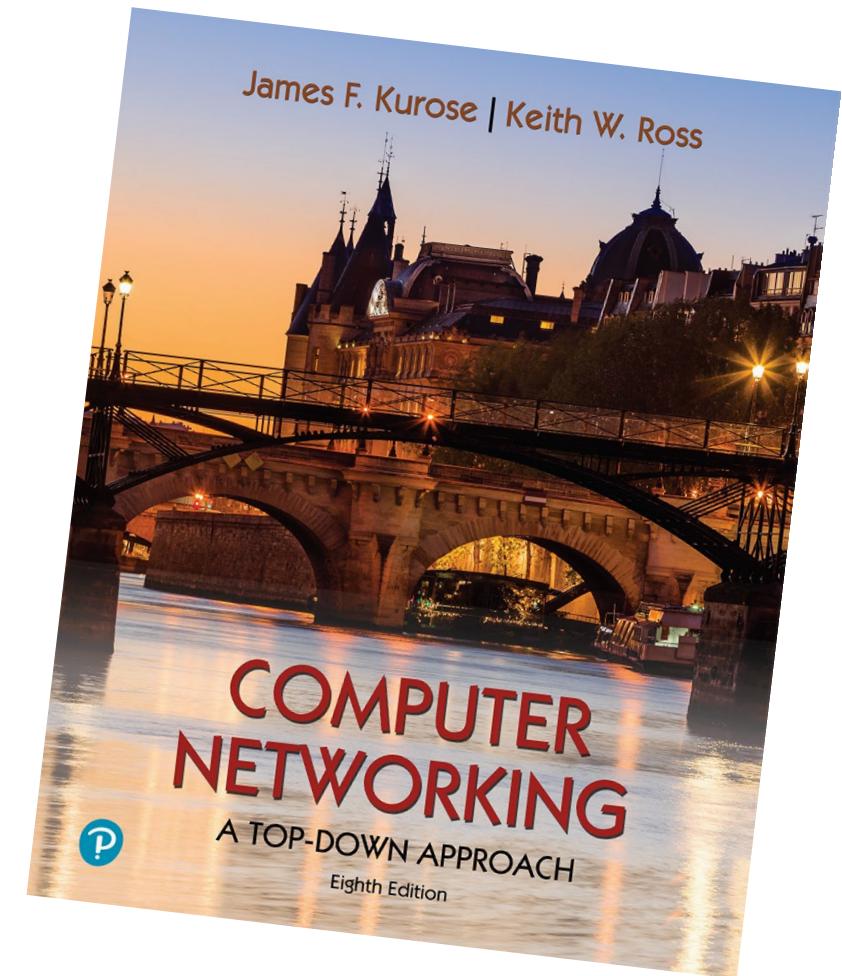
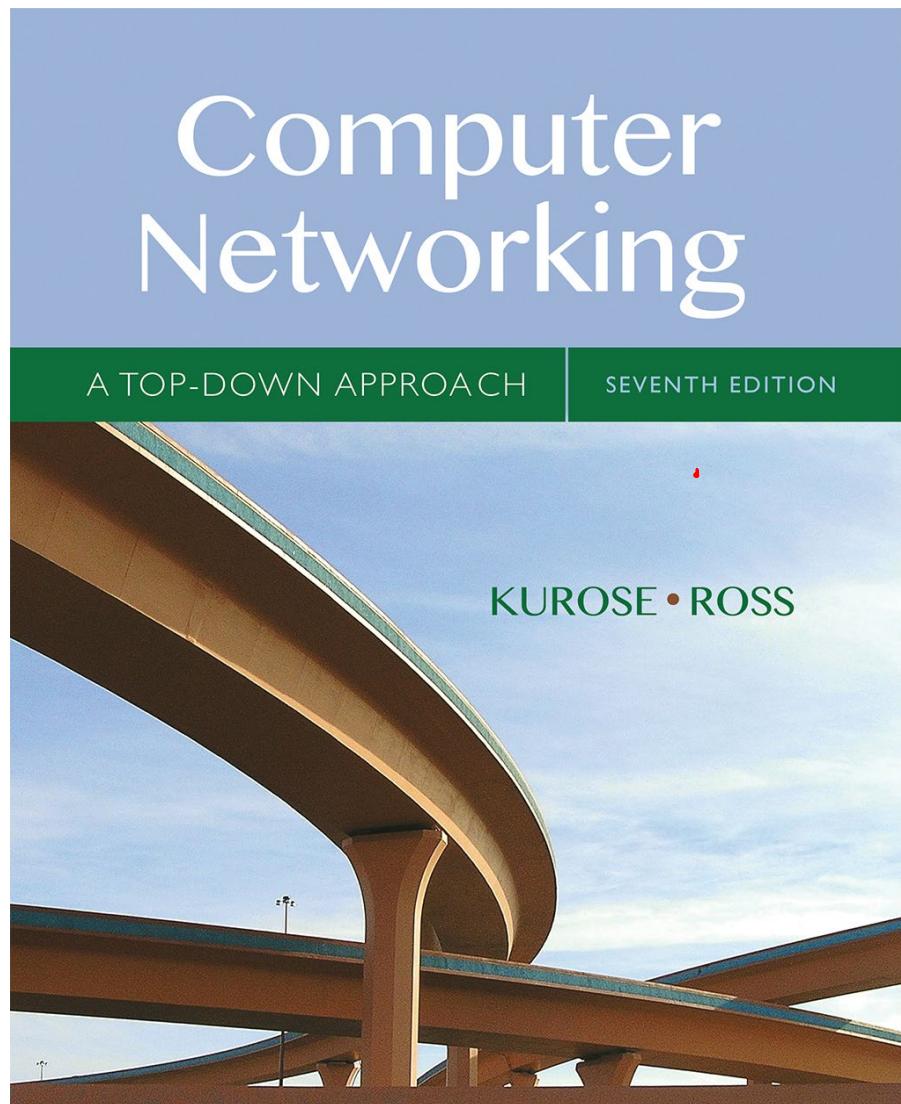


# CS5222 Computer Networks and Internets

Prof Weifa Liang

[weifa.liang@cityu-dg.edu.cn](mailto:weifa.liang@cityu-dg.edu.cn)

Slides based on book *Computer Networking: A Top-Down Approach.*



Slides based on book Computer Networking: A Top-Down Approach.

# Today's Lecture

## Goal:

- Overview of “big picture,” introduction to terminology
  - more depth & details *later*
- Approach:
  - use Internet as example



## Roadmap:

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

# The Internet: a “nuts and bolts” view

微观大意



Billions of connected computing *devices*:

- *hosts = end systems*
- running *network apps* at Internet's “edge”

