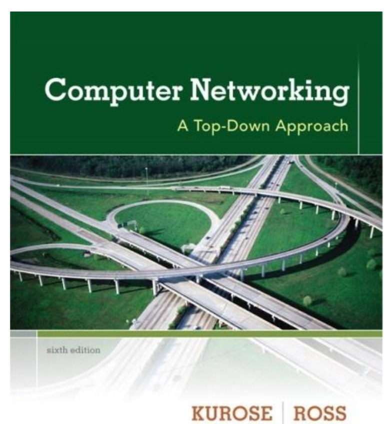
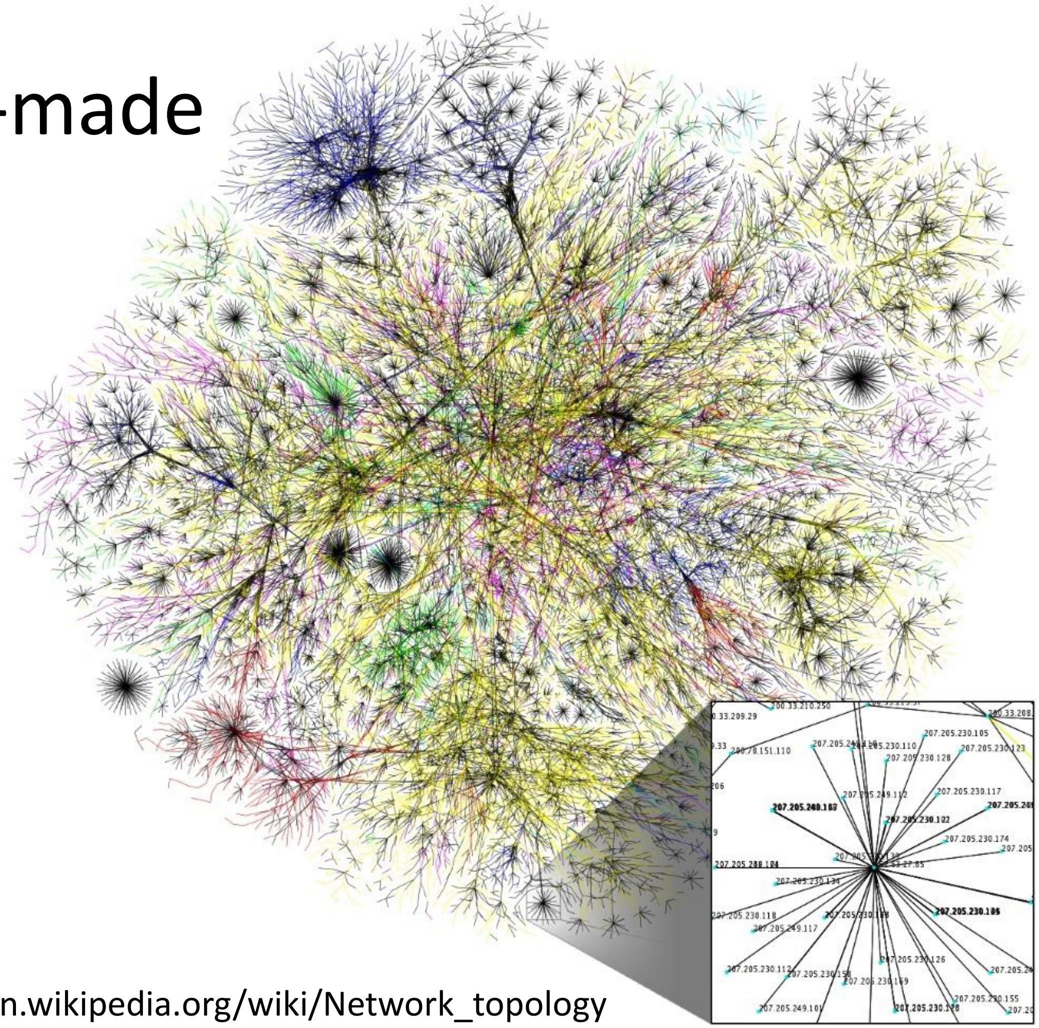
CSC 430/630
Computer Networking



Why this course?

- Most complex man-made technical system
- 93% of USA people use the Internet
- Over 60% of people on earth use the Internet



From https://en.wikipedia.org/wiki/Network_topology

Why this course? (cont.)

- Most popular IT companies now more or less related to computer network and the Internet
  - Especially unicorn companies


14°/10°

Chapter 1: Introduction

CSC 430/630

Chapter 1: roadmap

1.1 what is the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

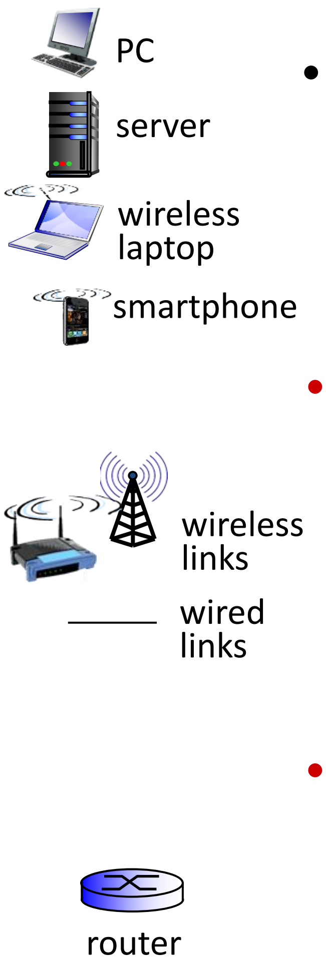
- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

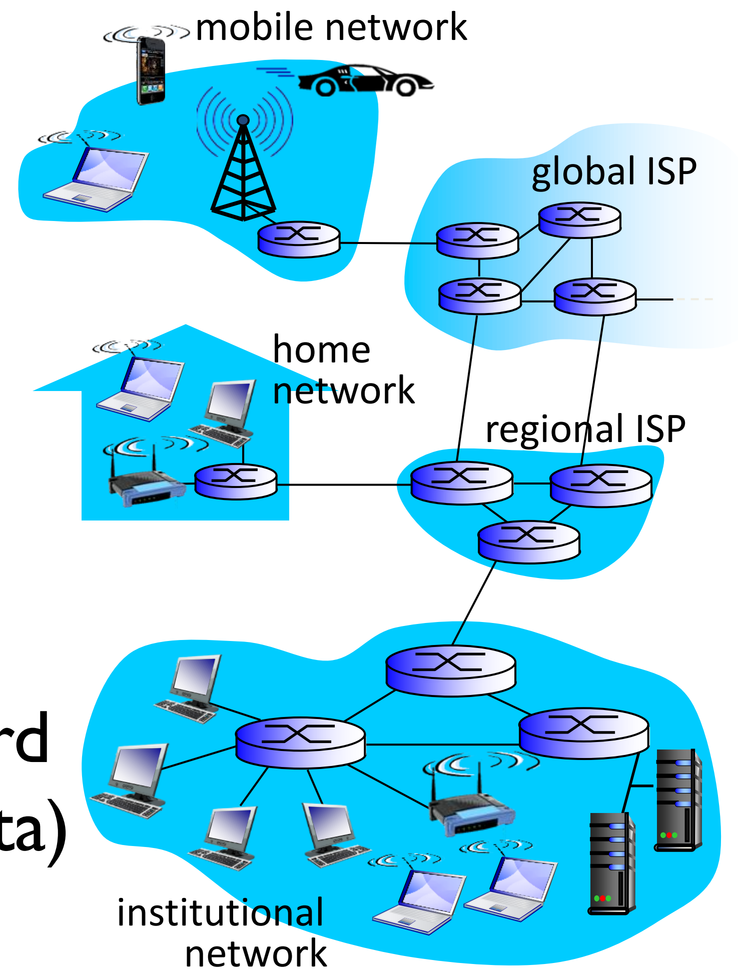
1.5 protocol layers, service models

1.6 networks under attack: security

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

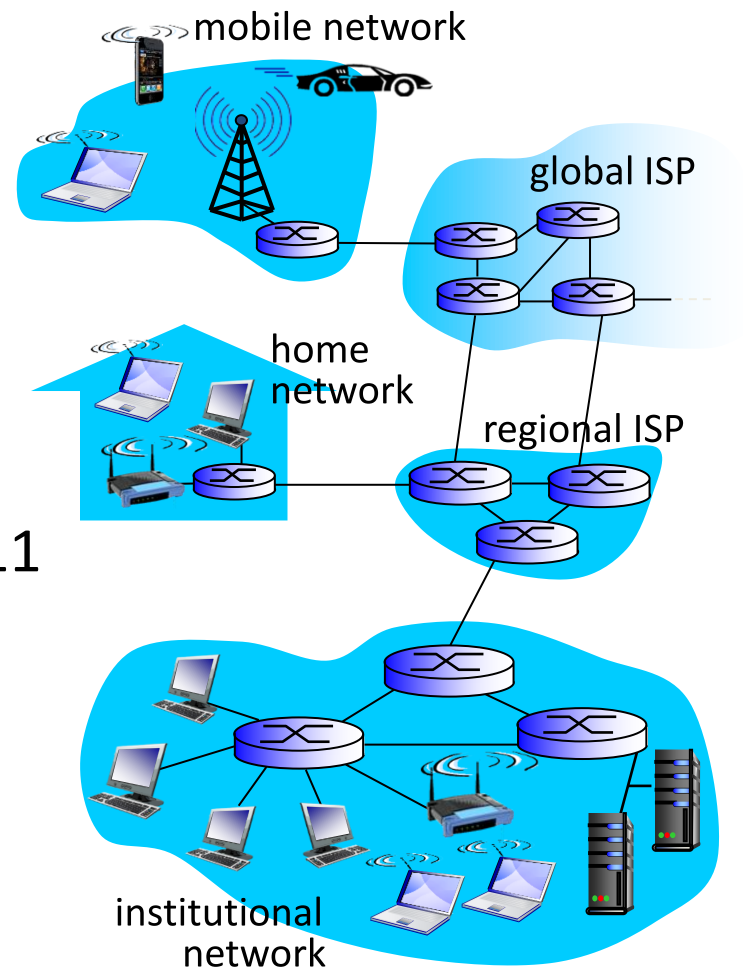


- millions of connected computing devices:
  - *hosts* = *end systems*
  - running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate: *bandwidth*
- *Packet switches:* forward packets (chunks of data)
  - *routers* and *switches*



