

CPSC 441

Computer Networks

Majid Ghaderi
Department of Computer Science
University of Calgary

Chapter 1: introduction

our goal:

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

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

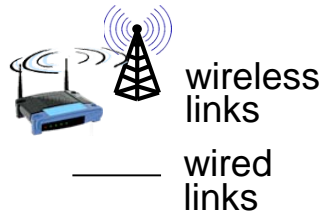
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

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



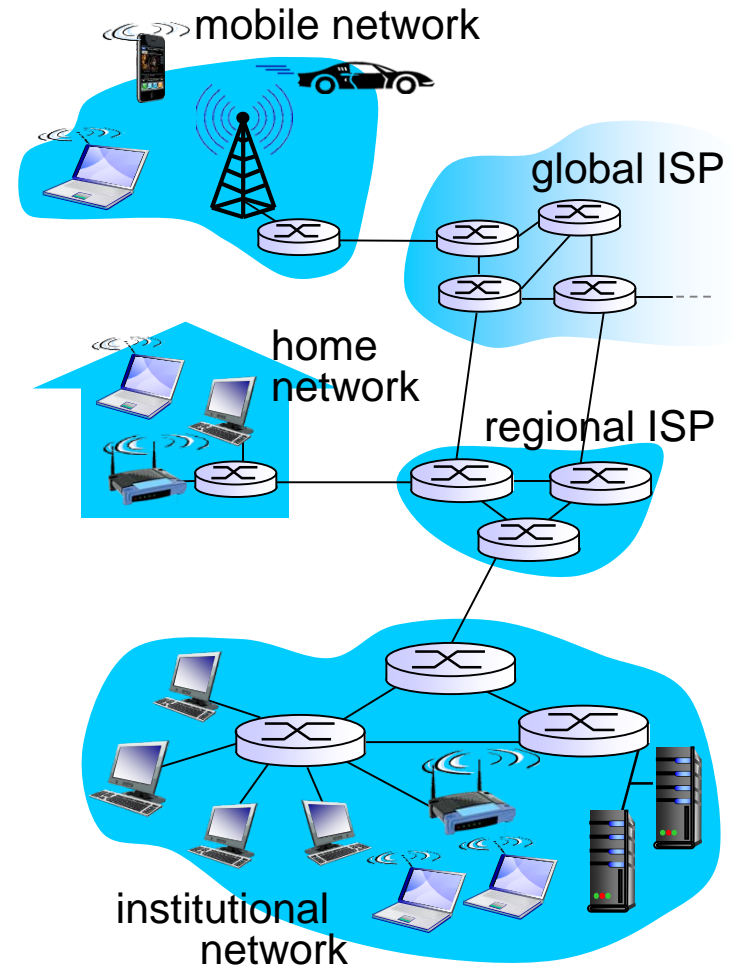
- ❖ millions of connected computing devices:
 - *hosts* = *end systems*
 - running *network apps*



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

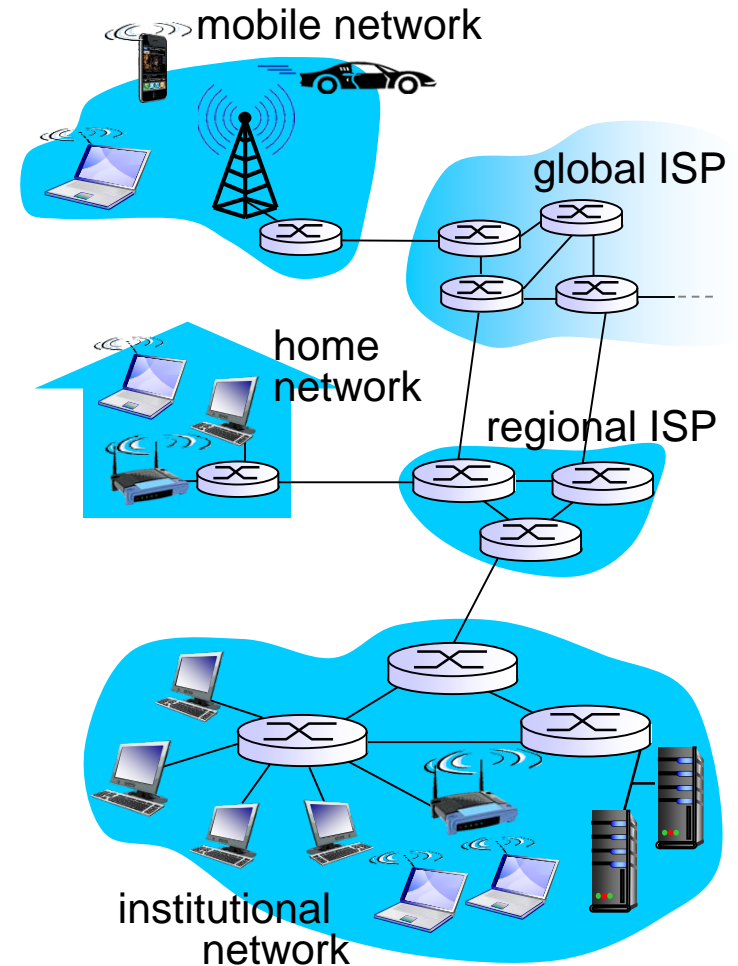


- ❖ *Packet switches*: forward packets (chunks of data)
 - *routers* and *switches*



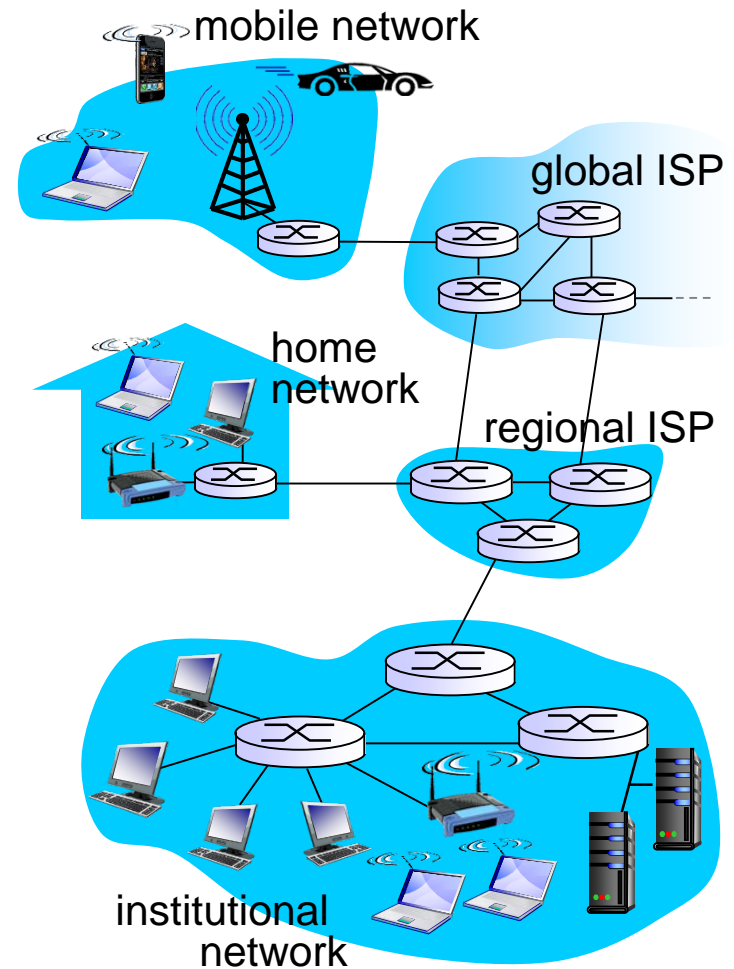
What's the Internet: “nuts and bolts” view

- ❖ *Internet: “network of networks”*
 - Interconnected ISPs
- ❖ *protocols* control sending, receiving of msgs
 - e.g., TCP, IP, HTTP, 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

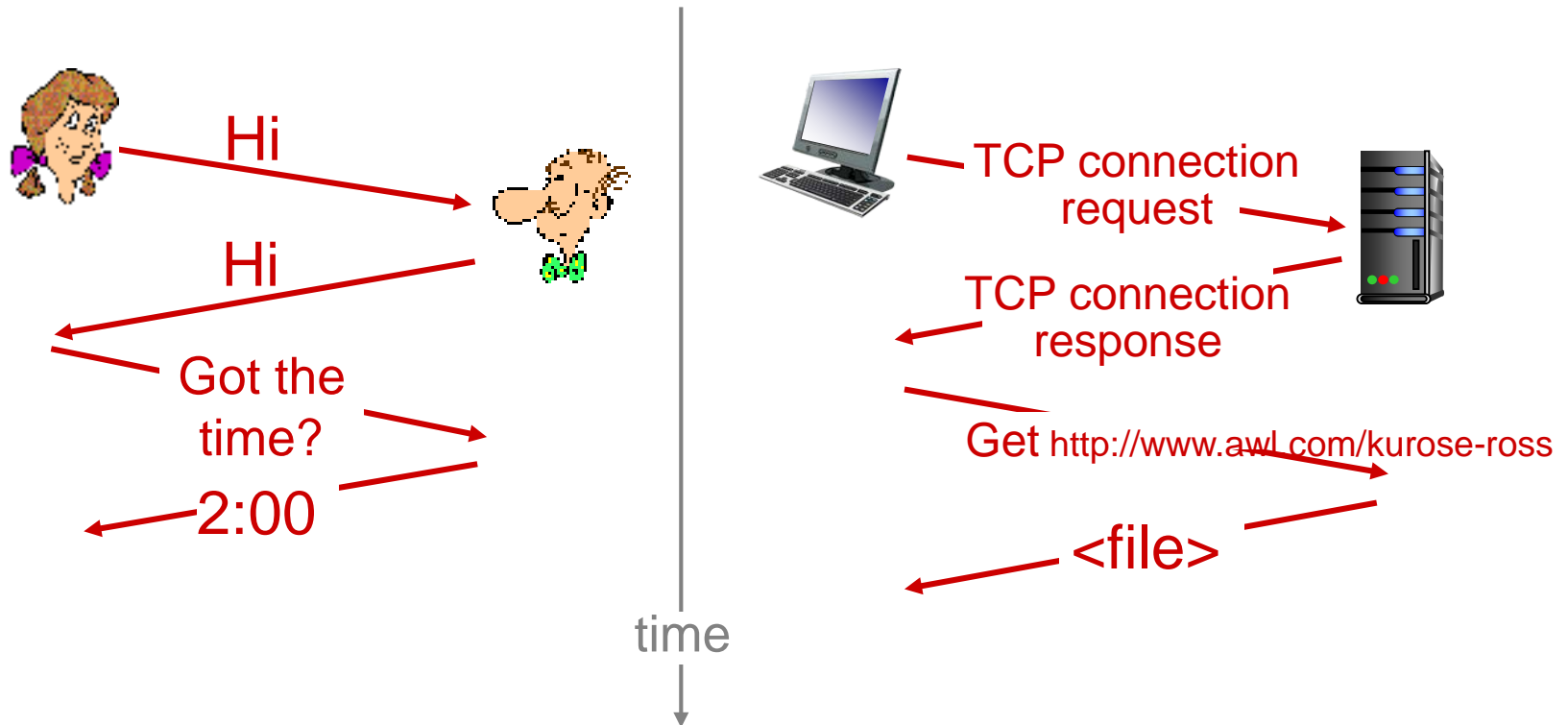
network protocols:

- ❖ 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:



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

A closer look at network structure:

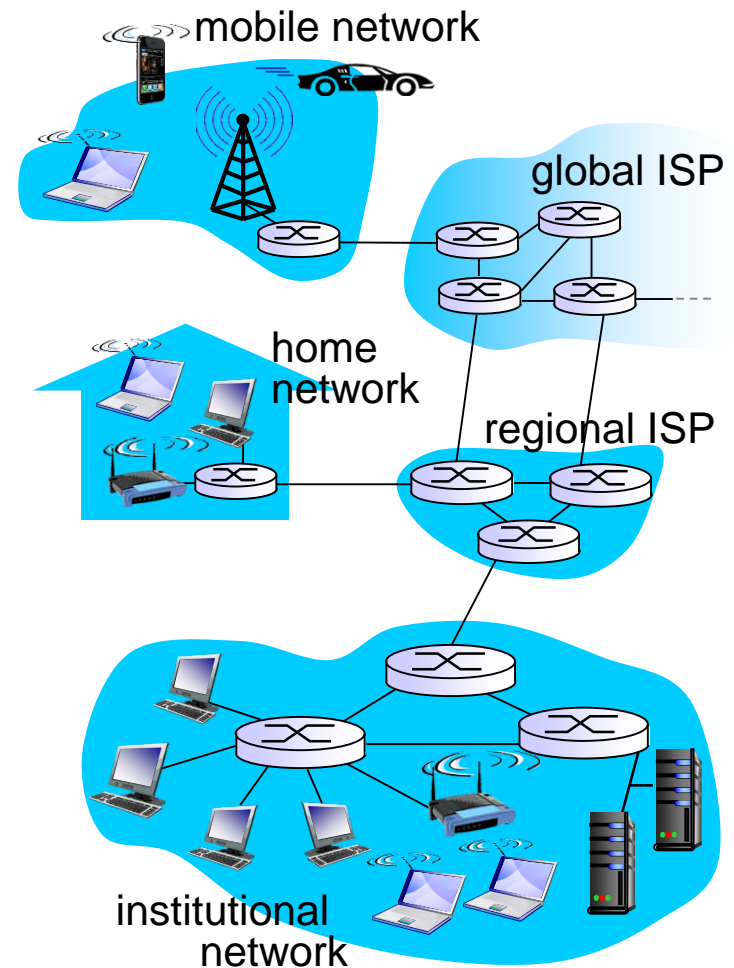
❖ *network edge:*

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

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

❖ *network core:*

- interconnected routers
- network of networks



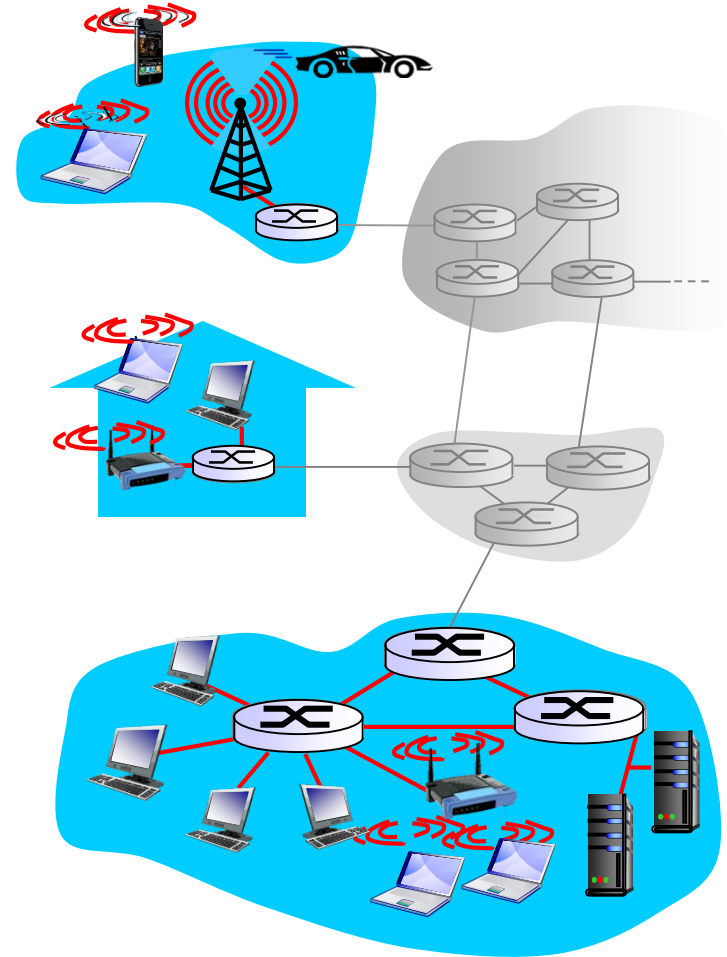
Access networks and physical media

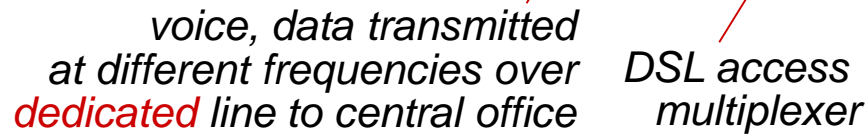
Q: How to connect end systems to edge router?

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

keep in mind:

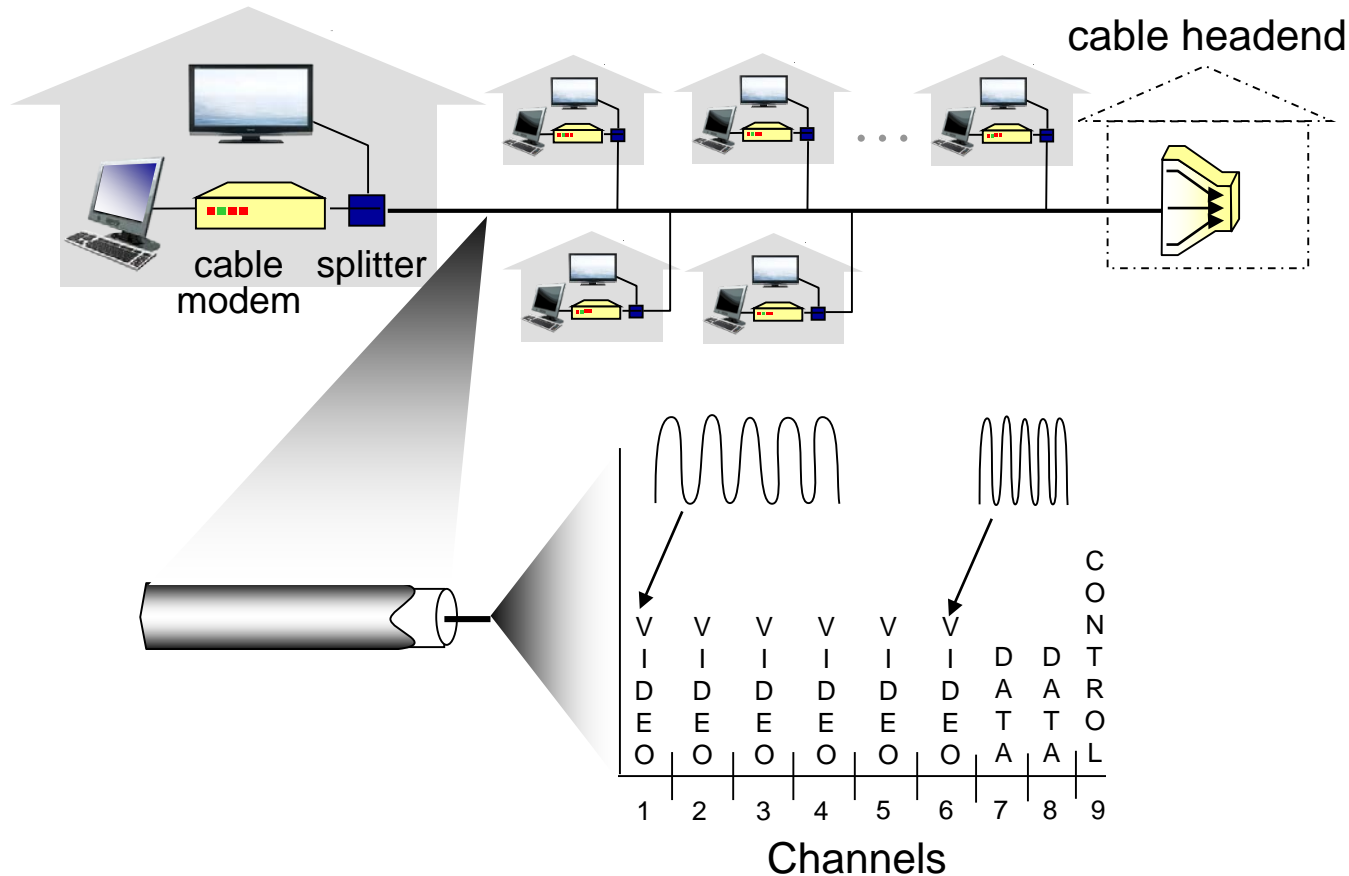
- ❖ bandwidth (bits per second) of access network?
- ❖ shared or dedicated?





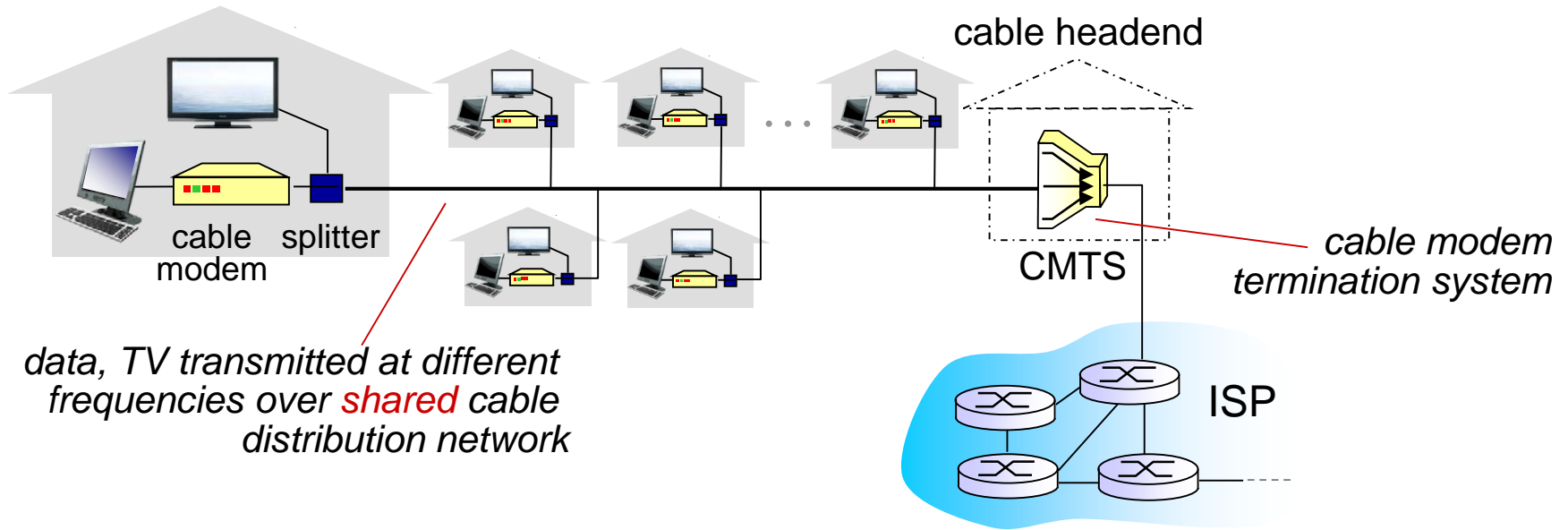
- ❖ use *existing* telephone line to central office DSLAM
 - data over DSL phone line goes to Internet
 - voice over DSL phone line goes to telephone net

Access net: cable network



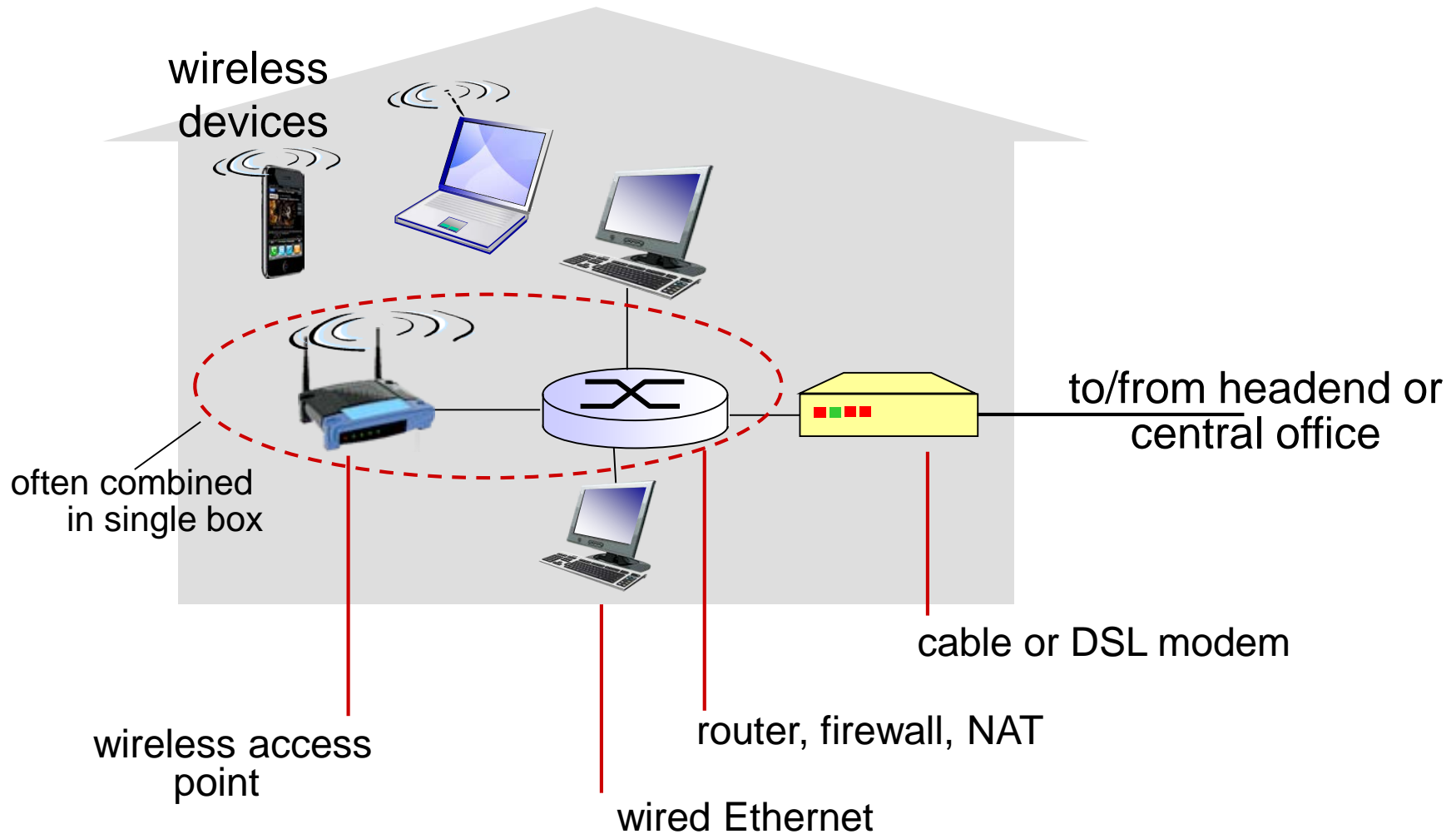
frequency division multiplexing: different channels transmitted in different frequency bands