**Packet switches**: forward packets (chunks of data)

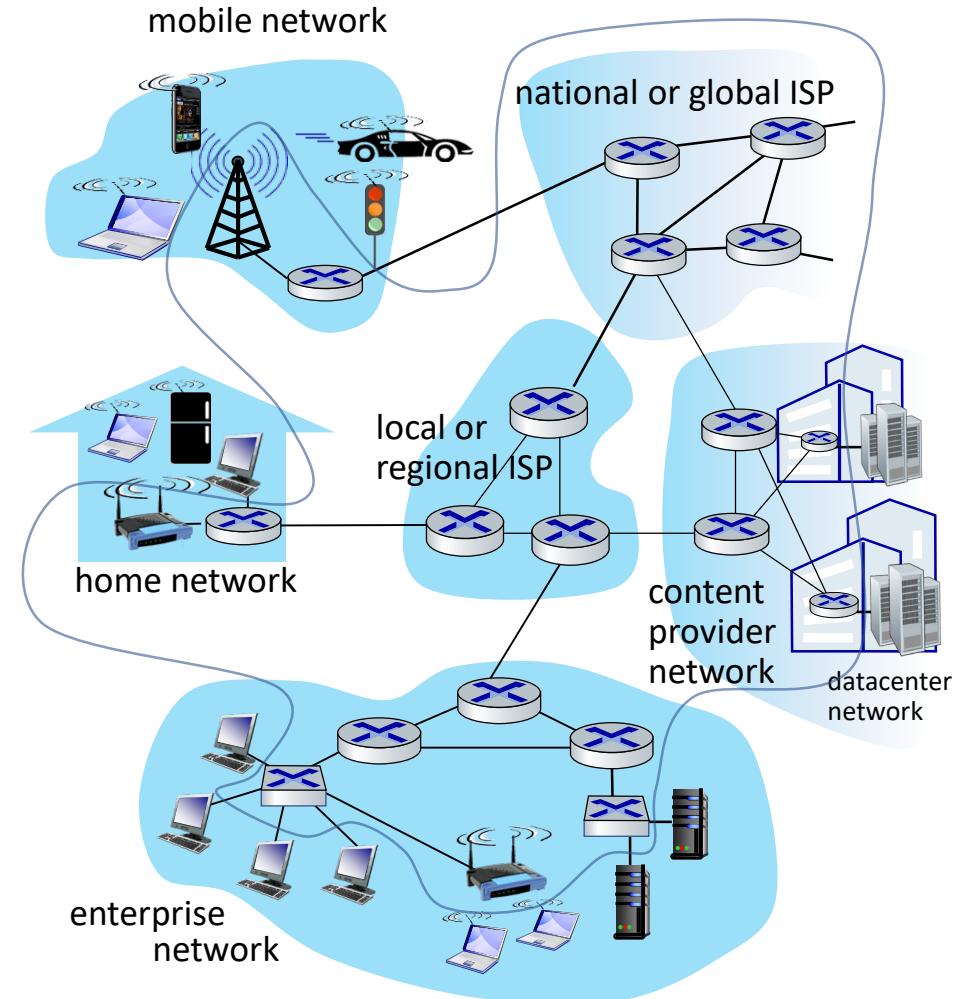
- *routers, switches*

**Communication links**

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

**Networks**

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



# “Fun” Internet-connected devices



Amazon Echo



Internet refrigerator



Security Camera



Internet phones



Slingbox: remote  
control cable TV  
电视盒子



sensorized,  
bed  
mattress  
床垫



Web-enabled toaster +  
weather forecaster



AR devices



Fitbit



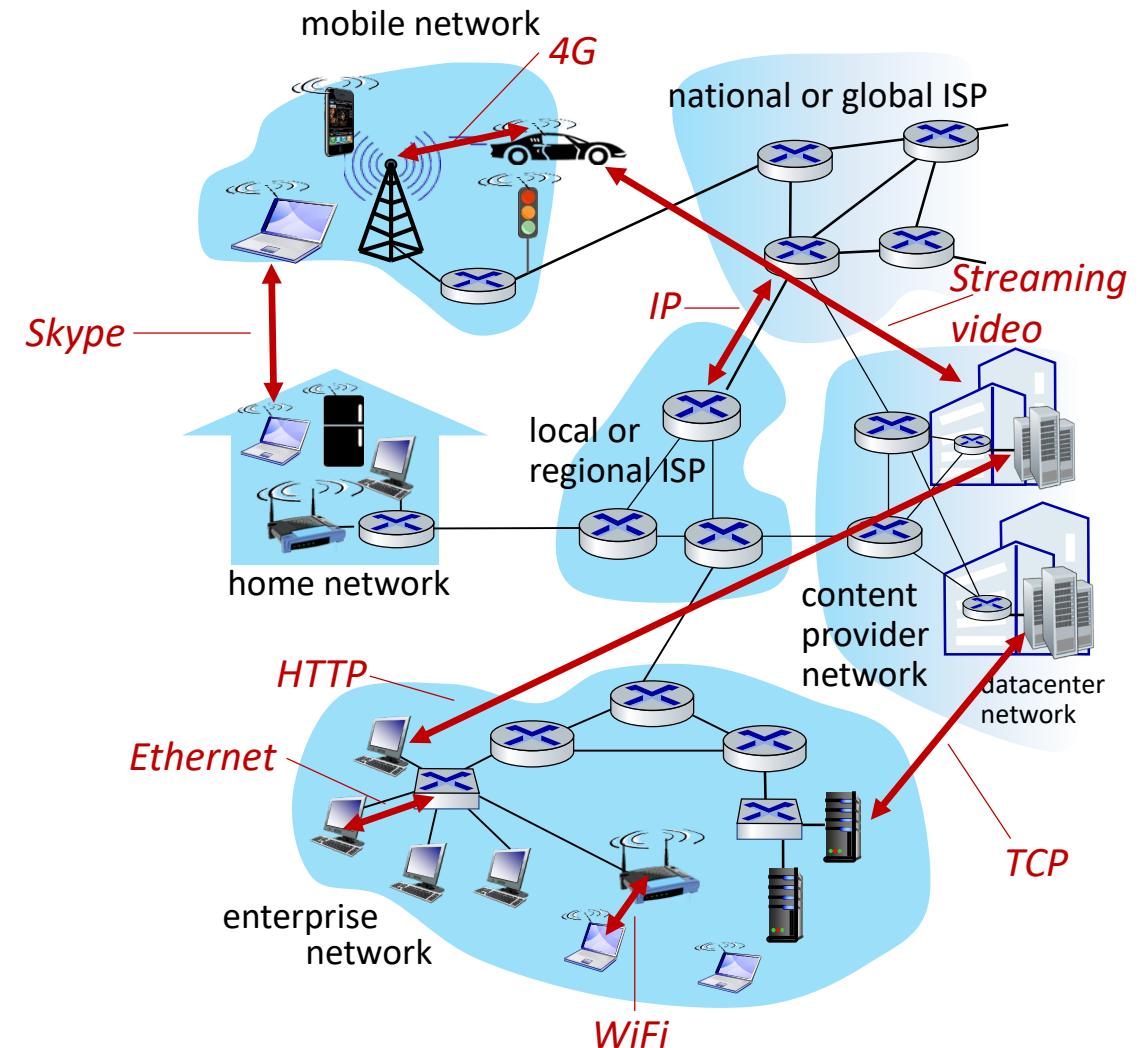
Pacemaker & Monitor  
起搏器



Tweet-a-watt:  
monitor energy use

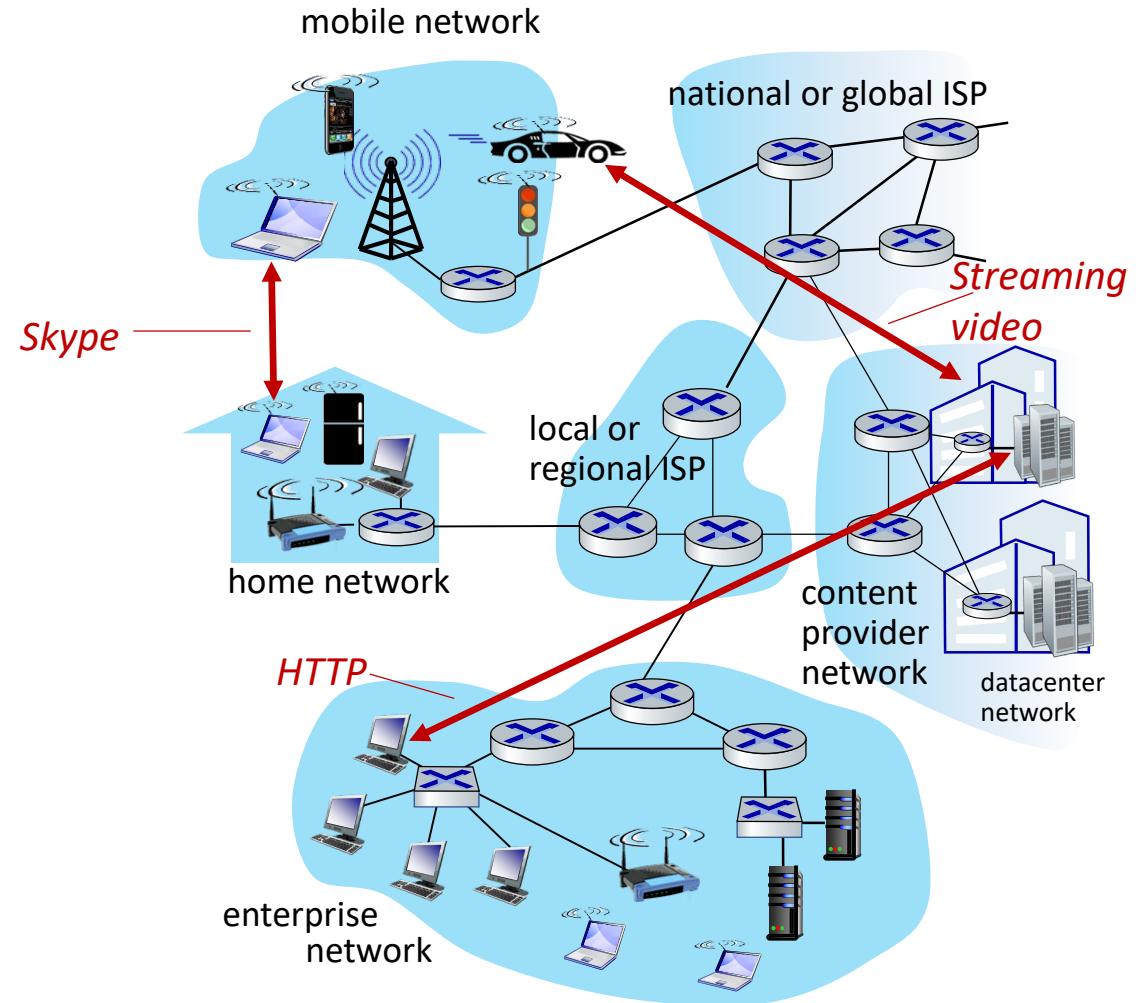
# The Internet: a “nuts and bolts” view

- *Internet: “network of networks”*
  - Interconnected **ISPs** Internet Service Provider
- *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 “service” view

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



# What's a protocol?

## *Human protocols:*

- “what’s the time?”
- “I have a question”
- introductions

... specific messages sent  
... specific actions taken  
when message received,  
or other events

## *Network protocols:*

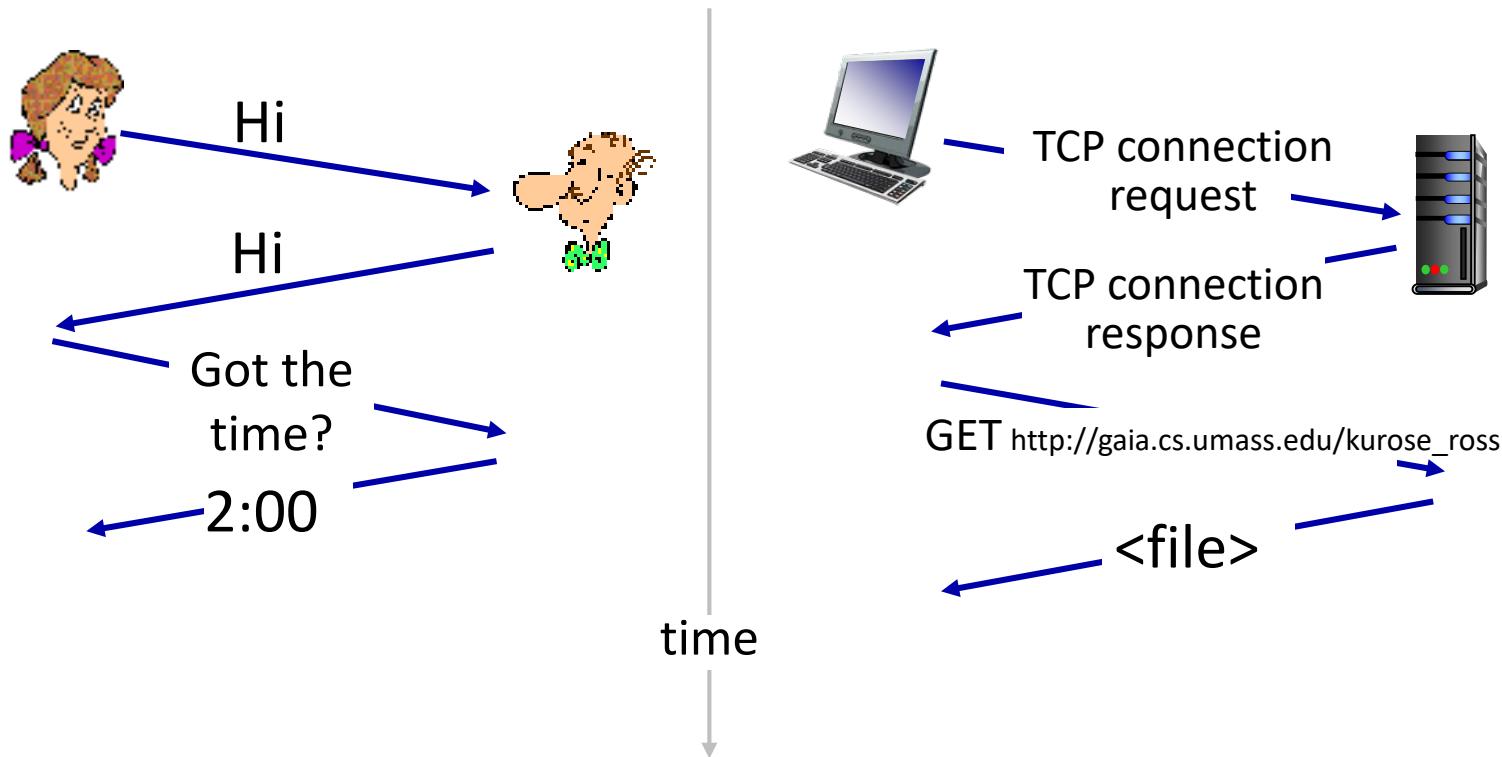
- computers (devices) rather than humans
- all communication activity in Internet governed by protocols

### *Protocols define...*

- *the format,*
- *order of messages sent and received among network entities,*
- *actions taken on msg transmission and/or msg receipt*

# Example

A human protocol and a computer network protocol:



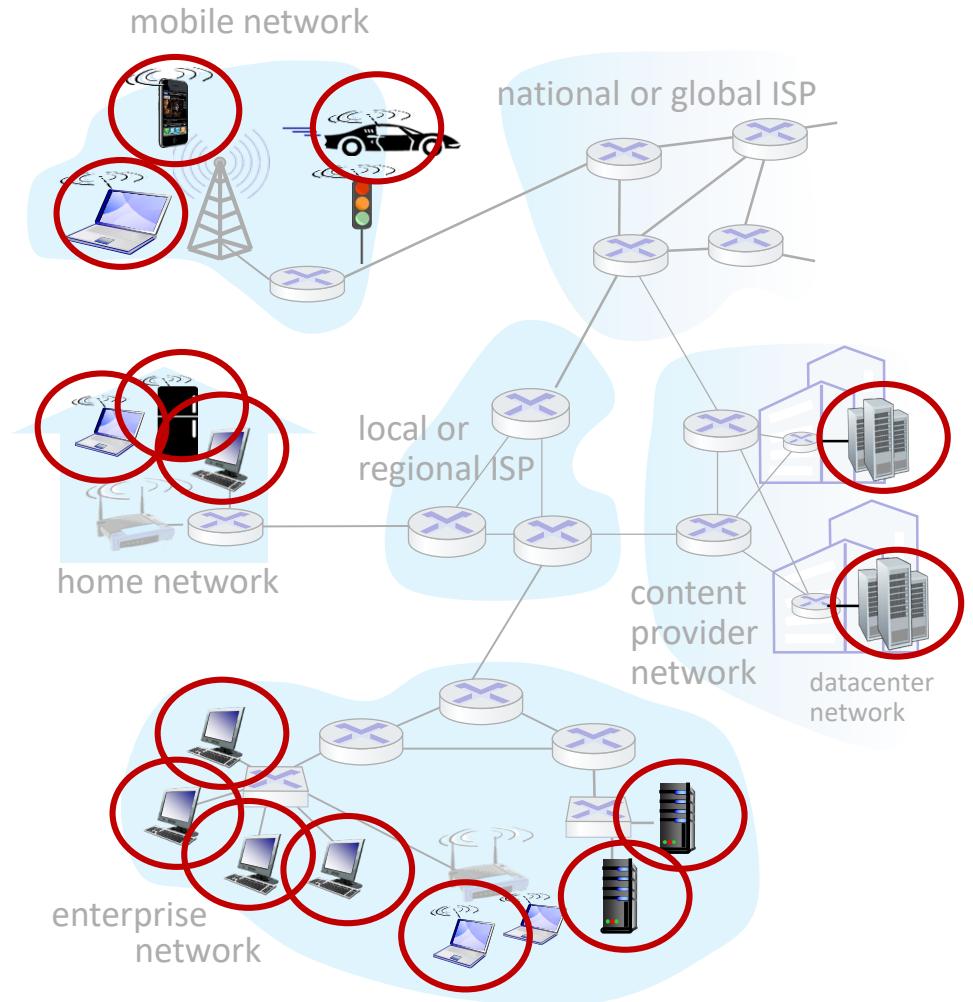
# Roadmap

- 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

# A closer look at Internet structure

## Network edge:

- hosts: clients and servers
- servers in data centers



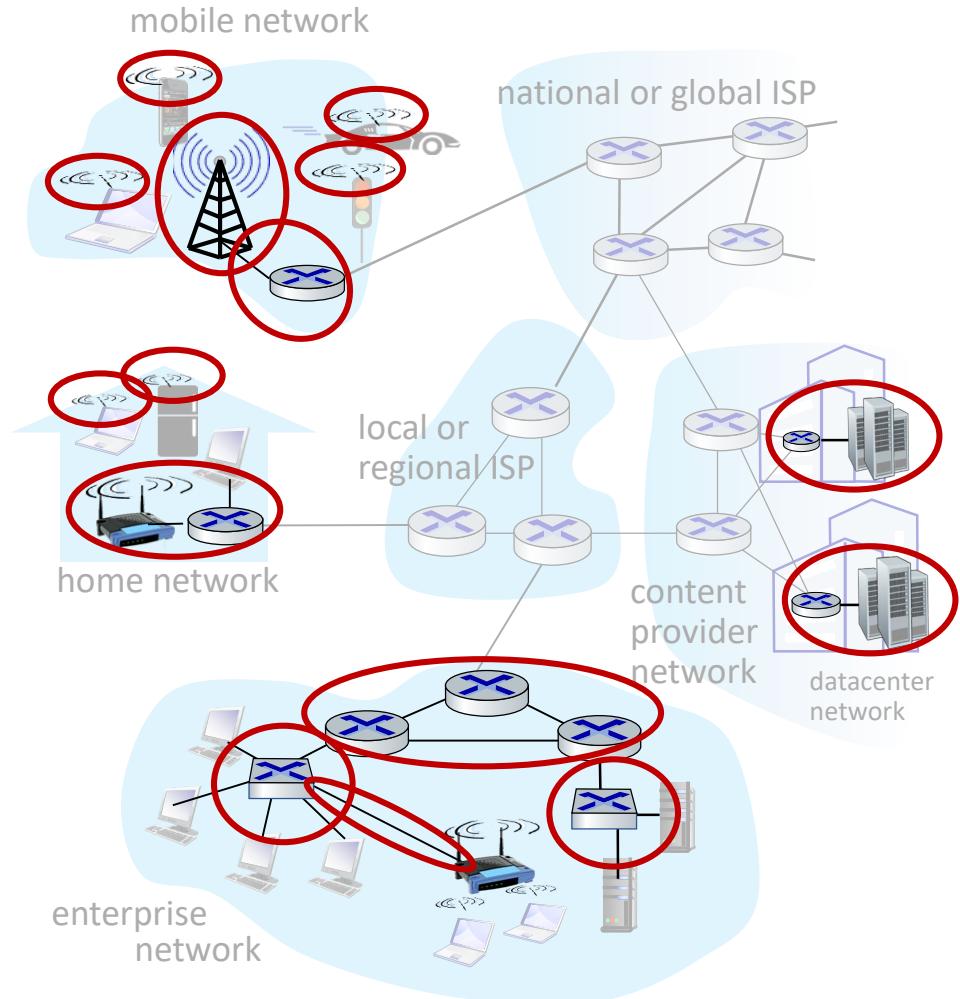
# A closer look at Internet structure

## Network edge:

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

## Access networks, physical media:

- wired, wireless communication links



# A closer look at Internet structure

## Network edge:

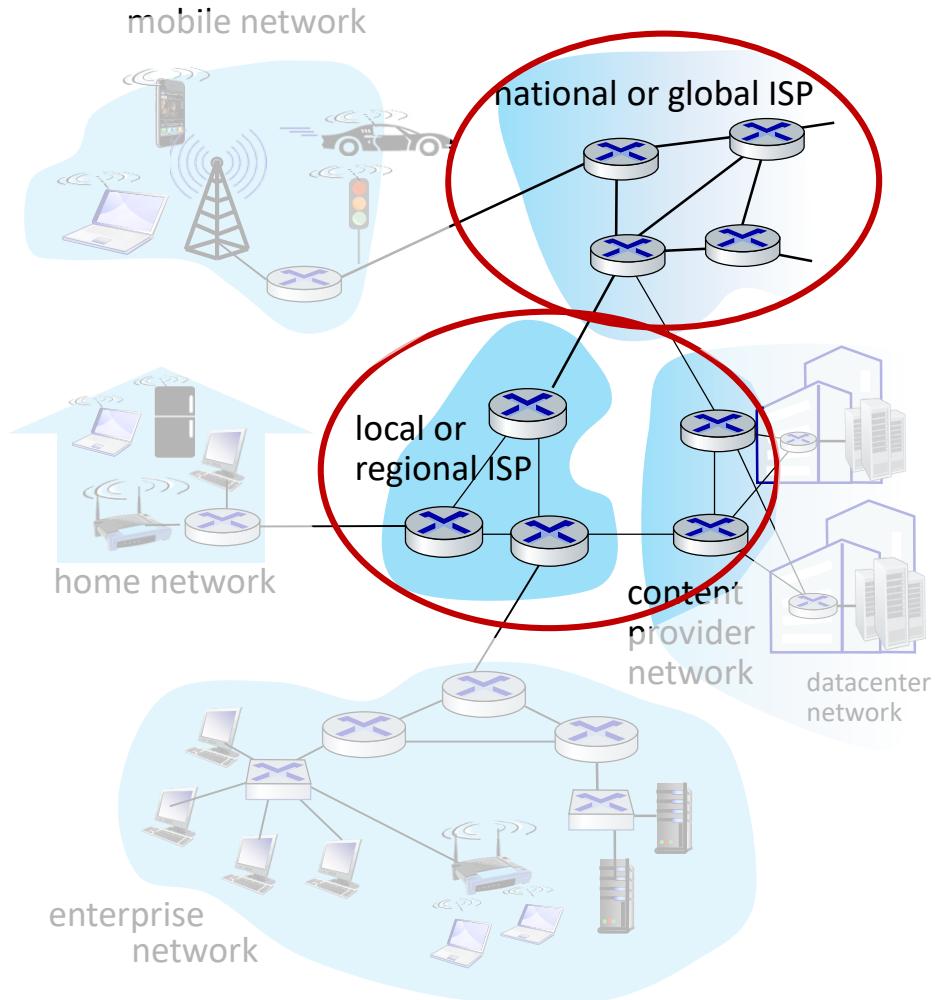
- hosts: clients and servers
- servers often in data centers