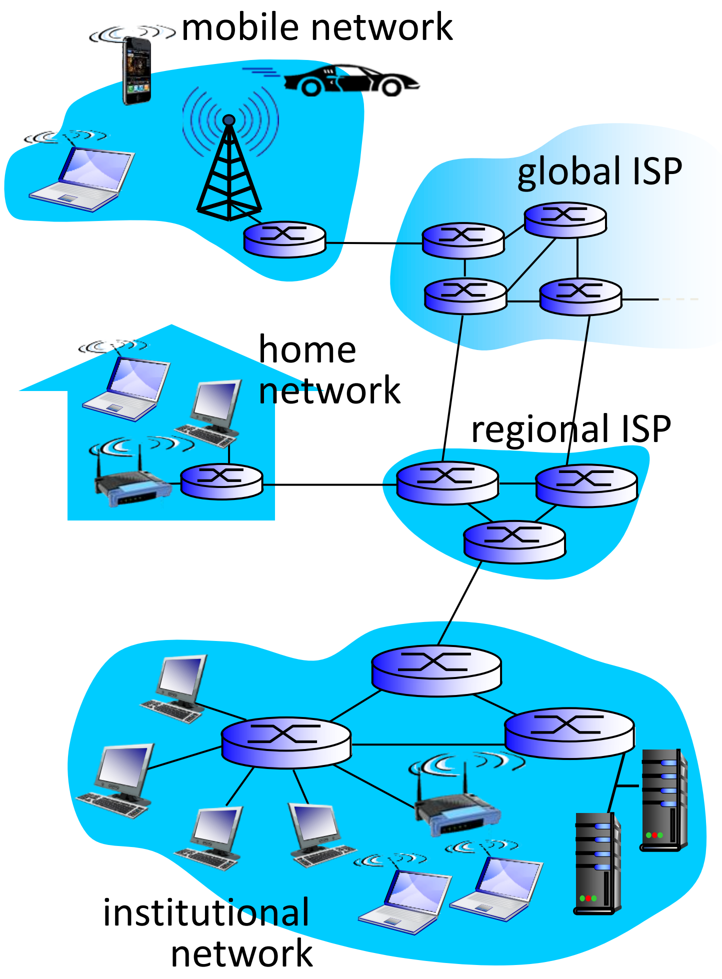
What's the Internet: "nuts and bolts" view

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

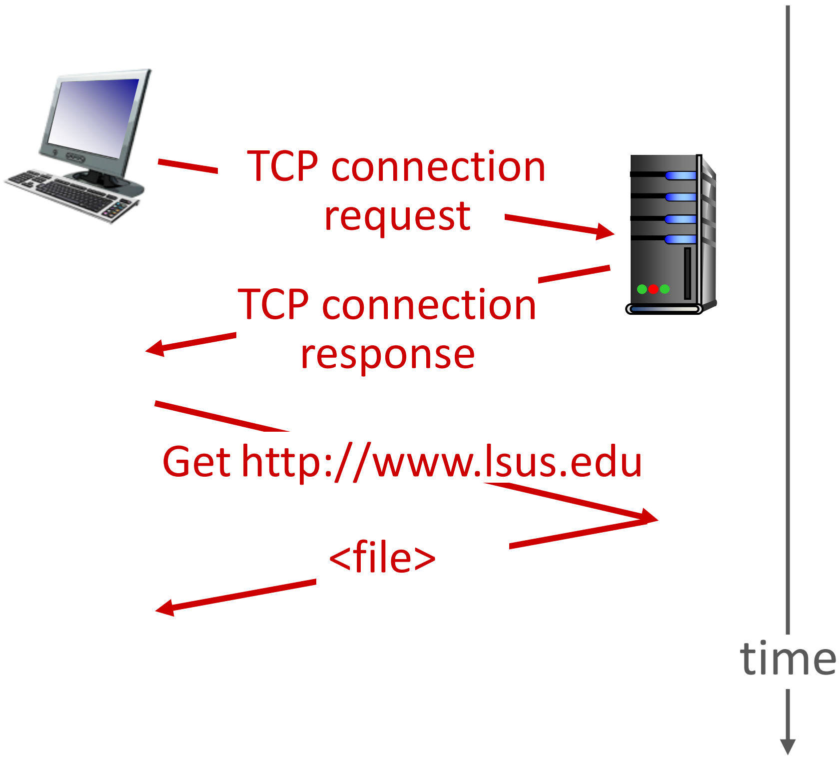


What's the Internet: a service view

- *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?



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

Chapter 1: roadmap

1.1 what is the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

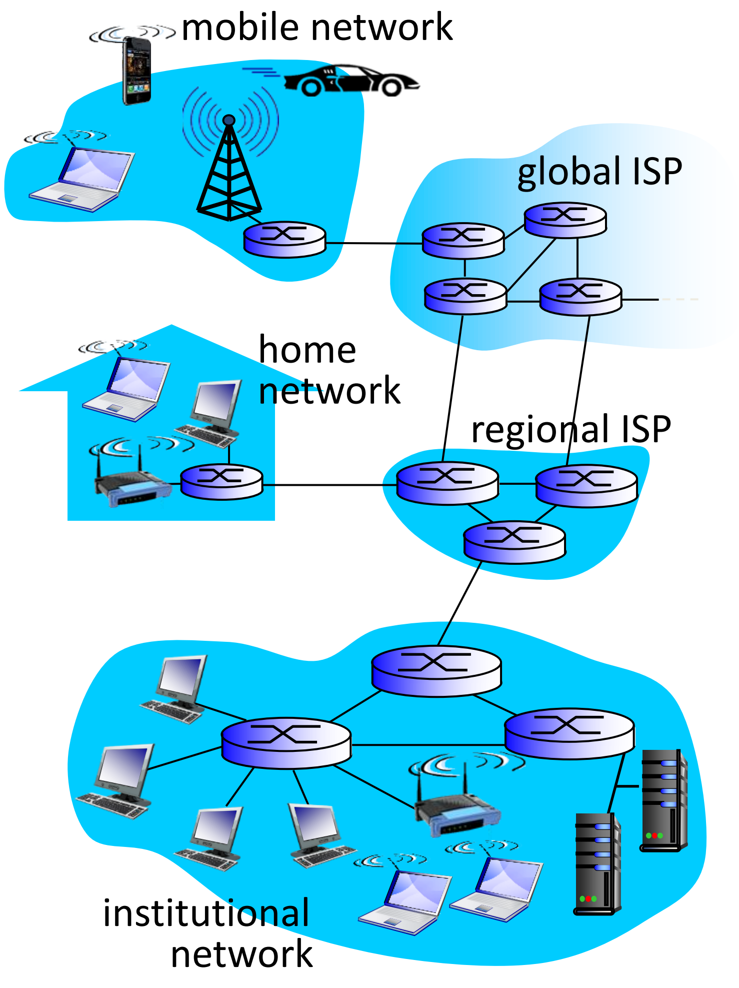
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

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



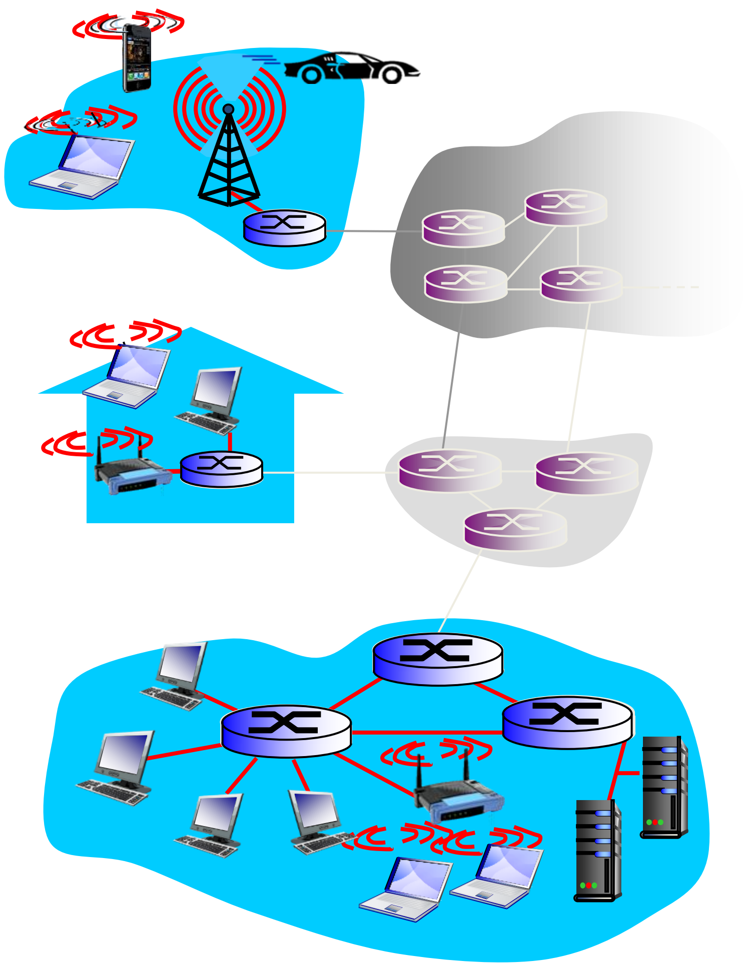
Access networks and physical media

Q: How to connect end systems to edge router?

