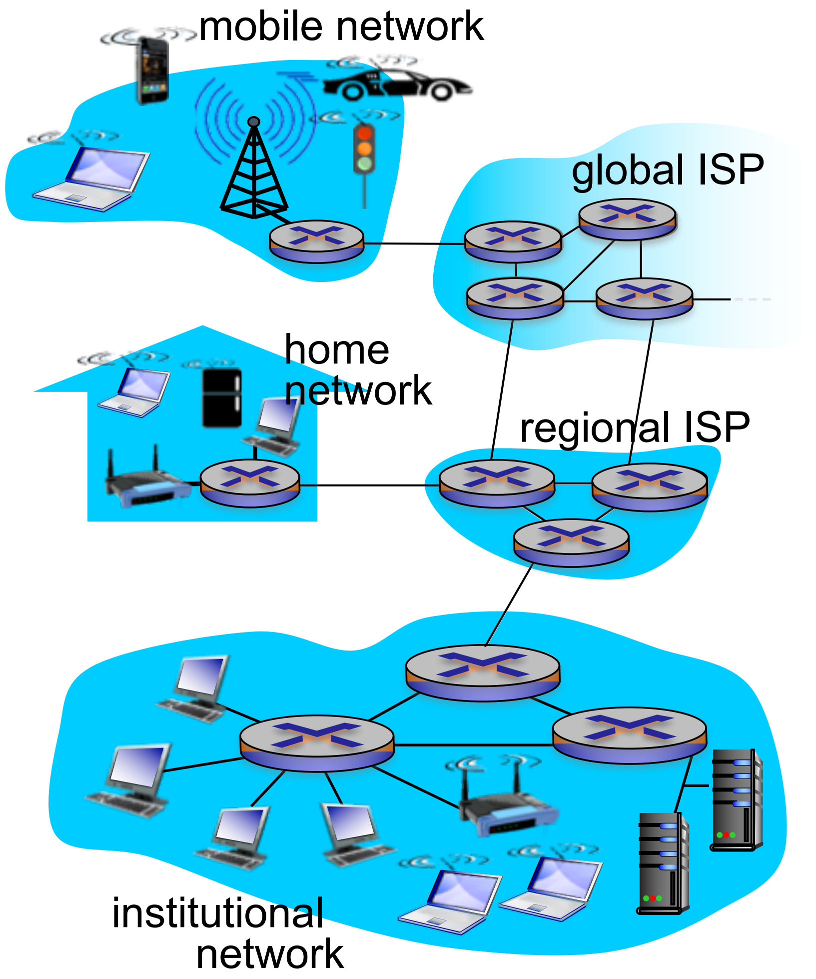
# Computer Networks and the Internet

CMPS 4750/6750: Computer Networks

# Outline

- What Is the Internet?
- Access Networks
- Packet Switching and Circuit Switching
- A closer look at delay, loss, and throughput
- Interconnection of ISPs
- Layered architecture

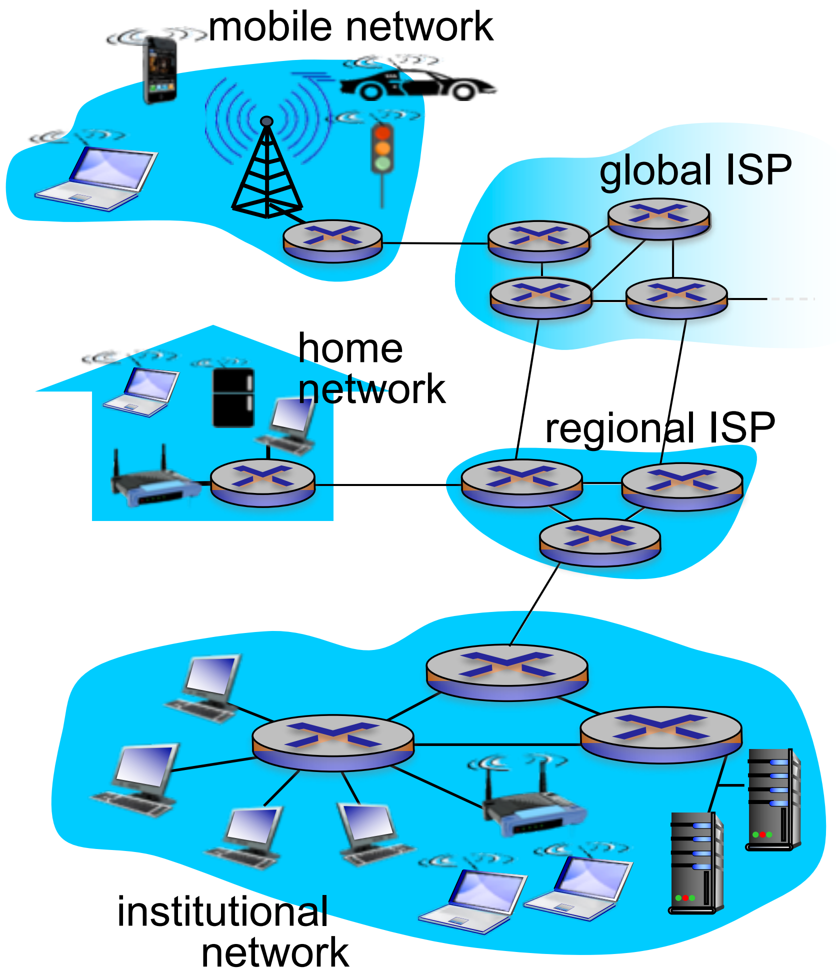
# A Nuts-and-Bolts View of the Internet




- Hosts = end systems
  - Running network apps
  - Billions of connected computing devices
- Communication links
  - copper, cables, fiber, radio, satellite
  - transmission rate (bit/sec), maximum distance
- Packet switches: forward packets
  - Routers and link-layer switches
  - ISP: a network of packet switches
- Internet: "network of networks"

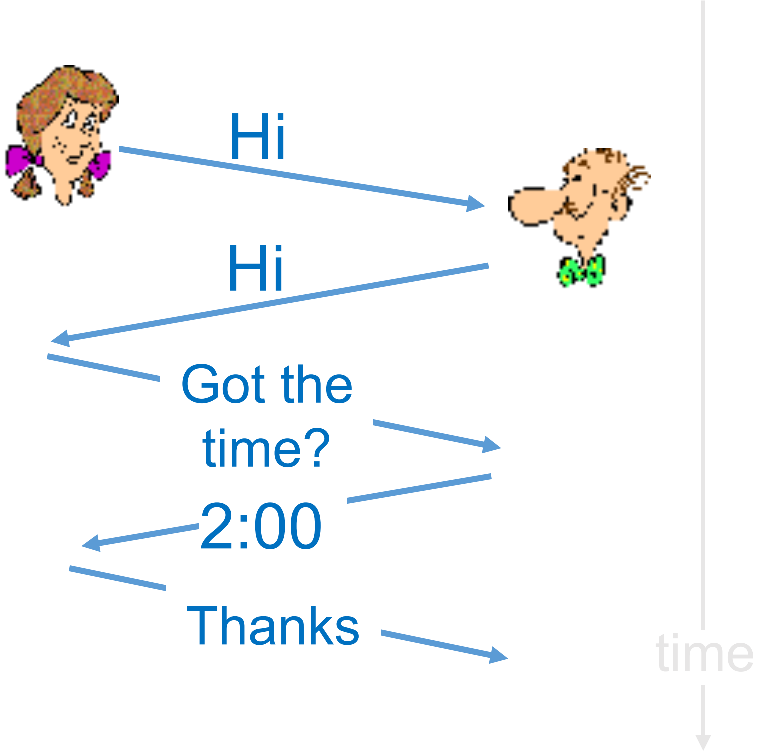
# A Service View of the Internet

- Infrastructure that provides services to network apps:
  - Web, email, messaging, games, e-commerce, social nets, maps, healthcare…
  - >1,500,000 apps in Google Play, most of which require network connections
- Provides programming interface to apps
  - Socket interface
  - Hooks that allows apps "connect" to each other
  - Provides service options: reliability, security, etc.

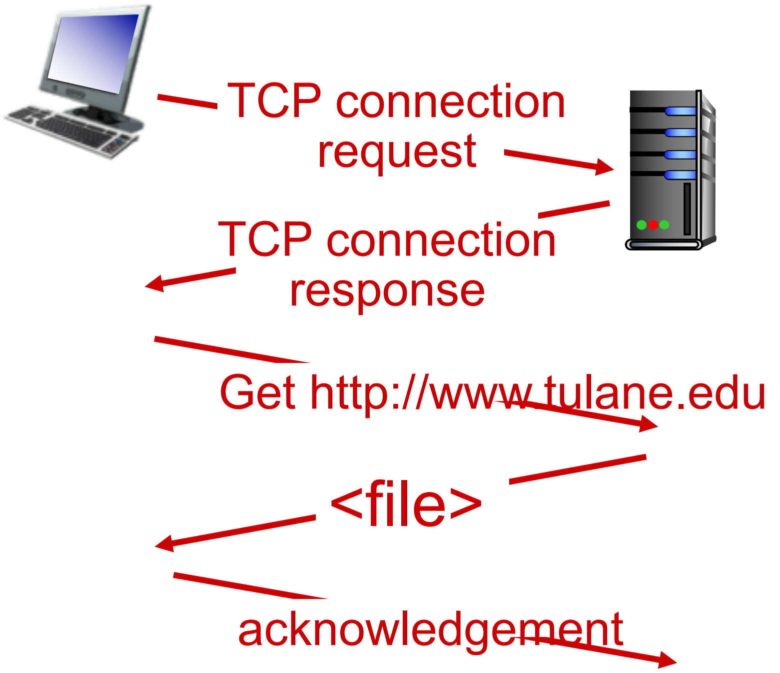


# What is a Protocol?

a human protocol:




a computer network protocol:

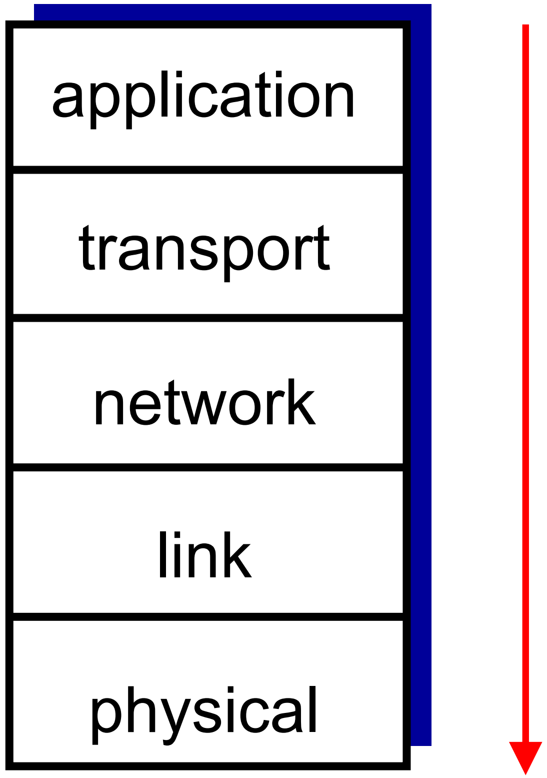


# What is a Network Protocol?

- A network protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other events.
- Protocol standardization
  - Most widely used protocols are defined in standards
  - Internet standards are developed by Internet Engineering Task Force (IETF) in the form of Request for Comments (RFCs)
  - Ethernet and wireless WiFi standards: IEEE 802 LAN/MAN Standards Committee
- Wireshark packet sniffer: a useful tool to learn protocols