## Access networks, physical media:

- wired, wireless communication links

## Network core:

- interconnected routers
- network of networks



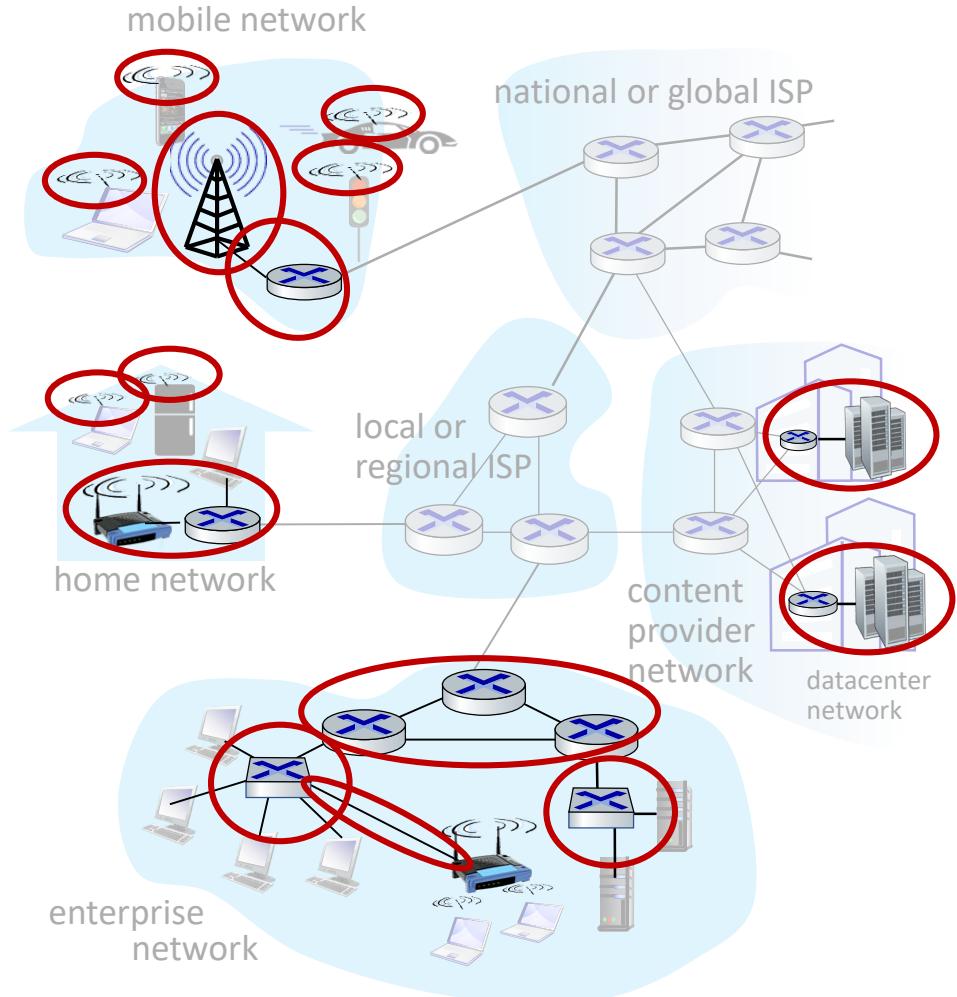
# Access networks and physical media

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

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

*What to look for:*

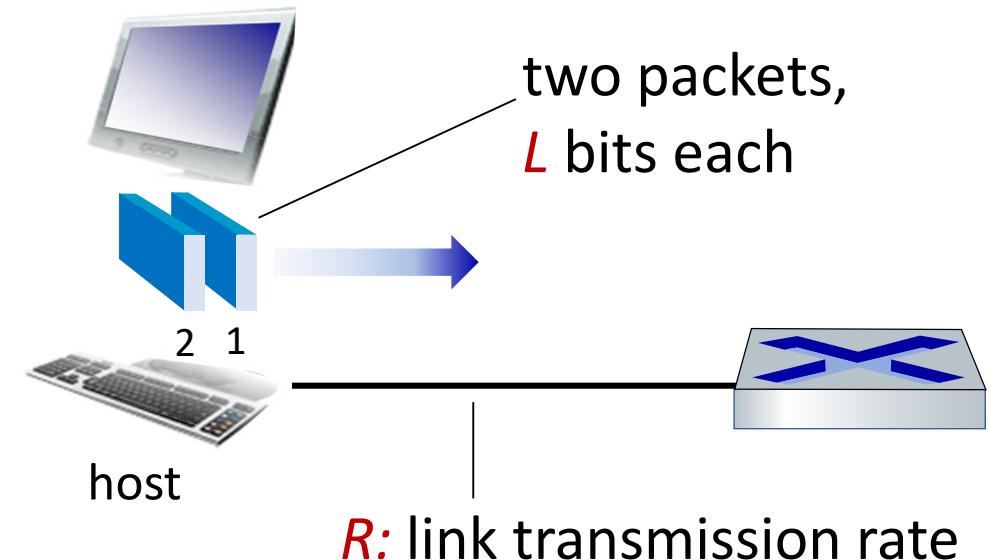
- transmission rate (bits per second) of access network?
- shared or dedicated access among users? 专用的



# 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*



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

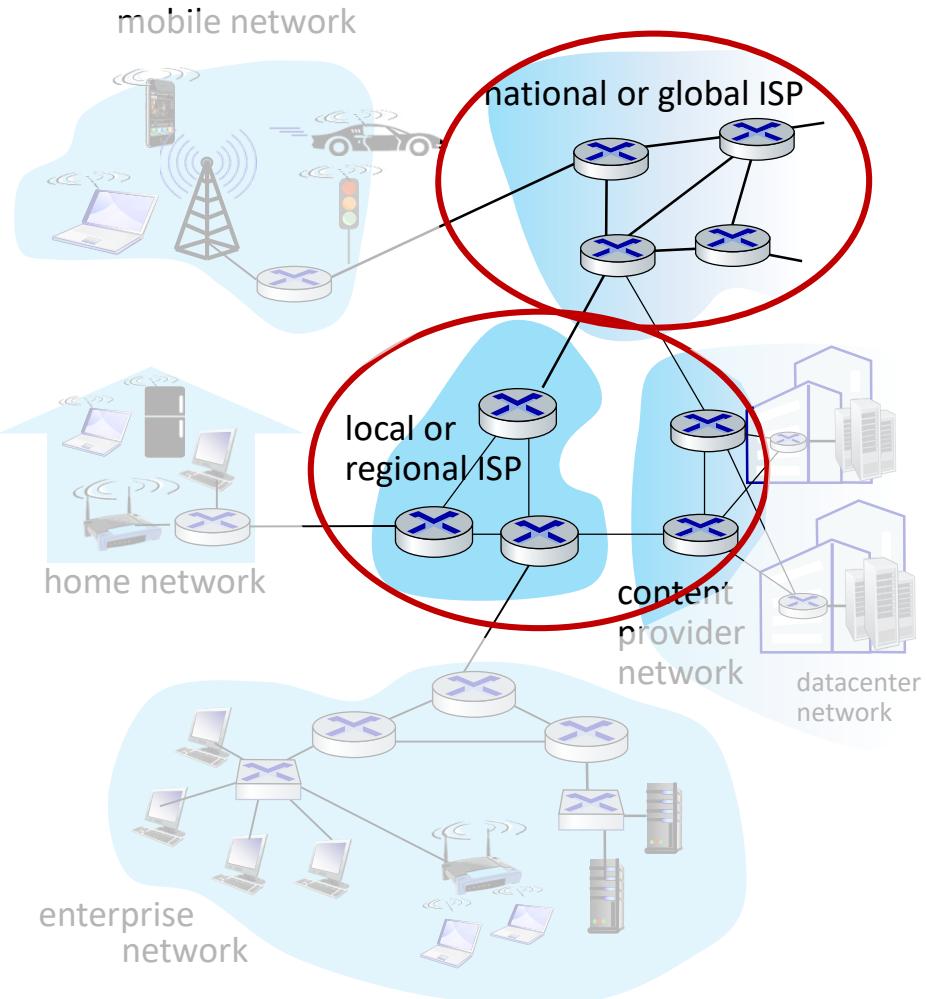
# Roadmap

- 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

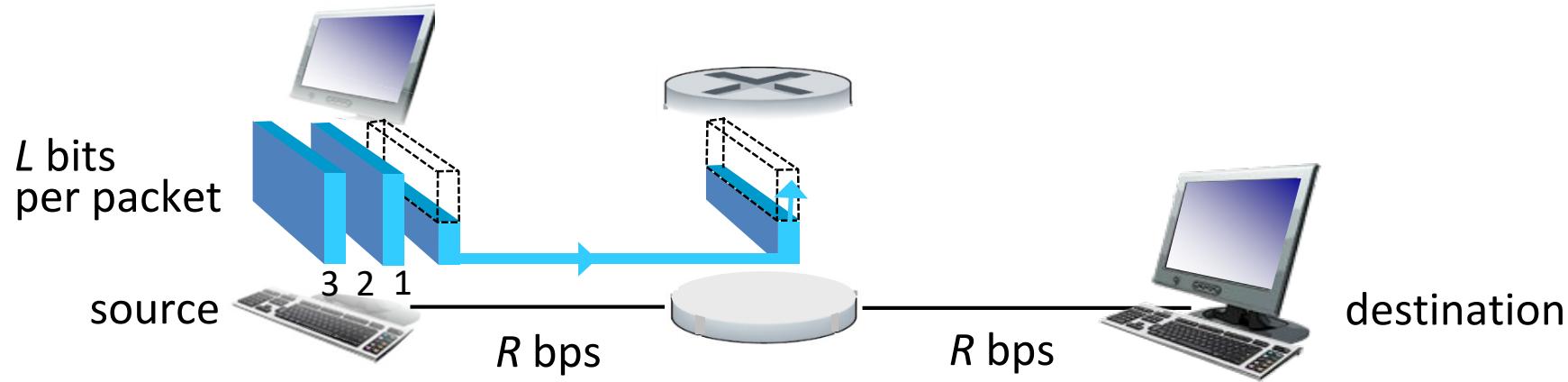
# The network core

- mesh of interconnected routers
- **packet-switching**: hosts split 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

这个forward是个动词



# Packet-switching: store-and-forward

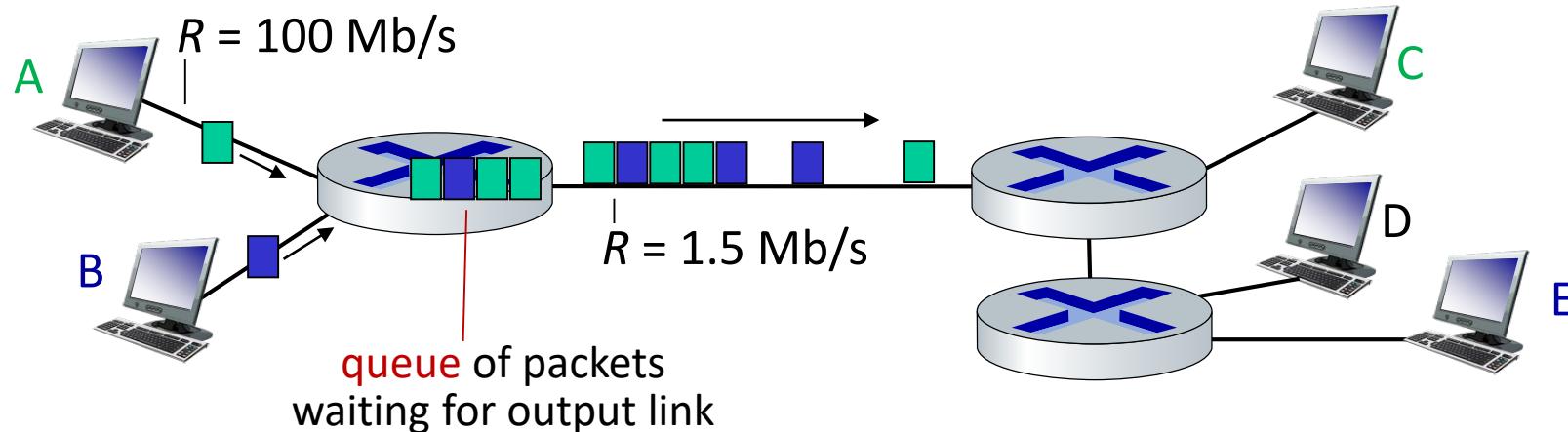


- **Store and forward:** entire packet must arrive at router before it can be transmitted on next link
- **Transmission delay:** takes  $L/R$  seconds to transmit (push out)  $L$ -bit packet into link at  $R$  bps
- **End-end delay:**  $2L/R$  (above), assuming a zero propagation delay (more on delay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