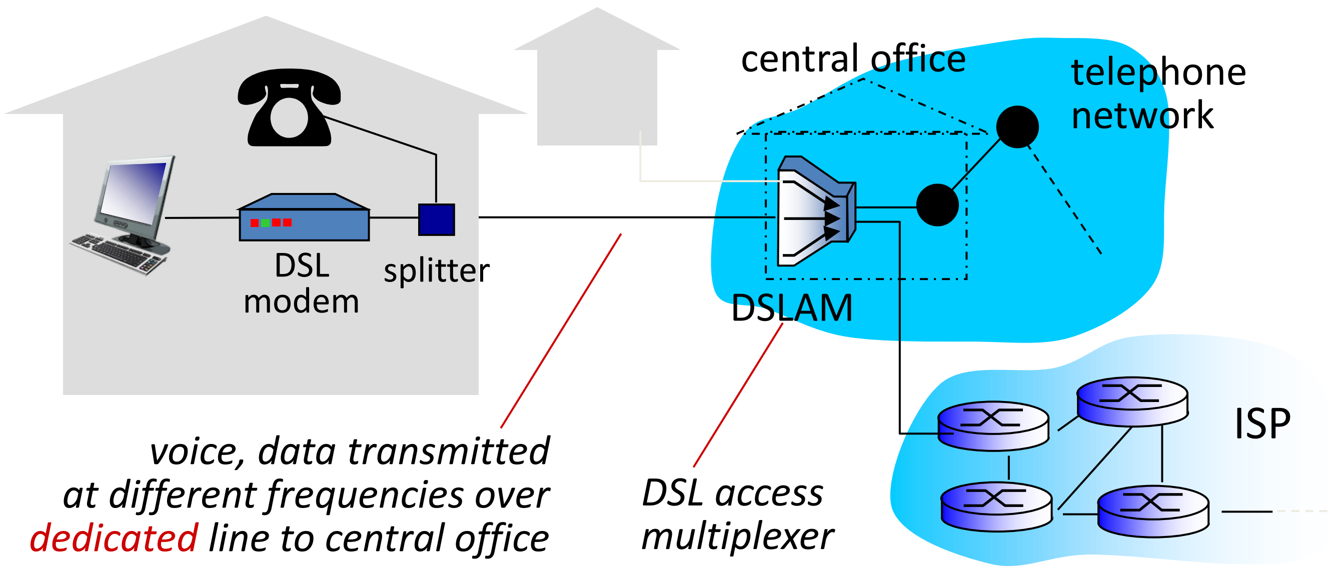
- residential access nets
- institutional access networks (school, company)
- mobile access networks

keep in mind:

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

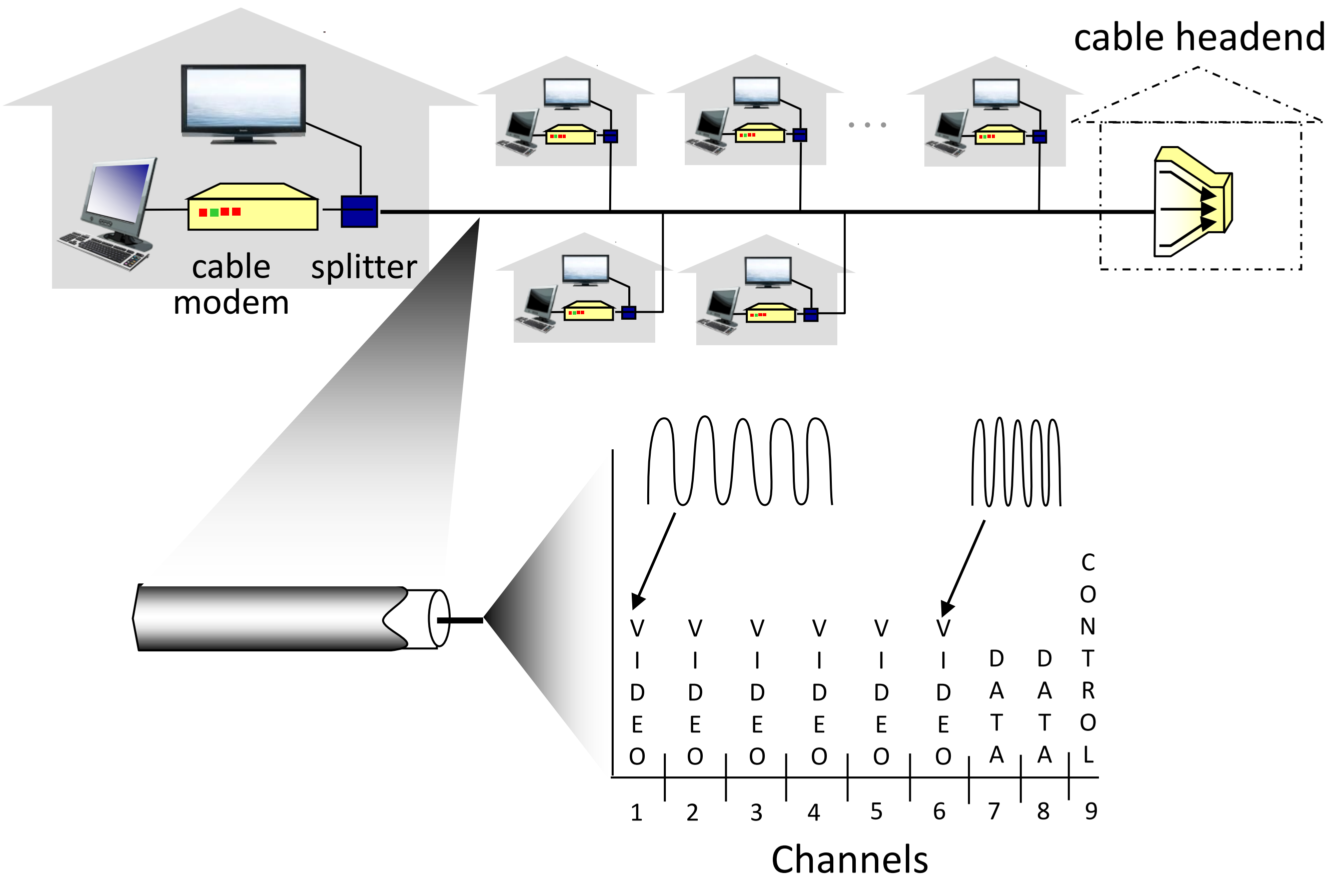


Access net: digital subscriber line (DSL)



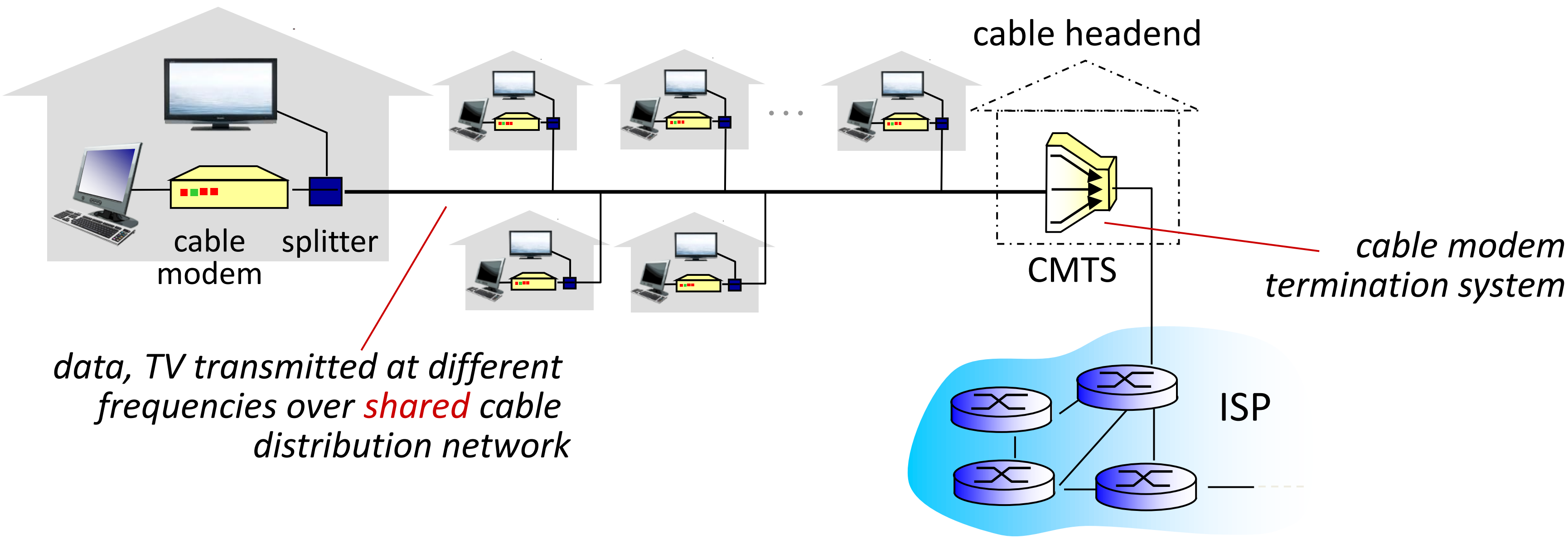
- < 2.5 Mbps upstream transmission rate (typically < 1 Mbps)
- < 24 Mbps downstream transmission rate (typically < 10 Mbps)