# Internet protocol stack

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

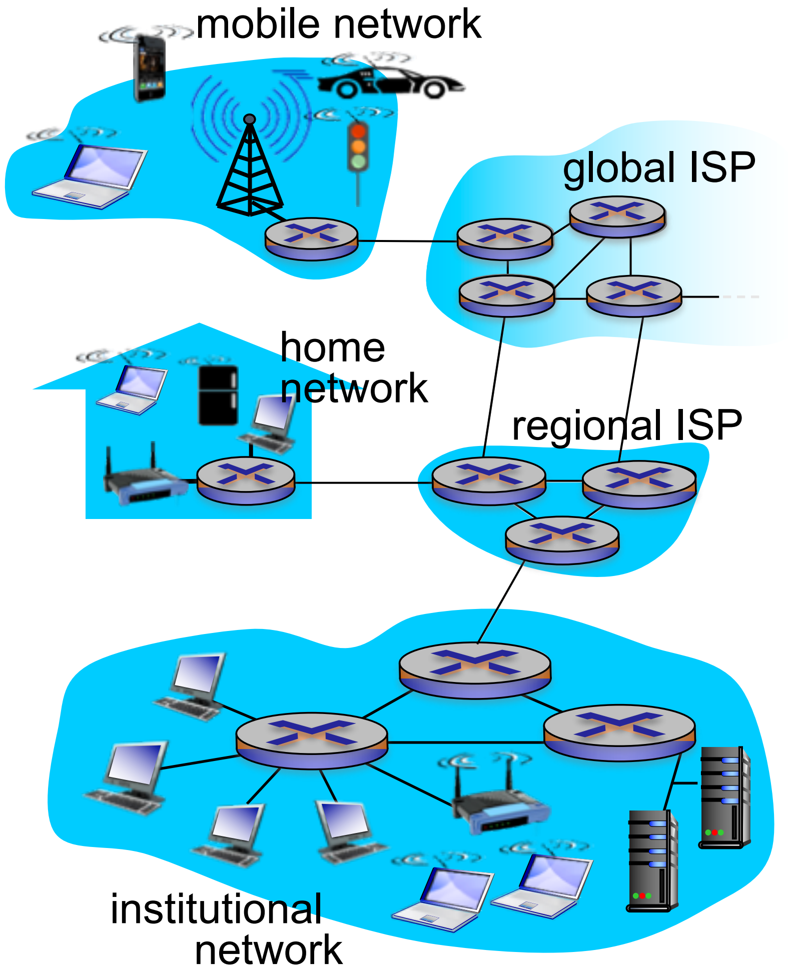


# Outline

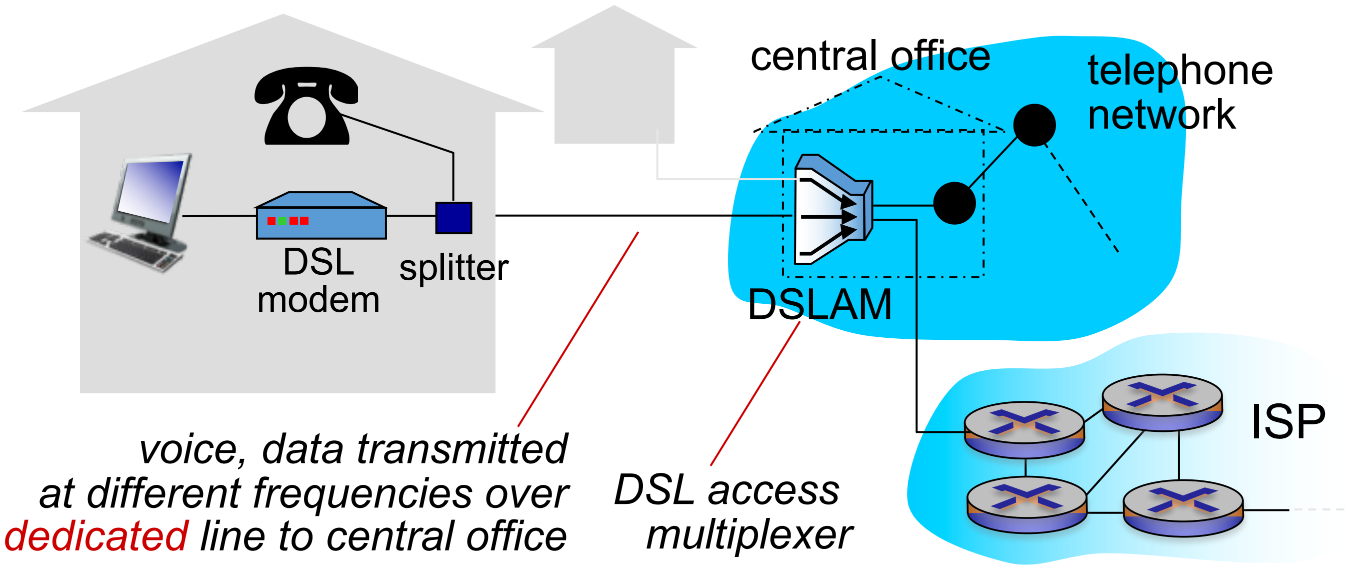
- What Is the Internet?
- Access Networks
- Packet Switching and Circuit Switching
- A closer look at delay, loss, and throughput
- Interconnection of ISPs
- Layered architecture

# A closer look at network structure

- Network Core
  - Interconnected routers
- Network Edge
  - access networks: connect hosts to the core
    - DSL, Cable, Ethernet, Wireless, Fiber to the home (FTTH), Satellite
  - hosts: clients and servers
    - clients: desktops, smartphones, smart devices
    - servers: service/content providers, often in data centers

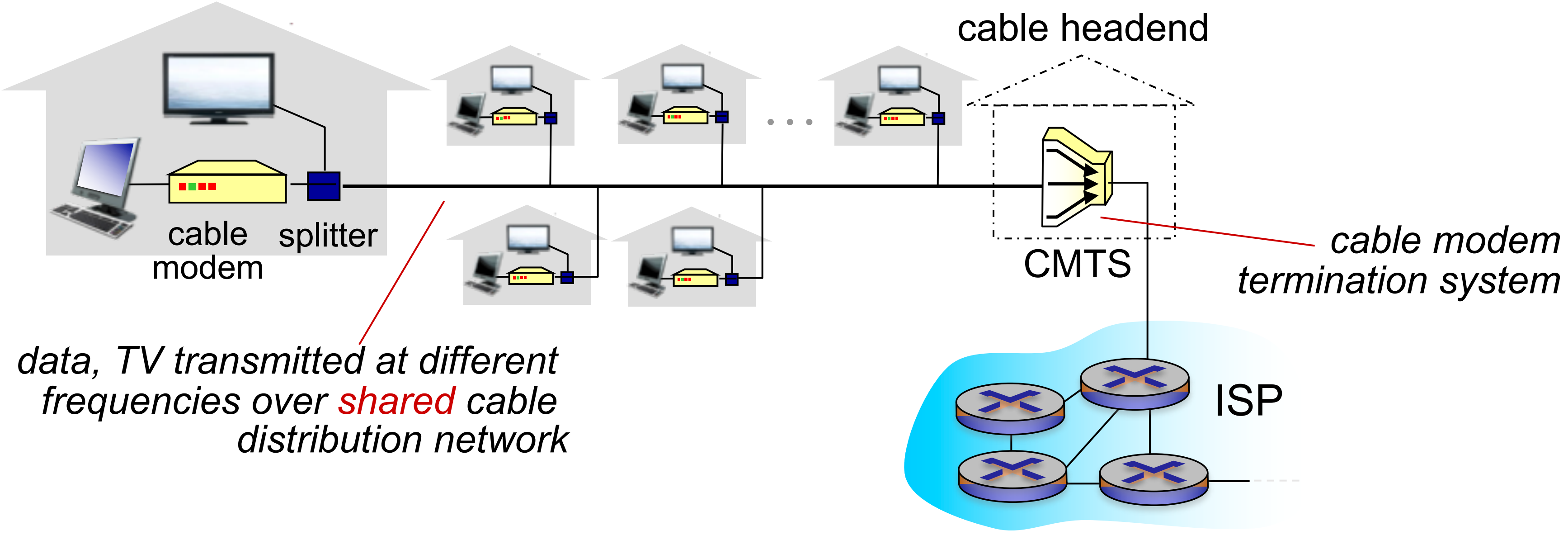


# Access network: digital subscriber line (DSL)



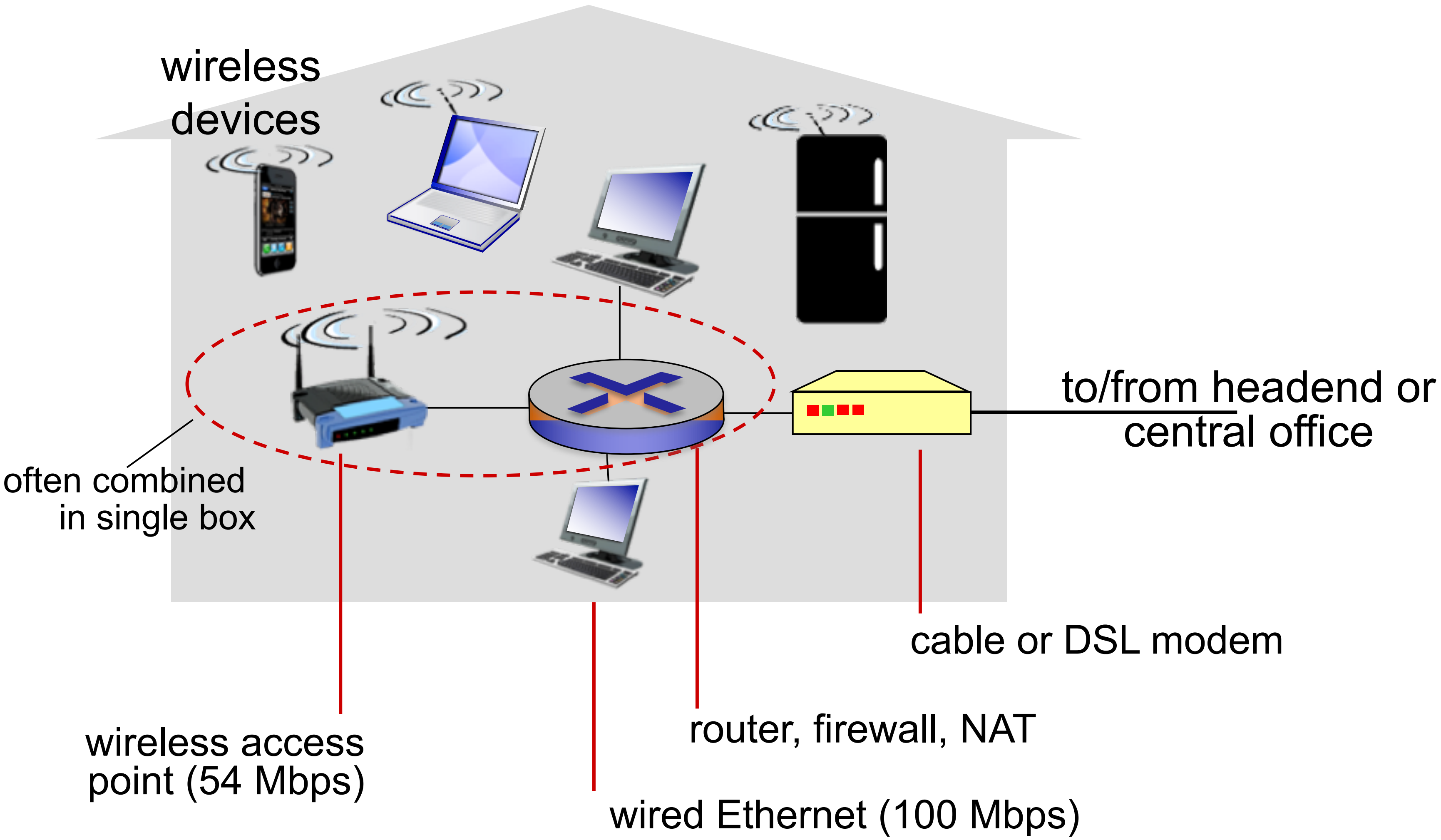

- Use *existing* telephone line to central office DSLAM
  - data over DSL phone line goes to Internet,
  - voice over DSL phone line goes to telephone net
- ADSL: asymmetric downstream and upstream rates

# Access network: cable network




- Homes *share access network* to cable headend
  - actual rate that each user receives can be significantly lower than the cable rate
  - multiple access protocol for upstream transmission

# Access network: home network



# Physical Media

- physical link: what lies between transmitter & receiver
- guided media:
  - signals propagate in solid media: twisted-pair copper wire, coaxial cable, fiber-optic cable
- unguided media:
  - signals propagate freely: terrestrial radio, satellite
- link rate: speed at which bits are transmitted
- bandwidth: the width of the range of frequencies
  - Ex:  if a telephone line can transmit signals over a range of frequencies from 300Hz to 1MHz ( = $10^6$Hz), its bandwidth is about 1MHz

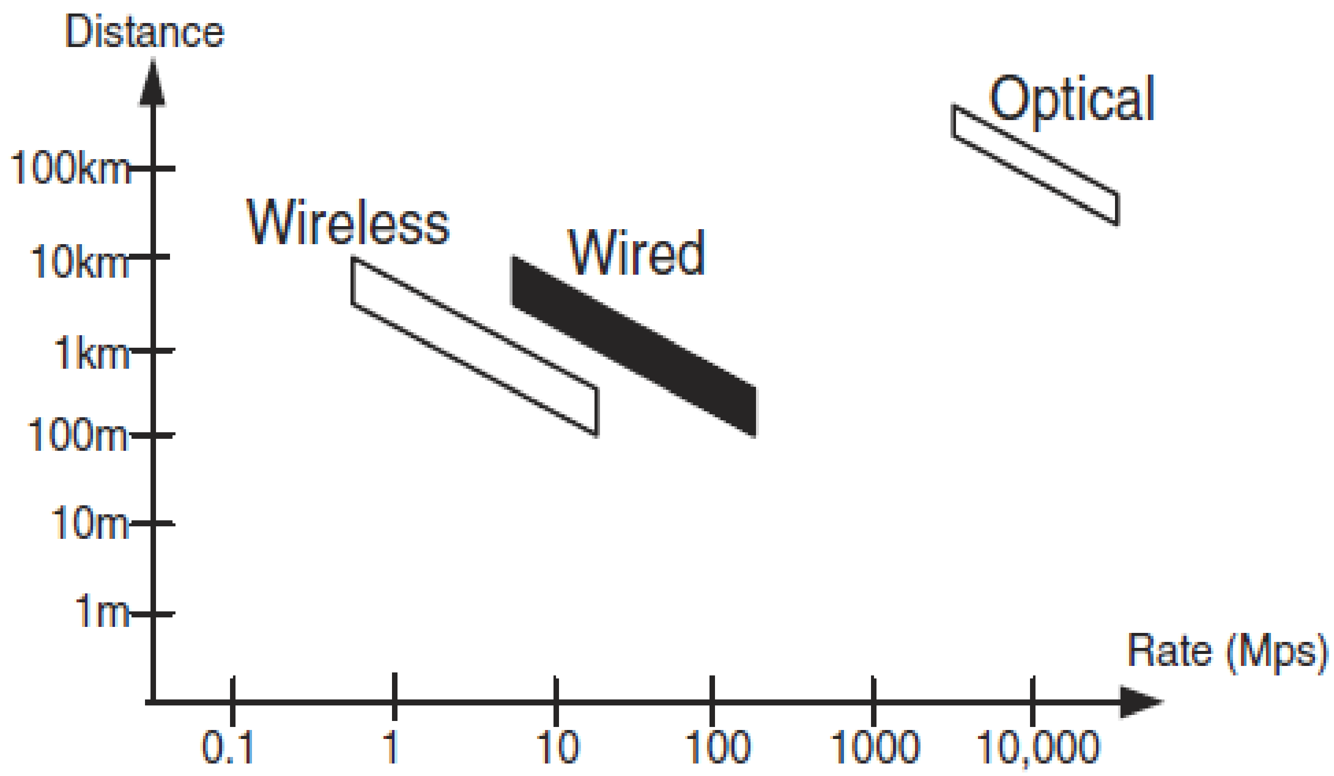
# Link rate

- Shannon Capacity: maximum reliable link rate

$$C = B\log_2(1 + \frac{S}{N}) \text{ bit per second}$$

  - $B$: bandwidth
  - $S$: power of the signal at the receiver (decreases with the length of the link)
  - $N$: power of the noise at the receiver
- Theoretical limit, hard to achieve in practice.

