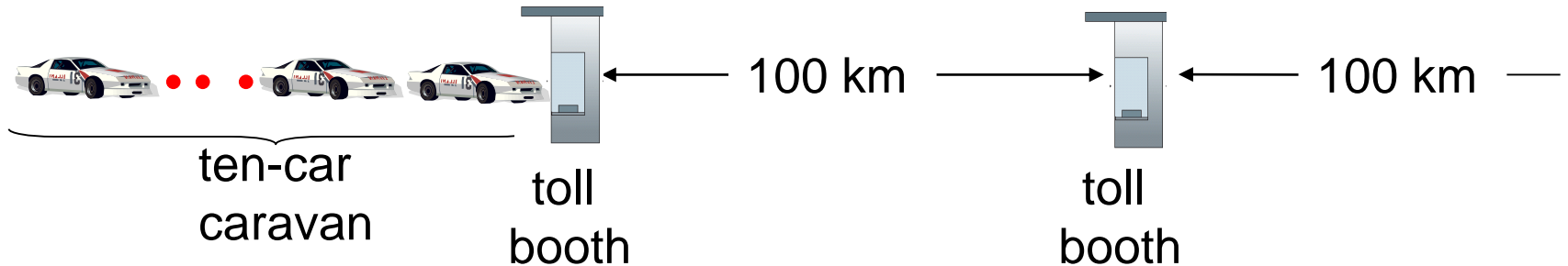


# Caravan analogy



- ❖ 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 =  $12 \times 10 = 120$  sec
- time for last car to propagate from 1st to 2nd toll booth:  $100 \text{ km} / (100 \text{ km/hr}) = 1$  hr
- A: 62 minutes

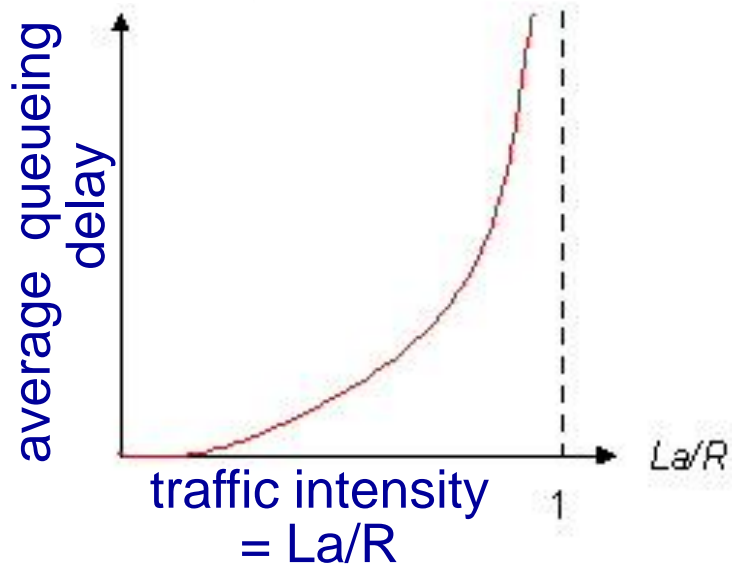
# Caravan analogy (more)



- ❖ 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.

# Queueing delay (revisited)

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



- ❖  $La/R \sim 0$ : avg. queueing delay small
- ❖  $La/R \rightarrow 1$ : avg. queueing delay large
- ❖  $La/R > 1$ : more “work” arriving than can be serviced, average delay infinite!



$La/R \sim 0$

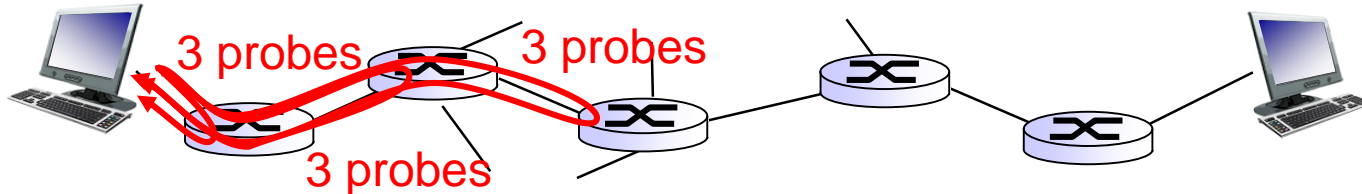


$La/R \rightarrow 1$

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

# “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

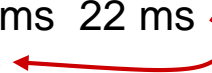
**traceroute:** gaia.cs.umass.edu to www.eurecom.fr