Access net: cable network

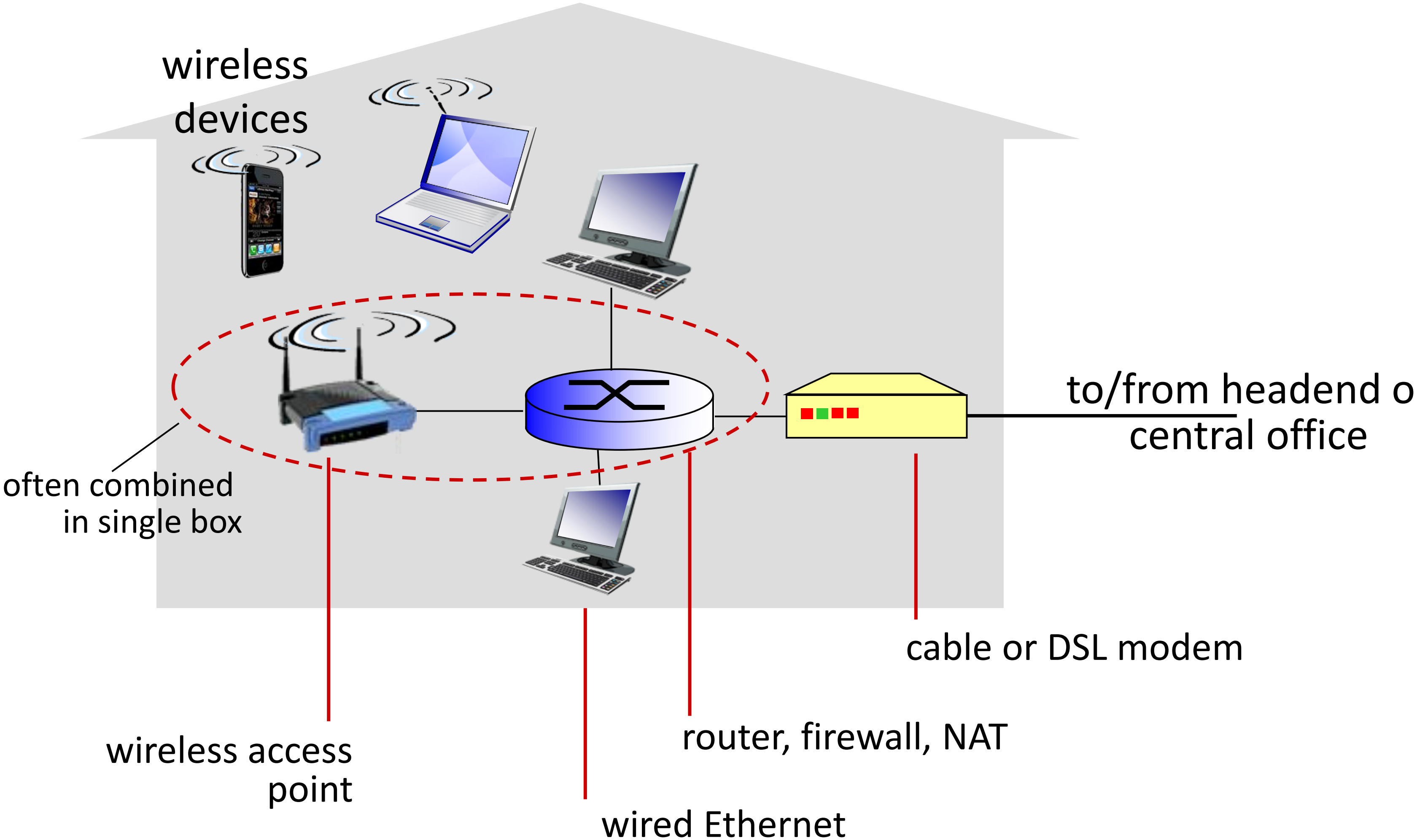


frequency division multiplexing: different channels transmitted in different frequency bands

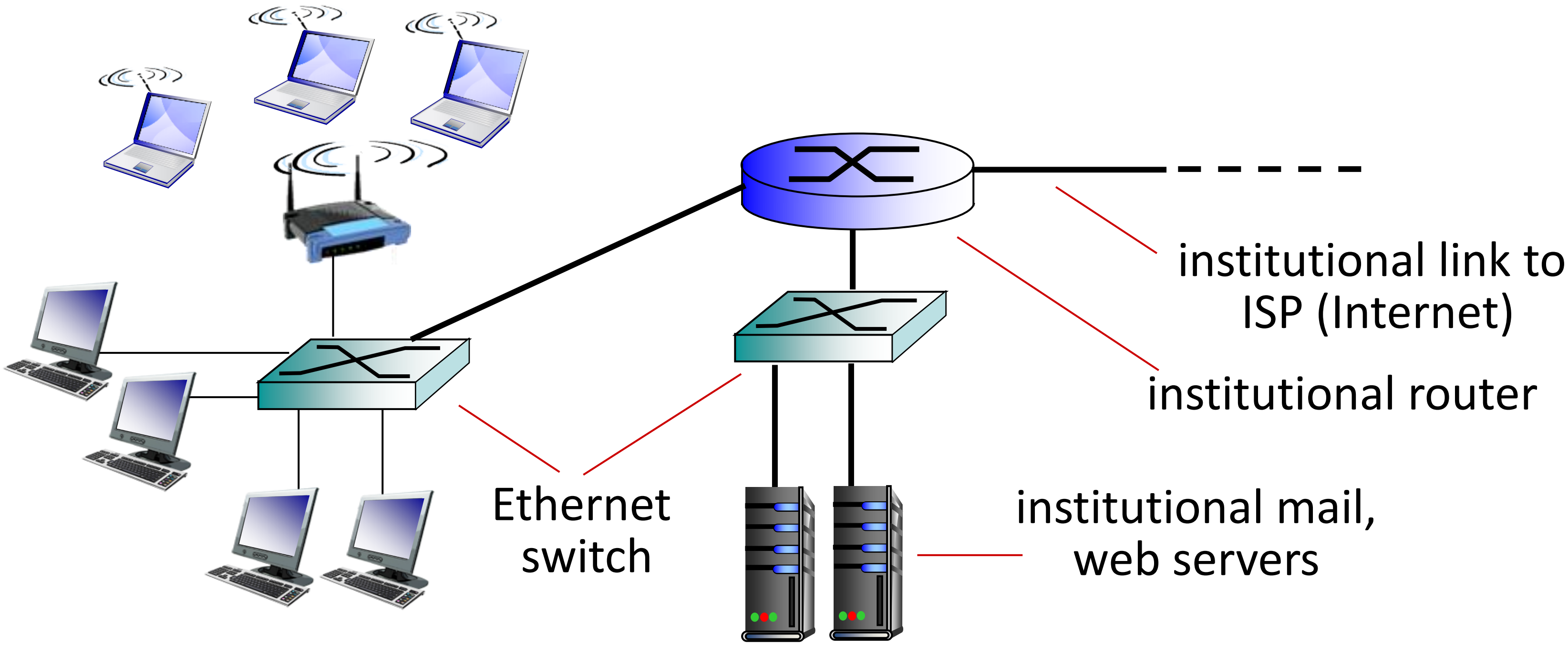
Access net: cable network



Access net: home network



Institutional access networks (Ethernet)



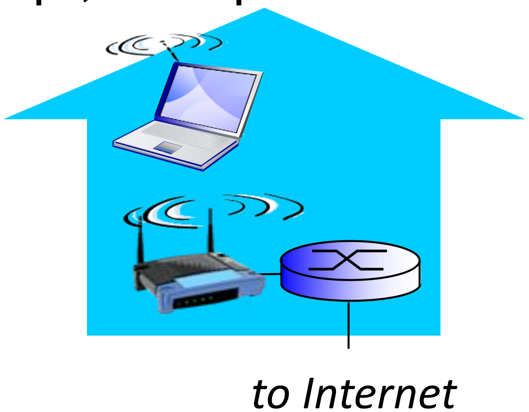
- 100Mbps, 1Gbps, 10Gbps transmission rates

Wireless access networks

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

wireless LANs:

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



wide-area wireless access

- provided by telecommunication (cellular) operator, 10's km
- between 1 Mbps to 1 Gbps
- 3G, 4G: LTE, 5G

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

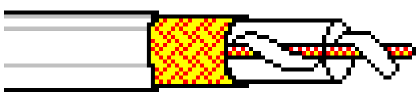
twisted pair (TP)

- Category 5: 100 Mbps, 1 Gpbs Ethernet (CAT5e)
- Category 6: 10Gbps
- Category 8/8.1/8.2: 25Gbps – 40Gbps