*One-hop numerical example:*

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

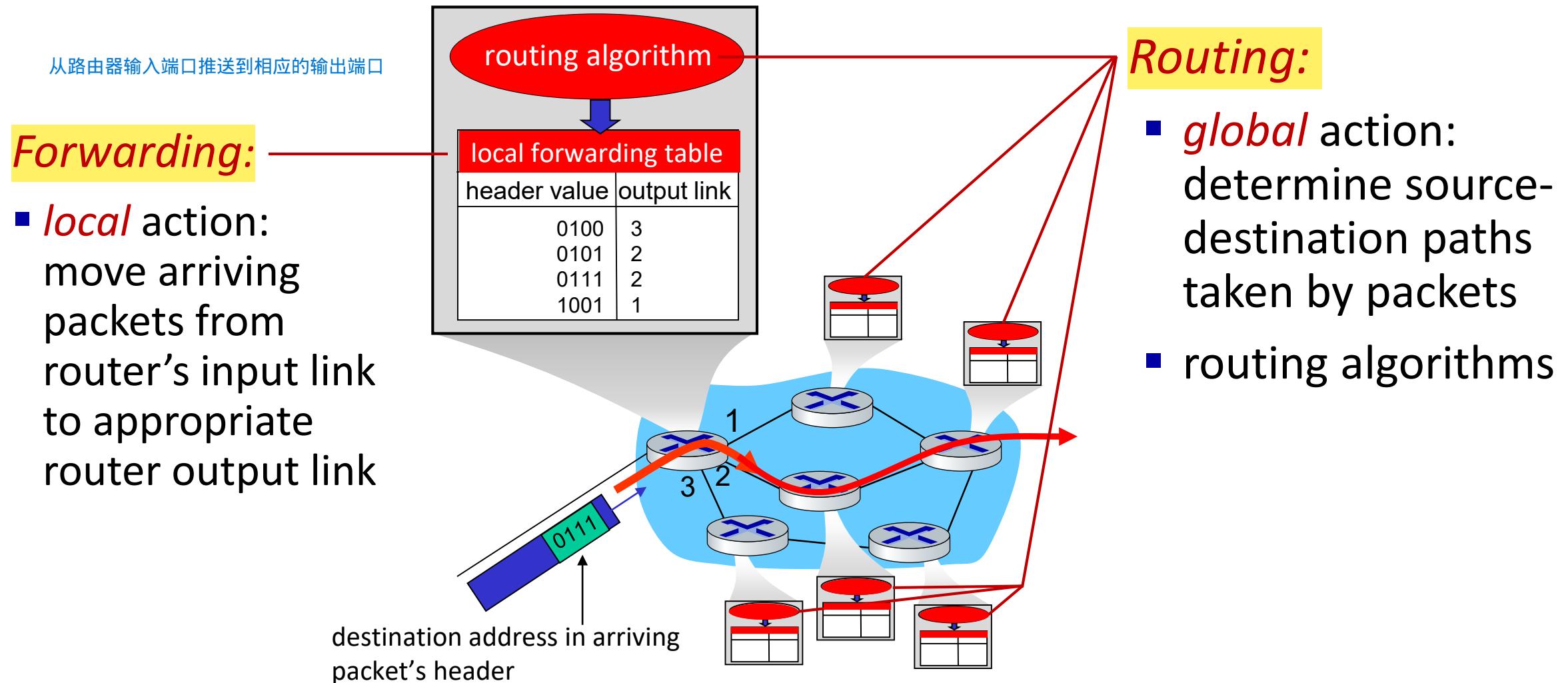
# Packet-switching: queuing delay, loss



*Packet queuing and loss:* if arrival rate (in bps) to link exceeds transmission rate (bps) of link for a 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

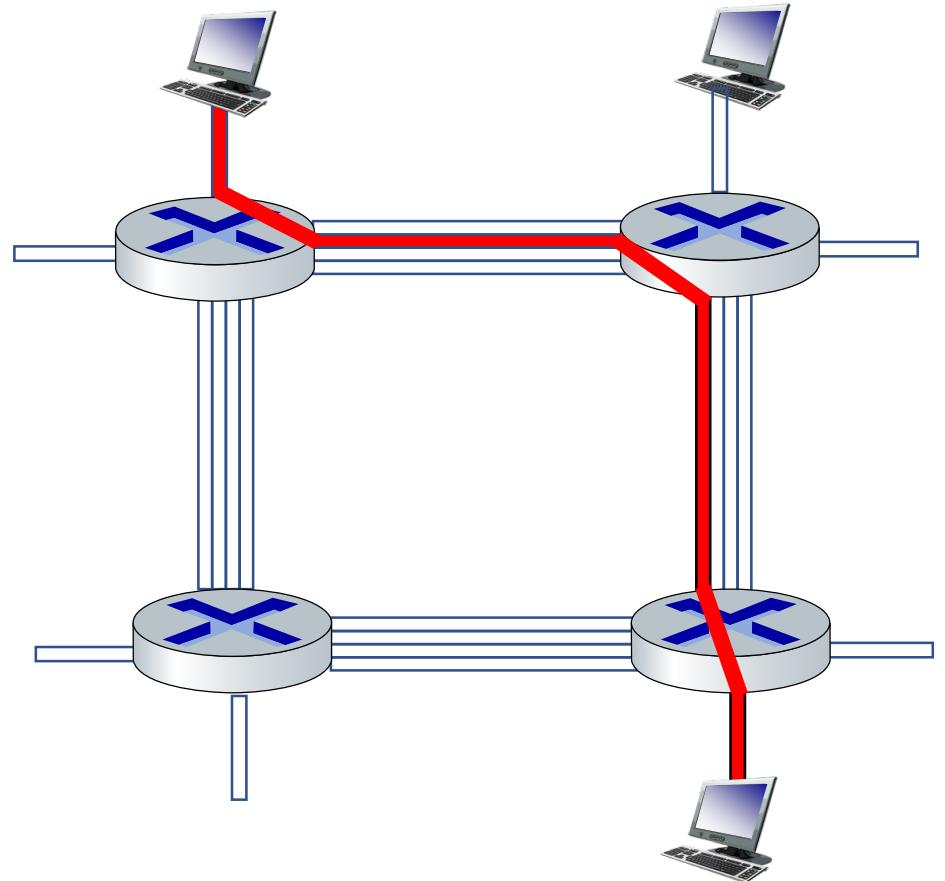
# Two key network-core functions



# Circuit Switching: alternative to packet switching

end-end resources reserved for “call” between source and destination

- commonly used in traditional telephone networks
- In Figure: 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
  - **guaranteed** performance
- circuit segment **idle** if not used by call (no sharing)  
闲置



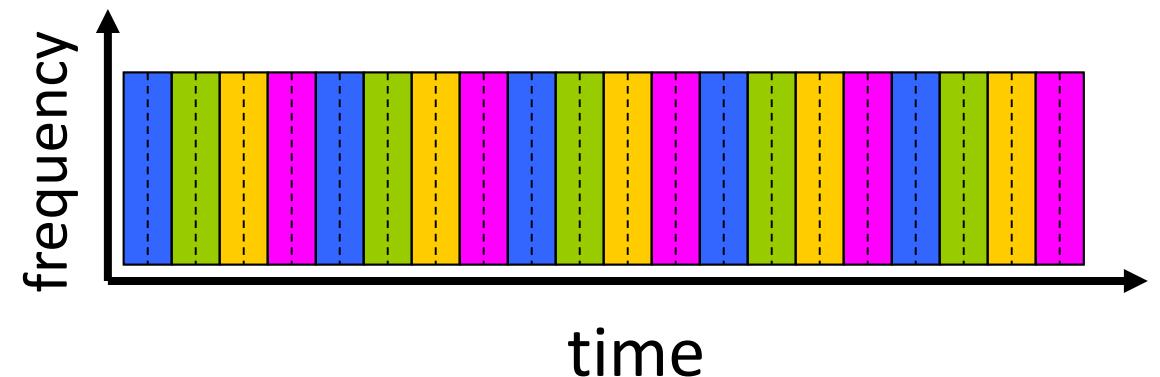
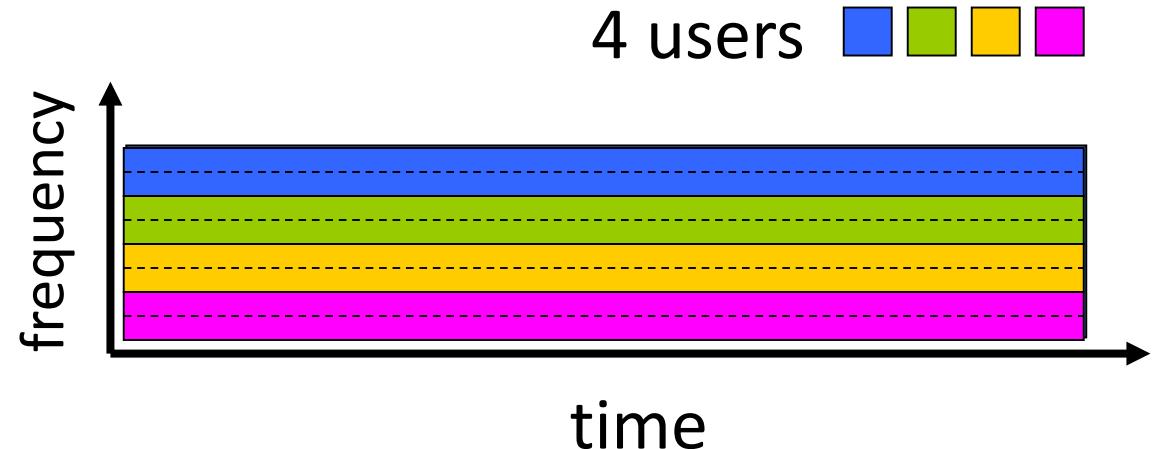
# Circuit switching: FDM and TDM

## Frequency Division Multiplexing (FDM)

- optical, electromagnetic frequencies divided into (narrow) frequency bands
- each call allocated its own band, can transmit at max rate of that narrow band

## Time Division Multiplexing (TDM)

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



# Numerical example

- How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?
  - All links are 1.536 Mbps
  - Each link uses TDM with 24 slots/sec
  - 500 msec to establish end-to-end circuit

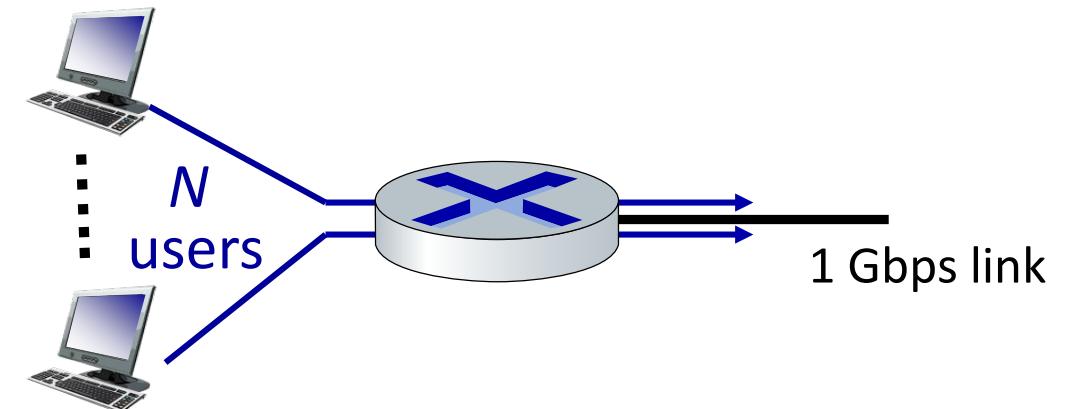
Let's work it out!

建立信道要500ms,  
然后1.536MPbps相当于被削弱成1/24，所以传输需要  
 $640,000\text{bits}/(1.536\text{Mbps}/24)=10\text{s}$   
所以要10.5s一共

# Packet switching versus circuit switching

Example:

- 1 Gb/s link
- each user:
  - 100 Mb/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?*

→ *packet switching allows more users to use network!*

# Packet switching versus circuit switching

Is packet switching always preferable?

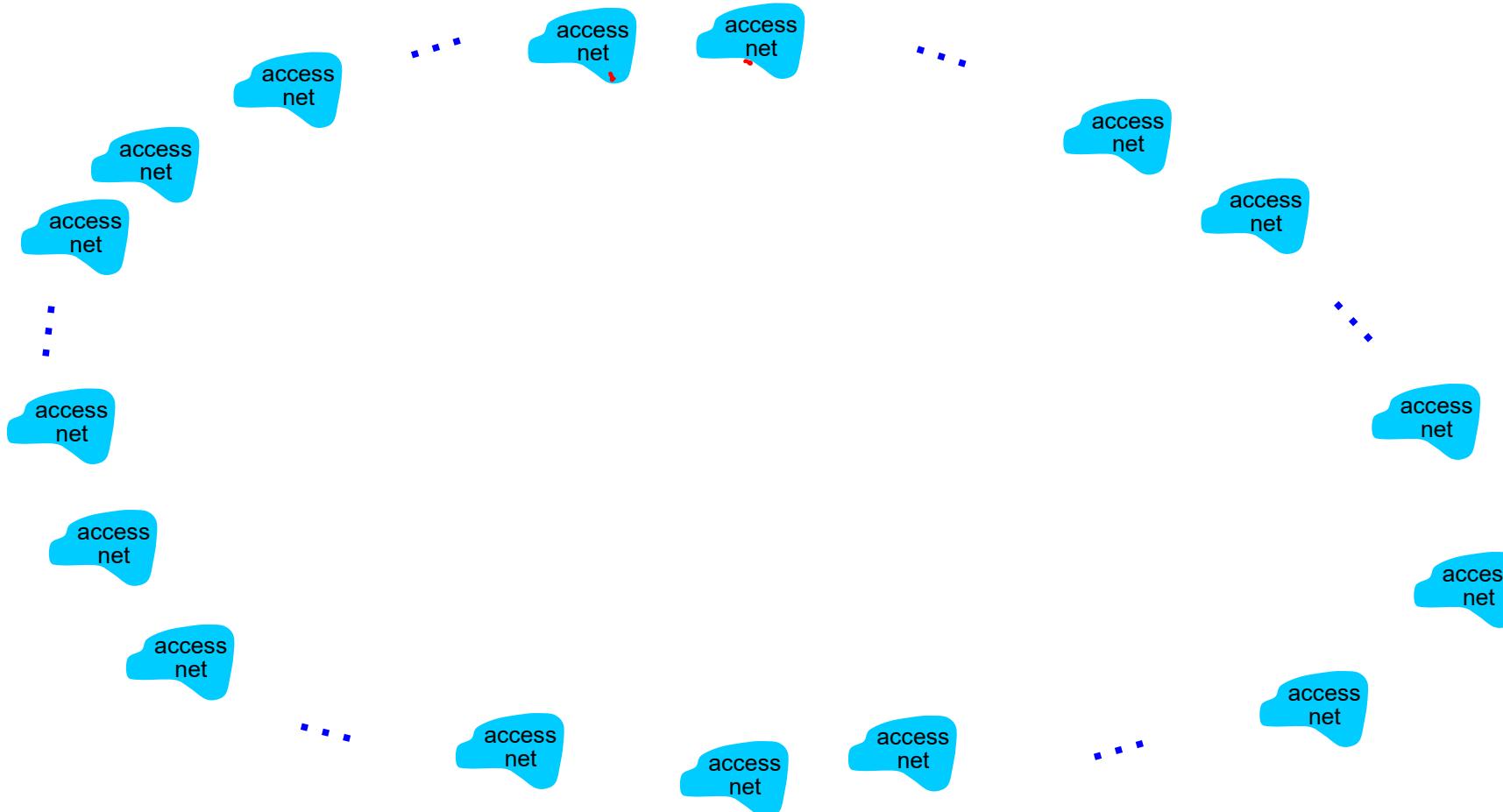
- great for “bursty” data – sometimes has data to send, but at other times not
  - resource sharing
  - simpler, no call setup
- **excessive congestion** 过度拥挤 possible: packet delay and loss due to buffer overflow
  - protocols needed for reliable data transfer, congestion control
- **Q: How to provide circuit-like behavior?**
  - bandwidth guarantees traditionally used for audio/video applications