3 delay measurements from  
gaia.cs.umass.edu to cs-gw.cs.umass.edu




1 cs-gw (128.119.240.254) 1 ms 1 ms 2 ms  
2 border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145) 1 ms 1 ms 2 ms  
3 cht-vbns.gw.umass.edu (128.119.3.130) 6 ms 5 ms 5 ms  
4 jn1-at1-0-0-19.wor.vbns.net (204.147.132.129) 16 ms 11 ms 13 ms  
5 jn1-so7-0-0-0.wae.vbns.net (204.147.136.136) 21 ms 18 ms 18 ms  
6 abilene-vbns.abilene.ucaid.edu (198.32.11.9) 22 ms 18 ms 22 ms  
7 nycm-wash.abilene.ucaid.edu (198.32.8.46) 22 ms 22 ms 22 ms  
8 62.40.103.253 (62.40.103.253) 104 ms 109 ms 106 ms  
9 de2-1.de1.de.geant.net (62.40.96.129) 109 ms 102 ms 104 ms  
10 de.fr1.fr.geant.net (62.40.96.50) 113 ms 121 ms 114 ms  
11 renater-gw.fr1.fr.geant.net (62.40.103.54) 112 ms 114 ms 112 ms  
12 nio-n2.cssi.renater.fr (193.51.206.13) 111 ms 114 ms 116 ms  
13 nice.cssi.renater.fr (195.220.98.102) 123 ms 125 ms 124 ms  
14 r3t2-nice.cssi.renater.fr (195.220.98.110) 126 ms 126 ms 124 ms  
15 eurecom-valbonne.r3t2.ft.net (193.48.50.54) 135 ms 128 ms 133 ms  
16 194.214.211.25 (194.214.211.25) 126 ms 128 ms 126 ms  
17 \* \* \*  
18 \* \* \*  
19 fantasia.eurecom.fr (193.55.113.142) 132 ms 128 ms 136 ms

trans-oceanic link



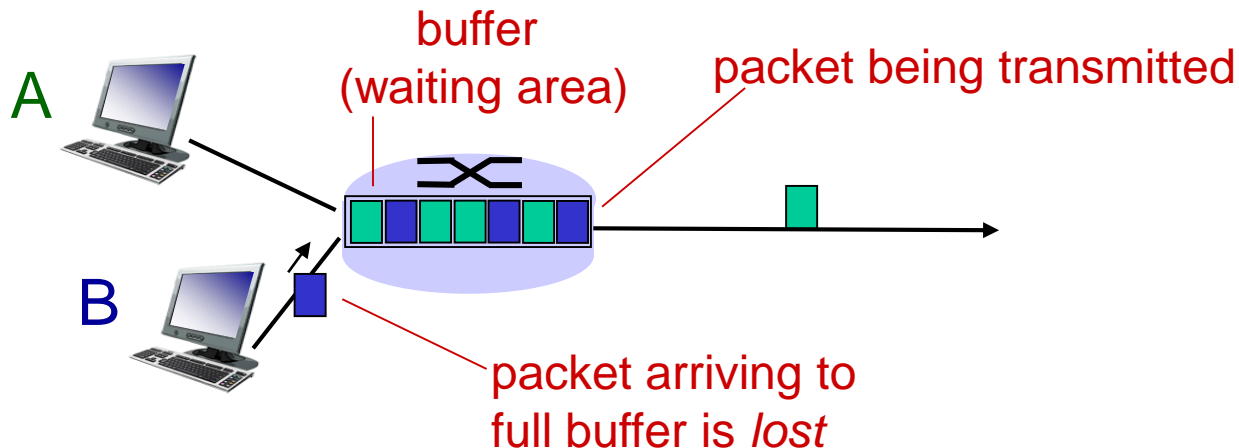
\* means no response (probe lost, router not replying)



\* Do some traceroutes from exotic countries at [www.traceroute.org](http://www.traceroute.org)