Physical media: coax, fiber

coaxial cable:

- broadband:
  - multiple channels on cable



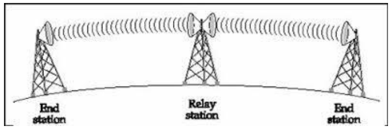
fiber optic cable:

- high-speed
  - e.g., 10's-100's Gbps 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)
  - 11Mbps, 54 Mbps, 11 Gbps
- wide-area (e.g., cellular)
  - 5G cellular: ~ 1 Gbps
- satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay

Chapter 1: roadmap

1.1 what is the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

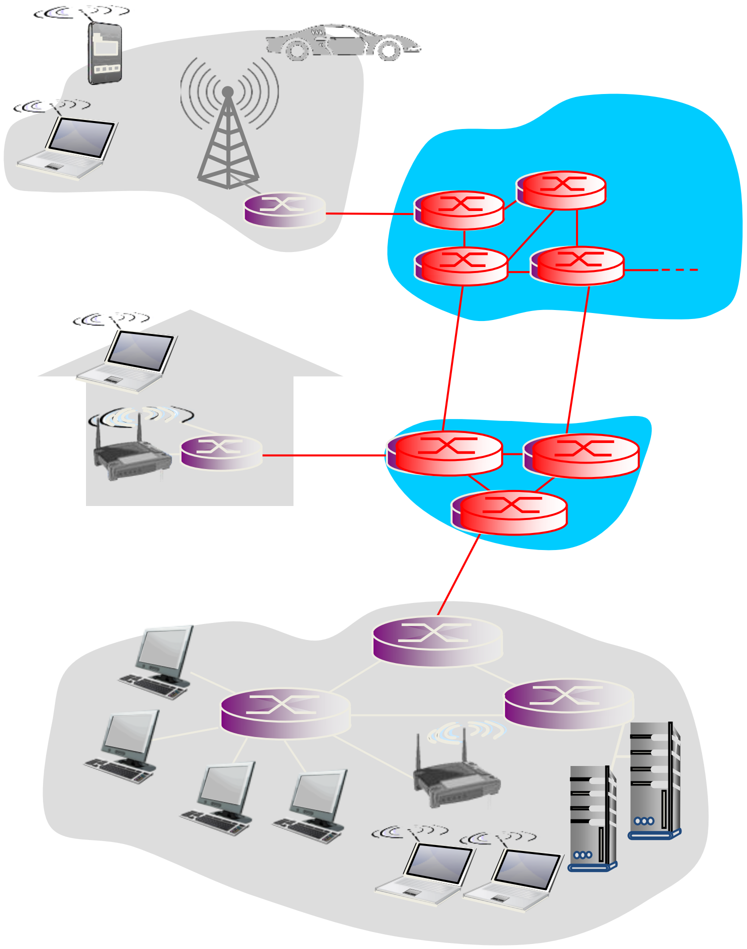
1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

The network core

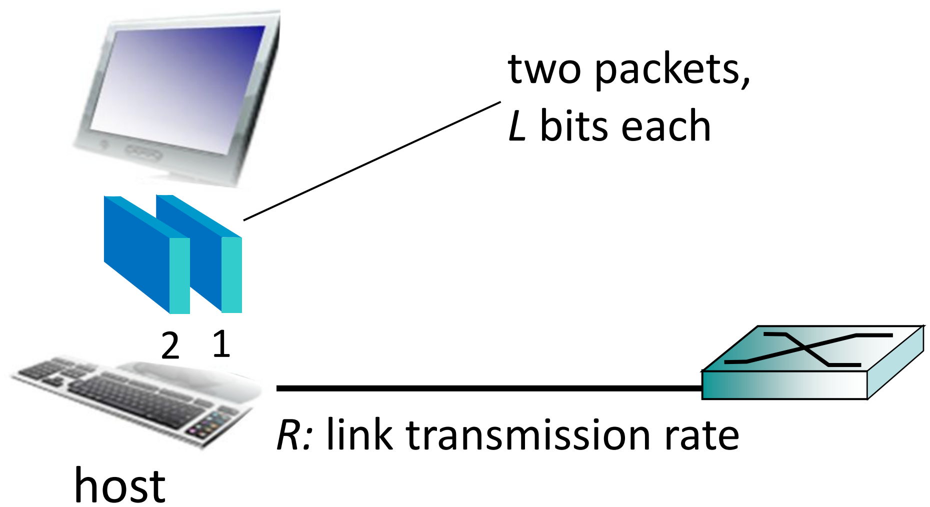
- mesh of interconnected routers
- packet-switching: hosts break application-layer messages into *packets*
  - forward packets from one router to the next, across links on path from source to destination
  - each packet transmitted at full link capacity



Host: sends *packets* of data

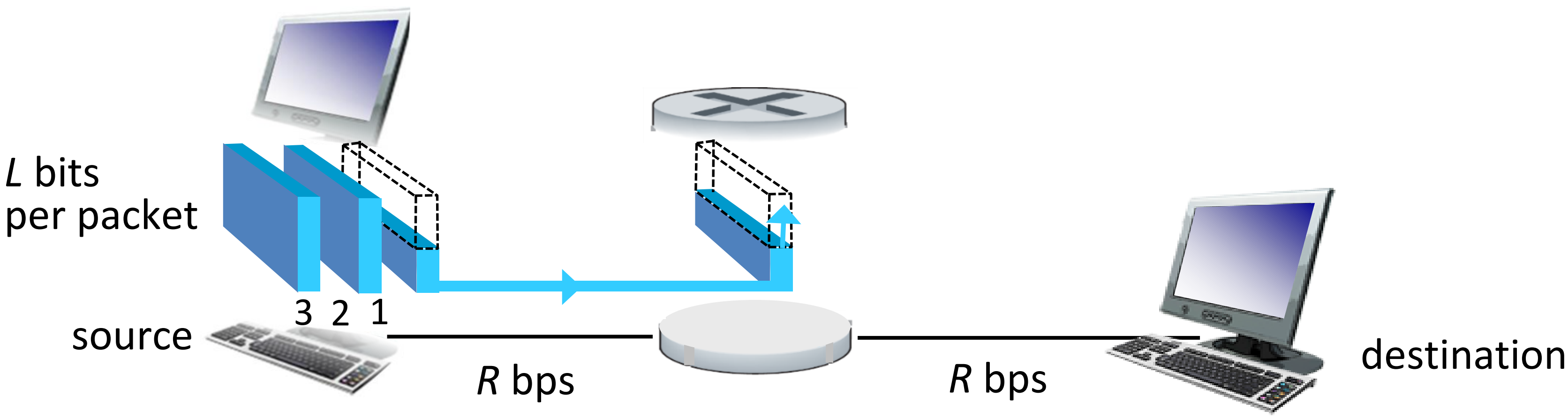
host sending function:

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



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

Packet-switching: store-and-forward



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

one-hop numerical example:

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

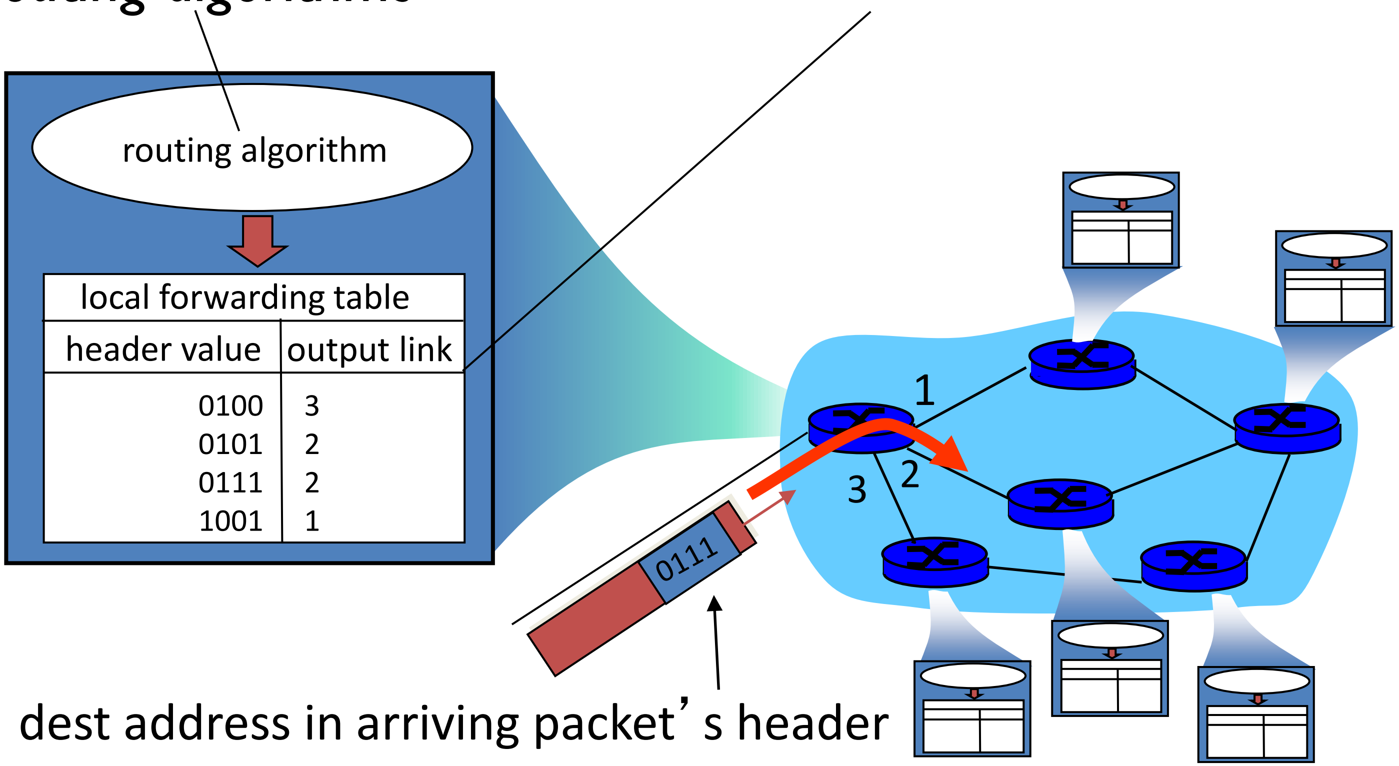
more on delay shortly …

Two key network-core functions

routing: determines source-destination route taken by packets

- *routing algorithms*

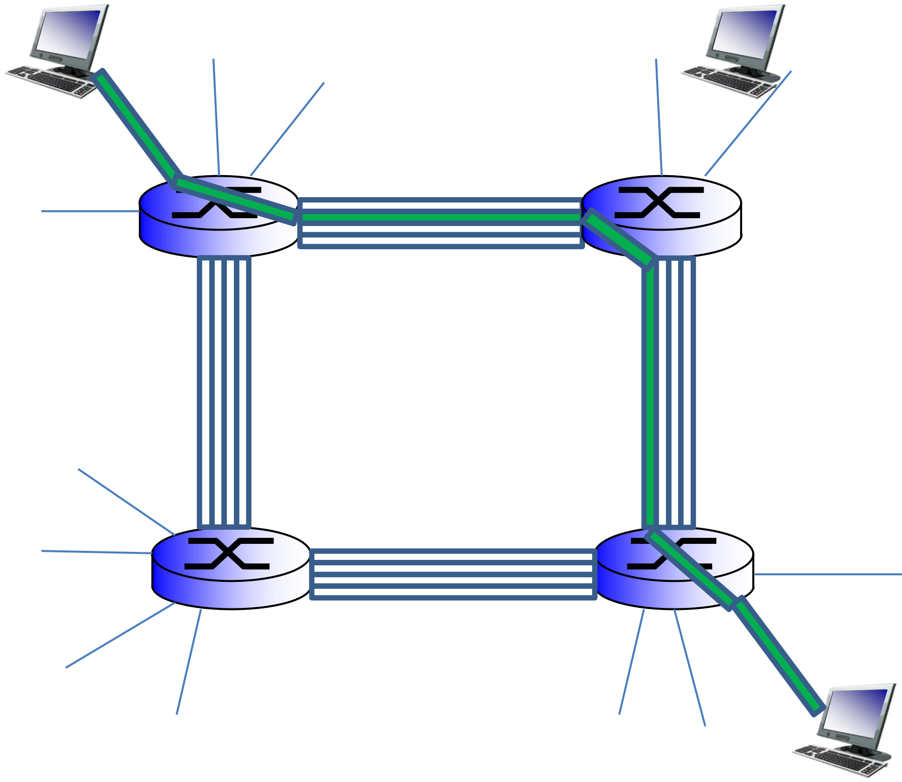
forwarding: move packets from router's input to appropriate router output



Alternative core: circuit switching

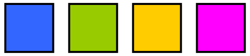
end-end resources allocated to, reserved for "call" between source & dest:

- 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

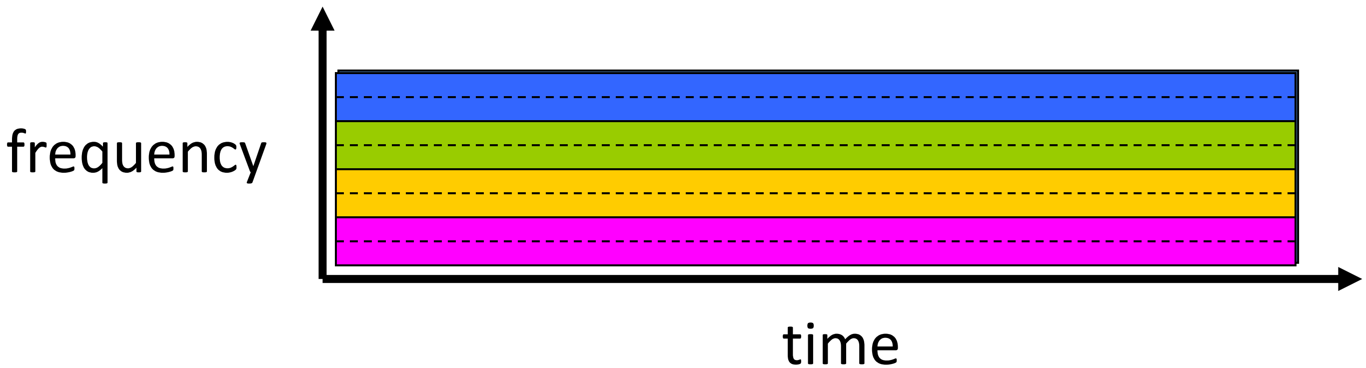


Circuit switching: FDM versus TDM

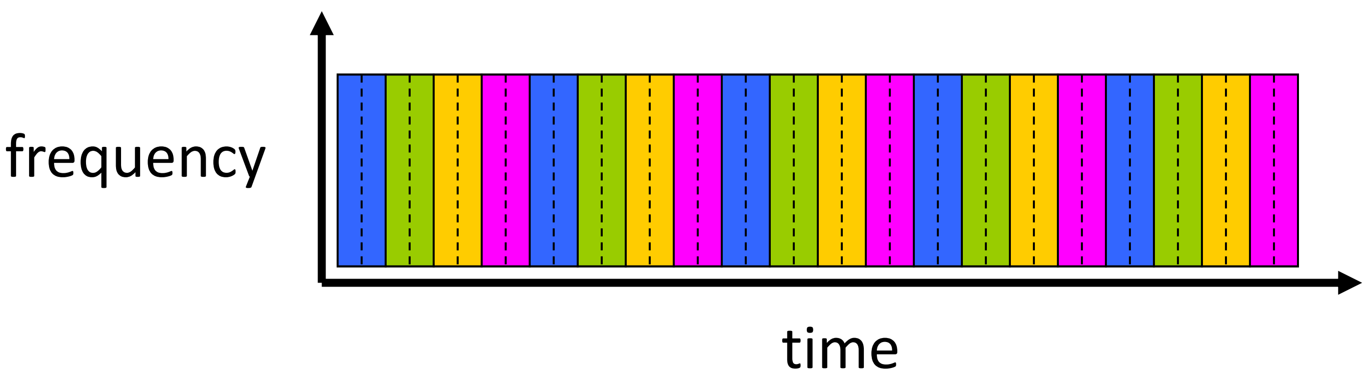
Example:

4 users 

FDM



TDM



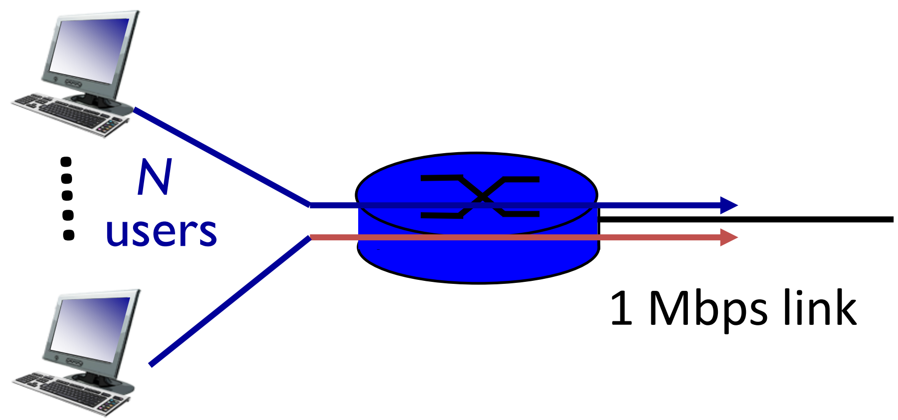
Packet switching versus circuit switching

packet switching allows more users to use network!

example:

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

- *circuit-switching:*
  – 10 users
- *packet switching:*
  – with 35 users, probability (> 10 active at same time) is less than .0004



Q: how did we get value 0.0004?

Q: what happens if > 35 users ?

Packet switching versus circuit switching

is packet switching a "slam dunk winner?"