# Internet structure: a “network of networks”

- Hosts connect to Internet via **access Internet Service Providers (ISPs)**
  - residential, enterprise (company, university, commercial) 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

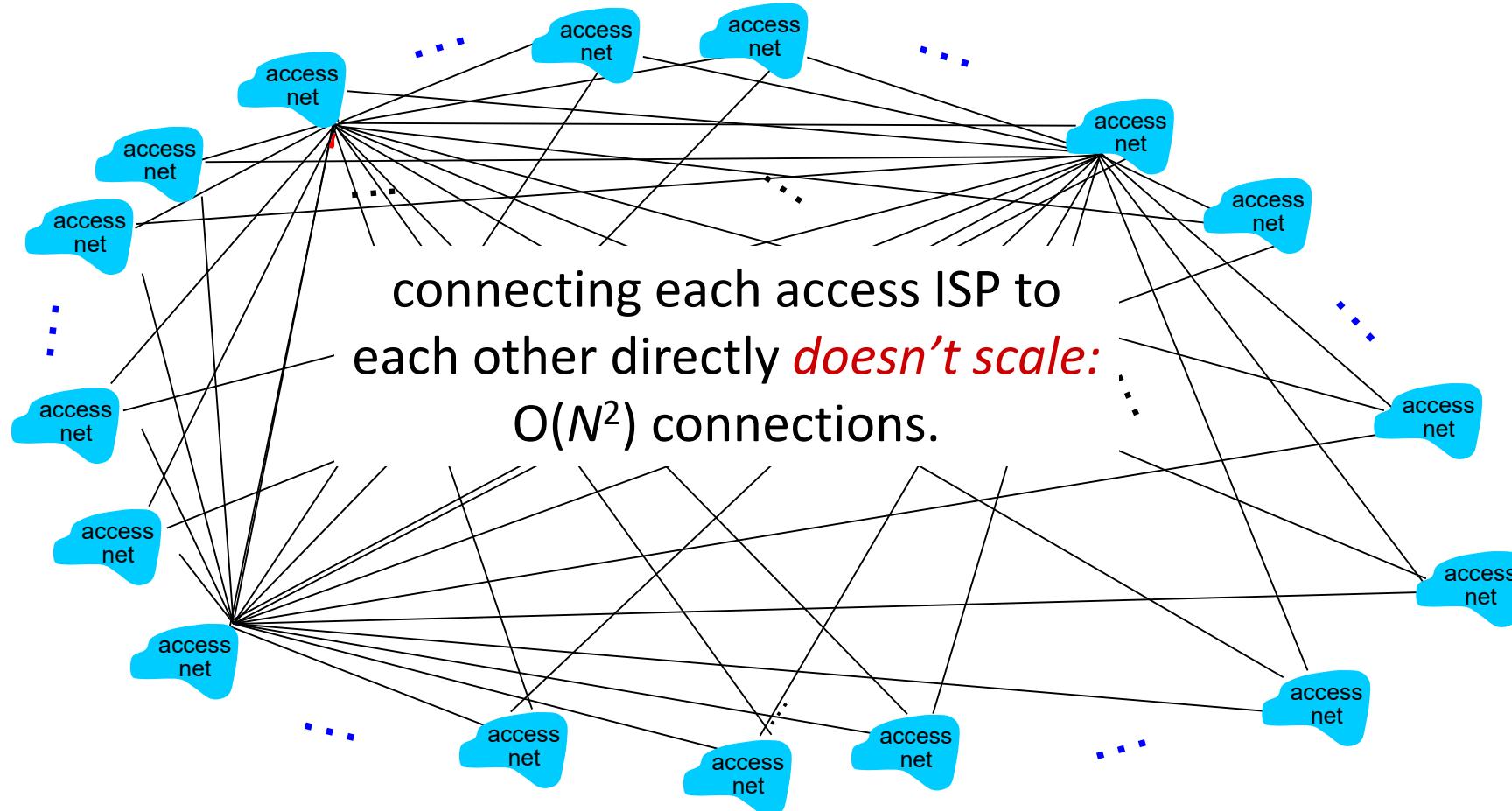
# Internet structure: a “network of networks”

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



# Internet structure: a “network of networks”

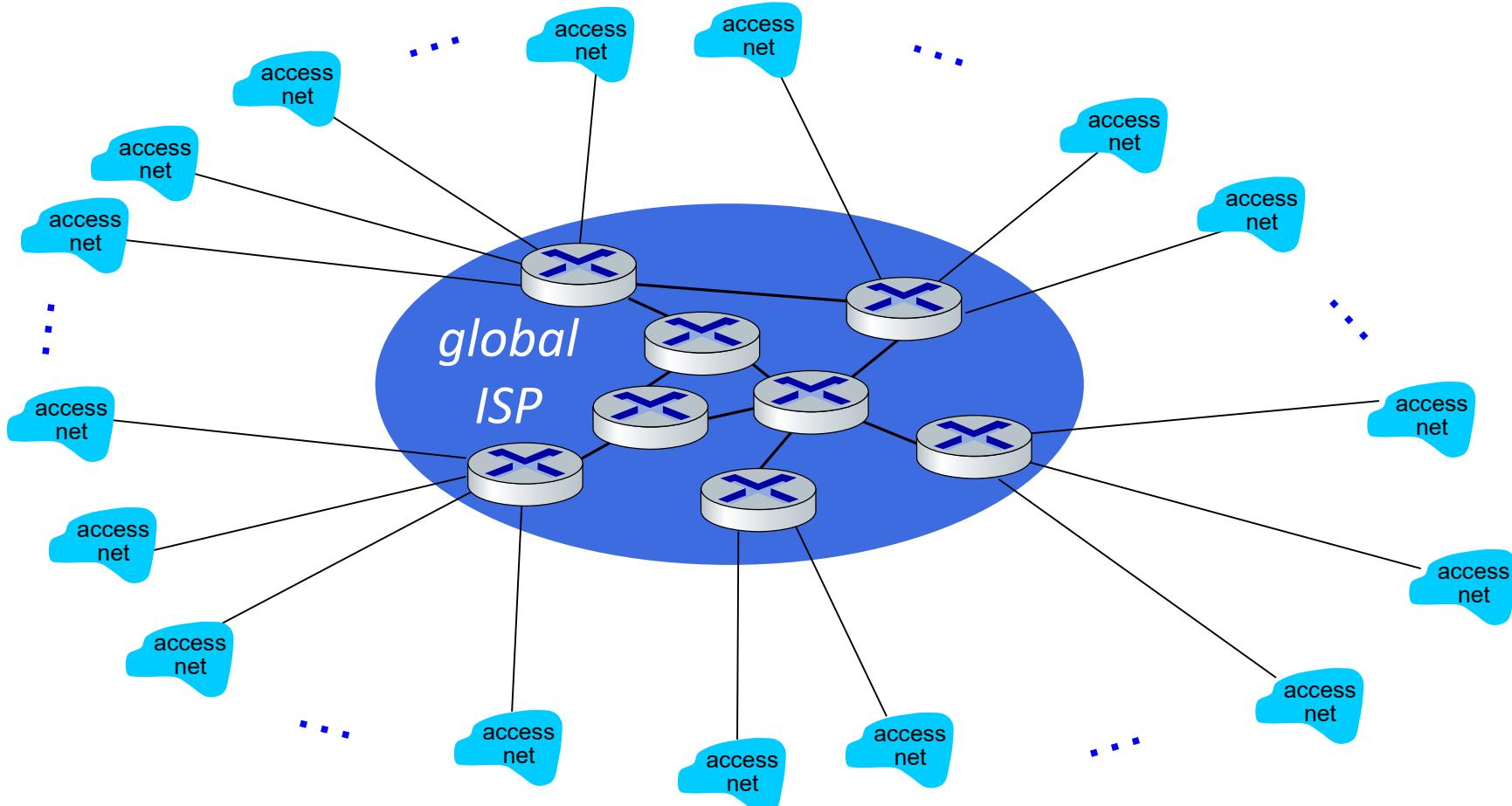
*Question:* given *millions* of access ISPs, how to connect them together?



# Internet structure: a “network of networks”

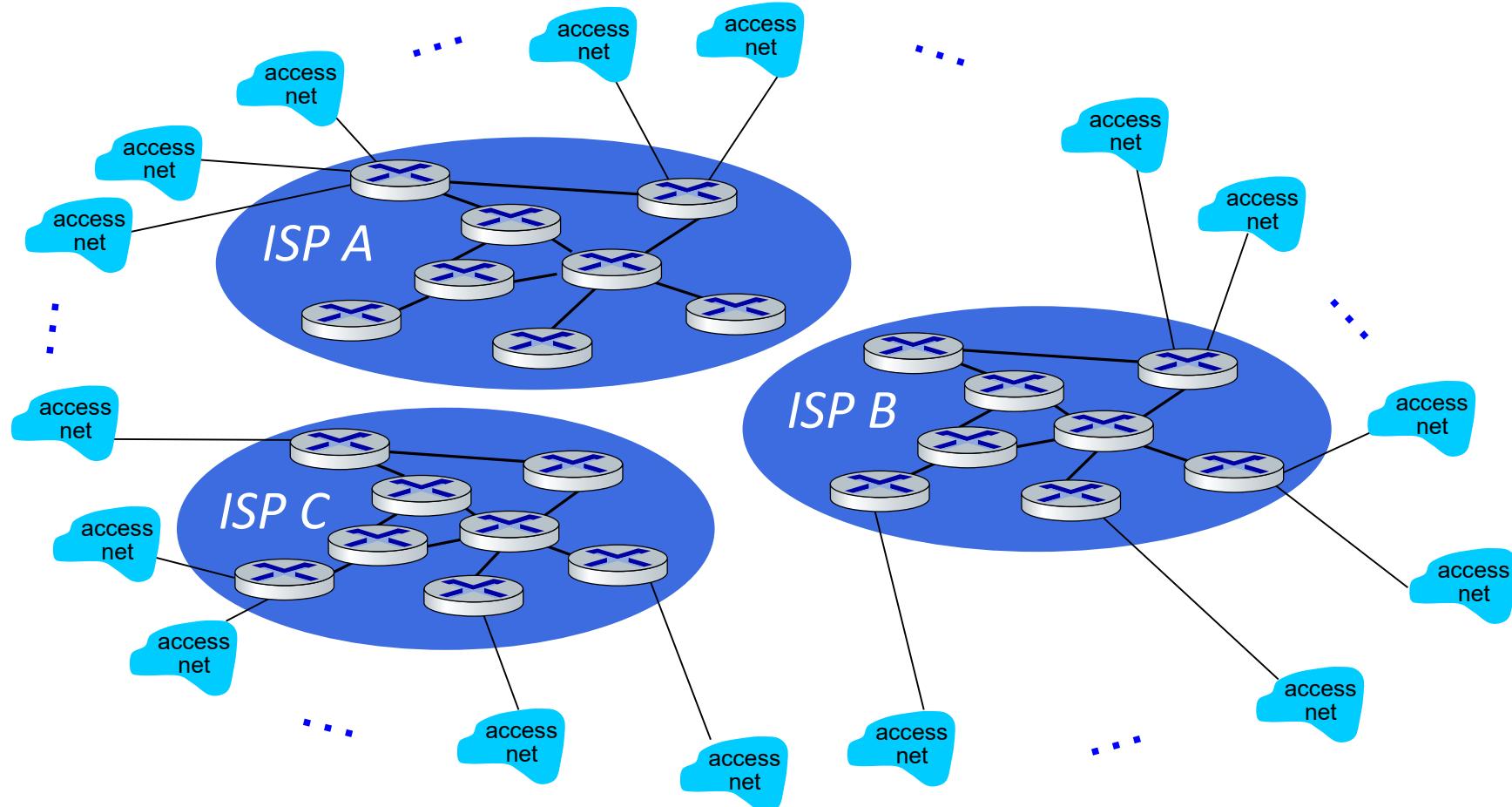
*Option: connect each access ISP to one global transit ISP?*

*Customer and provider ISPs have economic agreement.*



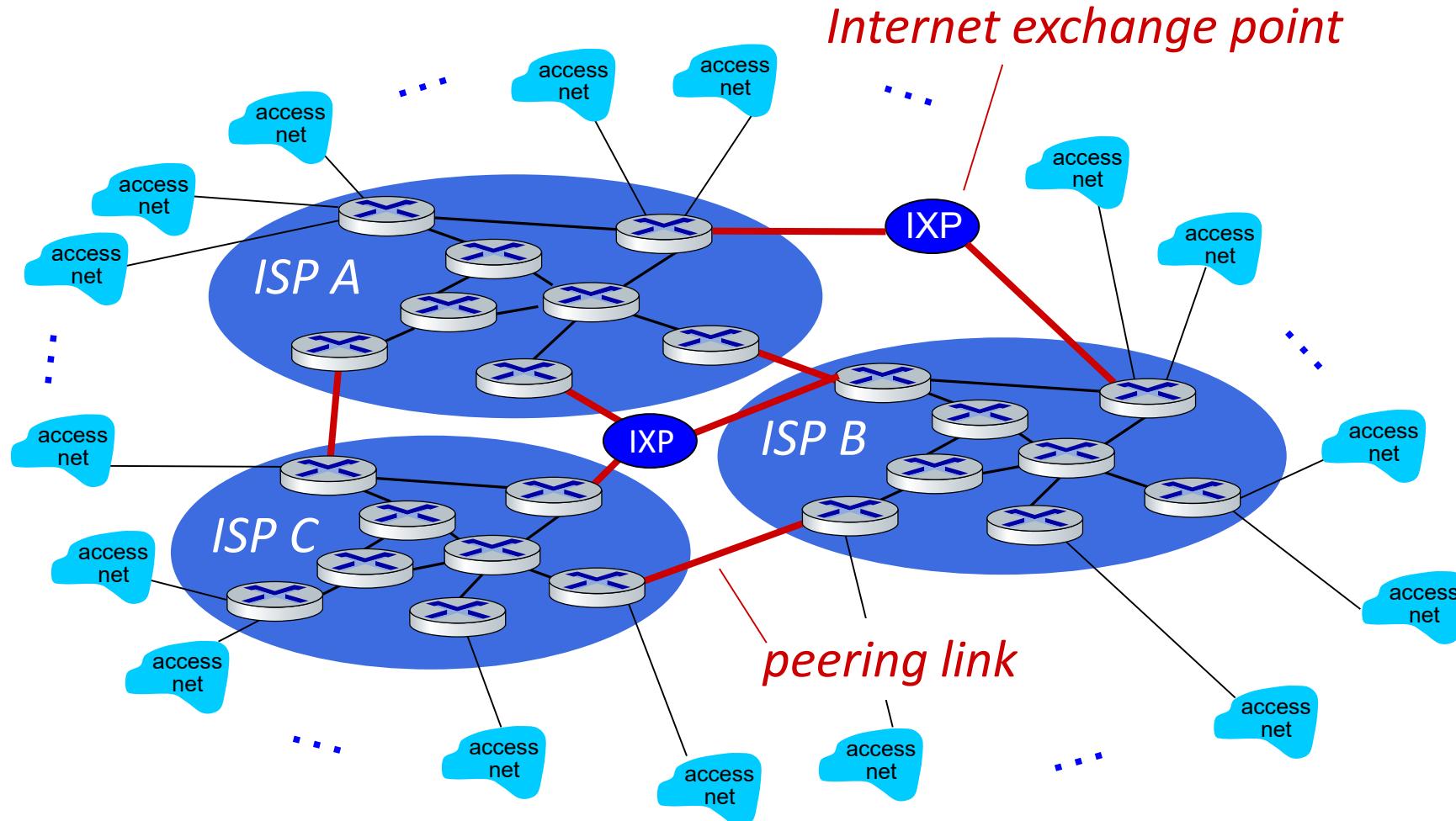
# Internet structure: a “network of networks”