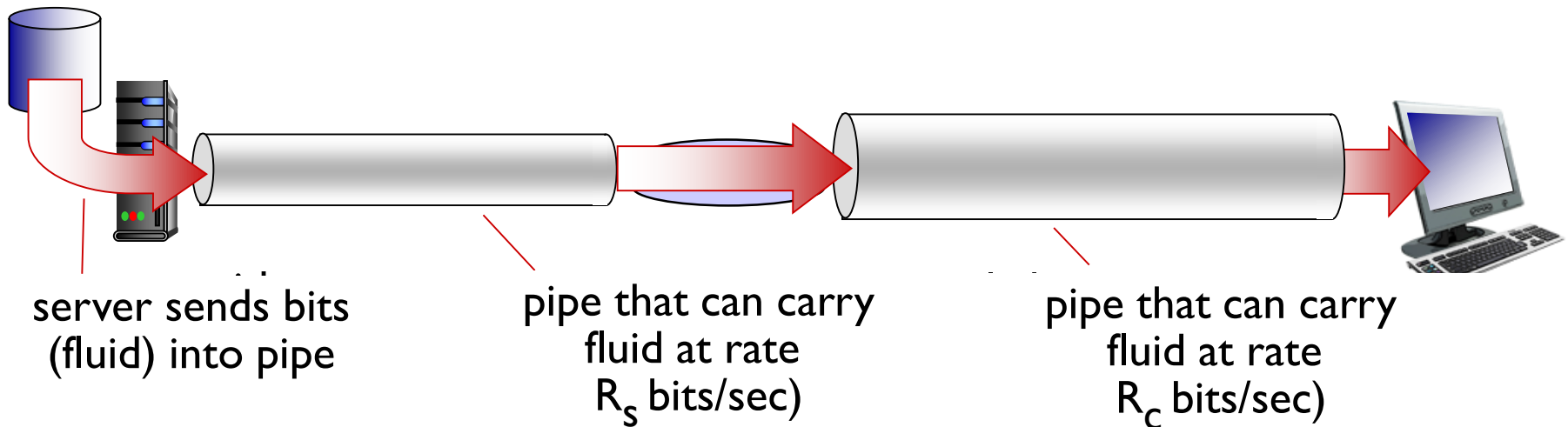
# Packet loss

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



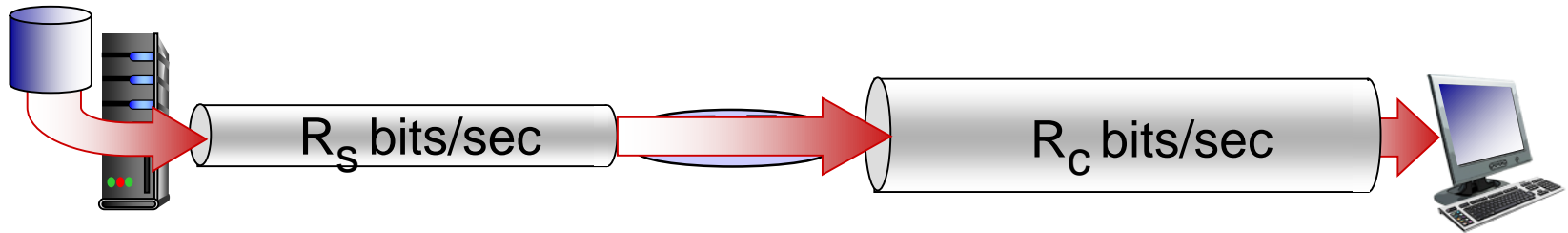
# Throughput

- ❖ *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

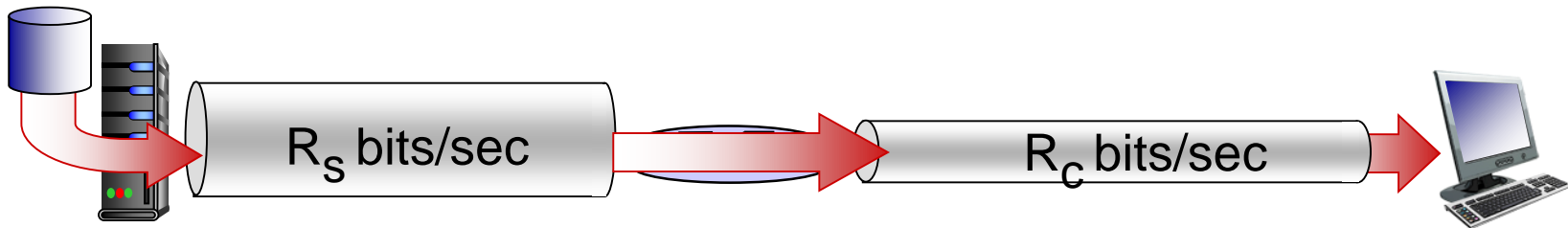