Access net: cable network

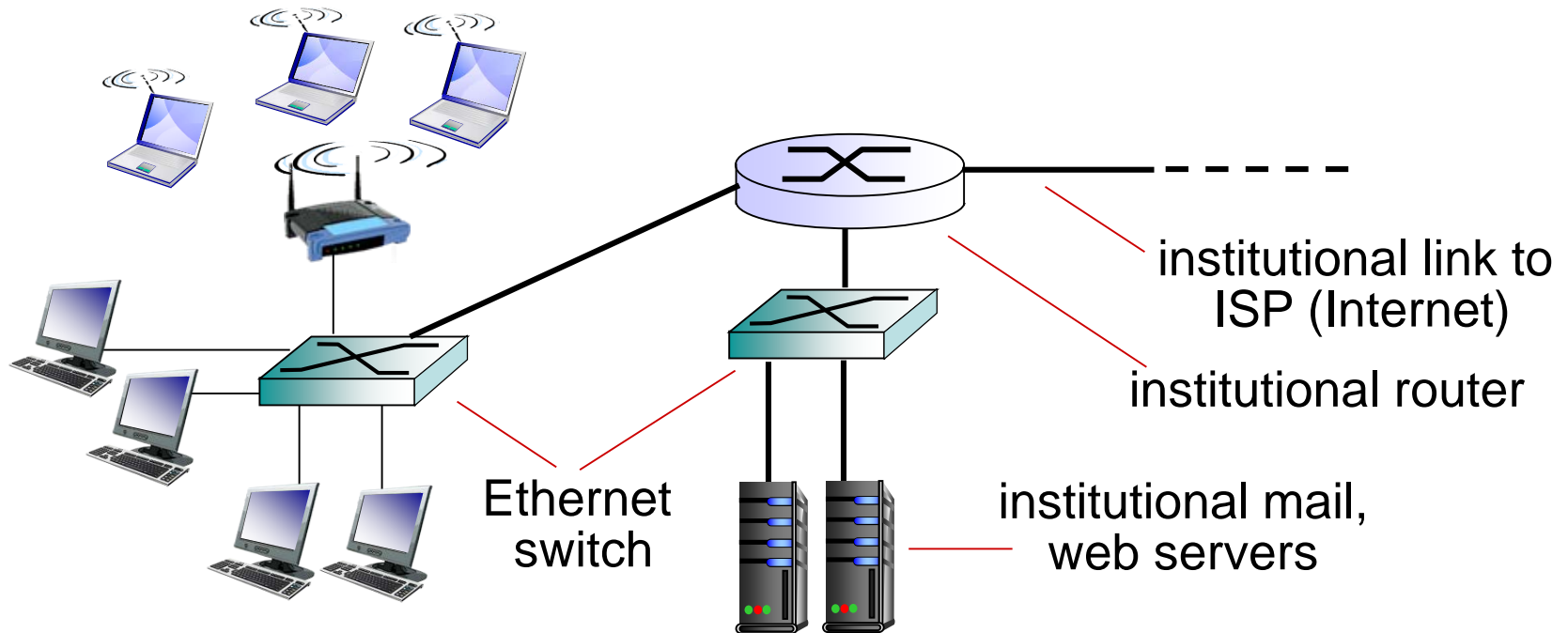


- ❖ **network** of cable, fiber attaches homes to ISP router
 - homes *share access network* to cable headend
 - unlike DSL, which has dedicated access to central office

Access net: home network



Enterprise access networks



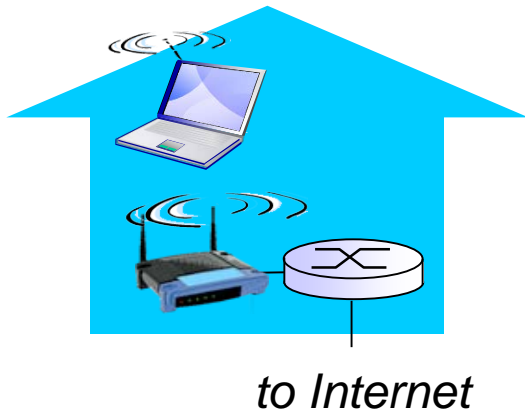
- ❖ typically used in companies, universities, etc
- ❖ 100Mbps, 1Gbps, 10Gbps transmission rates

Wireless access networks

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

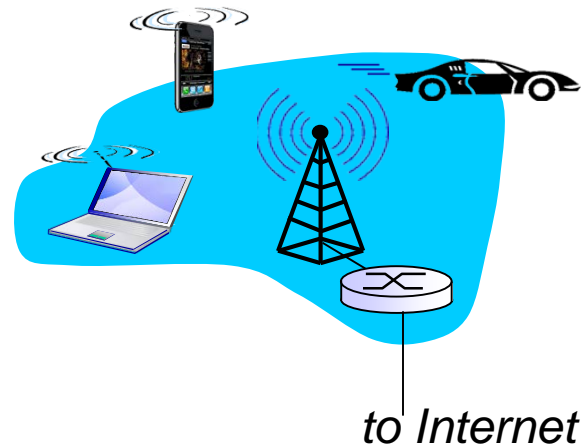
wireless LANs:

- within building (100 ft)
- 802.11 (WiFi): 100's Mbps transmission rate



wide-area wireless access

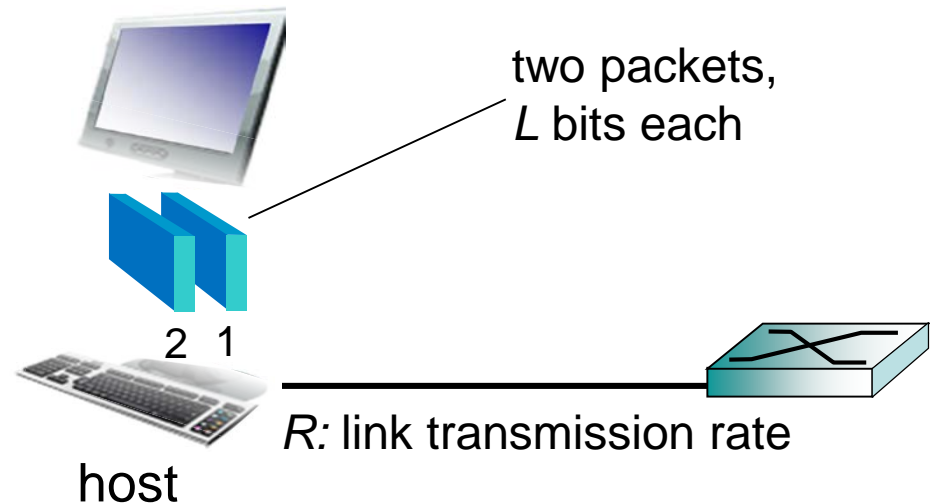
- provided by telco (cellular) operator, 10's km
- 10's Mbps
- 3G, 4G: LTE



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*



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

Chapter 1: roadmap

1.1 what is the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

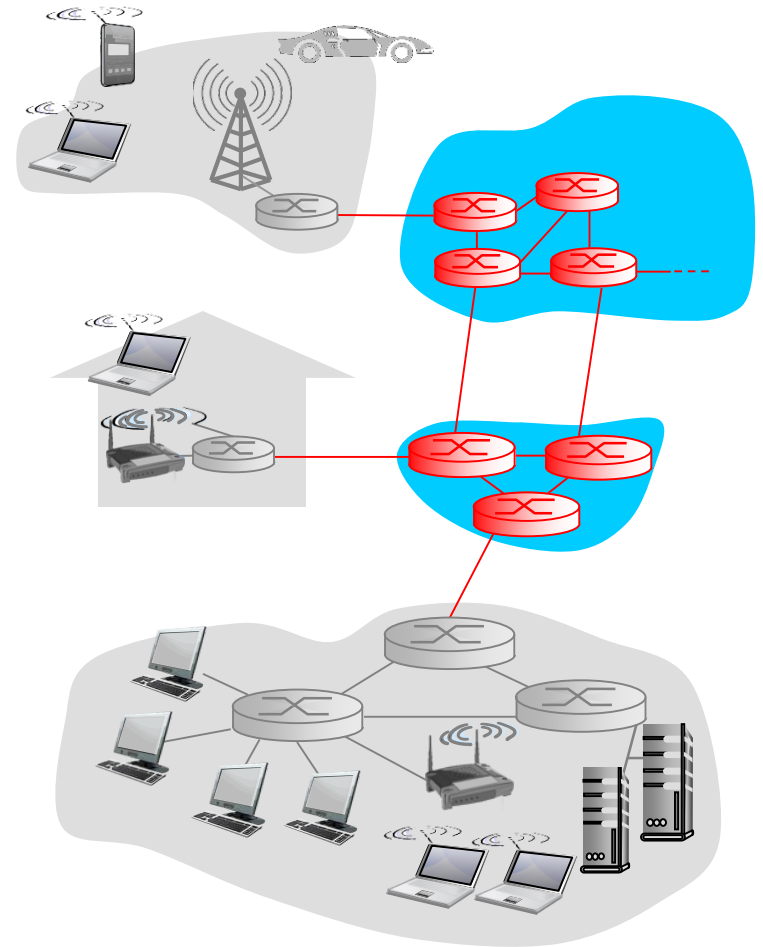
- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

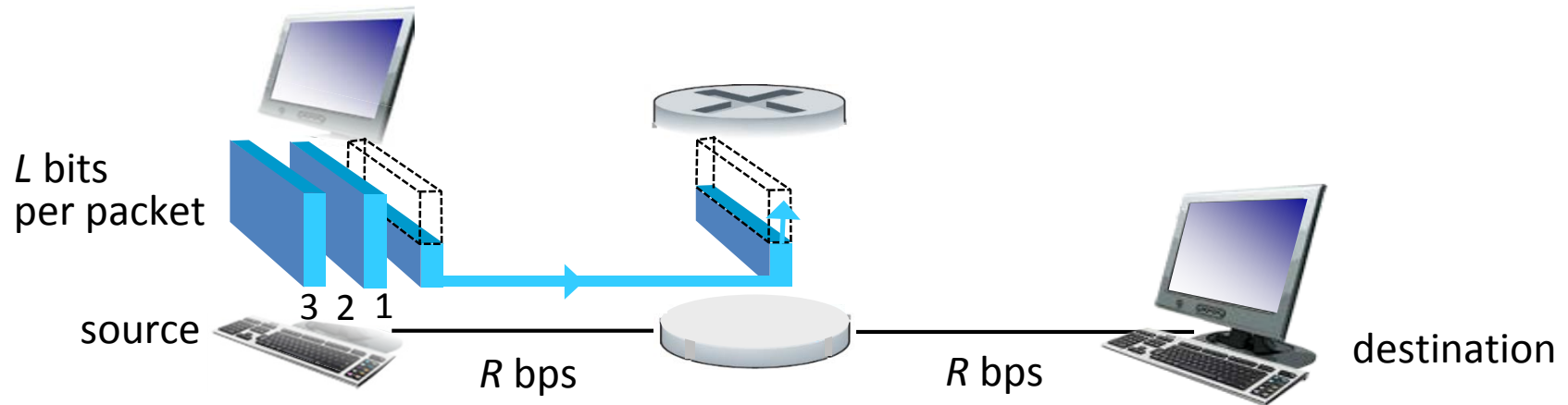
1.5 protocol layers, service models

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



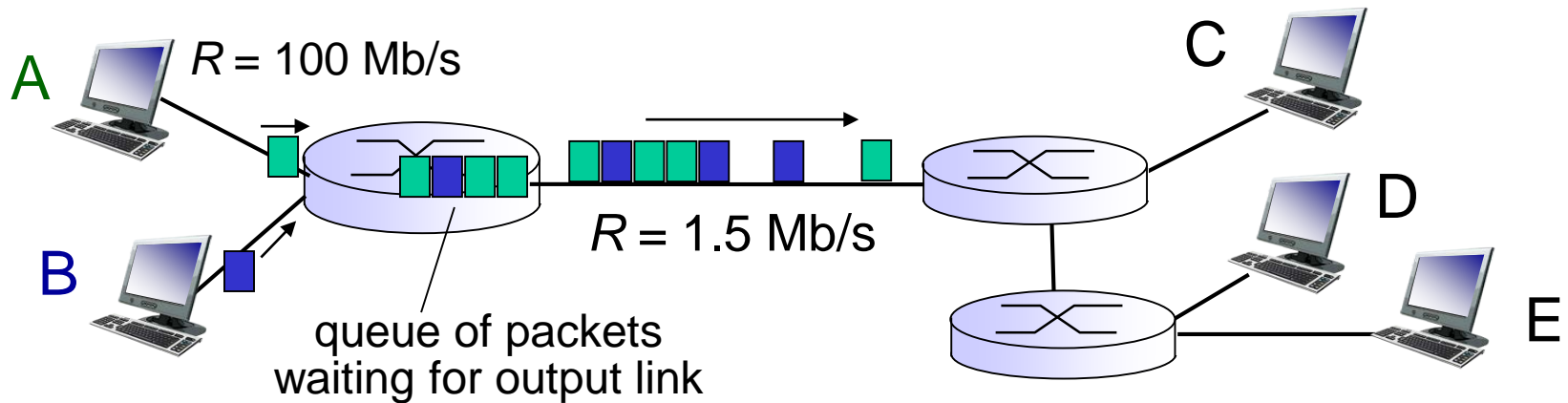
- ❖ takes L/R seconds to transmit (push out) L -bit packet into link at R bps
- ❖ *store and forward*: entire packet must arrive at router before it can be transmitted on next link
- ❖ end-end delay = $2L/R$ (assuming zero propagation delay)

one-hop numerical example:

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

} more on delay shortly ...