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



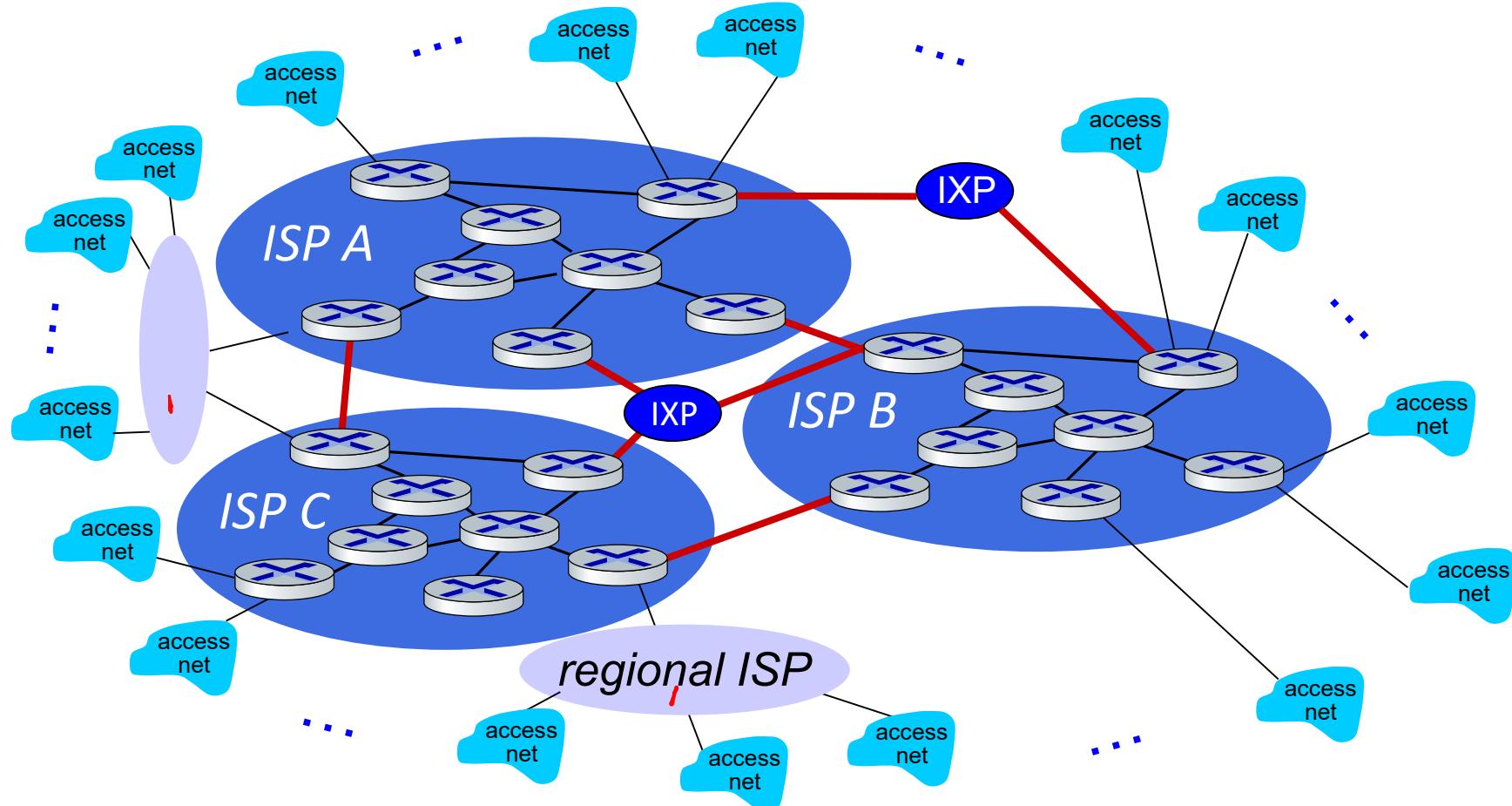
# Internet structure: a “network of networks”

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



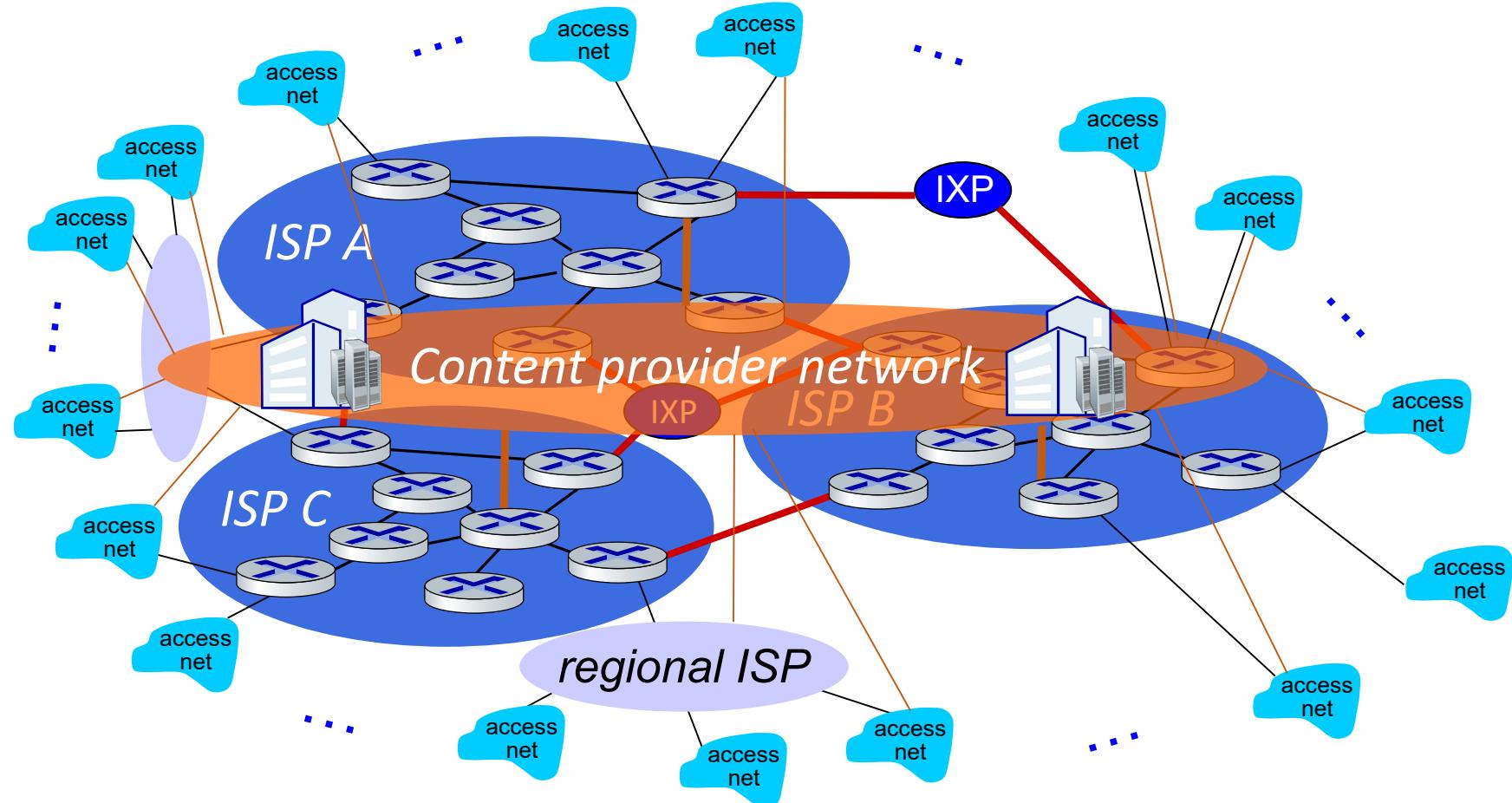
# Internet structure: a “network of networks”

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

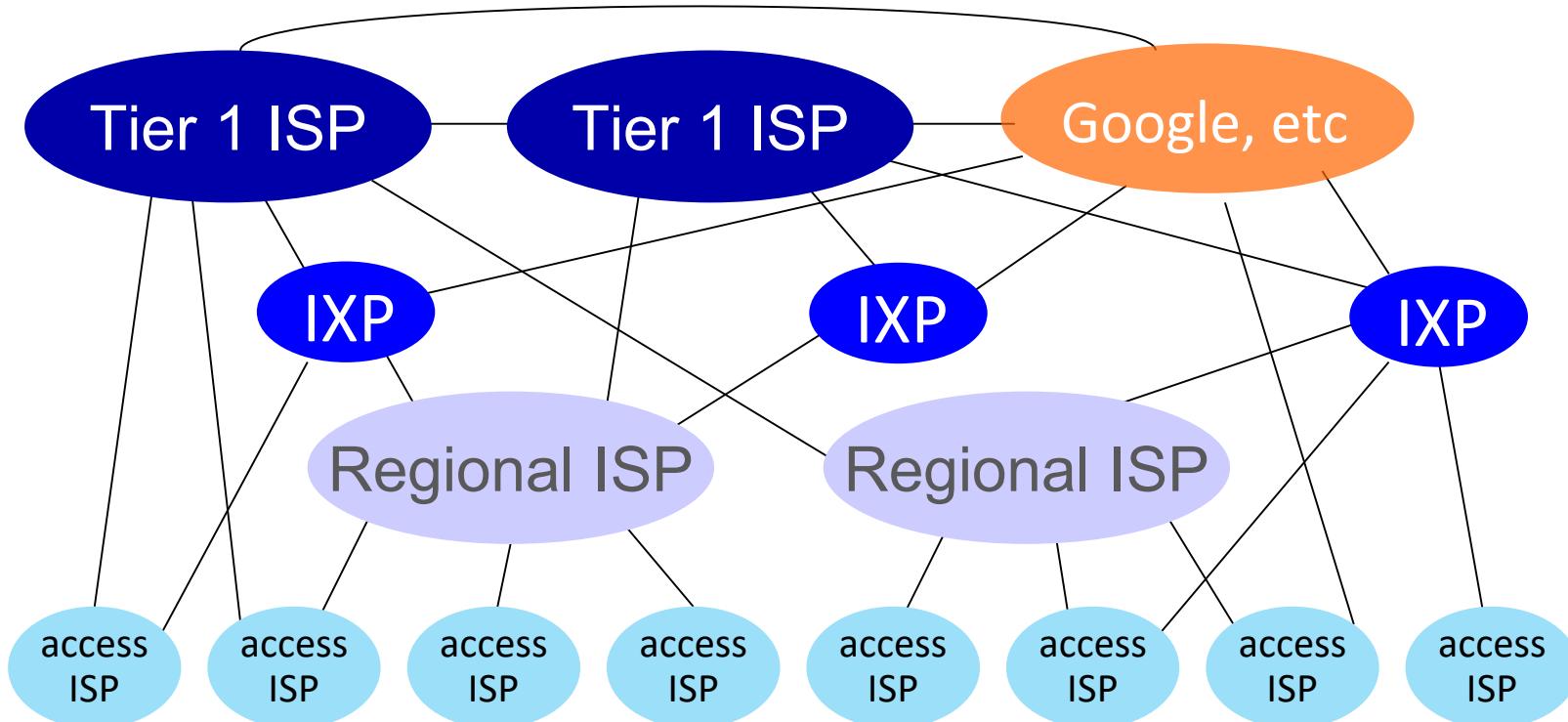


# Internet structure: a “network of networks”

... 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., AT&T, PCCW, NTT, Tata), 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

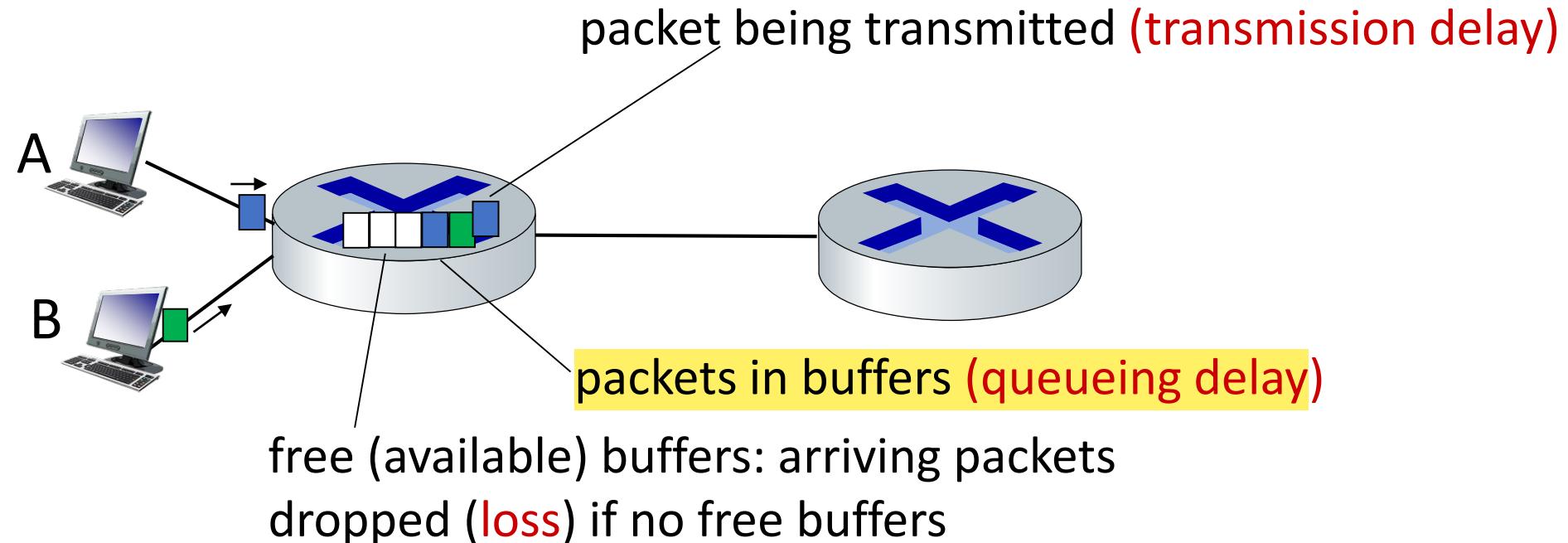
# Roadmap

- 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

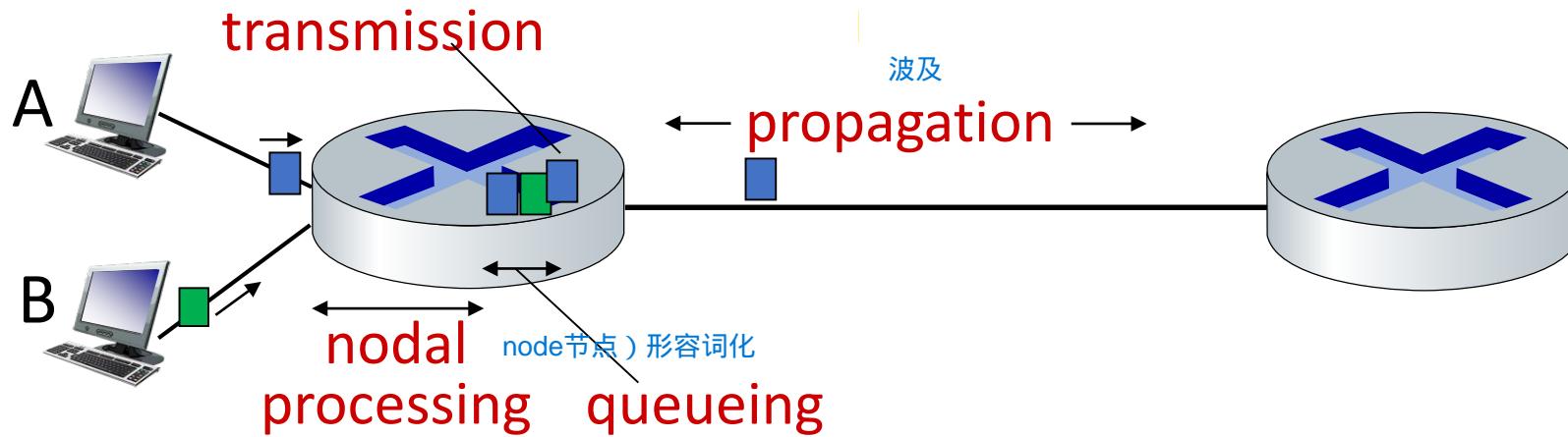
# How do packet loss and delay occur?