# Throughput (more)

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



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



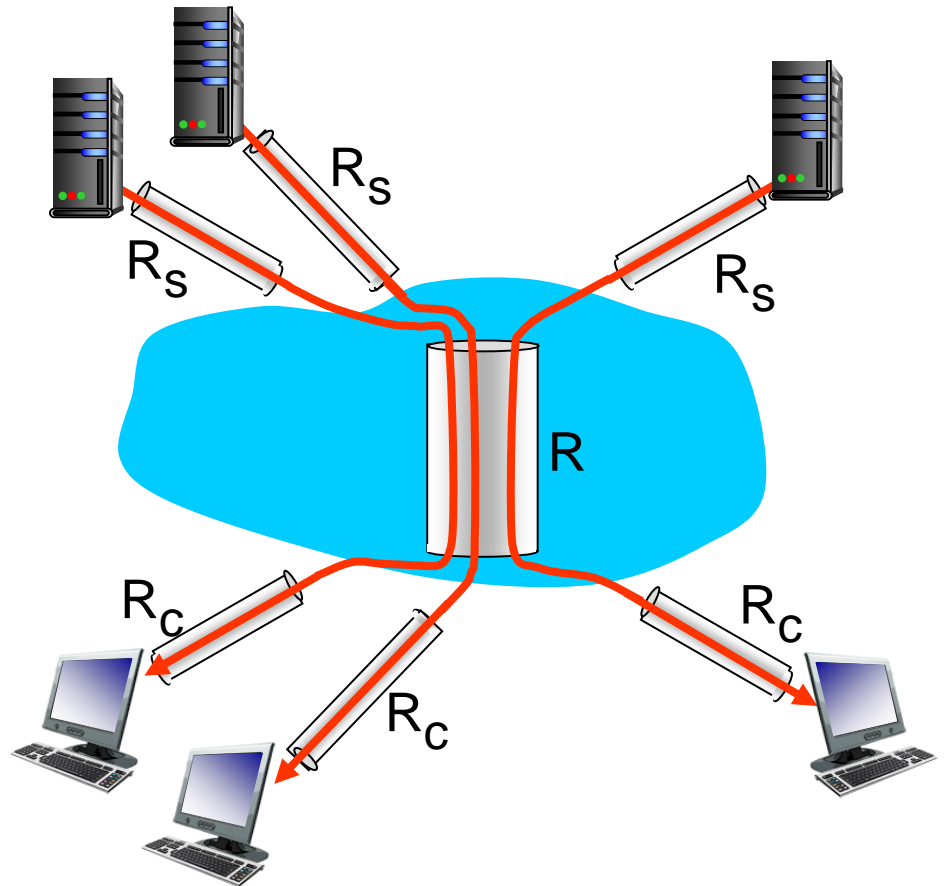
*bottleneck link*

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

# Protocol “layers”

*Networks are complex,  