Packet Switching: queueing delay, loss



queuing and loss:

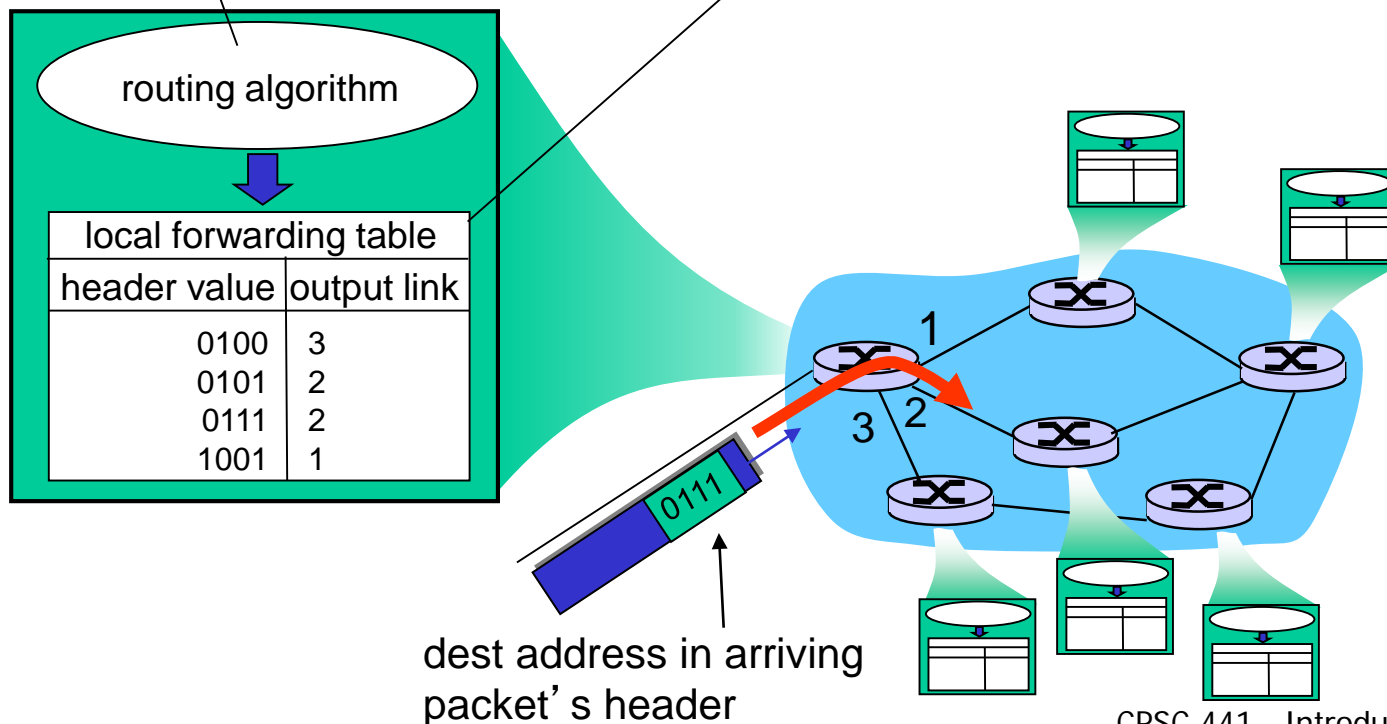
- ❖ If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
 - packets will queue, wait to be transmitted on link
 - packets can be dropped (lost) if memory (buffer) fills up

Two key network-core functions

routing: determines source-destination route taken by packets

- *routing algorithms*

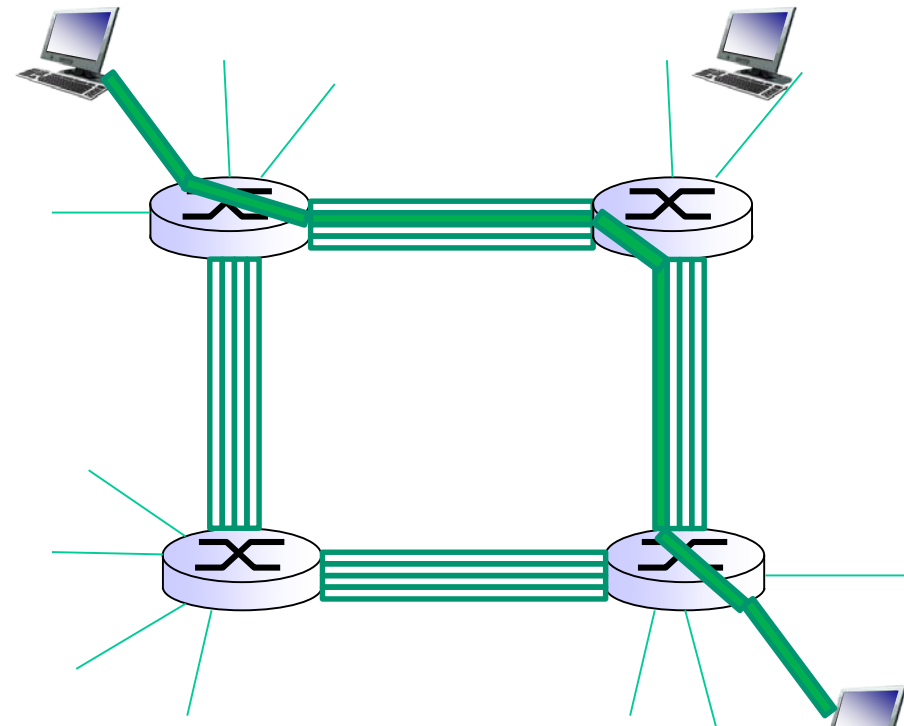
forwarding: move packets from router's input to appropriate router output



Alternative core: circuit switching

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

- ❖ In diagram, each link has four circuits.
 - call gets 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

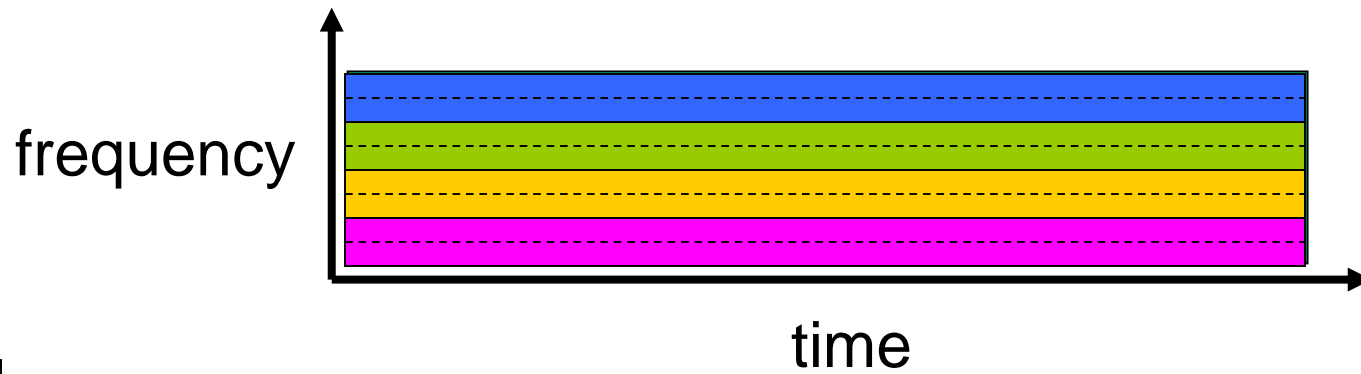


Circuit switching: FDM versus TDM

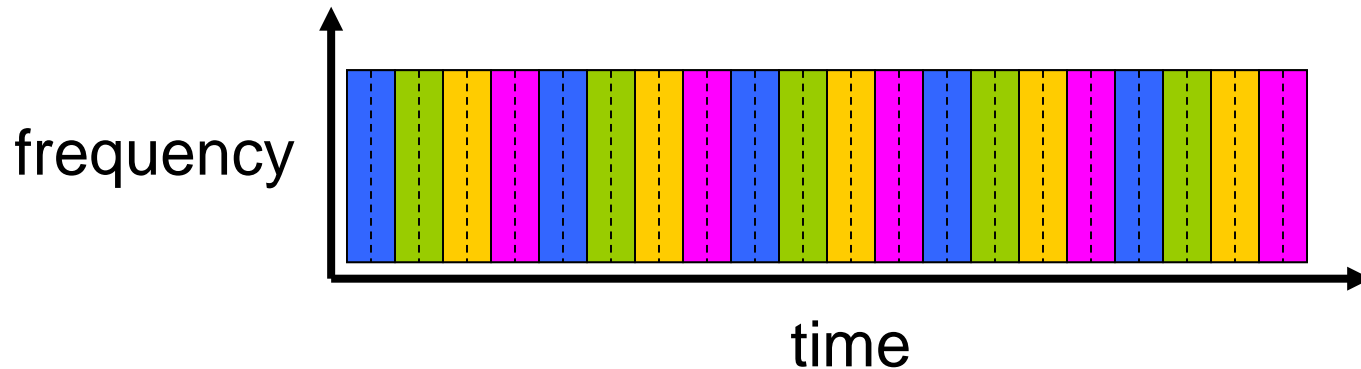
FDM

Example:

4 users



TDM

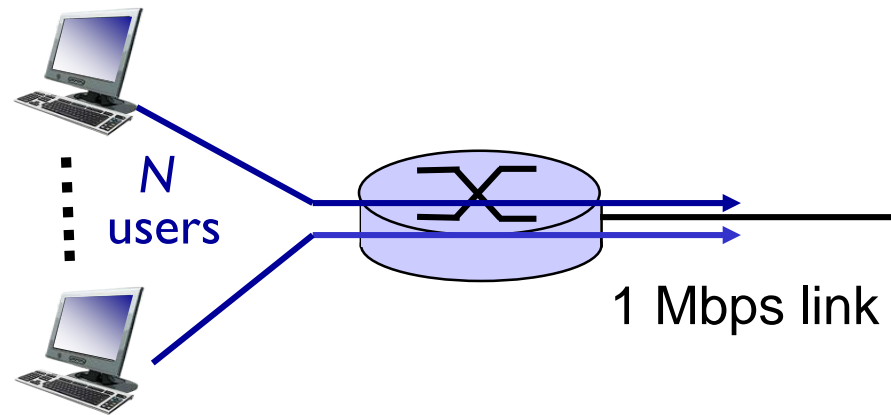


Packet switching versus circuit switching

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 0.0004

Packet switching versus circuit switching

is packet switching a “slam dunk winner?”

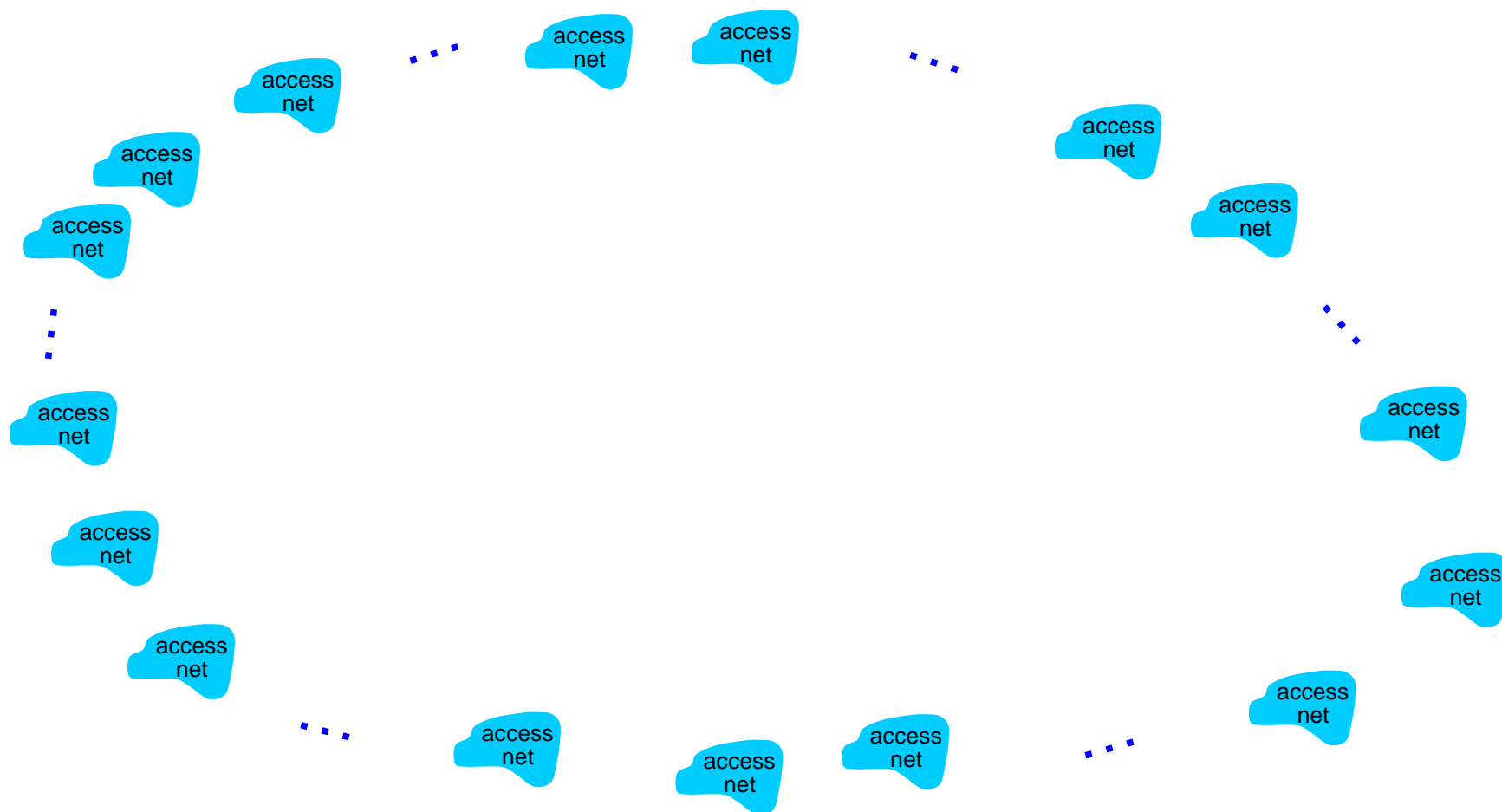
- ❖ great for bursty data
 - resource sharing
 - simpler, no call setup
- ❖ **excessive congestion possible:** packet delay and loss
 - protocols needed for reliable data transfer, congestion control
- ❖ **Q: How to provide circuit-like behavior?**
 - bandwidth guarantees needed for audio/video apps