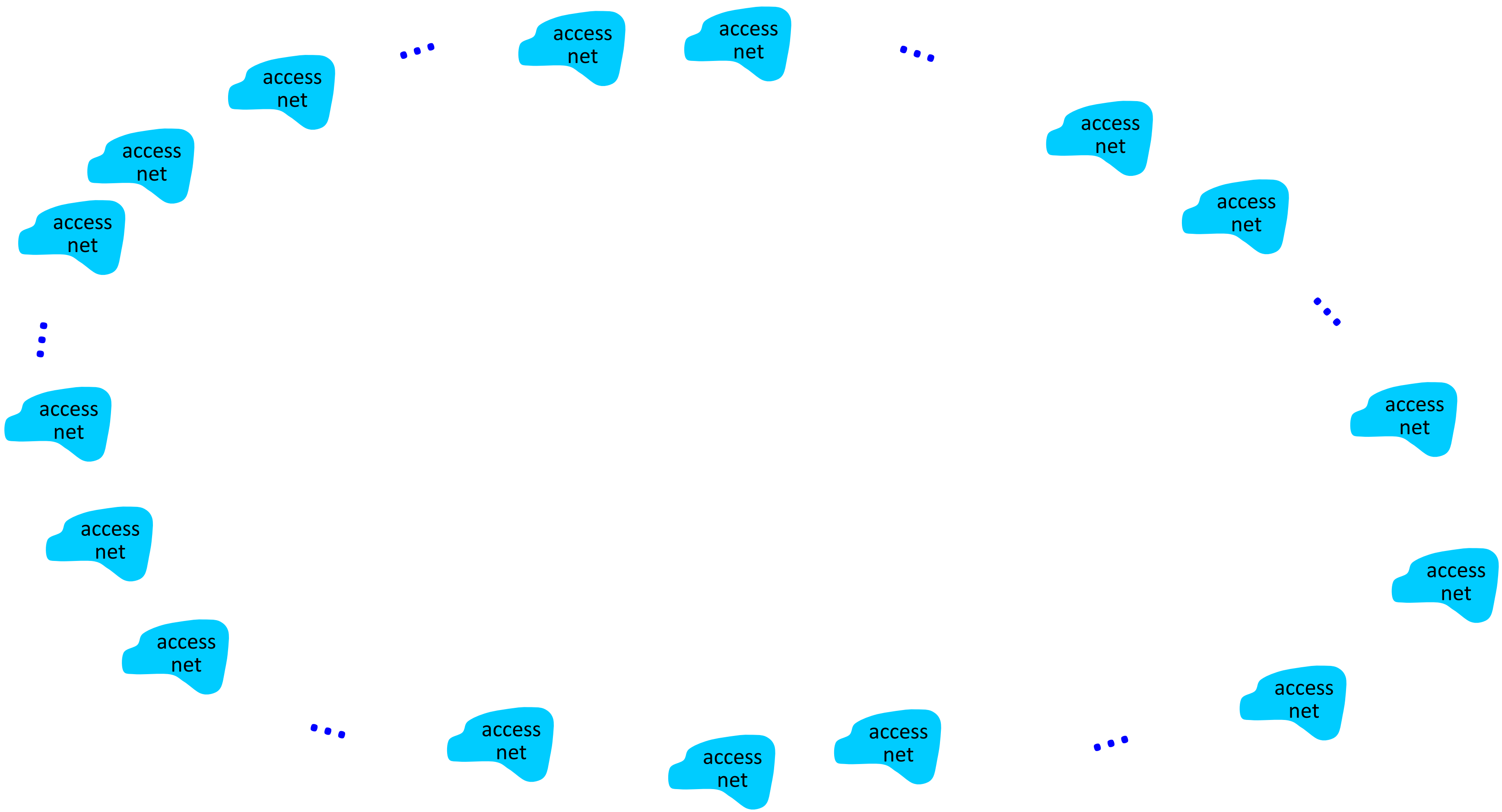
- great for bursty data
  - resource sharing
  - simpler, no call setup
- excessive congestion possible: packet delay and loss
  - protocols needed for reliable data transfer, congestion control
- *Q:* How to provide circuit-like behavior?
  - bandwidth guarantees needed for audio/video apps
  - still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

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

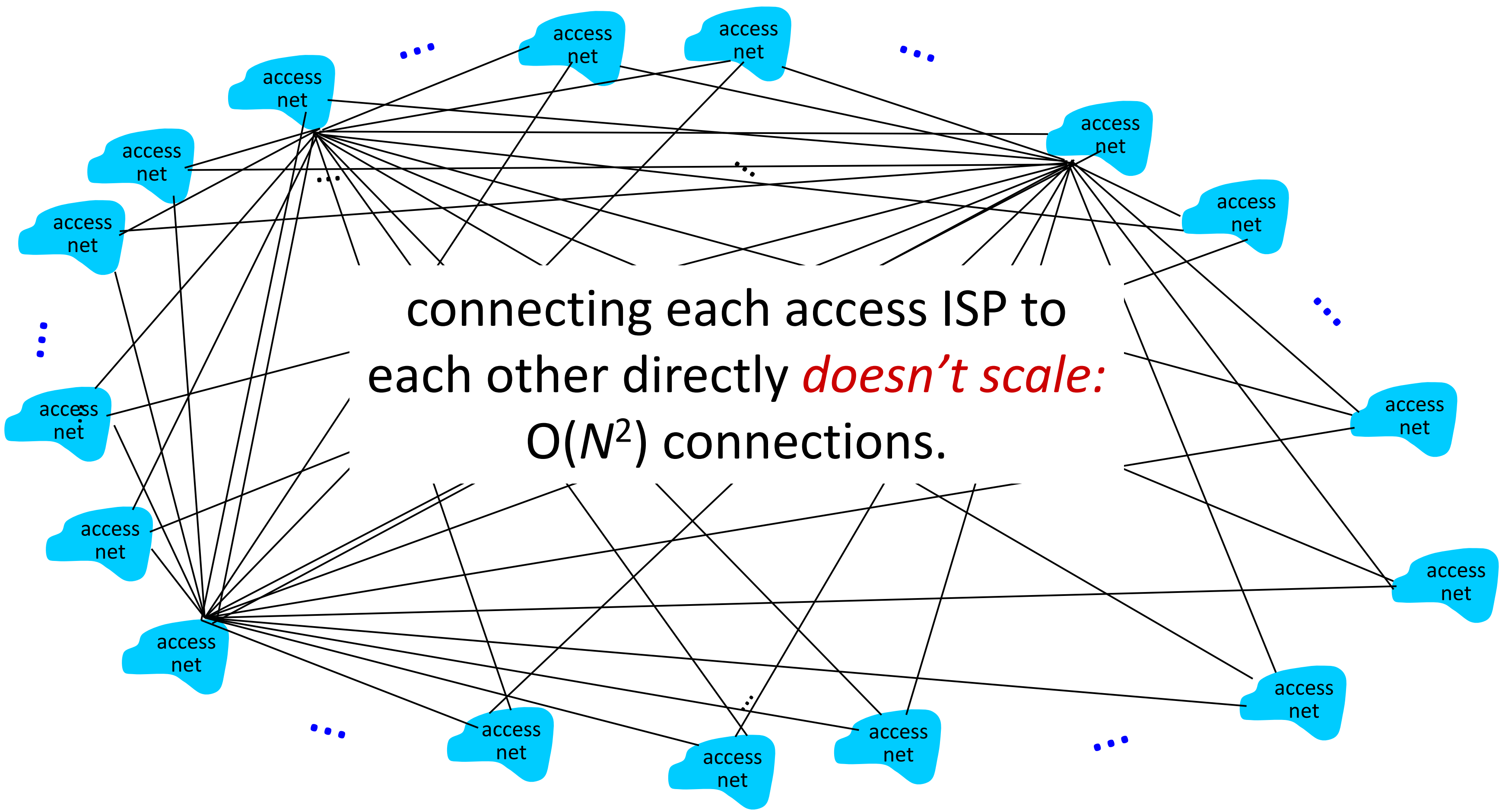
Internet structure: network of networks

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



Internet structure: network of networks

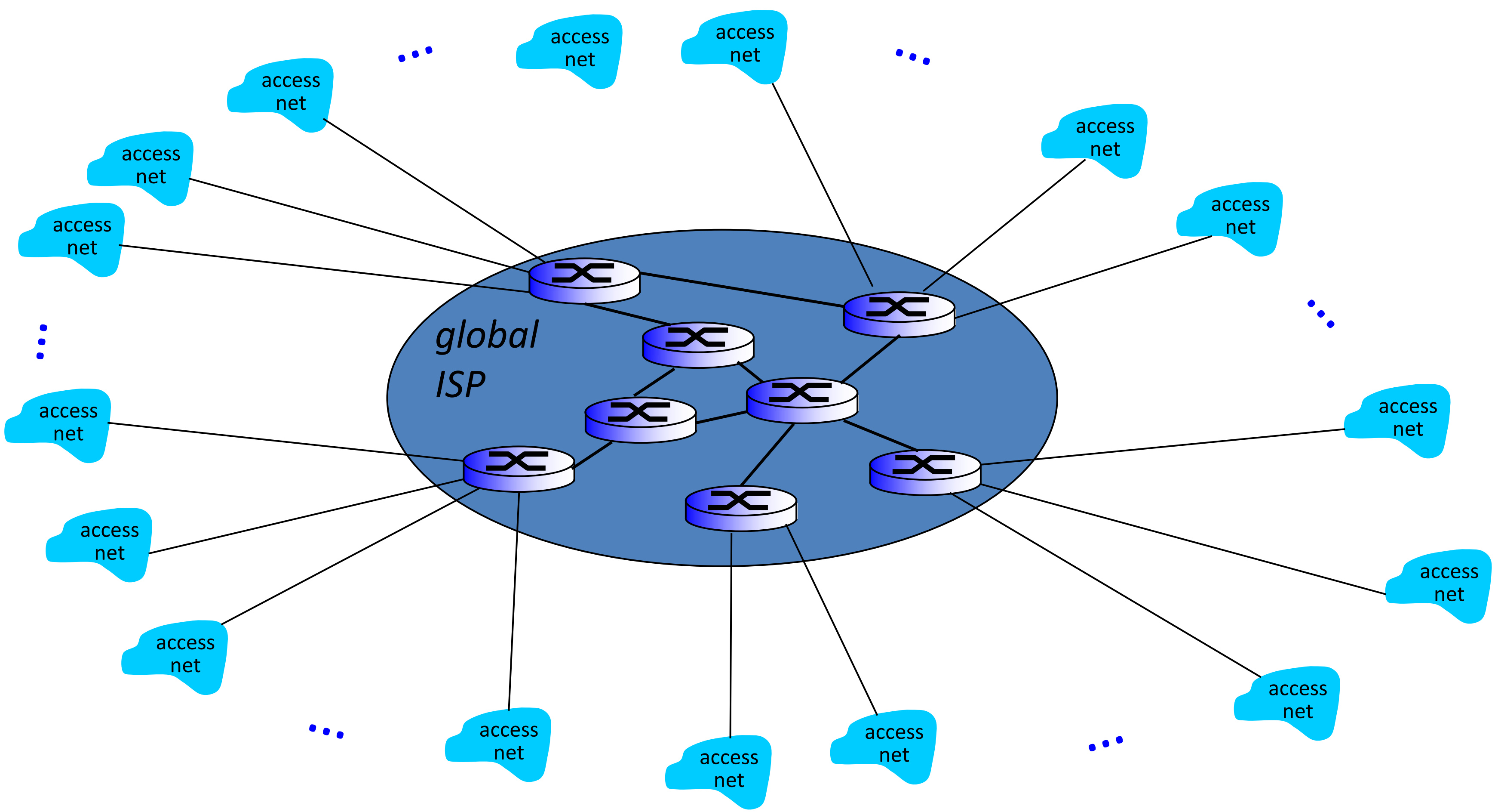
Option: connect each access ISP to every other access ISP?