with many “pieces”:*

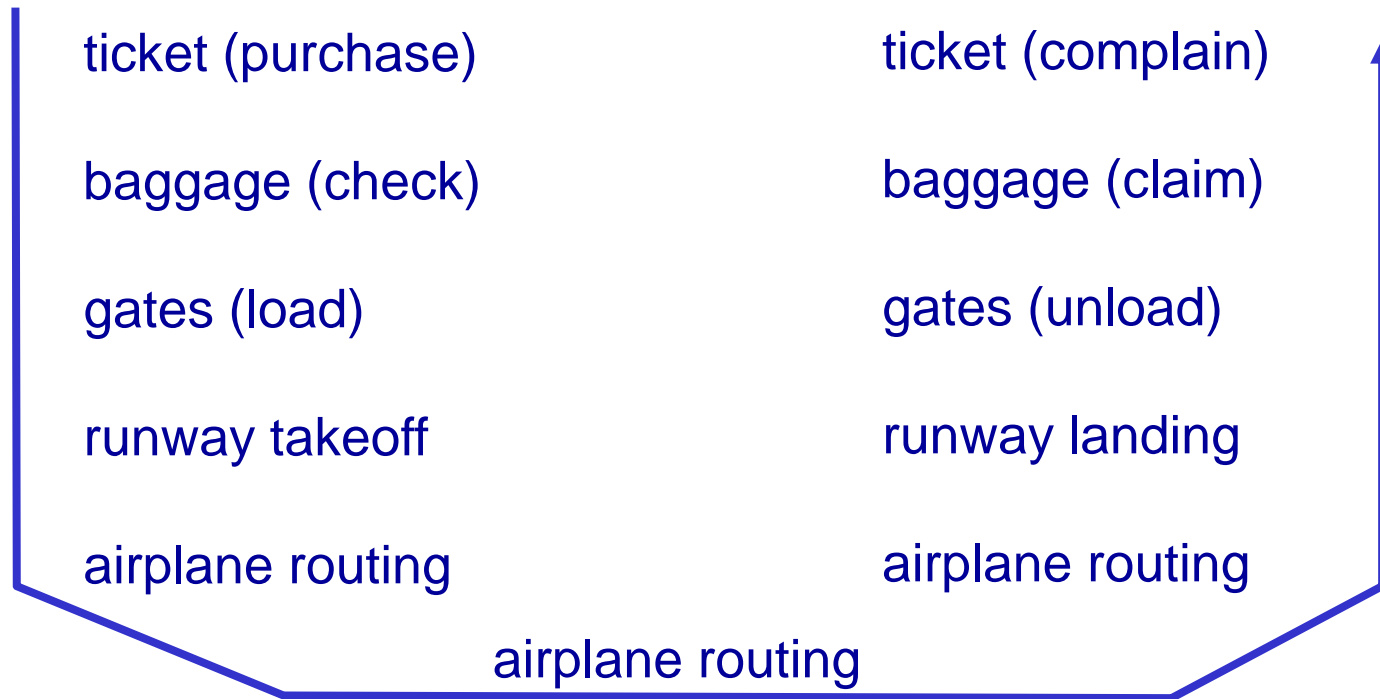
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

*Question:*

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

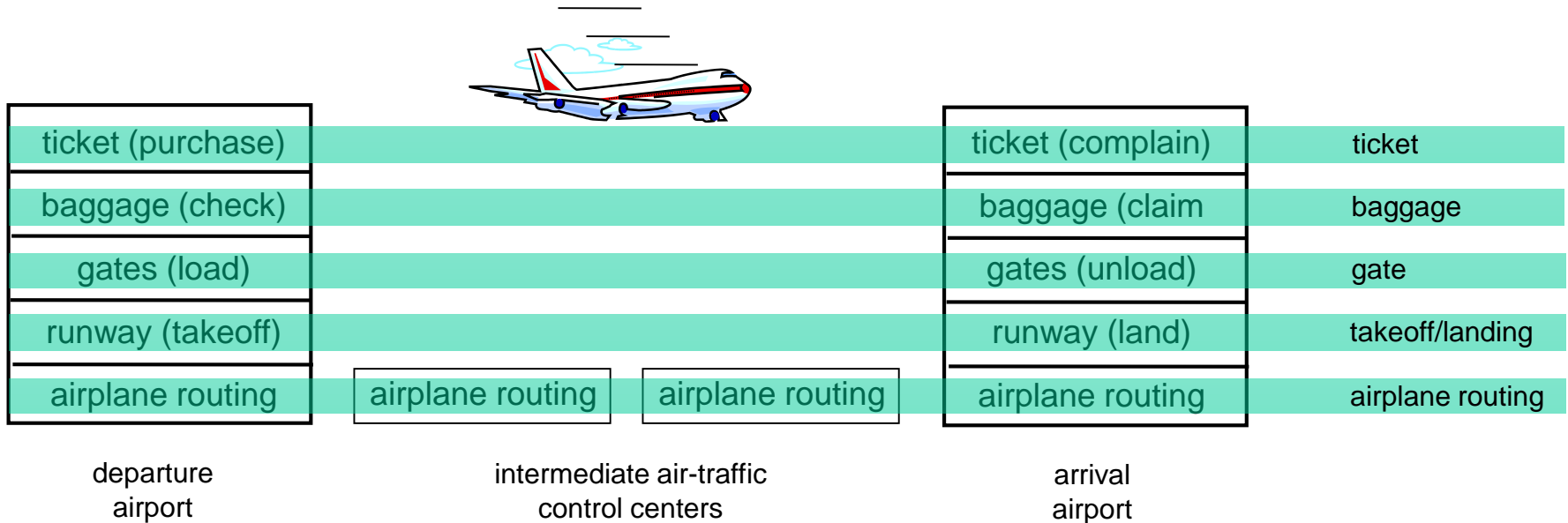
.... or at least our  
discussion of networks?