packets *queue* in router buffers

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



# Packet delay: four sources



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

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

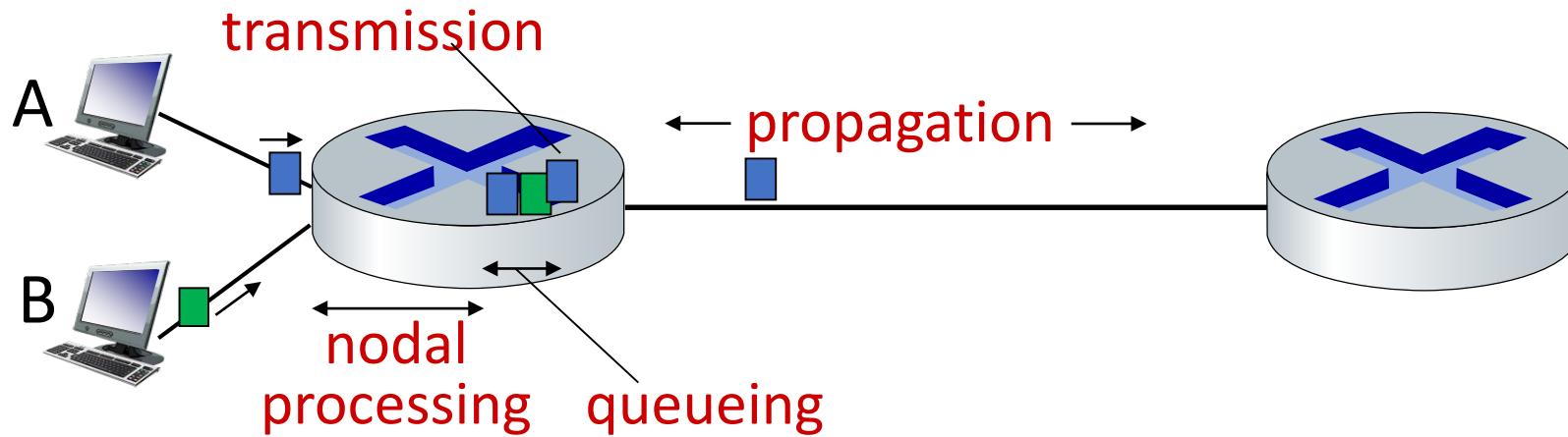
- check bit errors
- determine output link
- typically < msec

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

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

拥堵

# Packet delay: four sources



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

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

- $L$ : packet length (bits)
- $R$ : link *transmission rate (bps)*

$$\boxed{\mathbf{d}_{\text{trans}} = L/R}$$

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

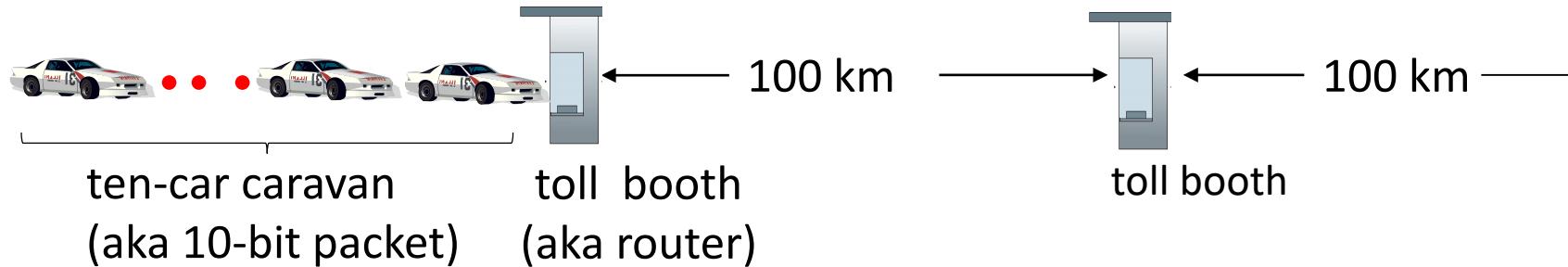
- $d$ : length of physical link
- $s$ : propagation speed ( $\sim 2 \times 10^8$  m/sec)

$$\boxed{\mathbf{d}_{\text{prop}} = d/s}$$

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

# Caravan analogy

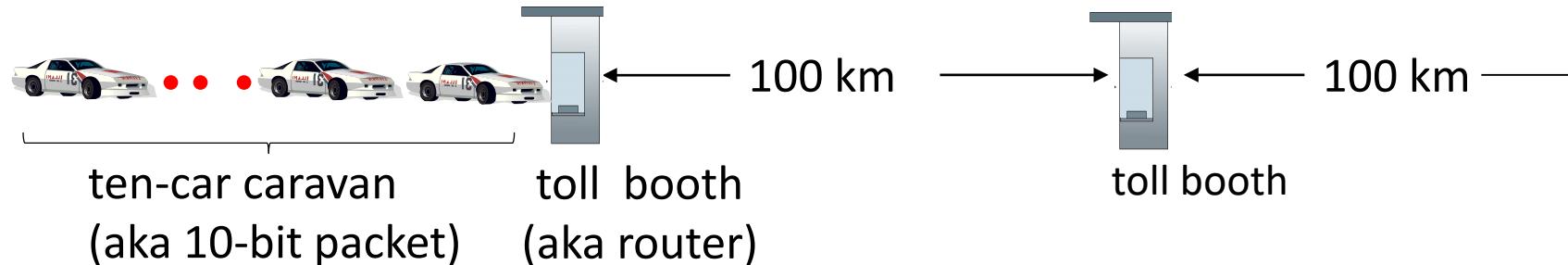
用宿营车队来模拟一下



- car = bit; caravan = packet
- cars “propagate” at 100 km/hr
- toll booth takes 12 sec to service car (bit transmission time)
- **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 some car to propagate from exit of 1st to 2nd toll both:  
 $100\text{km}/(100\text{km/hr}) = 1 \text{ hr}$
- **A: 62 minutes**

# Caravan analogy

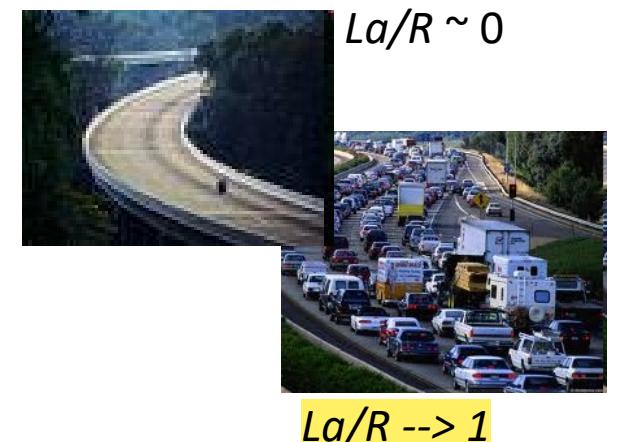
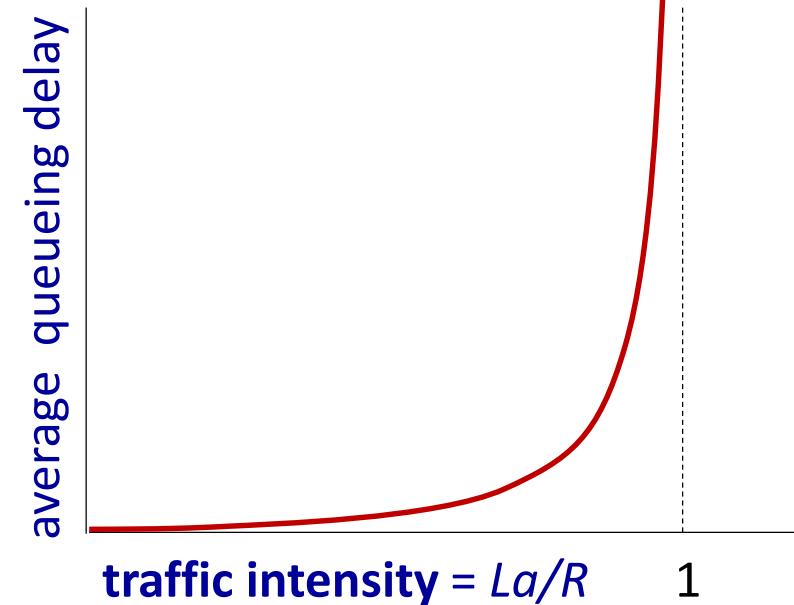


- suppose cars now “propagate” at 1000 km/hr
- and suppose toll booth now takes one min to service a car
- ***Q: Will cars arrive to 2nd booth before all cars serviced at first booth?***

**A: Yes!** after 7 min, first car arrives at second booth; three cars still at first booth

# Packet queueing delay (revisited)

- $R$ : link bandwidth (bps)
- $L$ : packet length (bits)
- $a$ : average packet arrival rate
- $La/R \sim 0$ : avg. queueing delay small
- $La/R \rightarrow 1$ : avg. queueing delay large
- $La/R > 1$ : more “work” arriving is more than can be serviced  
=> average delay infinite (in theory)!



# Packet queueing delay (additional details)

More realistic: Random arrival intervals.

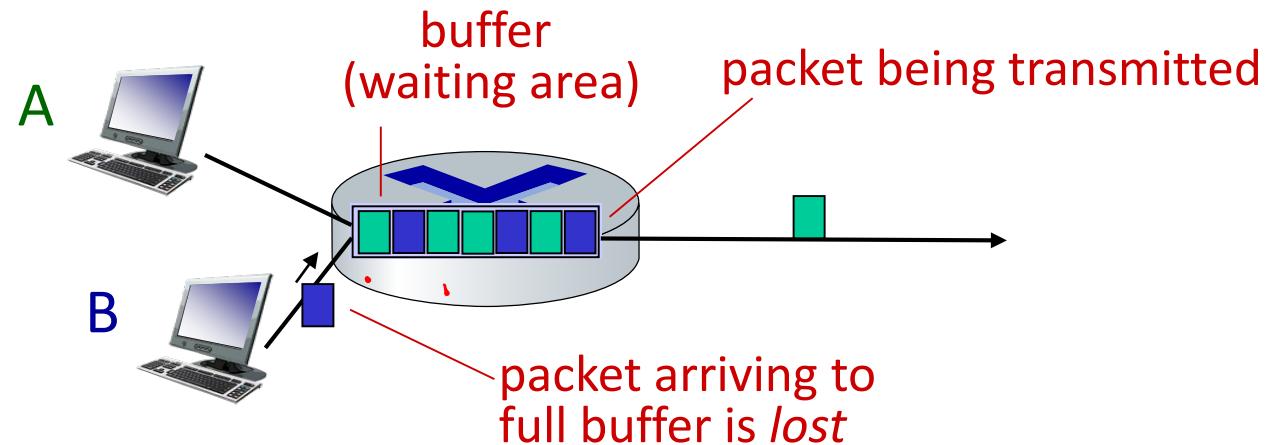
A closer look at case  $L_a/R \rightarrow 1$ . Two extreme scenarios:

1. What happens if **1** packet arrives every **( $L/R$ )** sec?  
→ No queueing delay. Why?

2. What happens if **N** packets arrive every **( $N L / R$ )** sec?  
→ Queueing delay grows with **N**. Why?

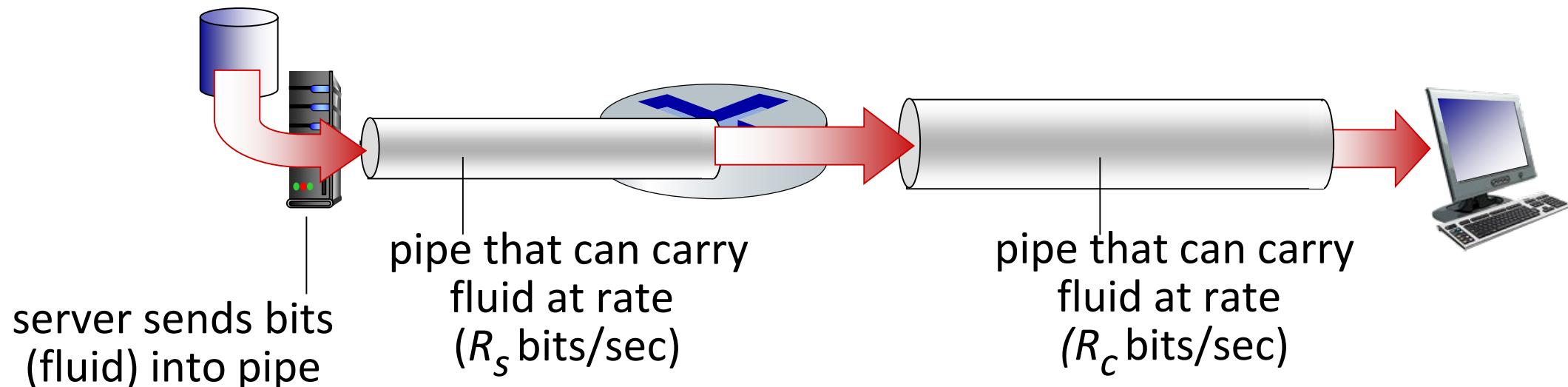
# Packet loss

- 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



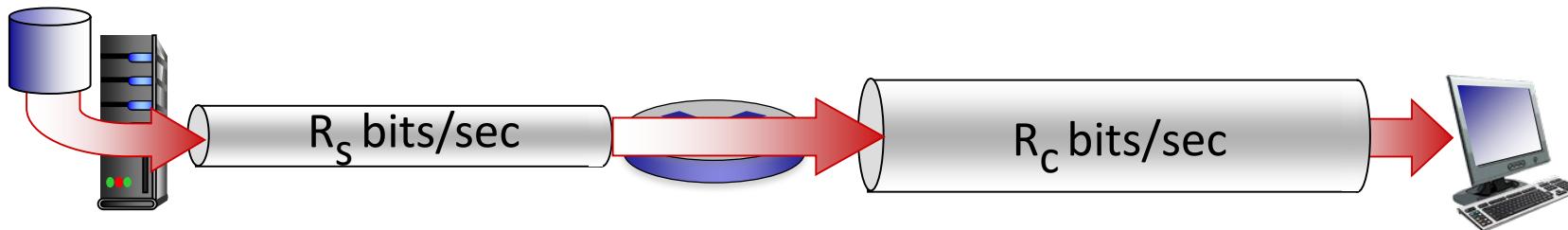
# Throughput

- **throughput:** rate (bits/time unit) at which bits are being sent from sender to receiver
  - *instantaneous:* <sup>即刻的</sup> rate at given point in time
  - *average:* rate over longer period of time.  
If file of  $F$  bits takes  $T$  sec until received then  $F/T$  bits/sec