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**

Internet structure: network of networks

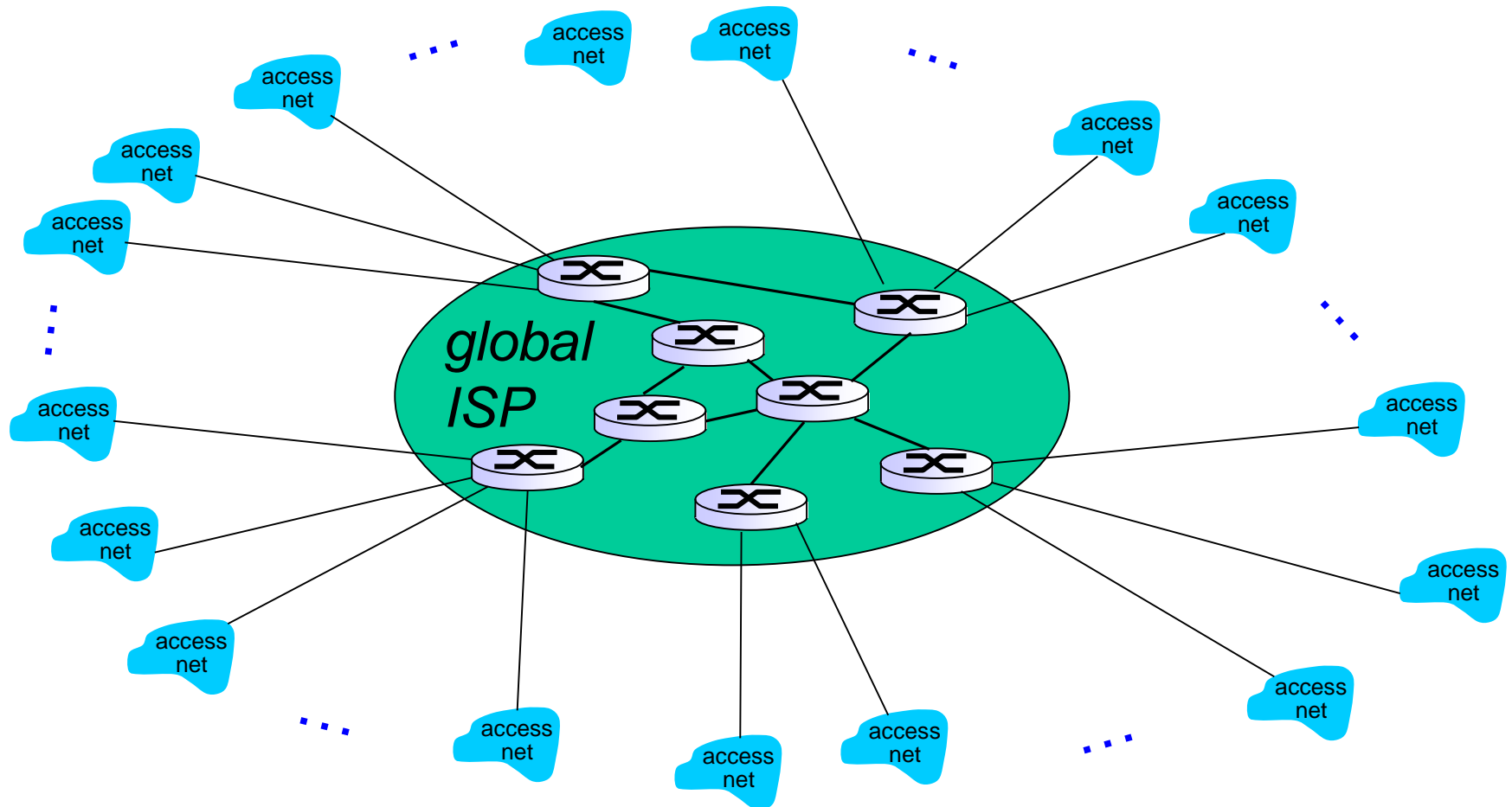
Question: given *millions* of access ISPs, how to connect them together?



Internet structure: network of networks

Option: connect each access ISP to a global ISP.

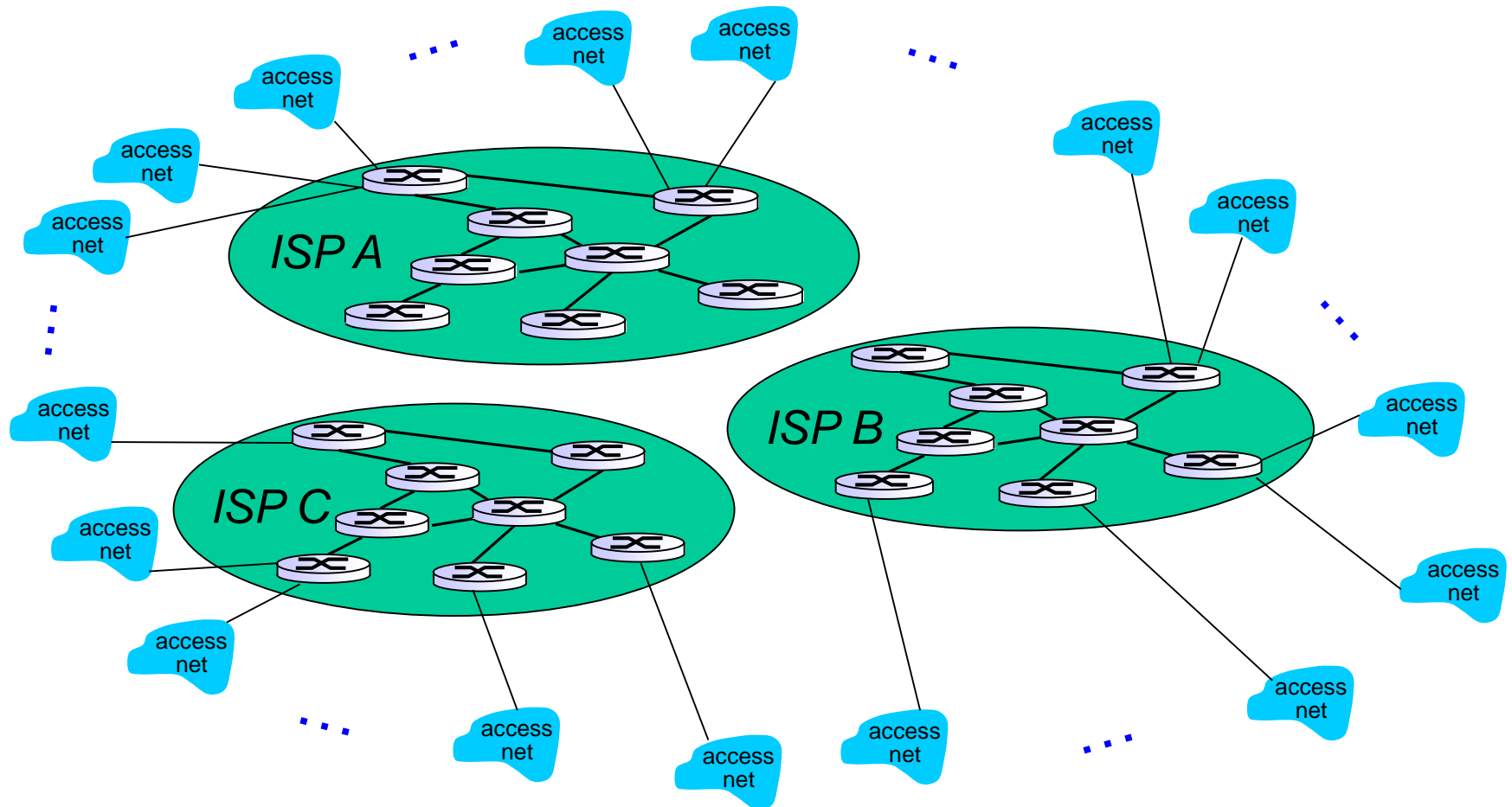
Customer and provider ISPs have economic agreement.



Internet structure: network of networks

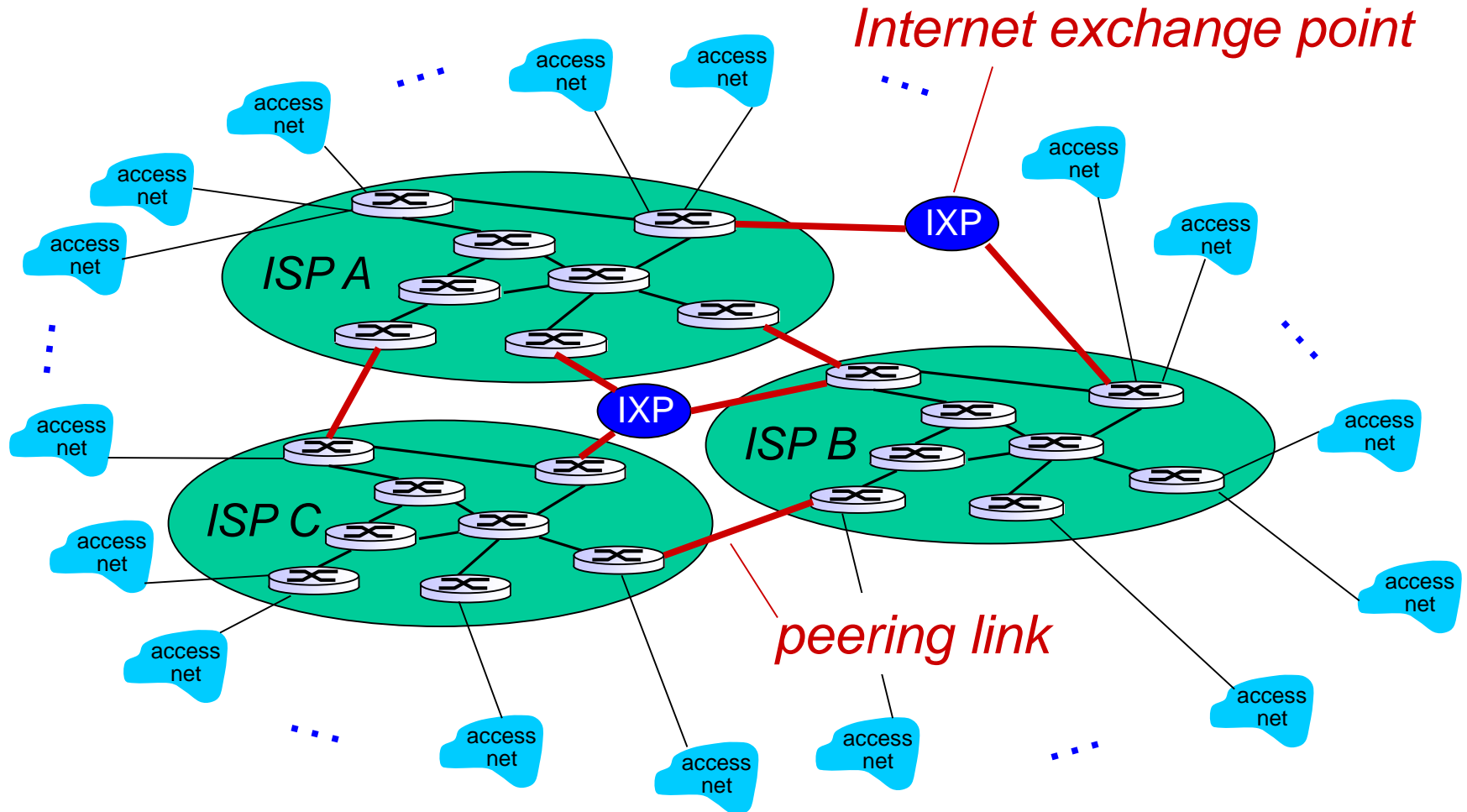
But if one global ISP is viable business, there will be competitors

....