# Organization of air travel



❖ a series of steps

# Layering of airline functionality



**layers:** each layer implements a service

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

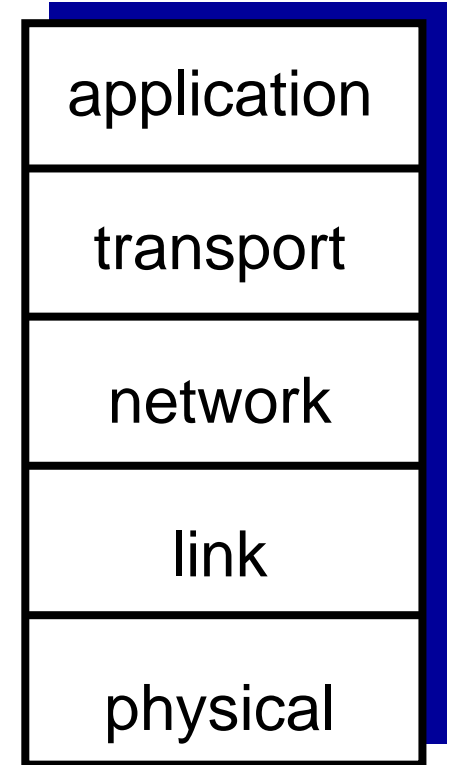
# Why layering?

dealing with complex systems:

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

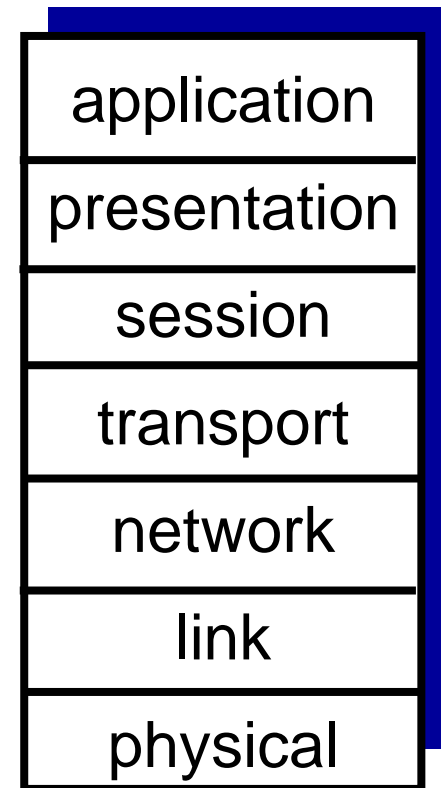
# Internet protocol stack

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

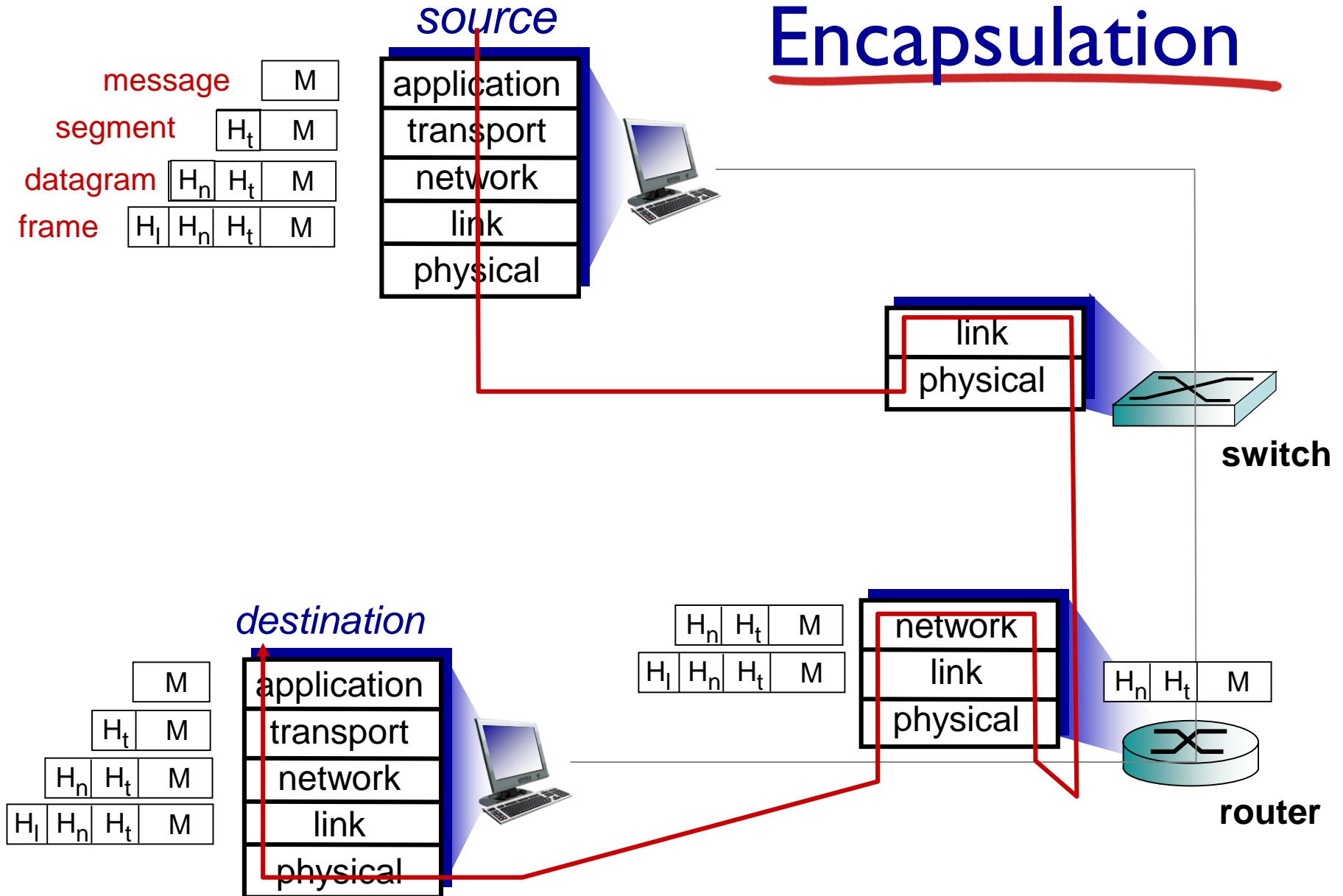


# ISO/OSI reference model

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



# Encapsulation





# Network security

## ❖ field of network security:

- how bad guys can attack computer networks
- how we can defend networks against attacks
- how to design architectures that are immune to attacks

## ❖ Internet not originally designed with (much) security in mind

- *original vision*: “a group of mutually trusting users attached to a transparent network” 😊
- Internet protocol designers playing “catch-up”
- security considerations in all layers!

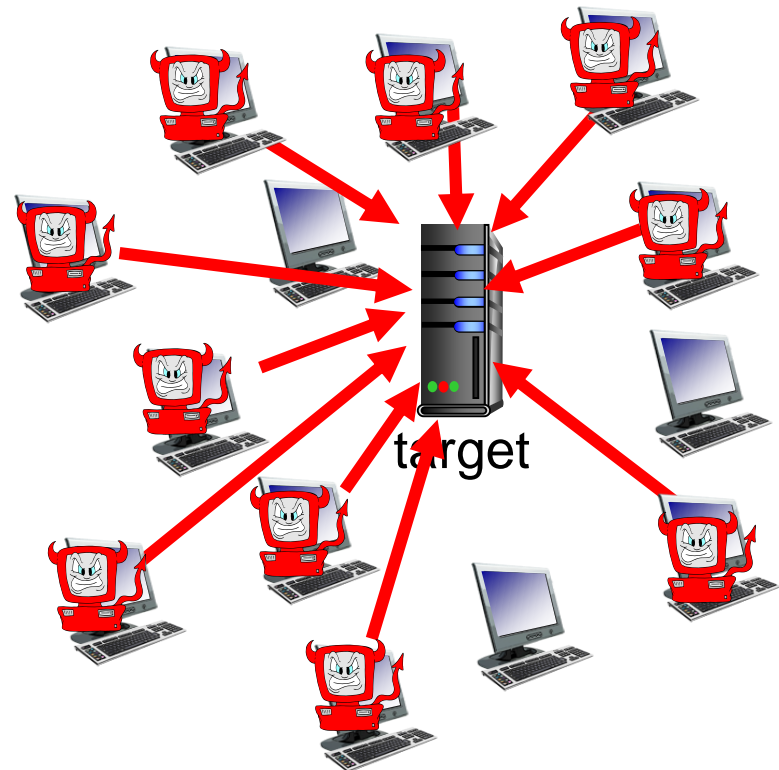
# Bad guys: put malware into hosts via Internet