# Link Characteristics



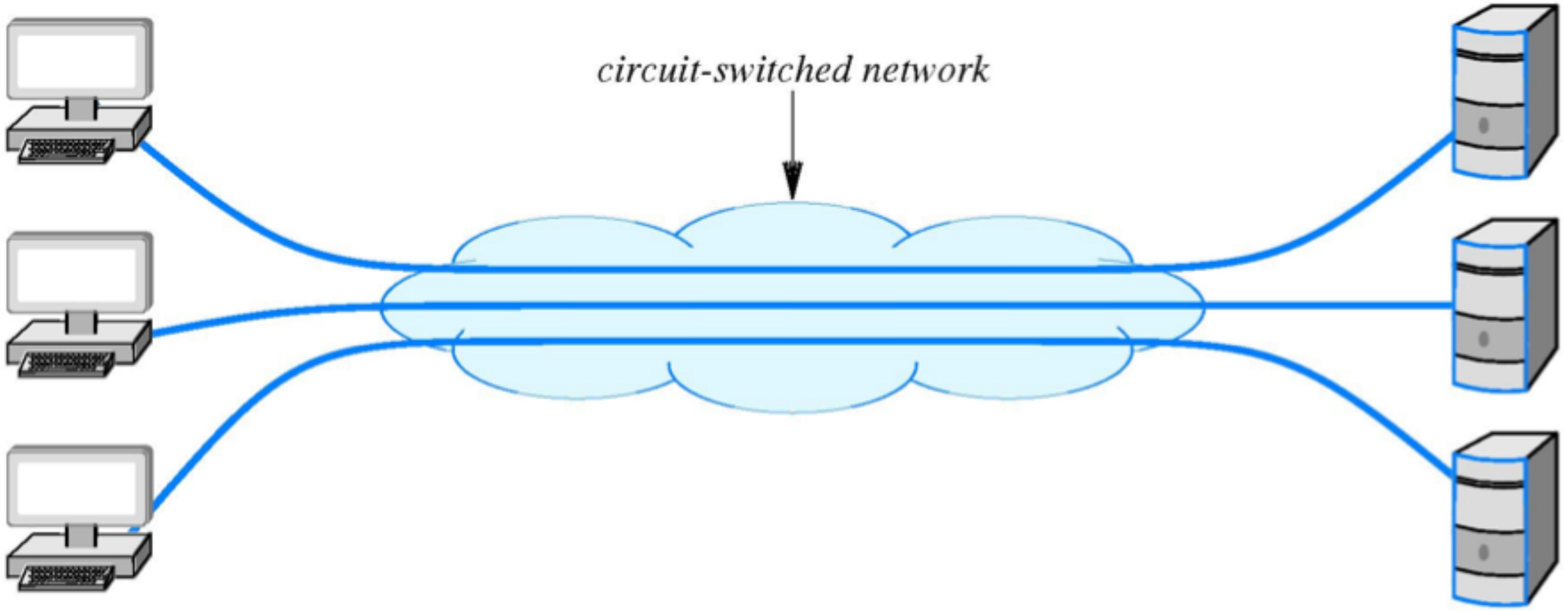

[Walrand and Parekh]

- Wired
  - DSL: a few Mbps up to 5km
  - Cable: 10 Mbps over 1km
  - Ethernet: 100 Mbps up to 110m
- Wireless
  - WiFi: tens of Mbps up to hundred meters
  - Cellular: 10 Mbps over a few km
- Optical: 10Gbps over 80km

# Outline

- What Is the Internet?
- Access Networks
- Packet Switching and Circuit Switching
- A closer look at delay, loss, and throughput
- Interconnection of ISPs
- Layered architecture

# Circuit Switching

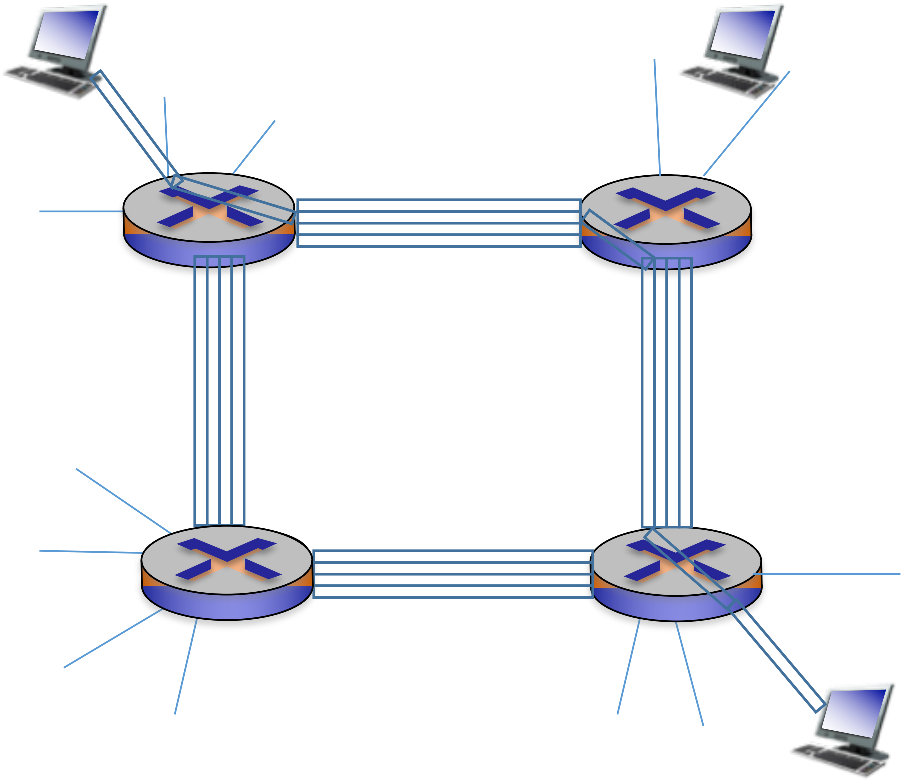



[Comer 6ed]

- dedicated resources

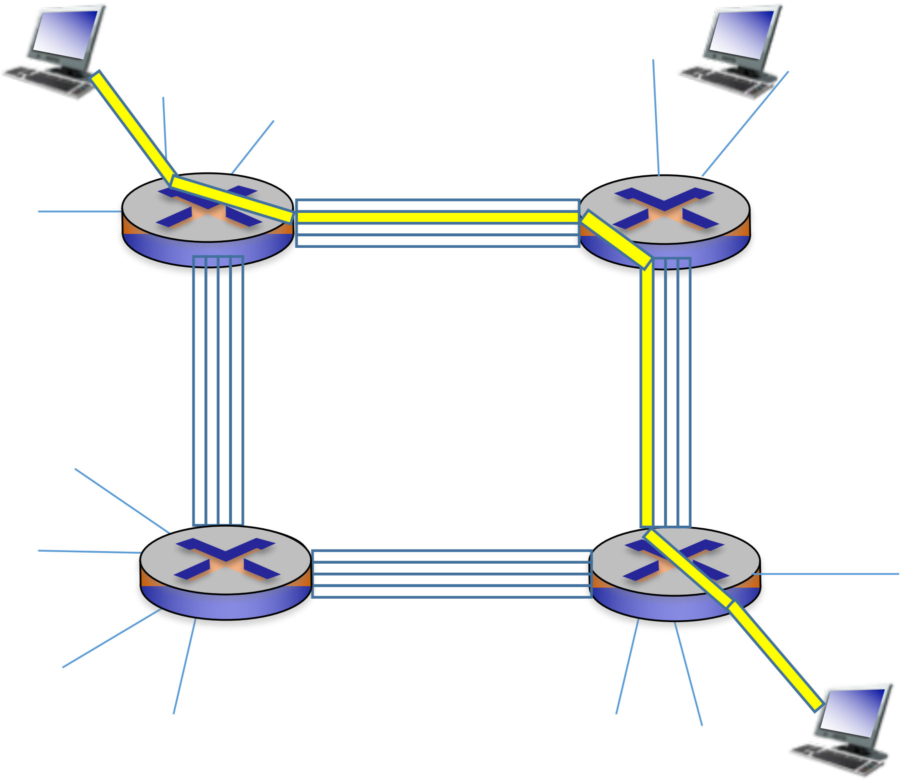
# Circuit Switching

- commonly used in traditional telephone networks
- resources reserved for "call" between source & dest:
  - resources: transmission rate, buffer, etc.
- In diagram, each link has four circuits



# Circuit Switching

- commonly used in traditional telephone networks
- resources reserved for "call" between source & dest:
  - resources: transmission rate, buffer, etc.
- In diagram, each link has four circuits
- dedicated resources
  - guaranteed performance
  - circuit segment idle if not used by call

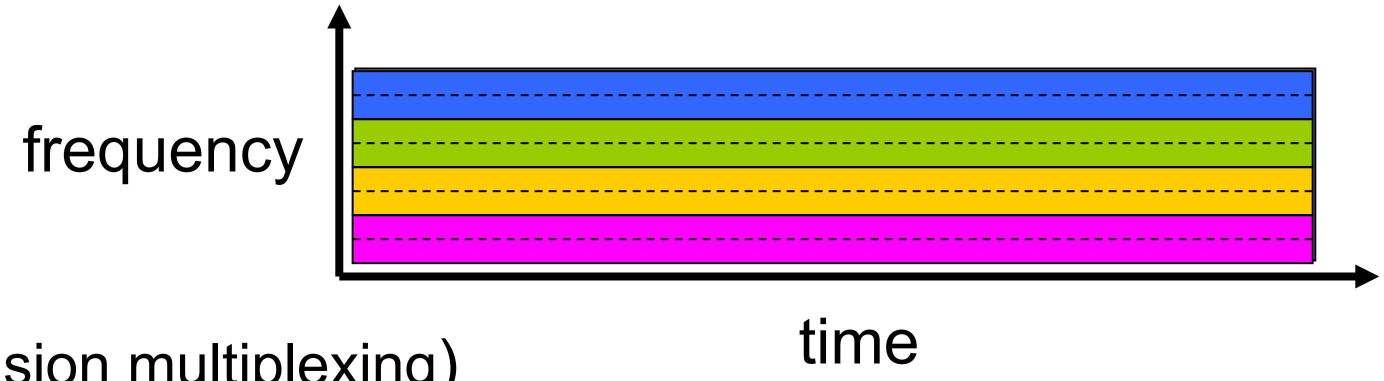


# Multiplexing in Circuit-Switched Networks

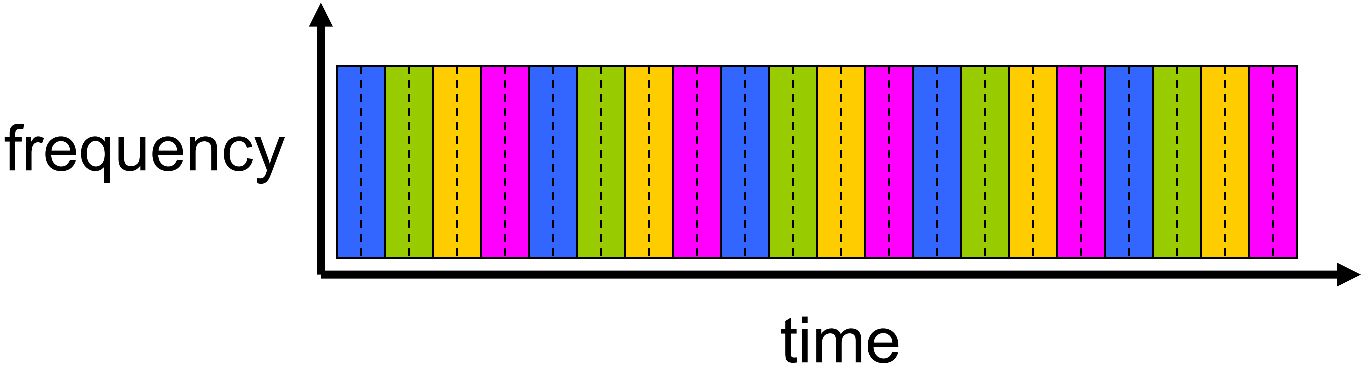
FDM (frequency-division multiplexing)

Example:

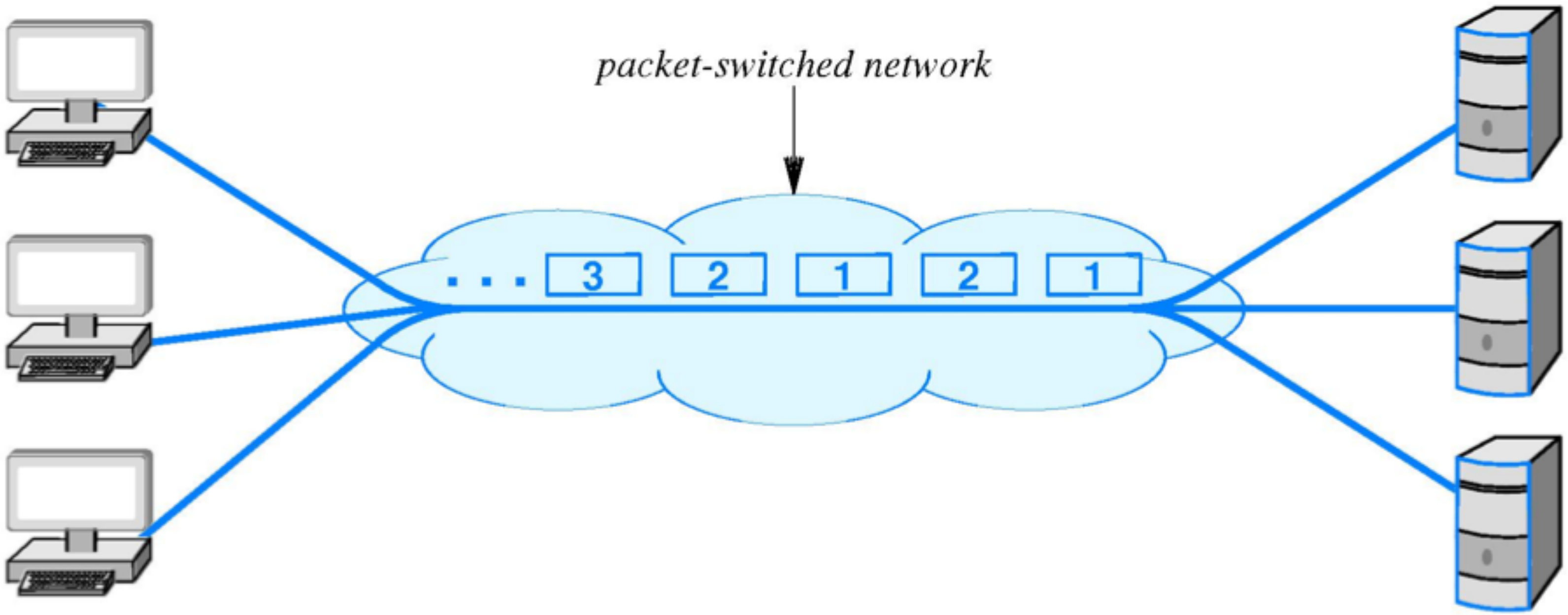
4 users




TDM (time-division multiplexing)



# Packet Switching

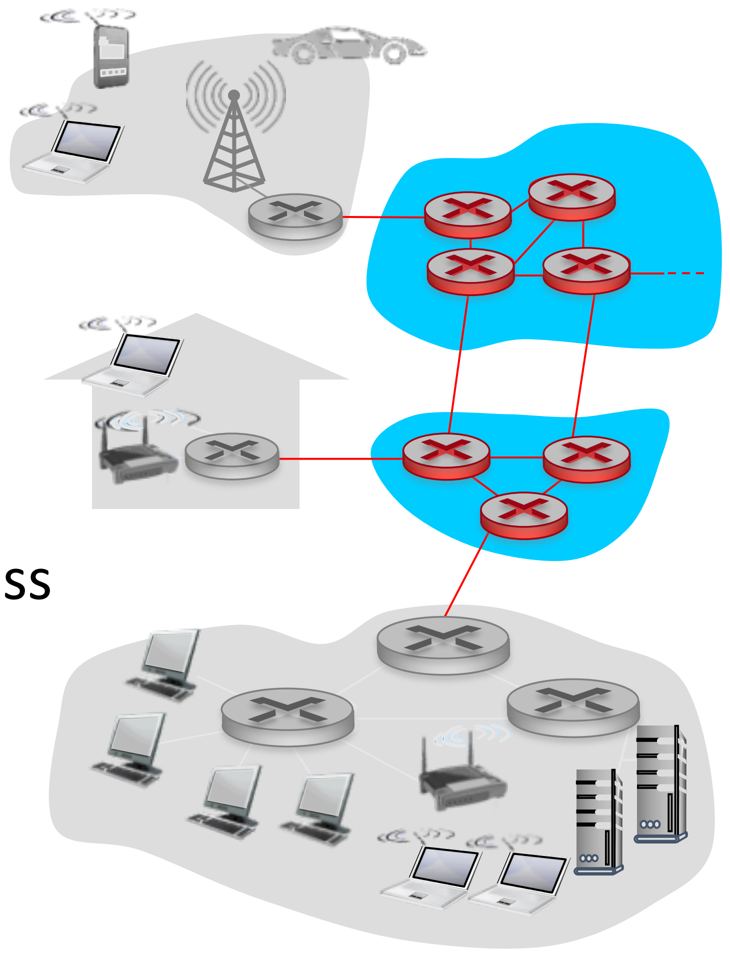



[Comer 6ed]

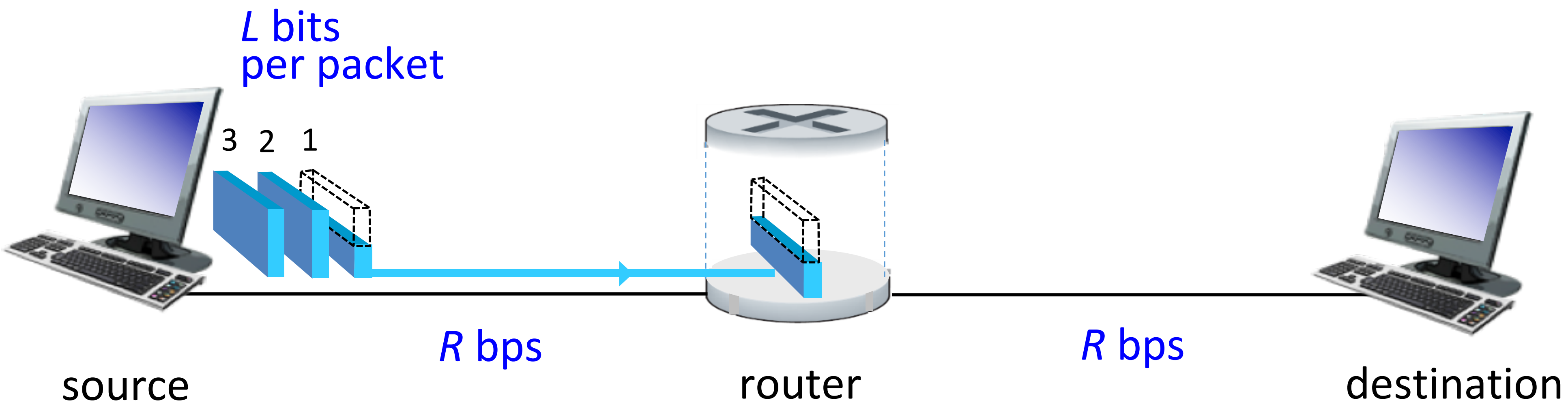
- statistical multiplexing
- resource pooling

# The Network Core

- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
  - A packet: header + payload (a set of bits)
  - forward packets from one router to the next, across links on path from source to destination
  - each packet transmitted at full link capacity

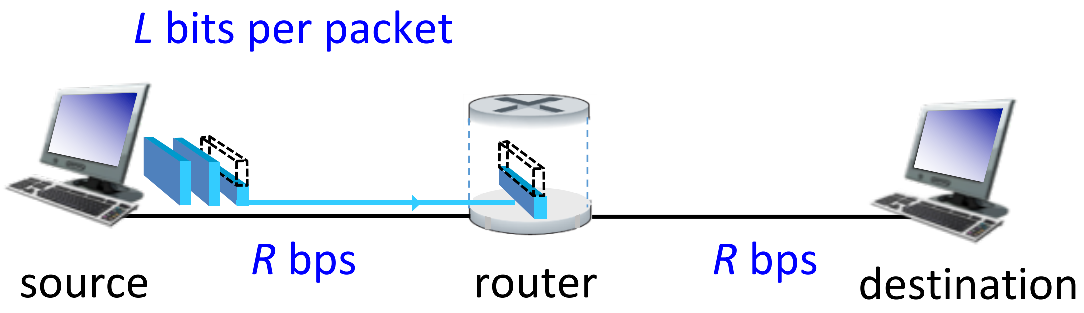


# Packet-switching: store-and-forward

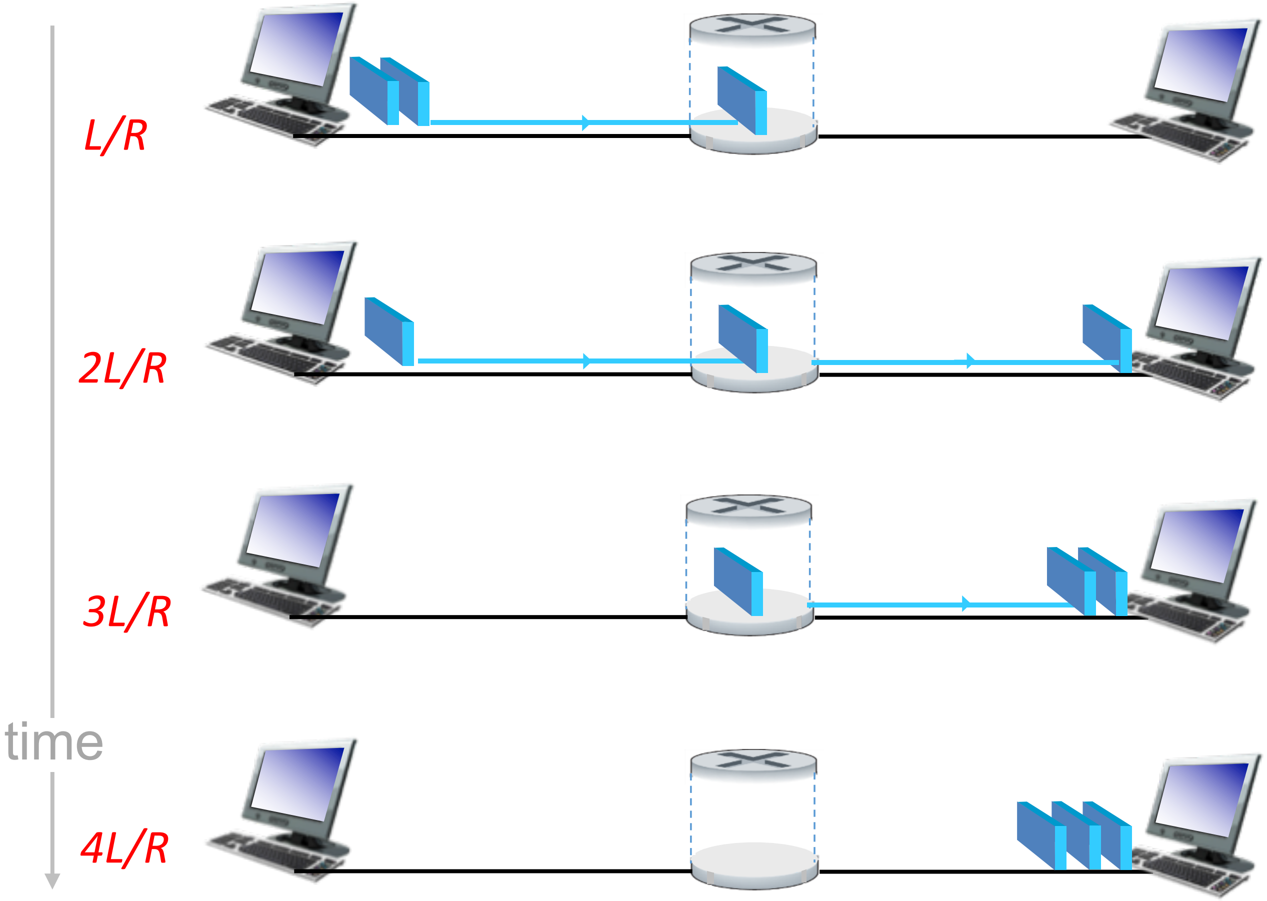



- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- takes *L*/*R* seconds to transmit (push out) *L*-bit packet into link at *R* bps
  - Ex: *R* = 7.5 Mbps, *L* = 1.5 Mbits, one-hop transmission delay = 0.2 sec
- End-to-end delay = *2L*/*R* (assuming zero propagation delay)

# Packet-switching: store-and-forward

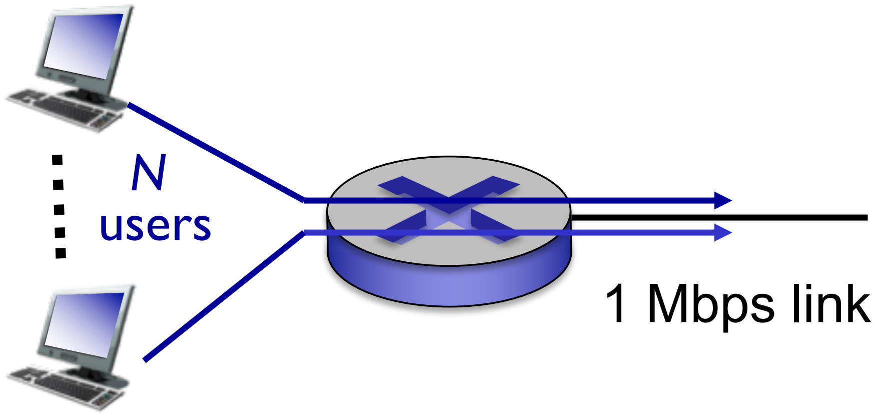



- *How long it takes for the destination to receive all the three packets?*
- *K packets? N links?*
- more on delay shortly …



# Packet Switching vs. Circuit Switching

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




- *How many users can be supported?*
  - circuit switching: 10 users
  - packet switching
    - Assume that users become active independently
    - with 35 users, probability that > 10 users active at same time is less than .0004