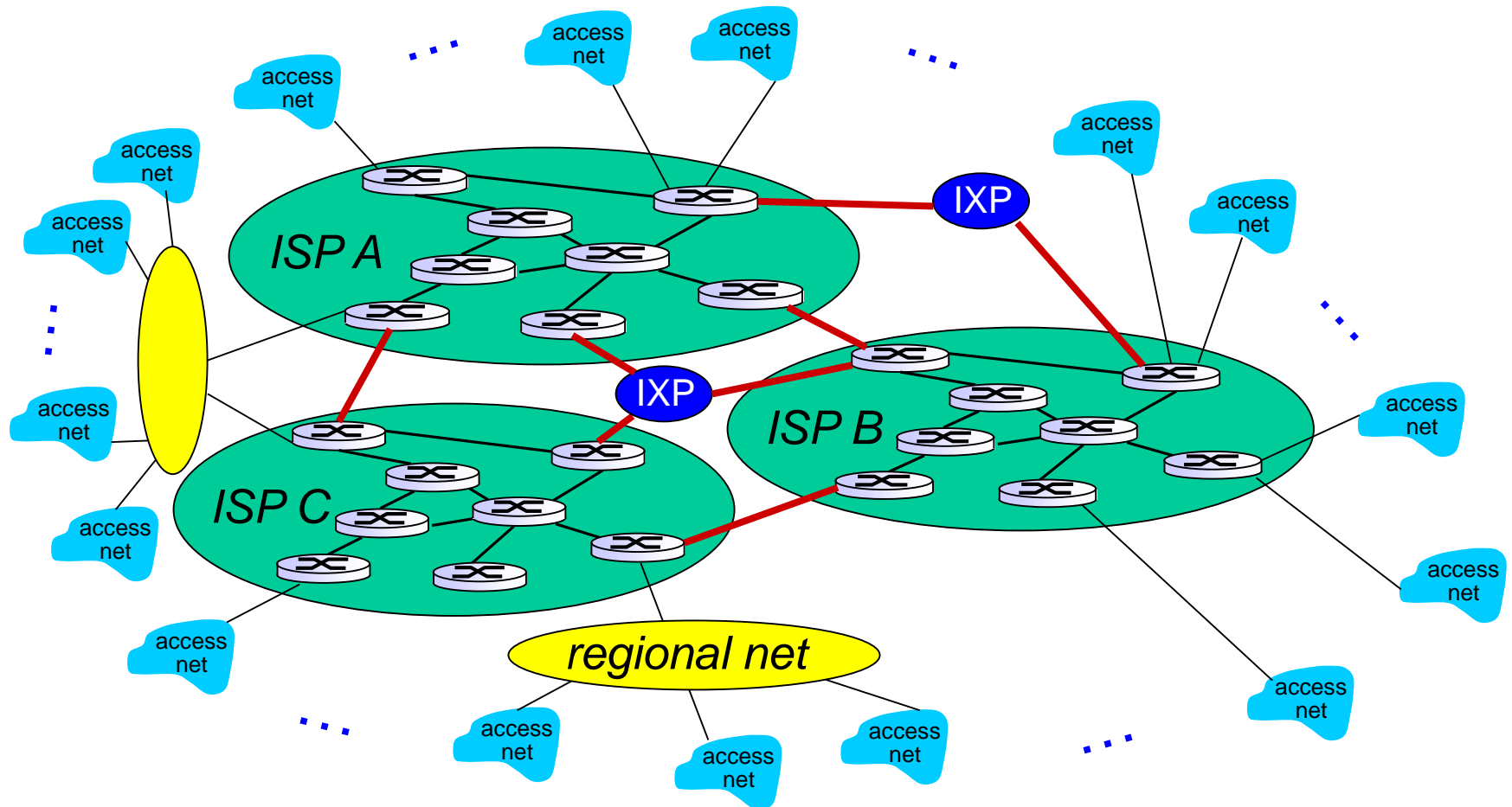
Internet structure: network of networks

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



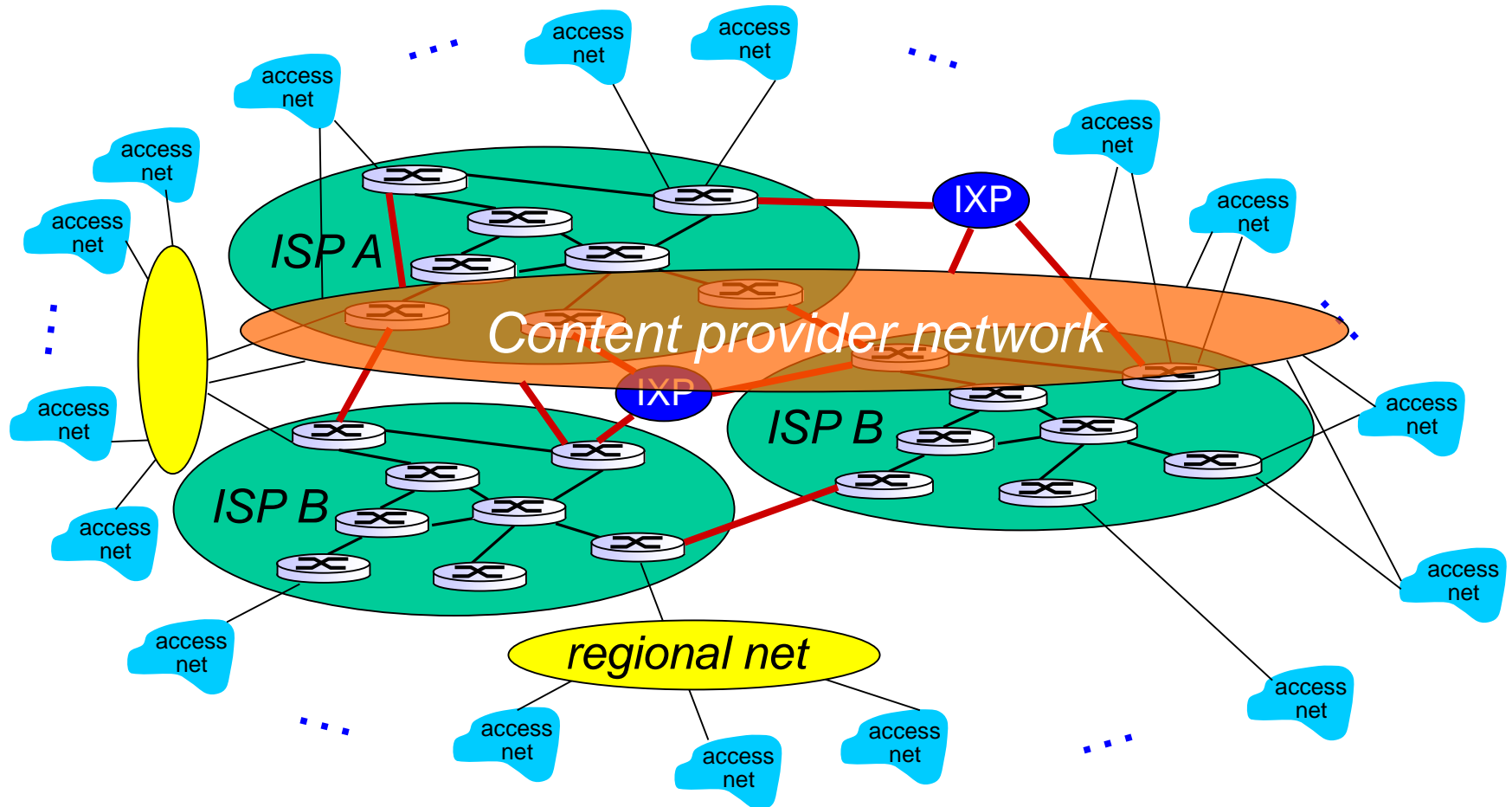
Internet structure: network of networks

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

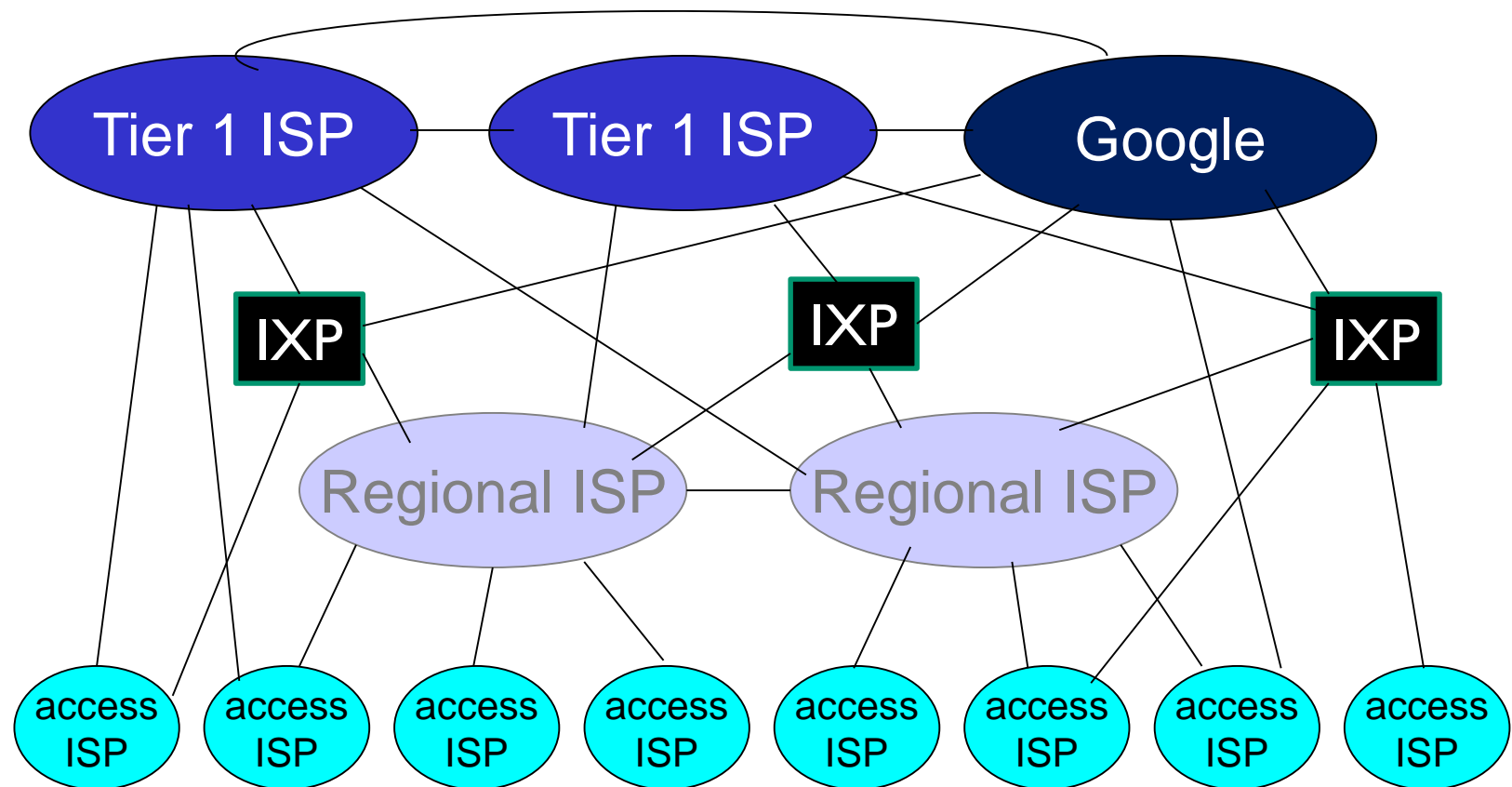


Internet structure: network of networks

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

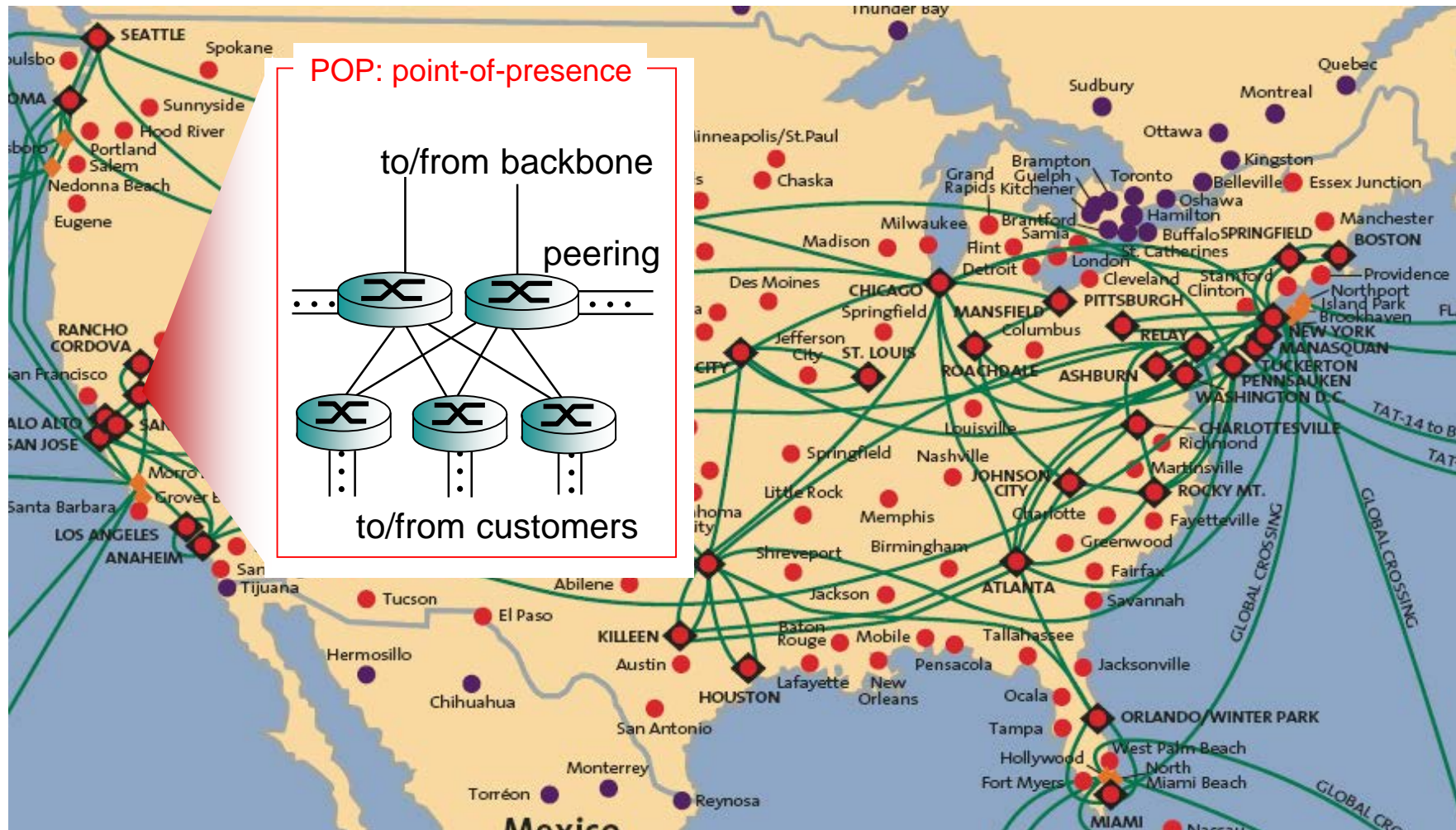


Internet structure: network of networks



- ❖ 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-1 ISP: e.g., Sprint



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

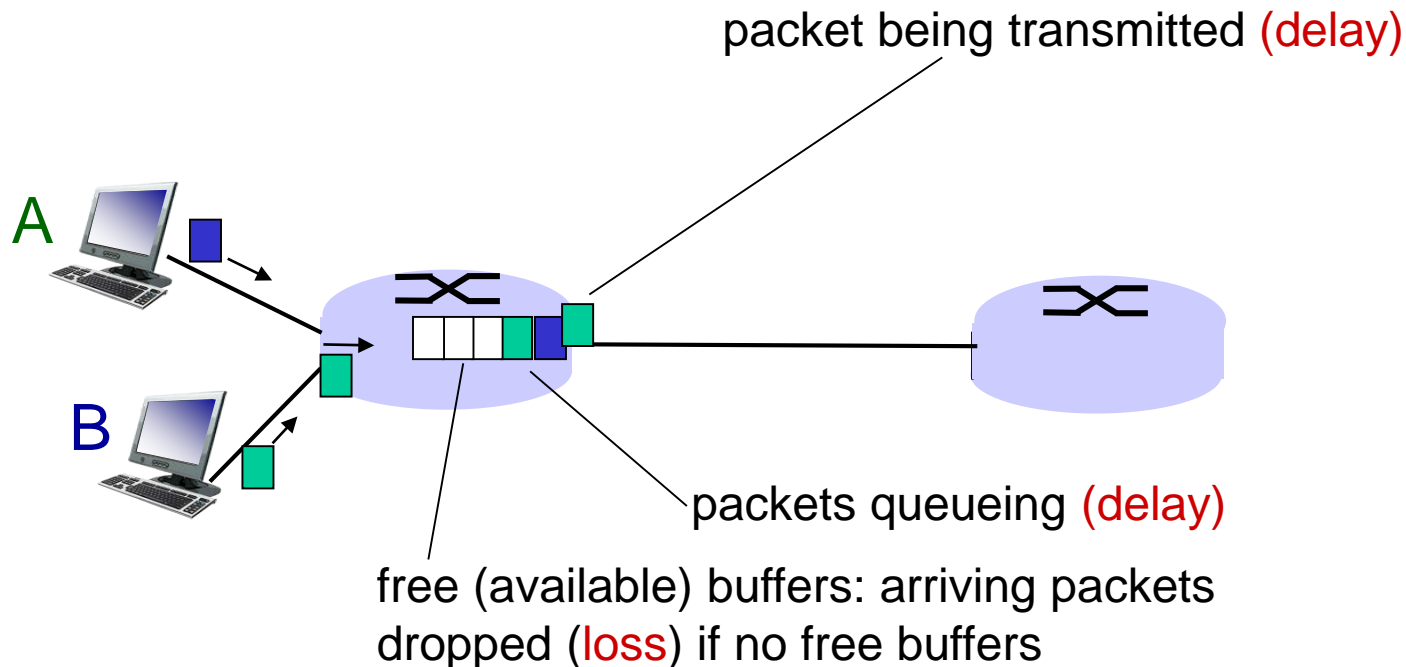
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