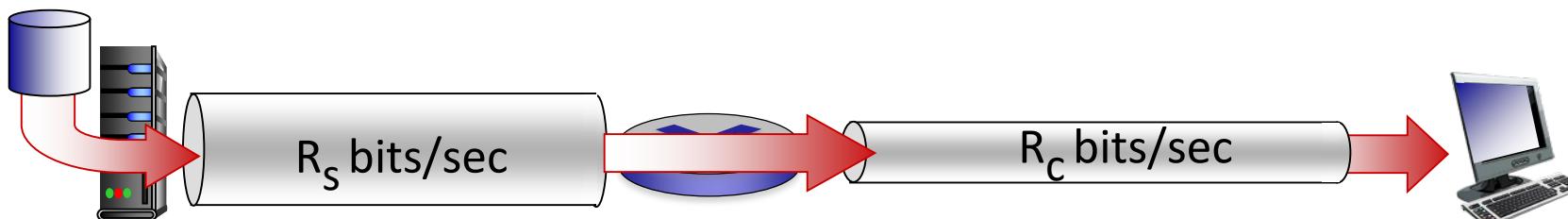


# Throughput

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



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

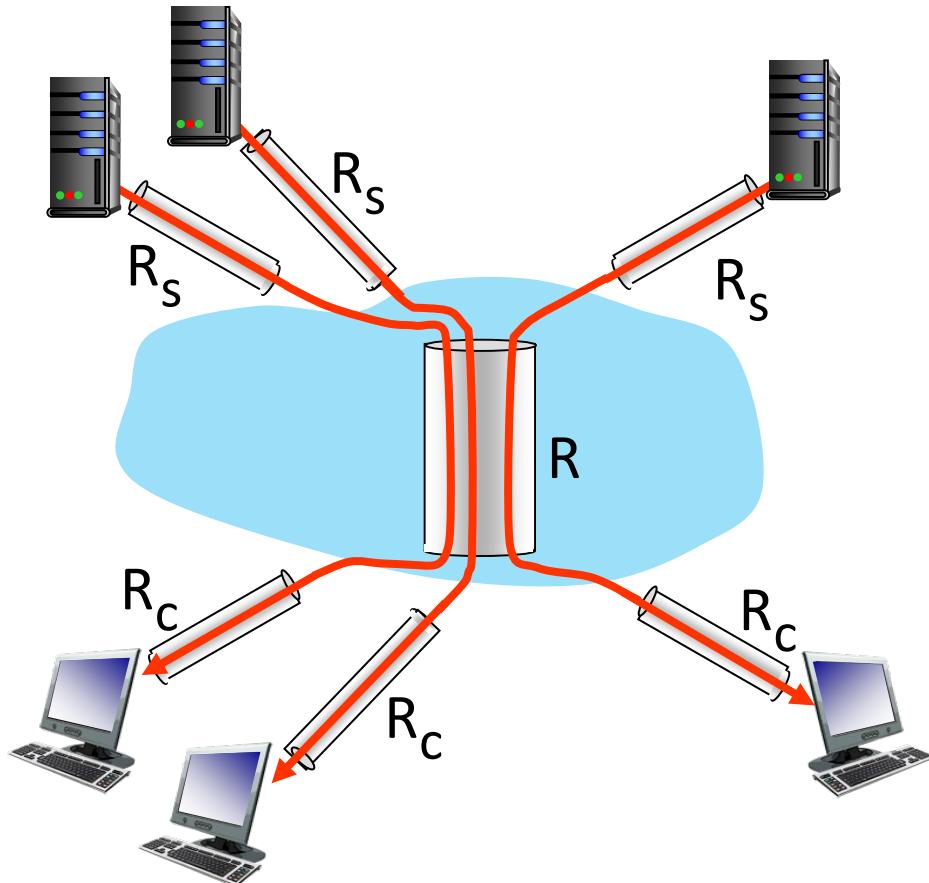


*bottleneck link*

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

# Throughput: network scenario

情节，方案



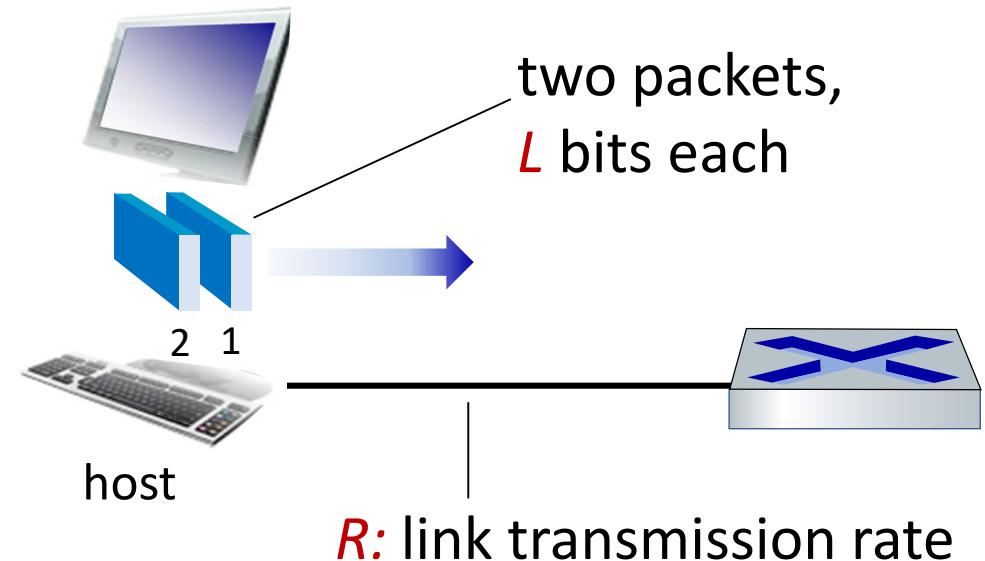
10 connections (fairly) share  
backbone bottleneck link  $R$  bits/sec

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

# 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*



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

# 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**

# Protocol “layers” and reference models

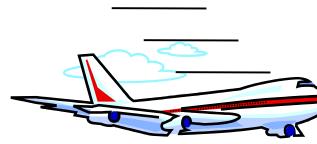
*Networks are complex,  
with many “pieces”:*

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

*Question:*

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

# Example: organization of air travel



ticket (purchase)  
baggage (check)  
gates (load)  
runway takeoff  
airplane routing

ticket (complain?)  
baggage (claim)  
gates (unload)  
runway landing  
airplane routing

airplane routing

airline travel: a series of steps, involving many services

# Example: organization of air travel



*layers*: each layer implements a service

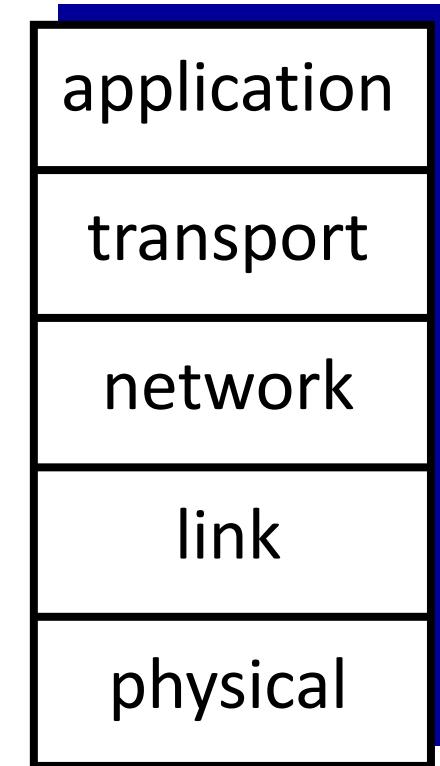
- by performing internal-layer actions, and
- relying on services provided by layer below

# Why layering?

- Allows us to discuss well-defined, specific part of complex system.  
模块化
- 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 airline system

# Internet protocol stack

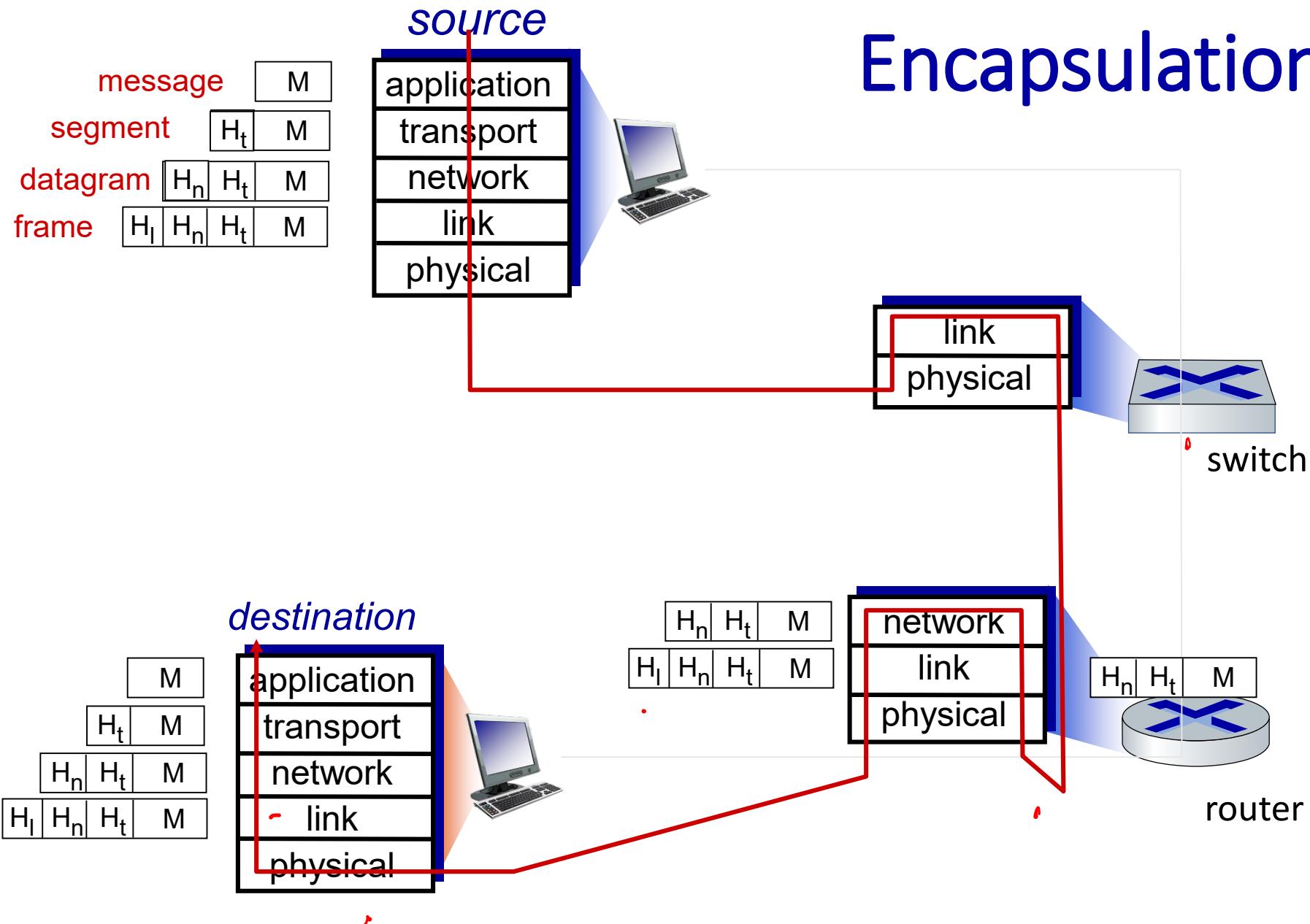
- *application*: supporting network applications
  - HTTP, IMAP, SMTP, ...
- *transport*: process-process data transfer
  - TCP, UDP
- *network*: routing of datagrams from source to destination
  - IP, routing protocols
- *link*: data transfer between neighboring network elements
  - Ethernet, 802.11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



数据报文

# Encapsulation

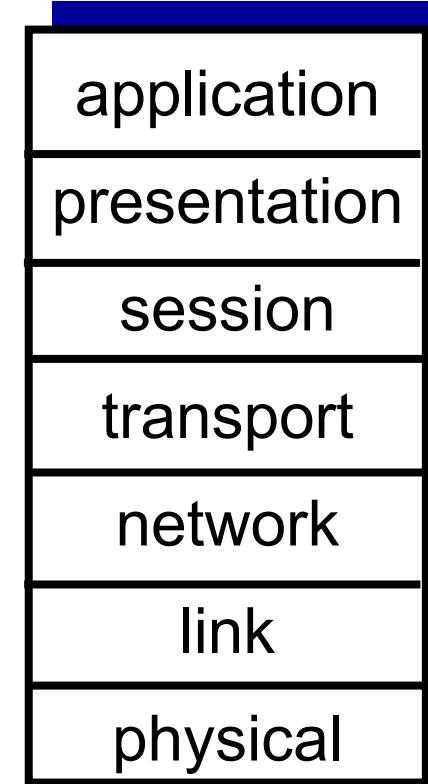
封装（这里是简而言之的意思？？）



# 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  
编码加密
- *session*: synchronization, checkpointing, recovery of data exchange
- Internet stack “missing” these layers!
  - these services, *if needed*, must be implemented in application



The seven layer OSI/ISO reference model