# Packet Switching vs. Circuit Switching

| | Circuit Switching | Packet Switching |
|---|---|---|
| Resource allocation | reserved | on demand |
| Routing | fixed routing | flexible routing |
| Resource sharing | FDM/TDM | statistical multiplexing |
| Performance guarantee | yes | no (“best effort” only) |

flexible routing → robust to attacks

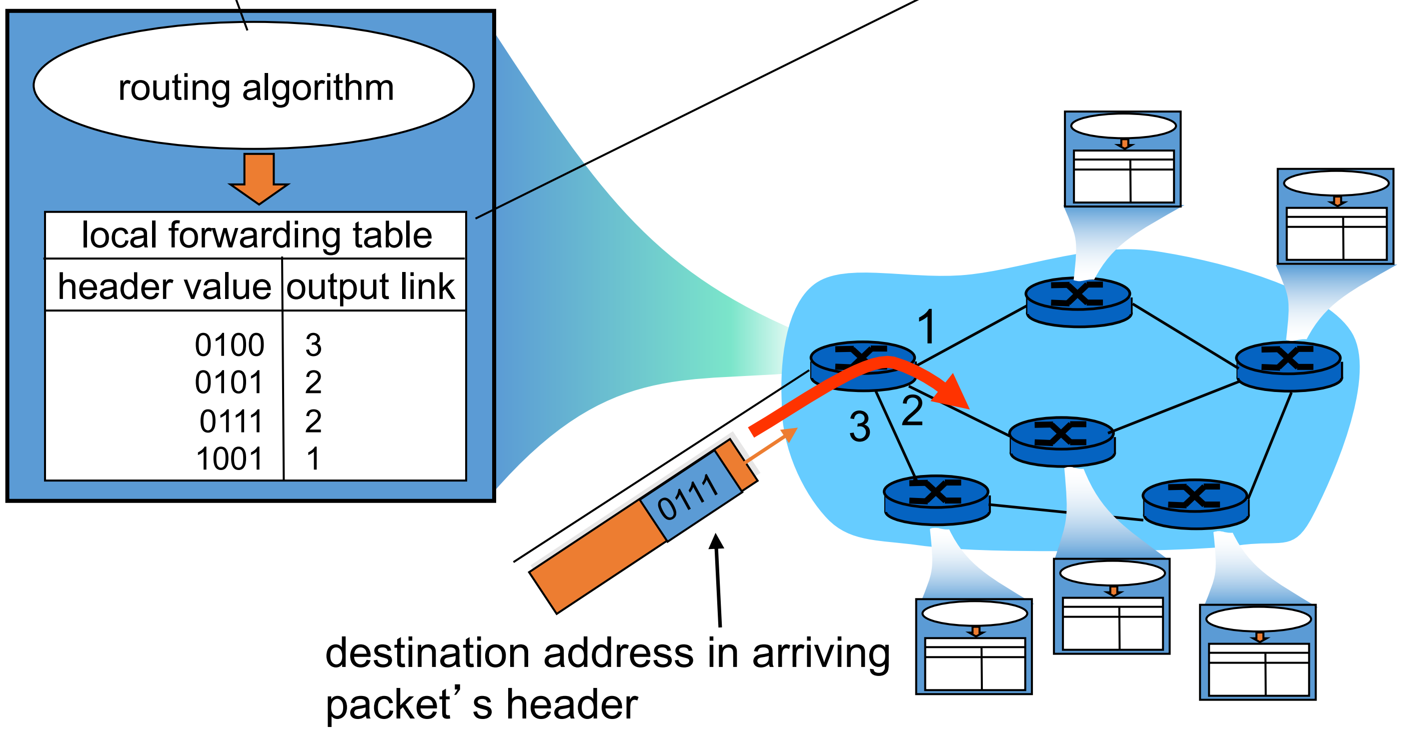
statistical multiplexing → better for bursty traffic

# Key network-core functions

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

- *routing algorithms*

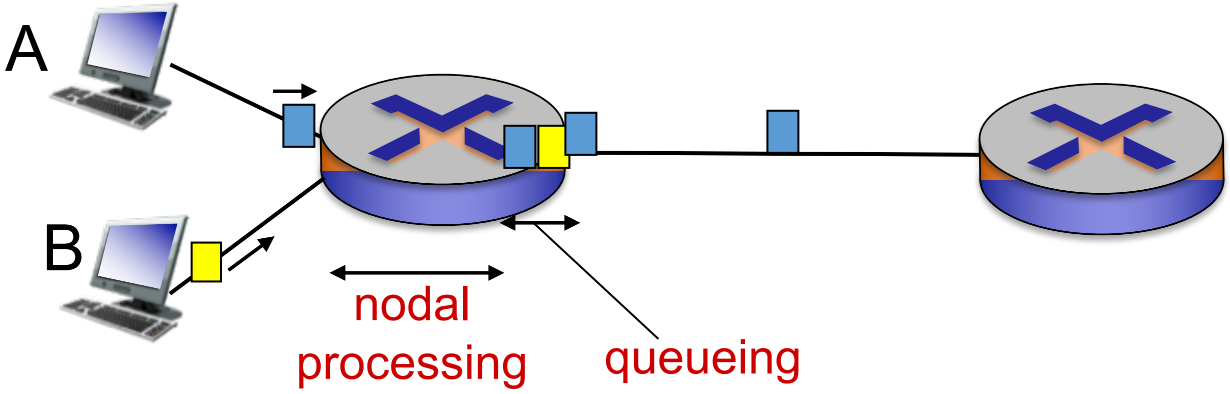
*forwarding:* move packets from router's input to appropriate router output



# Outline

- What Is the Internet?
- Access Networks
- Packet Switching and Circuit Switching
- A closer look at delay, loss, and throughput
- Interconnection of ISPs
- Layered architecture

# Four sources of packet delay



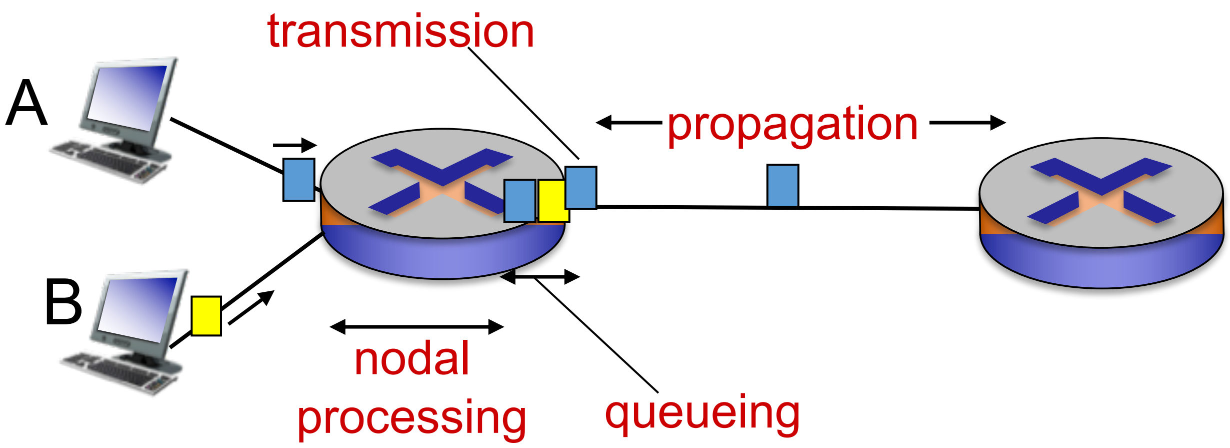

$d_{proc}$: nodal processing

- 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




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

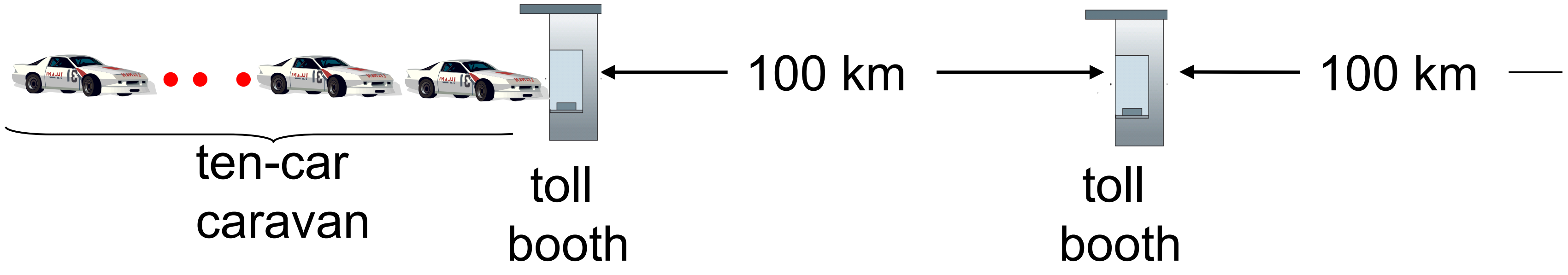
$d_{trans}$: transmission delay:

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

$d_{prop}$: propagation delay:

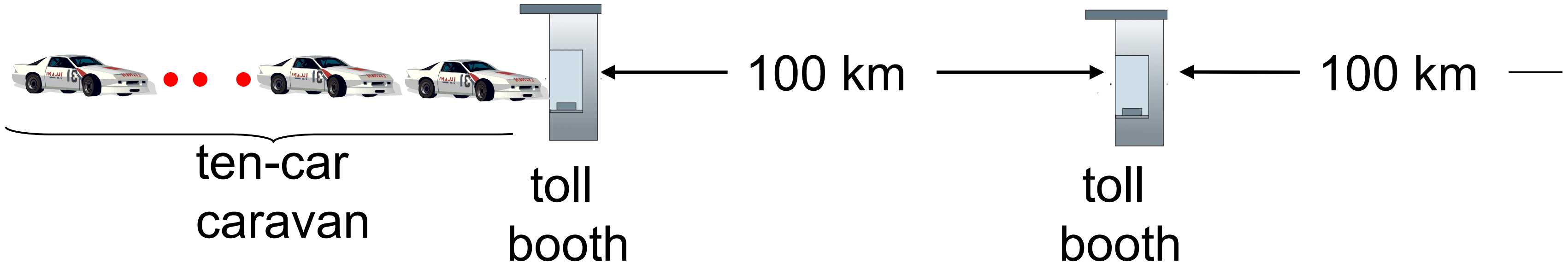
- $d$: length of physical link
- $s$: propagation speed (~$2 \times 10^8$ m/sec)
- $d_{prop} = d/s$

# Caravan analogy



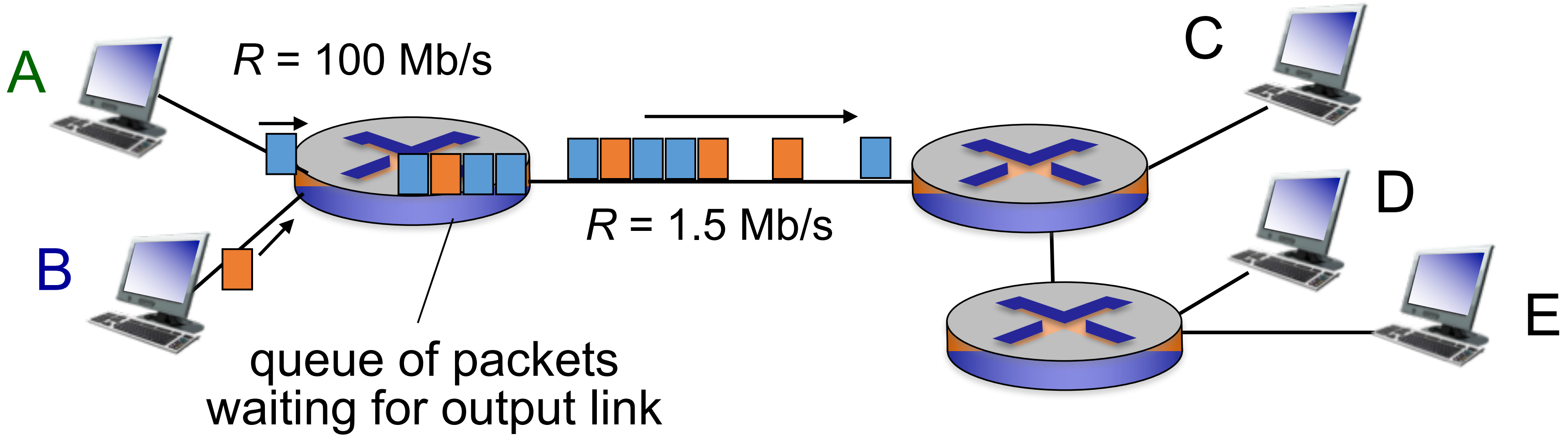

- cars “propagate” at 100 km/hr
- toll booth takes 12 sec to service one car (bit transmission time)
- car ~ bit; caravan ~ packet
- *Q:* How long until caravan is lined up before 2nd toll booth?
- time to “push” entire caravan through toll booth onto highway = 12×10 = 120 sec
- time for first car to propagate from 1st to 2nd toll both: 100km/(100km/hr)= 1 hr
- *A:* 62 minutes

# Caravan analogy (cont.)




- suppose cars now "propagate" at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- *Q:* Will cars arrive to 2nd booth before all cars serviced at first booth?
  - *A: Yes!* after 7 min, first car arrives at second booth; three cars still at first booth

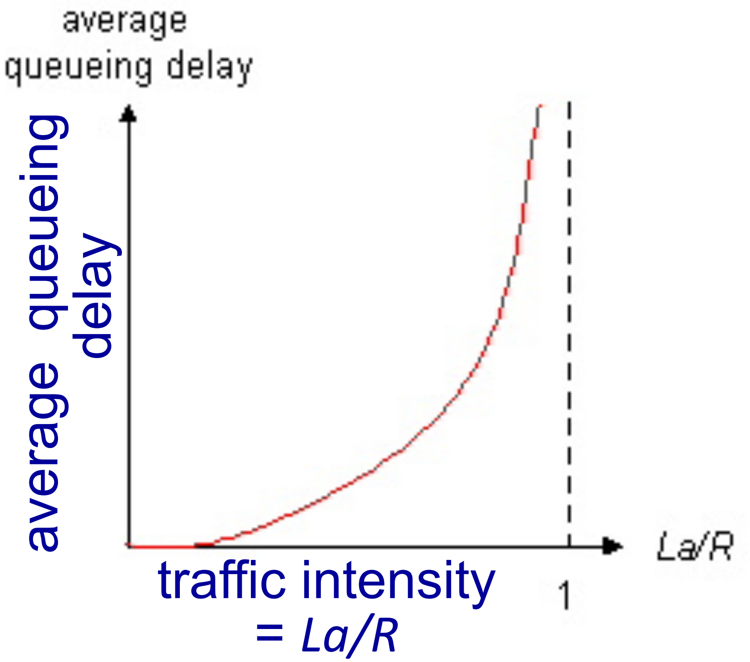
# Queueing delay and packet loss




- Each output link has a queue (buffer) of finite space
- An arriving packet will queue when link is busy
- Packet loss will occur when the output queue is full

# Queueing delay

- *R:* link bandwidth (bps)
- *L:* packet length (bits)
- *a*: average packet arrival rate
- traffic intensity = *La/R*



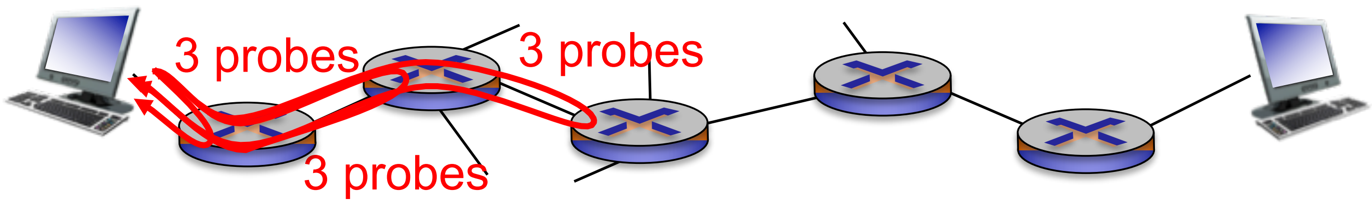

- *La/R* ~ 0: avg. queueing delay small
- *La/R* -> 1: avg. queueing delay large
- *La/R* > 1: more "work" arriving than can be serviced, average delay infinite!