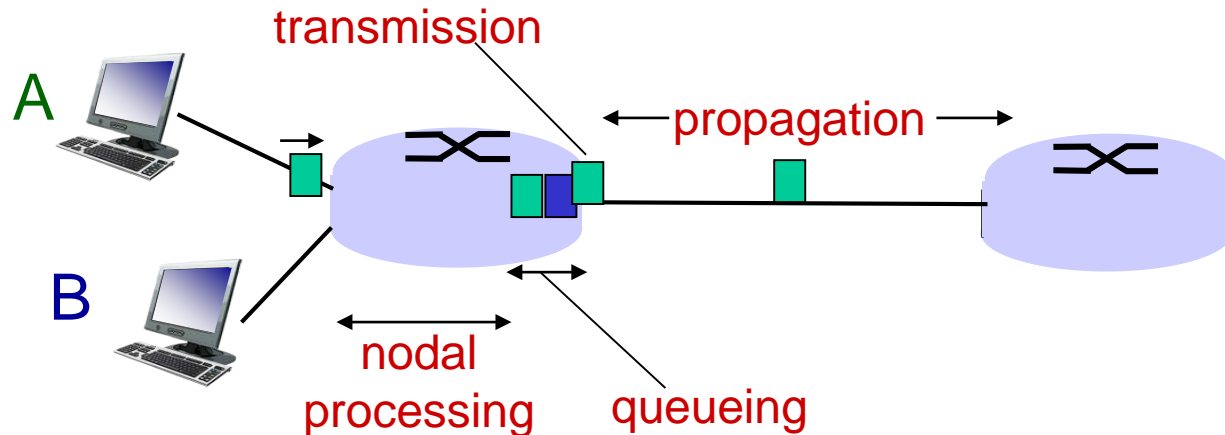
How do loss and delay occur?

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



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

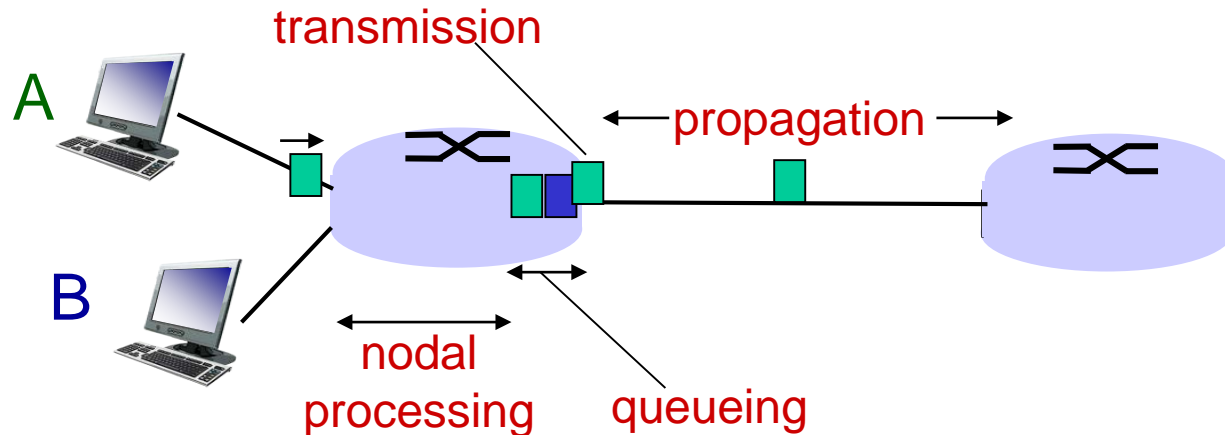
d_{proc} : nodal processing

- 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



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

d_{trans} : transmission delay:

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

d_{prop} : propagation delay:

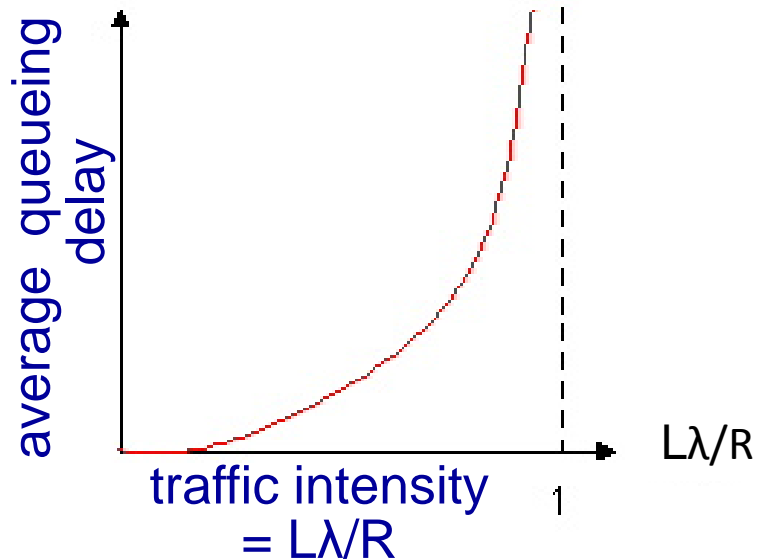
- d : length of physical link
- s : propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
- $d_{\text{prop}} = d/s$

d_{trans} and d_{prop}
very different

Queueing delay (revisited)