- ❖ malware can get in host from:
  - *virus*: self-replicating infection by receiving/executing object (e.g., e-mail attachment)
  - *worm*: self-replicating infection by passively receiving object that gets itself executed
- ❖ **spyware malware** can record keystrokes, web sites visited, upload info to collection site
- ❖ infected host can be enrolled in **botnet**, used for spam. DDoS attacks

# Bad guys: attack server, network infrastructure

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

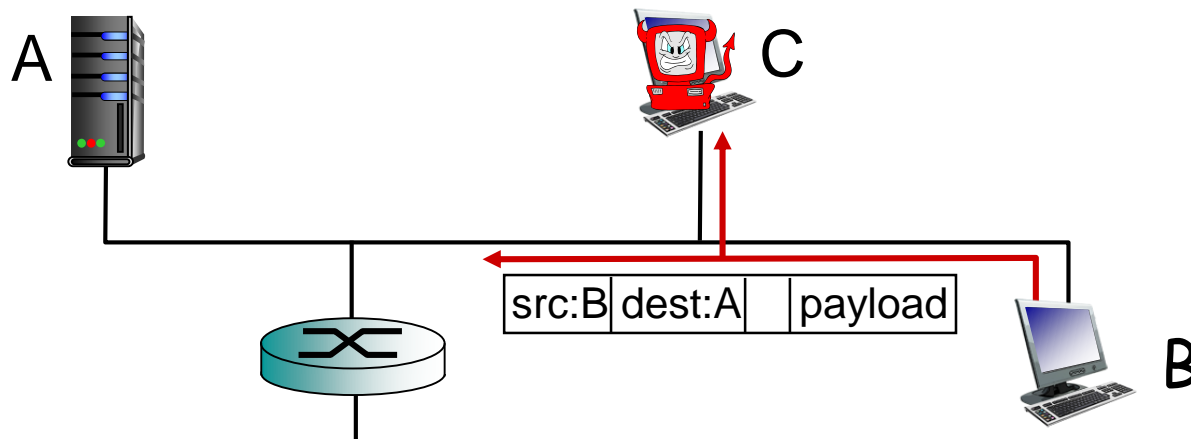
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



# Bad guys can sniff packets

## *packet “sniffing”:*

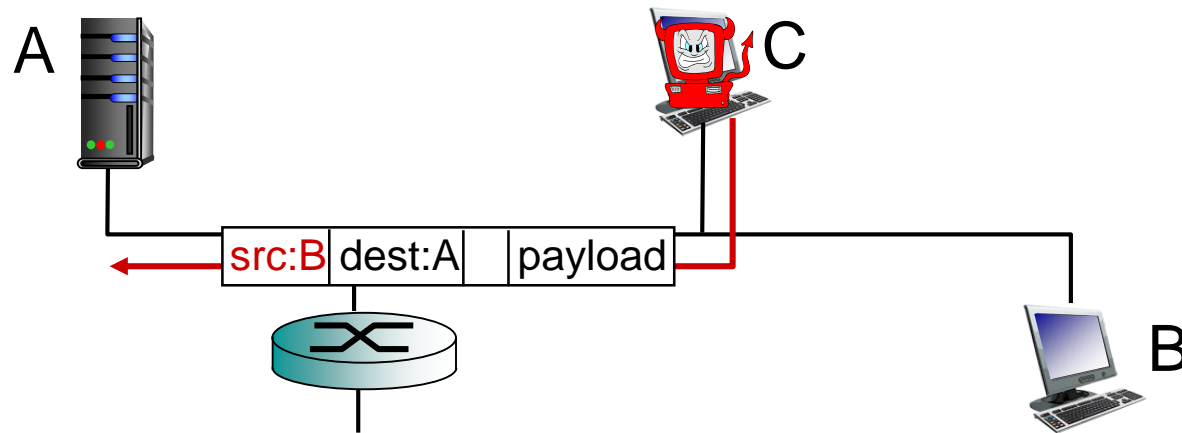
- broadcast media (shared ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- ❖ wireshark software used for end-of-chapter labs is a (free) packet-sniffer

# Bad guys can use fake addresses

*IP spoofing*: send packet with false source address



*... lots more on security (throughout, Chapter 8)*