*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 $i$ = *1, 2, 3, …*
  - sender sends three packets that will reach $i$-th router on path towards destination
  - router $i$ will return packets to sender
  - sender times interval between transmission and reply



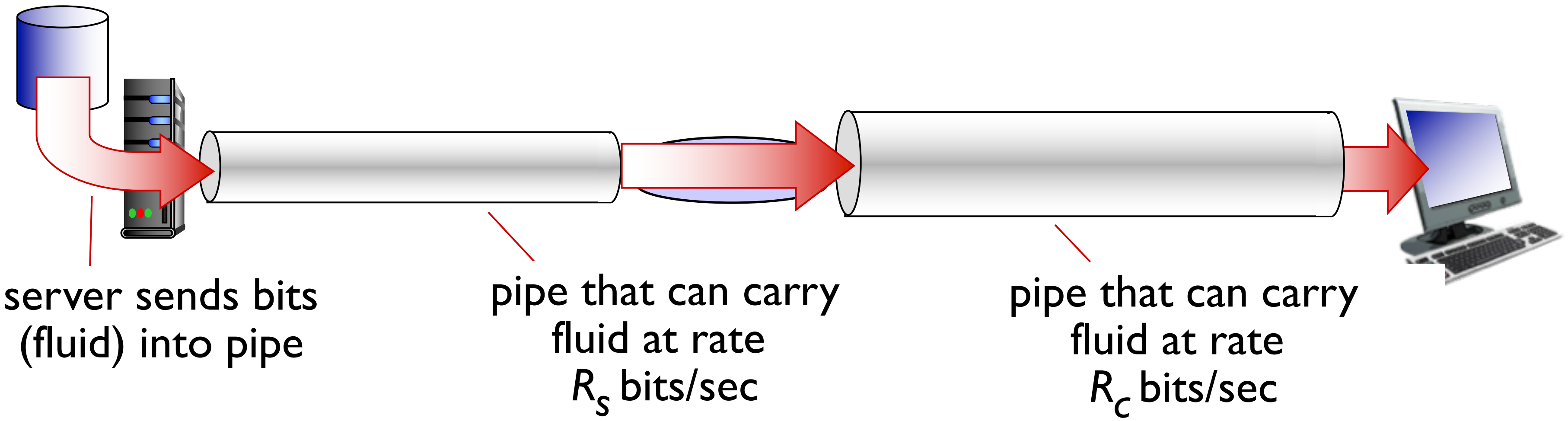
# traceroute www.cs.cmu.edu

```
 1 010-stanley-d2001-8024.tulane.net (129.81.132.62) 1.436 ms 0.963 ms 0.978 ms
 2 172.21.0.137 (172.21.0.137) 4.432 ms
   011-pyramid-8208.tulane.net (172.24.0.37) 0.386 ms
   172.21.0.137 (172.21.0.137) 0.331 ms
 3 gn-7050.tulane.net (172.24.1.150) 1.237 ms
   172.21.1.150 (172.21.1.150) 1.194 ms
   gn-7050.tulane.net (172.24.1.150) 1.105 ms
 4 bu-960.tulane.net (129.81.255.97) 1.355 ms
   bu-960.tulane.net (129.81.255.105) 1.195 ms
   bu-960.tulane.net (129.81.255.97) 1.001 ms
 5 lhno-1368-tulp.loni.org (208.100.127.193) 1.149 ms 1.168 ms 2.061 ms
 6 10.240.57.1 (10.240.57.1) 6.596 ms 6.784 ms 6.684 ms
 7 rtr.houh.net.internet2.edu-et-10-2-0.loni.org (208.100.127.2) 7.021 ms 6.801 ms 6.794 ms
 8 et-7-0-0.4079.sdn-sw.jack.net.internet2.edu (162.252.70.41) 19.465 ms 19.506 ms 19.413 ms
 9 et-3-3-0.4079.rtsw.atla.net.internet2.edu (162.252.70.42) 25.360 ms 25.176 ms 25.151 ms
10 ae-4.4079.rtsw.wash.net.internet2.edu (198.71.45.7) 37.893 ms 38.209 ms 38.048 ms
11 et-7-0-0.4079.sdn-sw.phil.net.internet2.edu (162.252.70.118) 40.827 ms 40.943 ms 40.998 ms
12 204.238.76.33 (204.238.76.33) 40.941 ms 40.964 ms 40.831 ms
13 204.238.76.46 (204.238.76.46) 41.038 ms 41.144 ms 41.332 ms
14 162.223.17.79 (162.223.17.79) 59.186 ms 51.167 ms 50.998 ms
15 core0-pod-i-dcns.gw.cmu.net (128.2.0.193) 51.291 ms 51.098 ms 51.171 ms
16 pod-d-dcns-core0.gw.cmu.net (128.2.0.210) 51.341 ms 51.554 ms 51.517 ms
17 scs-web-lb.andrew.cmu.edu (128.2.42.95) 51.647 ms 51.623 ms 51.521 ms
```

3 delay measurements

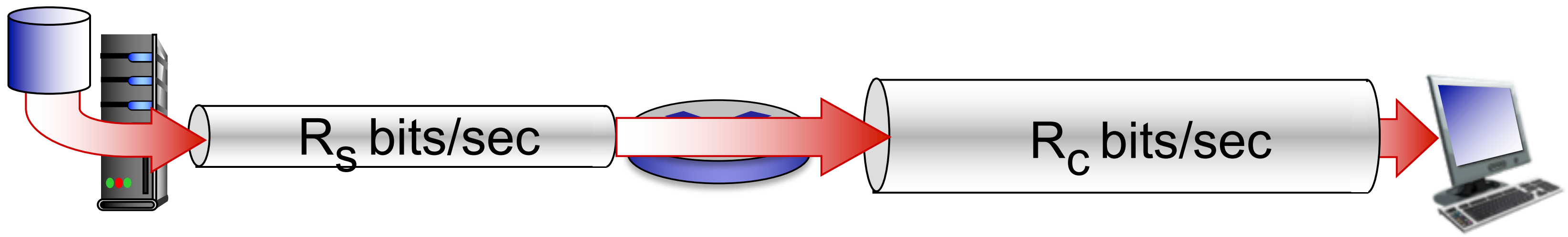
# Throughput

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

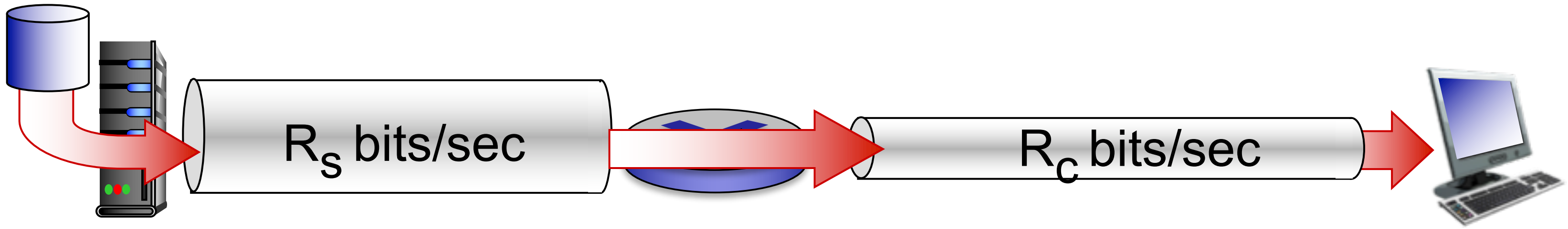


# Throughput

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



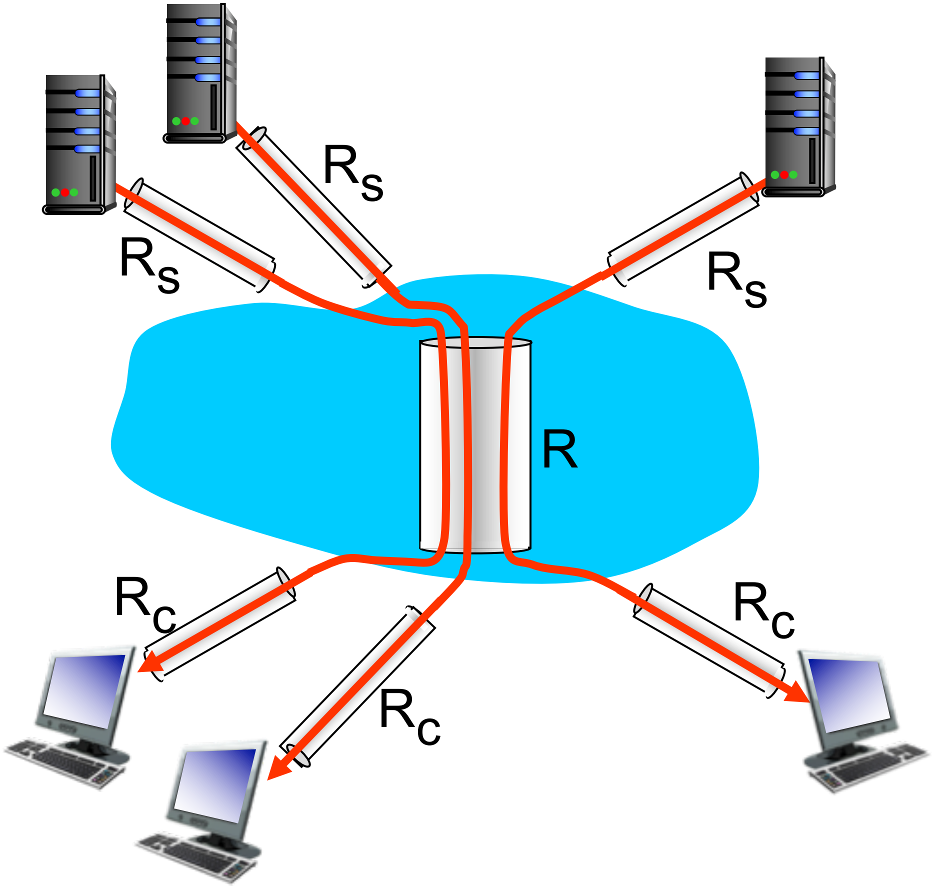

- $R_s > R_c$ What is average end-end throughput?




*bottleneck link*

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

# Throughput: Internet scenario




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

- What is the per-connection end-end throughput?

$$\min(R_c, R_s, R/10)$$

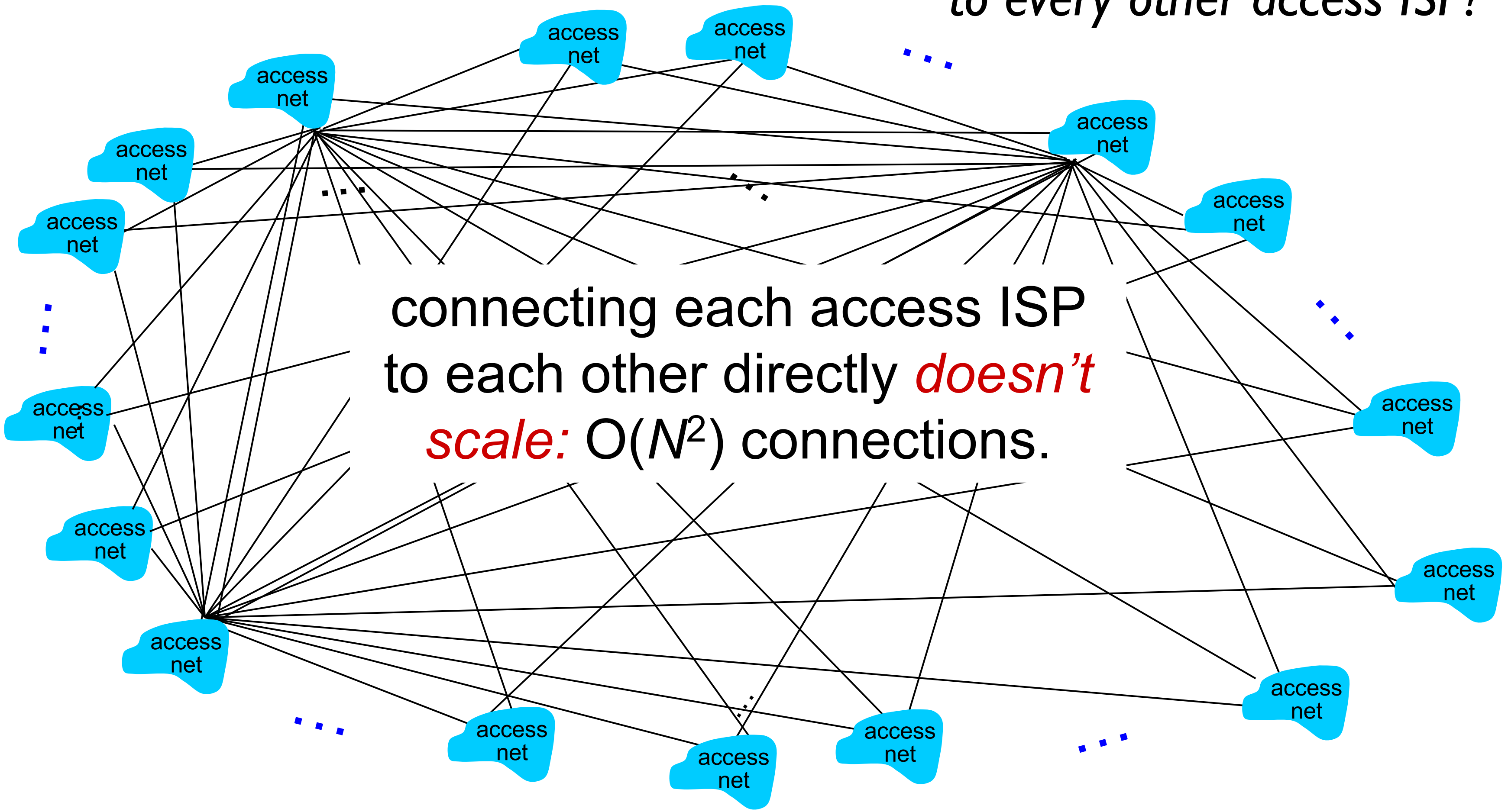
- in practice: $R_c$ or $R_s$ is often bottleneck