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

traffic intensity = $L\lambda/R$



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



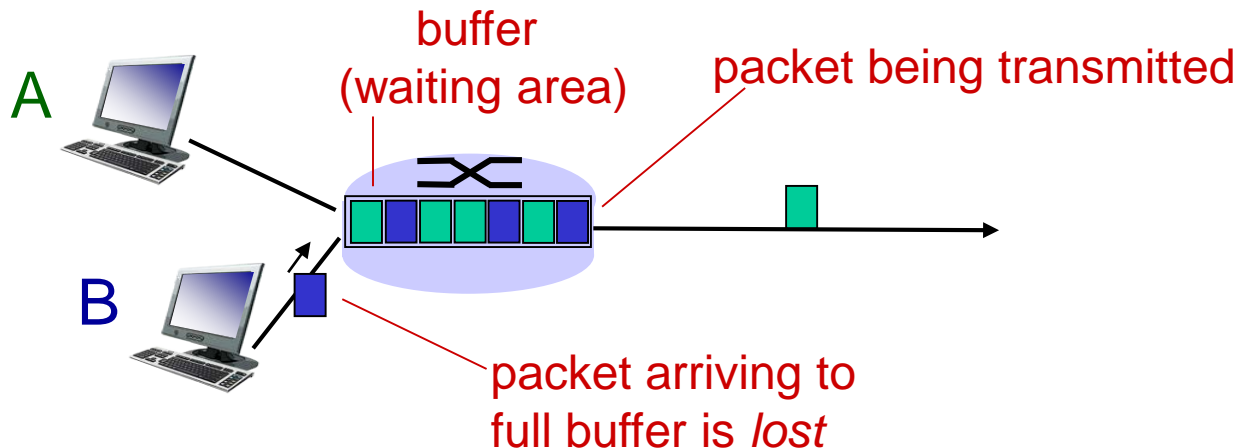
$L\lambda/R \sim 0$



$L\lambda/R \rightarrow 1$

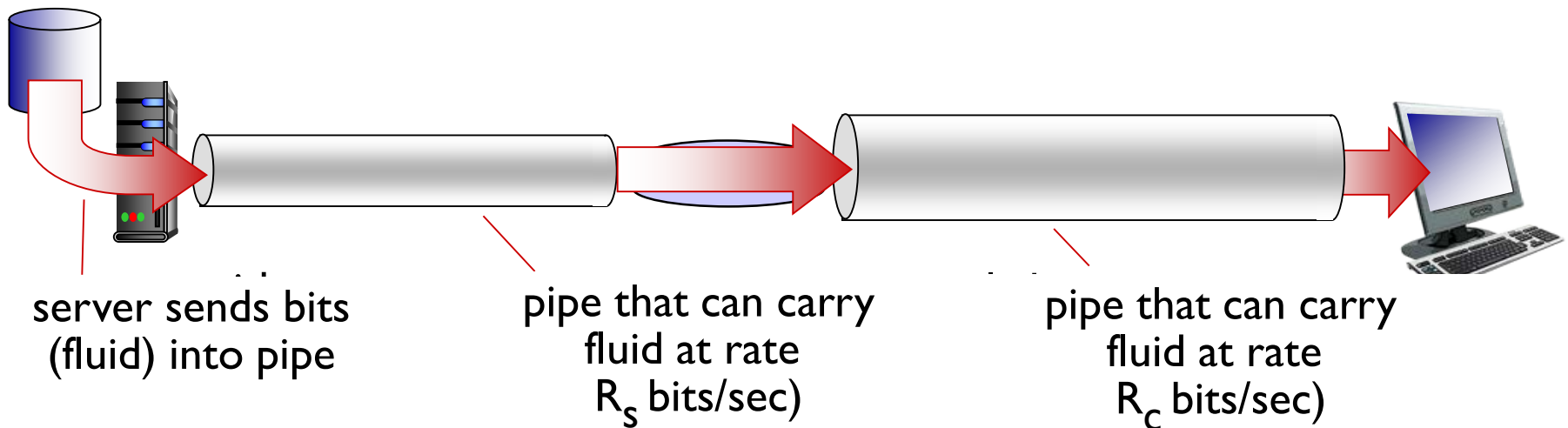
Packet loss

- ❖ 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



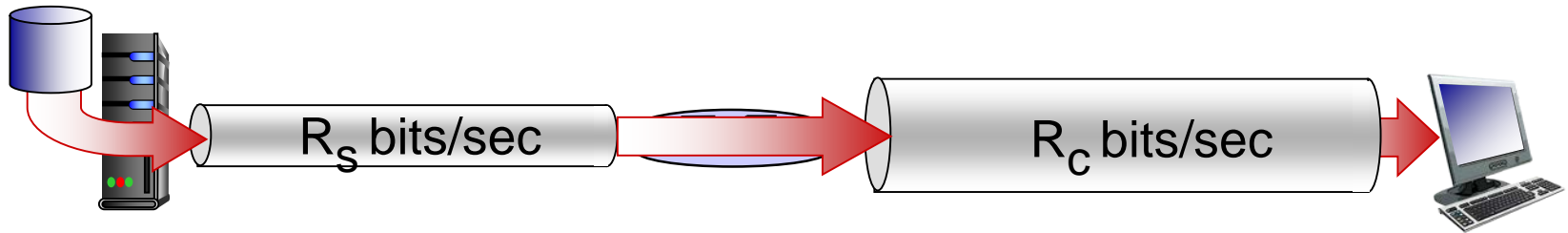
Throughput

- ❖ *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

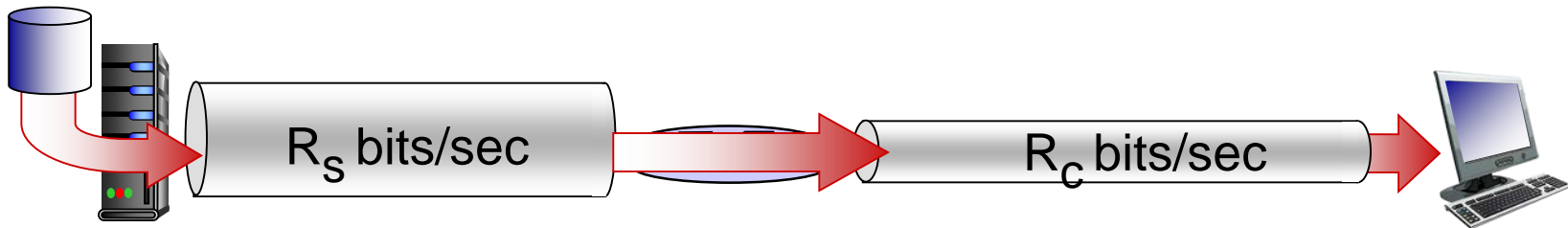


Throughput (more)

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



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

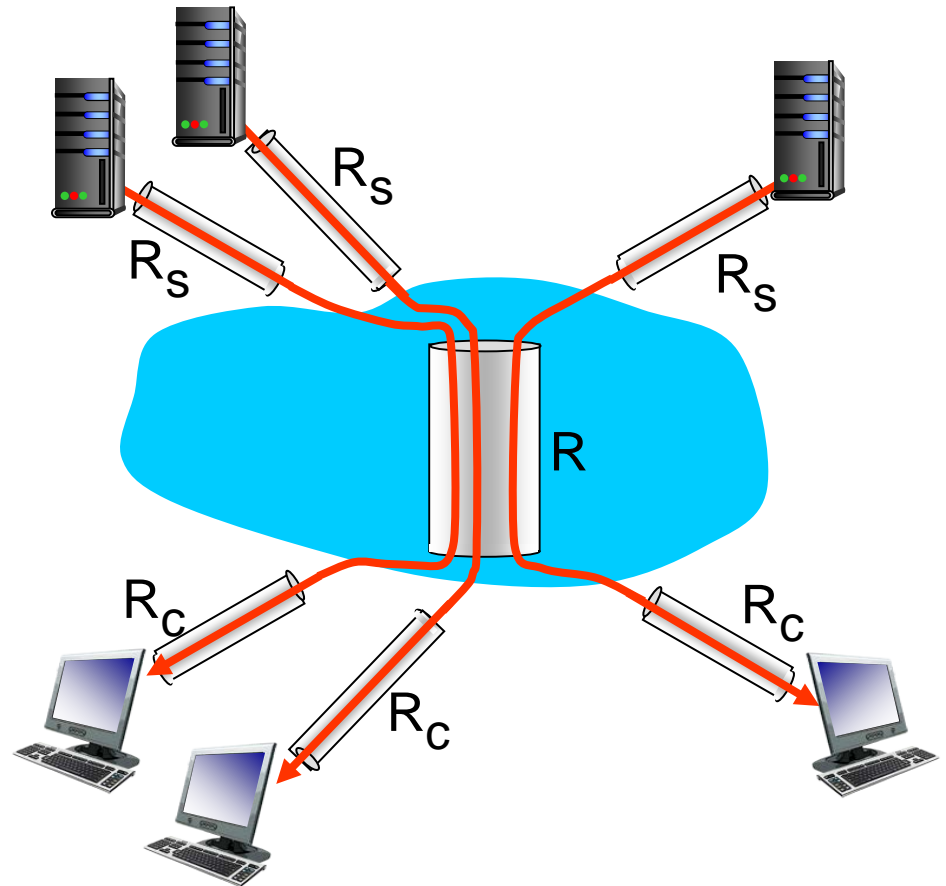


bottleneck link

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

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

Protocol “layers”

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

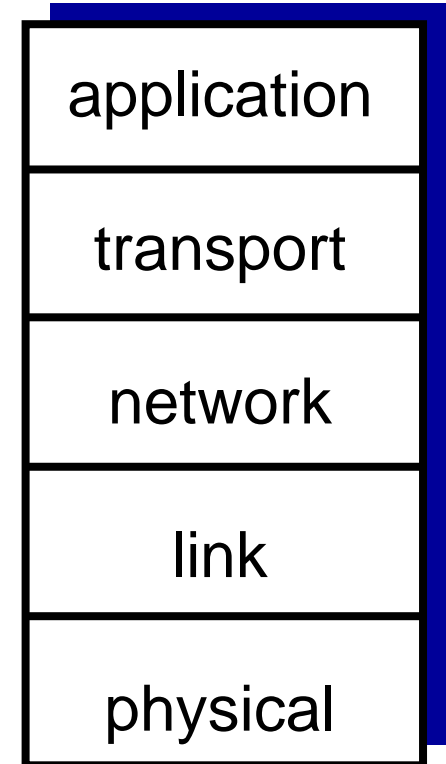
- hosts
- routers
- links of various media
- applications
- protocols
- hardware, software

Question:

How to *organize*
structure of network?

Protocol “layers”

*Organize network
components into “layers”*



layers: each layer implements a service

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

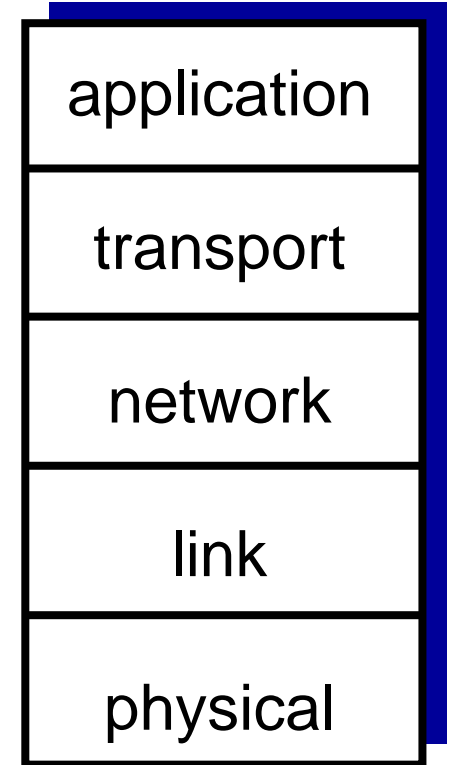
Why layering?

dealing with complex systems:

- ❖ explicit structure allows identification, relationship of complex system's pieces
- ❖ modularization eases maintenance, updating of system
 - change of implementation of layer's service transparent to rest of system
- ❖ layering considered harmful?

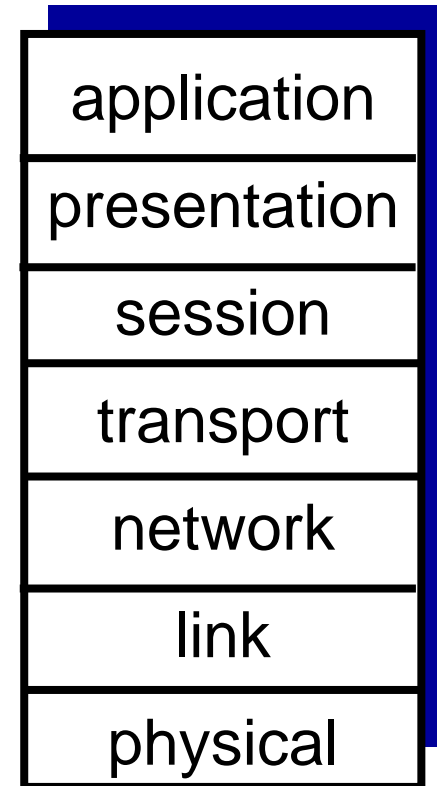
Internet protocol stack

- ❖ *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, 802.11 (WiFi), PPP
- ❖ *physical*: bits “on the wire”

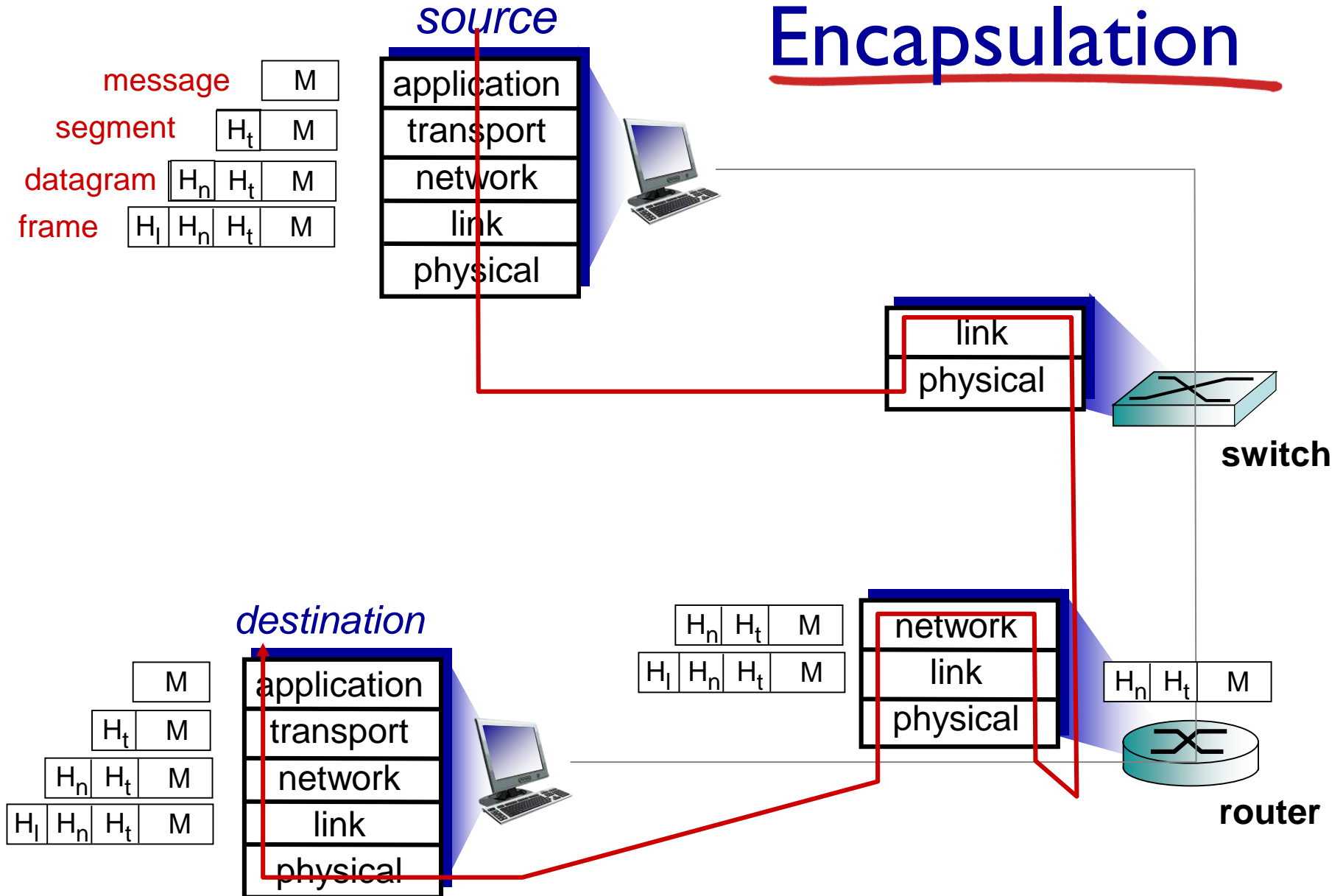


ISO/OSI reference model

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



Encapsulation



Acknowledgement

These notes are adapted
from the publishers material.

All material copyright 1996-2016
J. F Kurose and K. W. Ross
All Rights Reserved.