# Outline

- What Is the Internet?
- Access Networks
- Packet Switching and Circuit Switching
- A closer look at delay, loss, and throughput
- Interconnection of ISPs
- Layered architecture

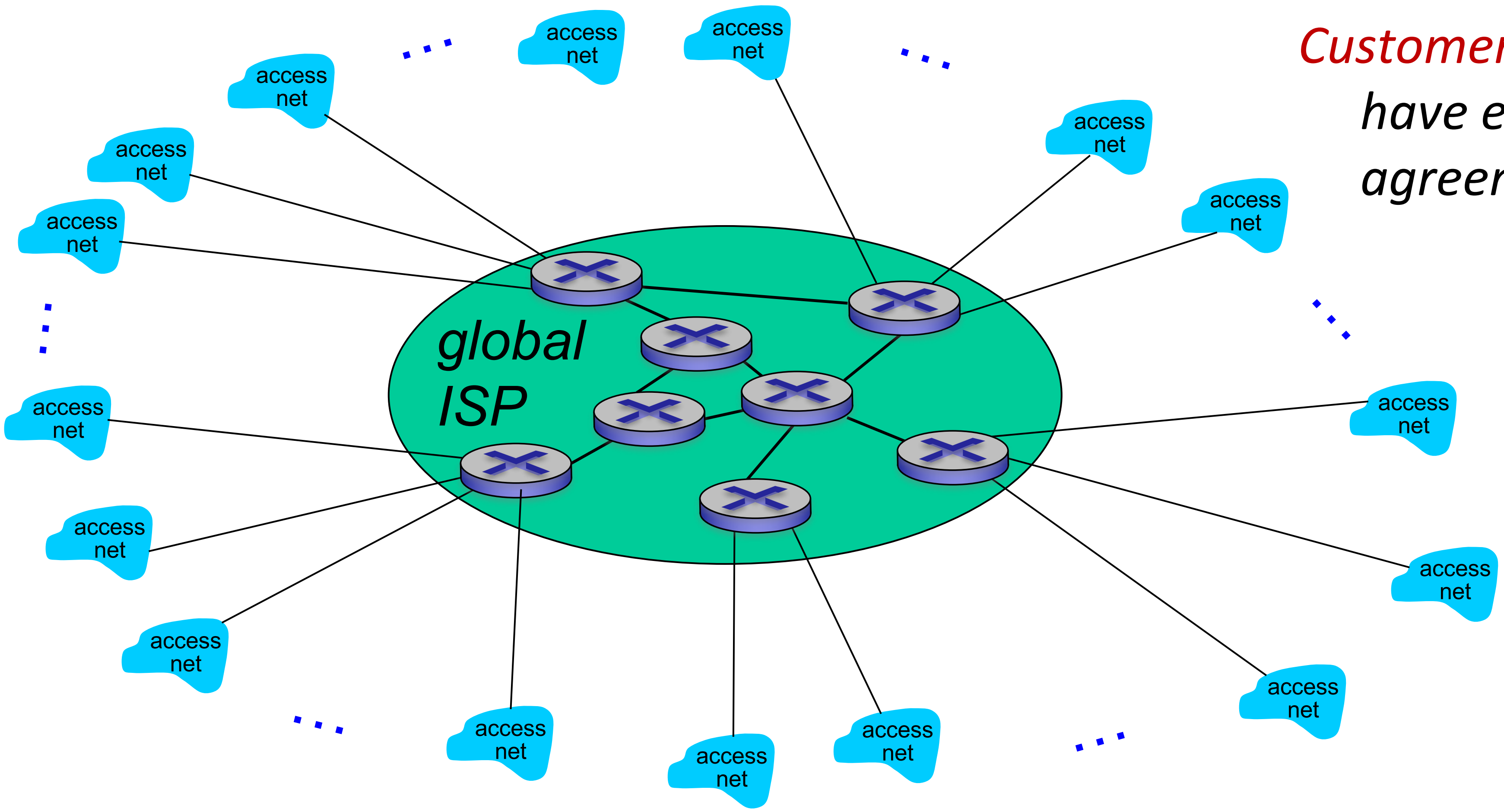
# Internet structure: network of networks

*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.

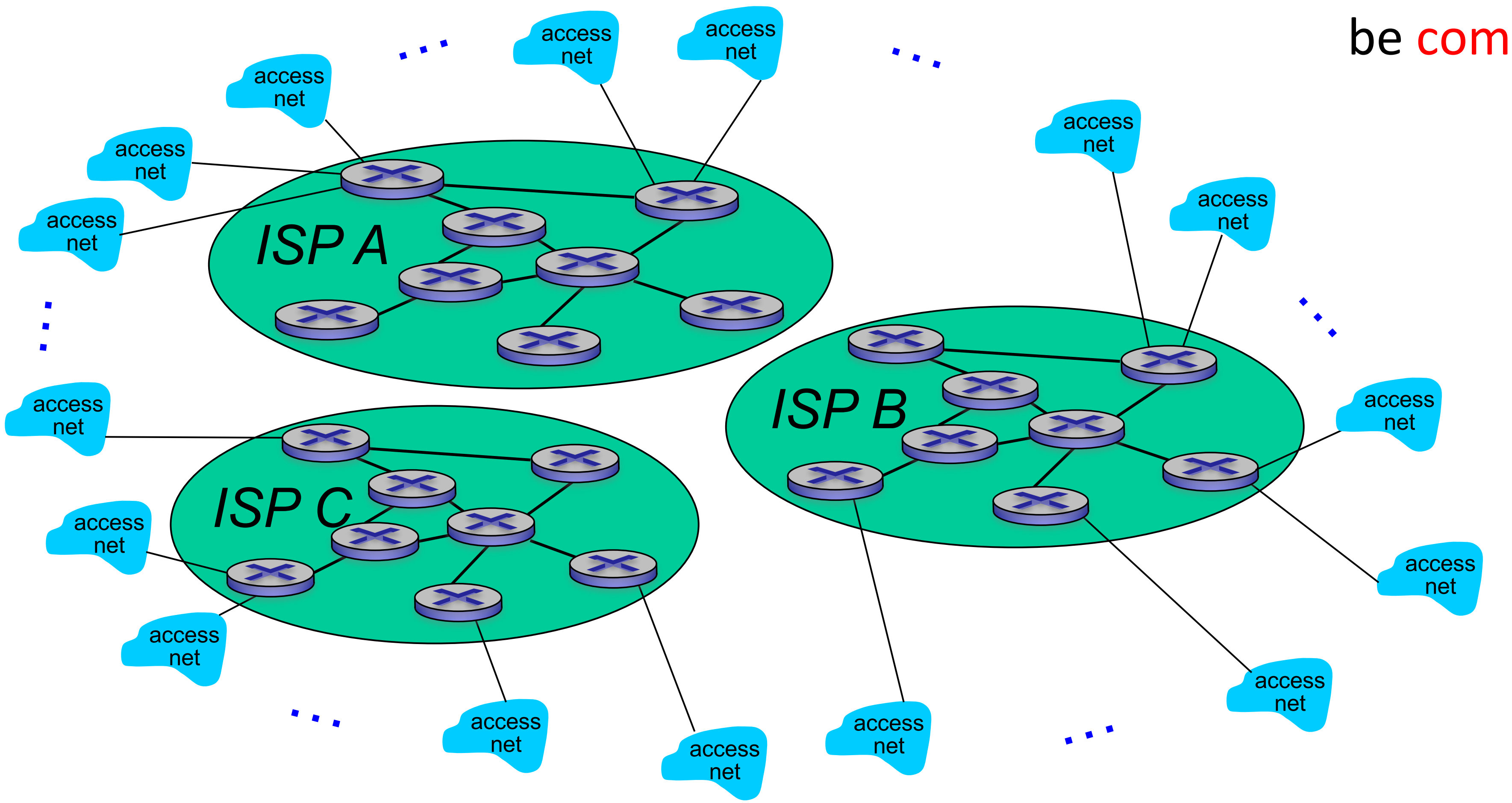
# Internet structure: network of networks




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

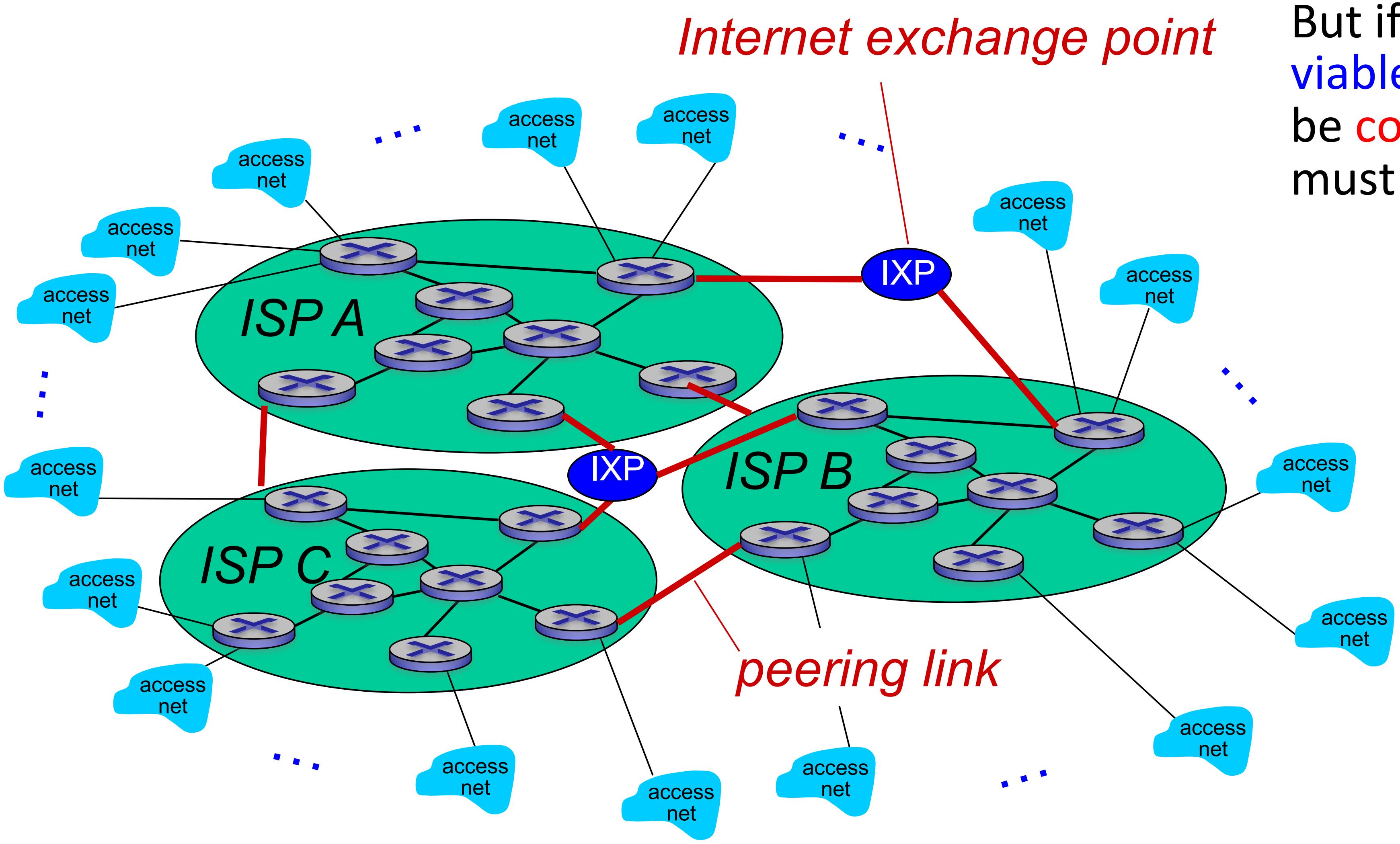
*Customer and provider ISPs have economic agreement.*

# Internet structure: network of networks



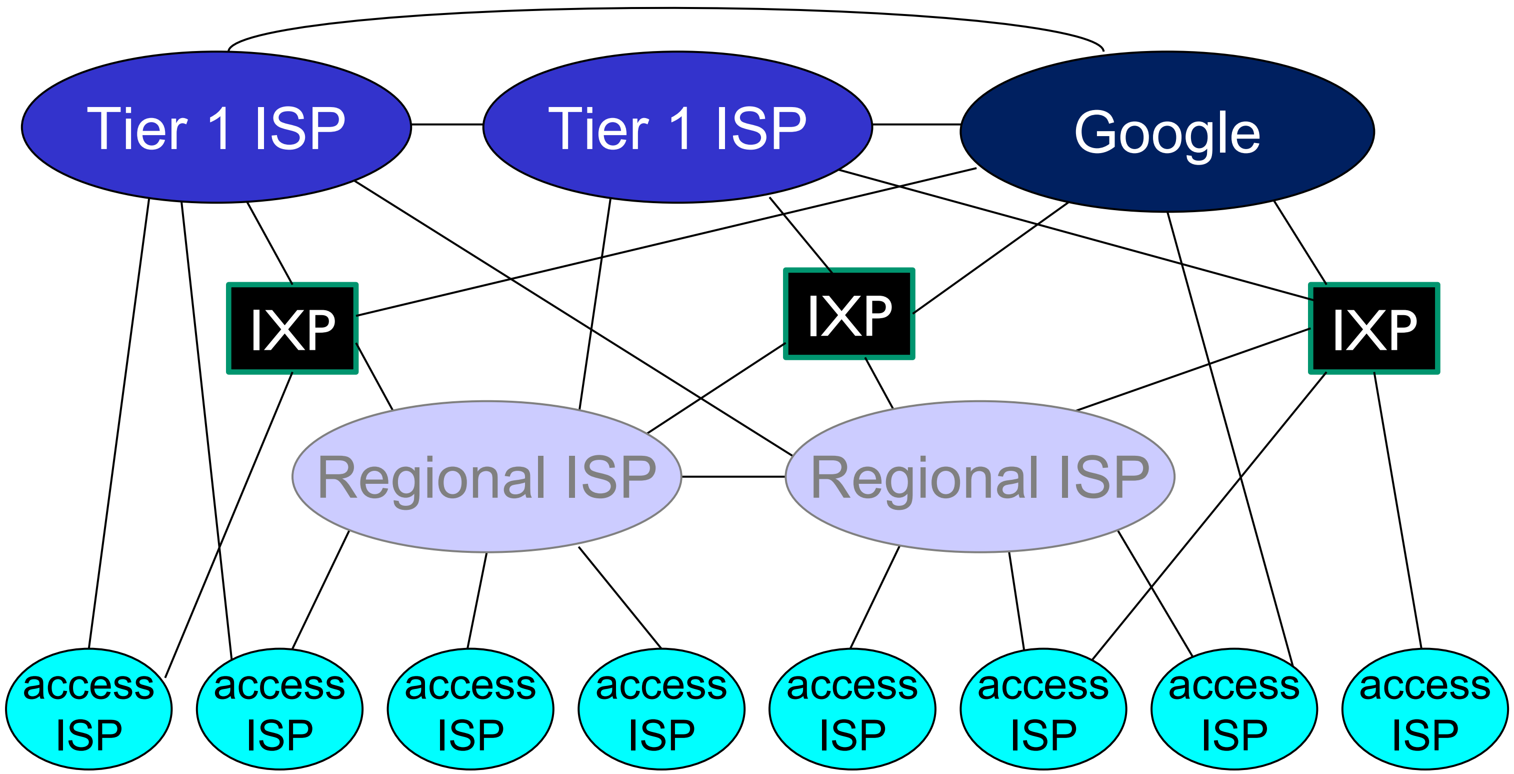

But if one global ISP is viable business, there will be competitors ….

# 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




- at center: small # of well-connected large networks
  - "tier-1" commercial ISPs (e.g., AT&T, Verizon, CenturyLink), national & international coverage
  - content provider network (e.g., Google): private network that connects it data centers to Internet, often bypassing tier-1, regional ISPs

# Outline

- What Is the Internet?
- Access Networks
- Packet Switching and Circuit Switching
- A closer look at delay, loss, and throughput
- Interconnection of ISPs
- Layered architecture

# Protocol layers

*Networks are complex, with many "pieces":*

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

*Various services provided:*

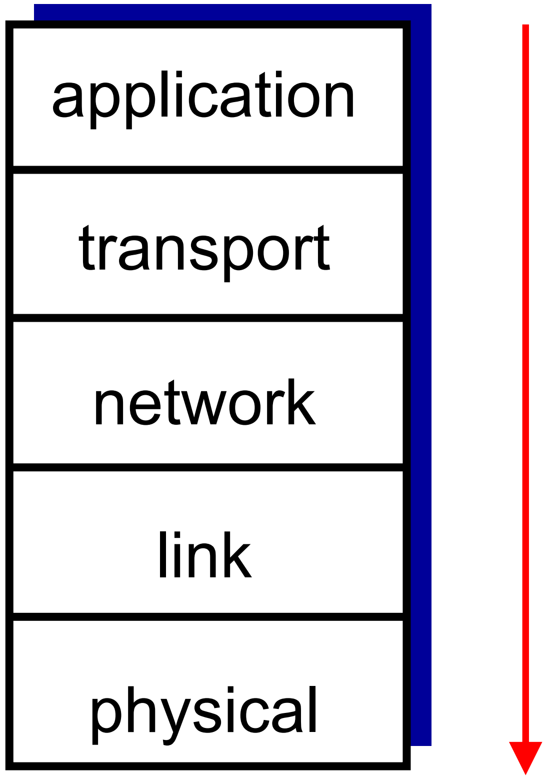
- Move bits over various media: multiple access, error control
- Packet switching: forwarding & routing
- Reliable transfer: retransmission, flow control, congestion
- Quality of service: delay, throughput, security

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

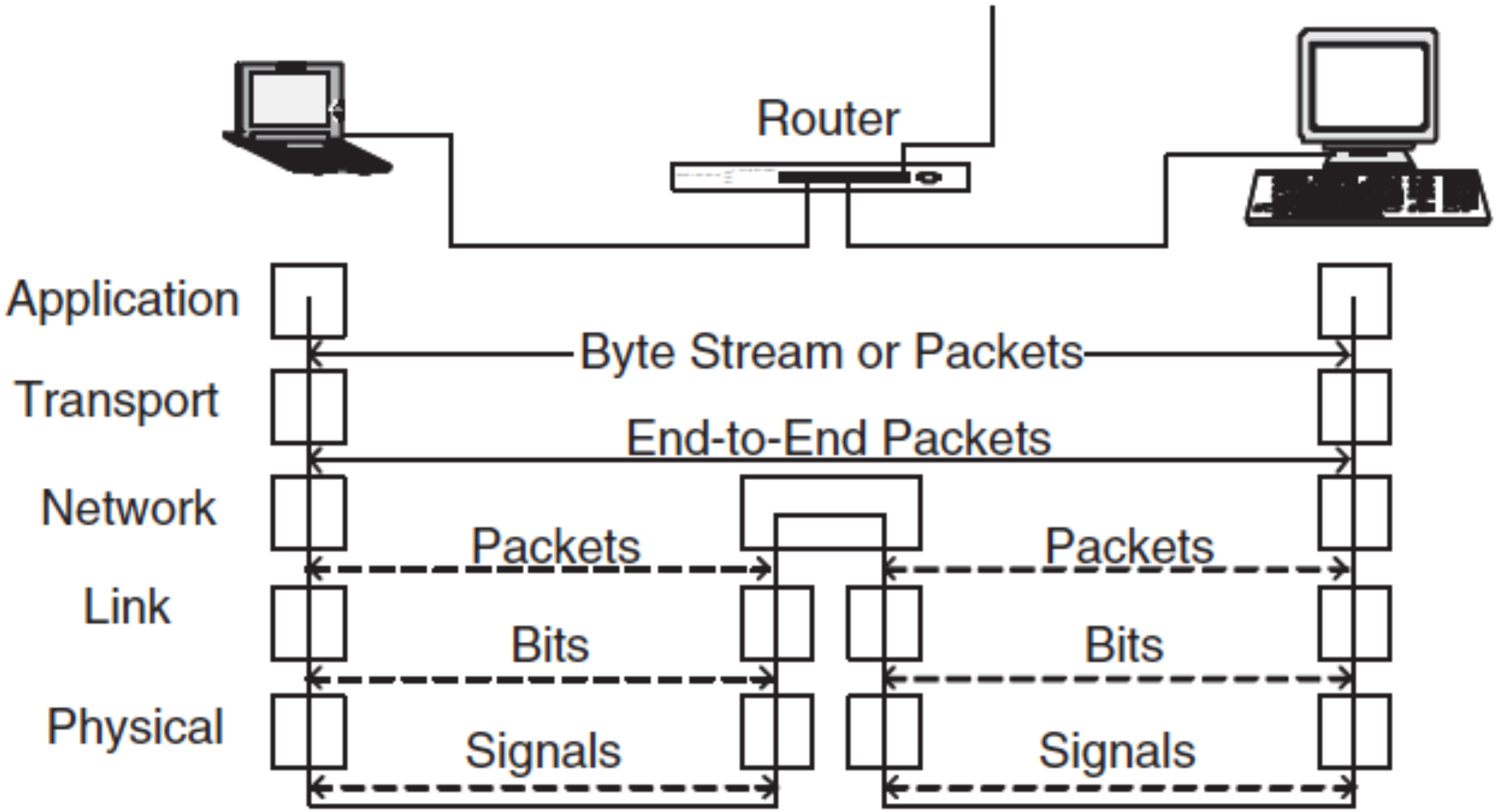
…. or at least our discussion of networks?

# Internet protocol stack

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

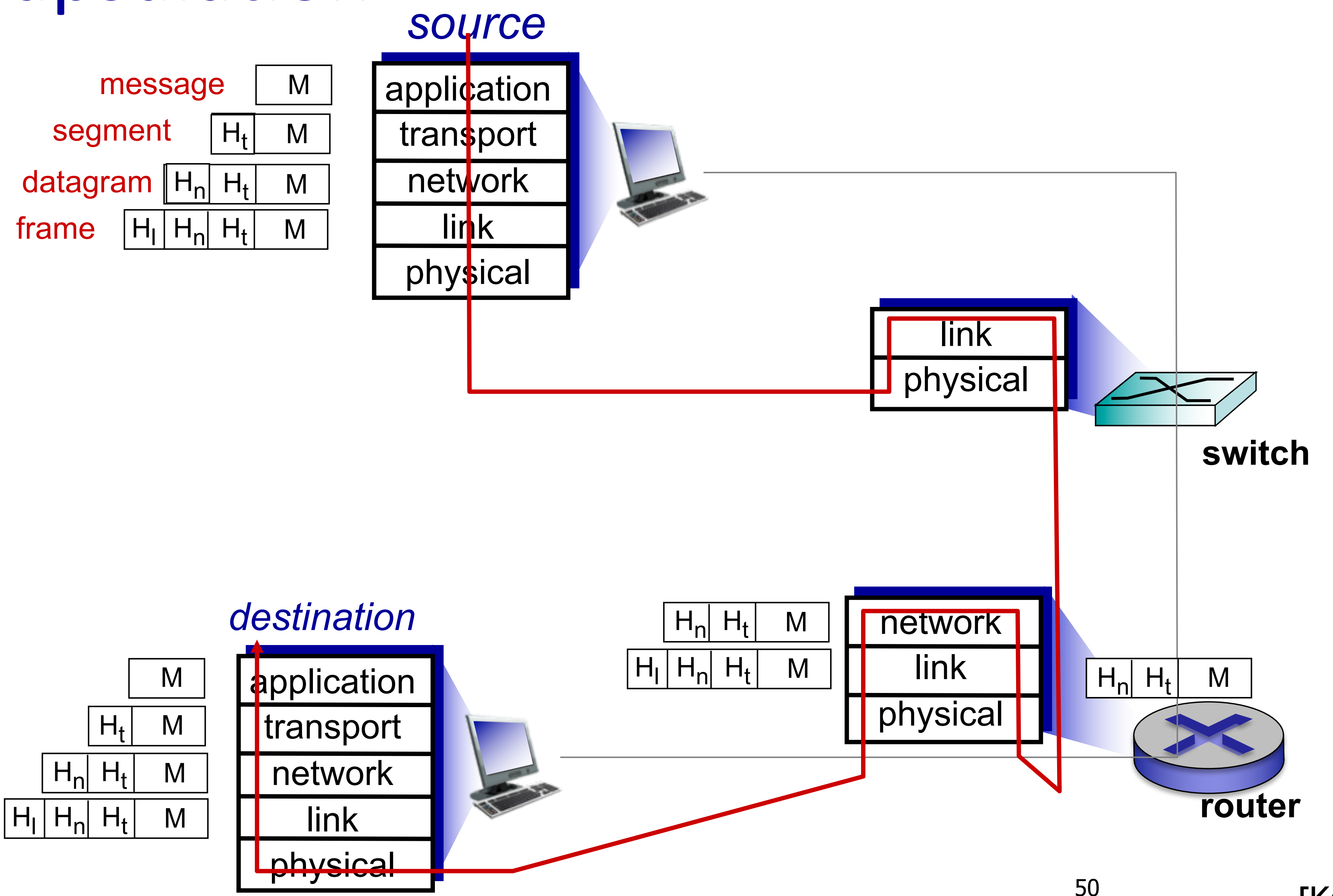


# Internet protocol stack




[Walrand and Parekh]

# Encapsulation






[Kurose and Ross]

# End-to-End Principle:

- “tasks should not be performed by routers if they can be performed by the end devices”
  - e.g., reliable transfer, congestion control are implemented by end systems
- stateless routers: router considers one packet at a time, no connection information -> robustness & scalability

# Why layering?

- simplify design
  - explicit structure allows identification of relationship of complex system's pieces
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system
- layering considered harmful?
  - overhead
  - loss of efficiency: "cross-layer" approaches