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

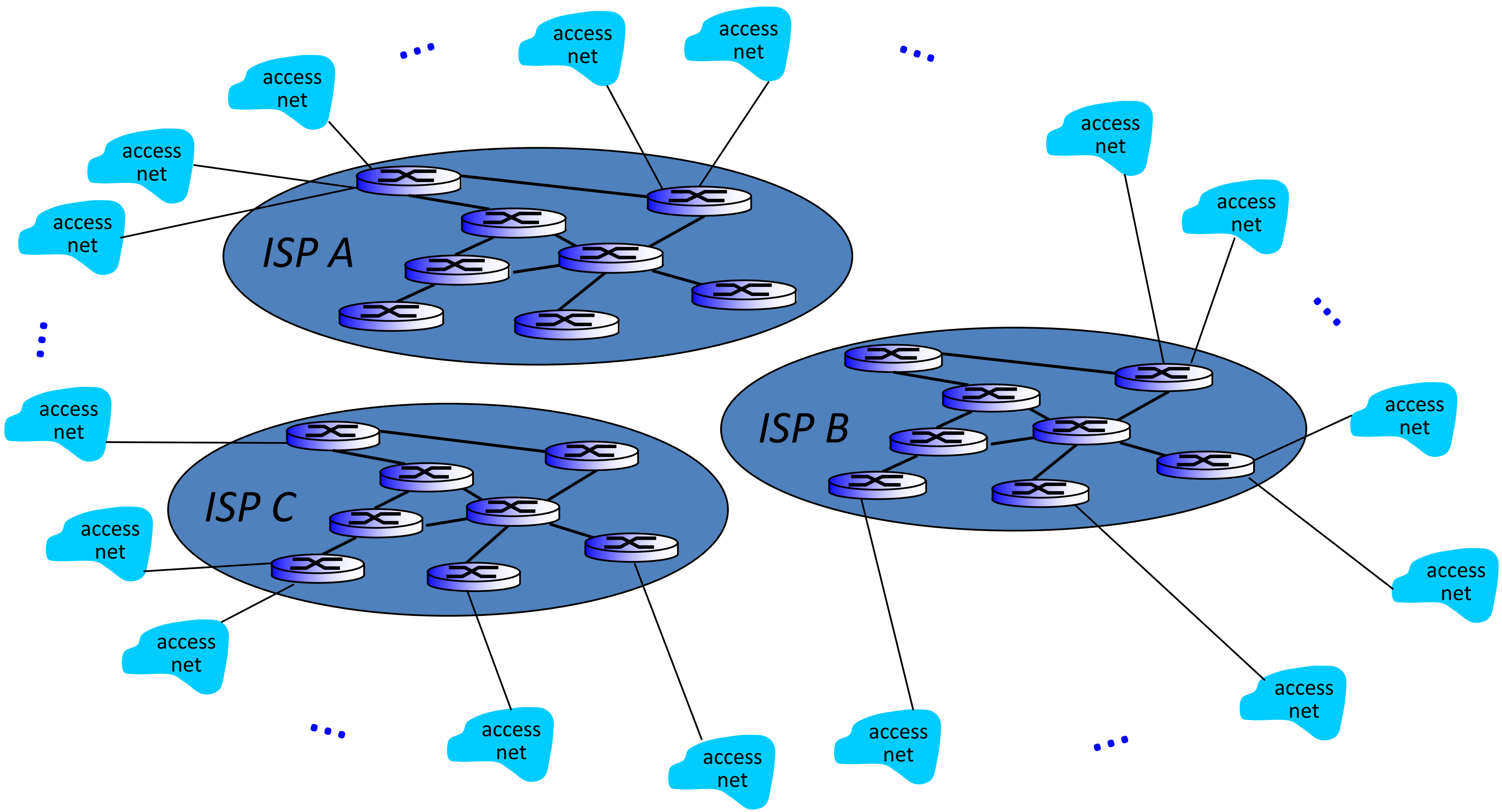
Internet structure: network of networks

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



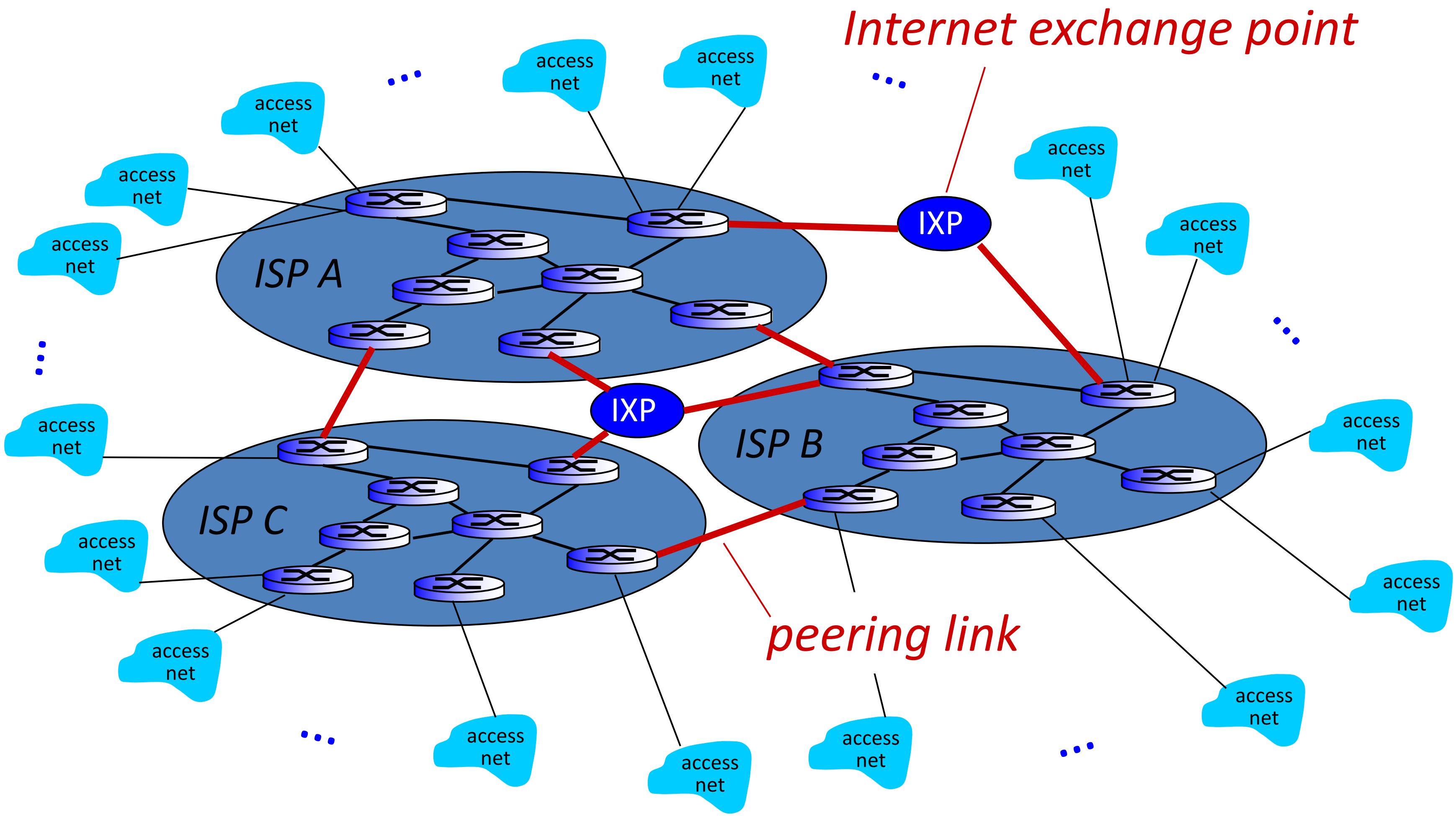
Internet structure: network of networks

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



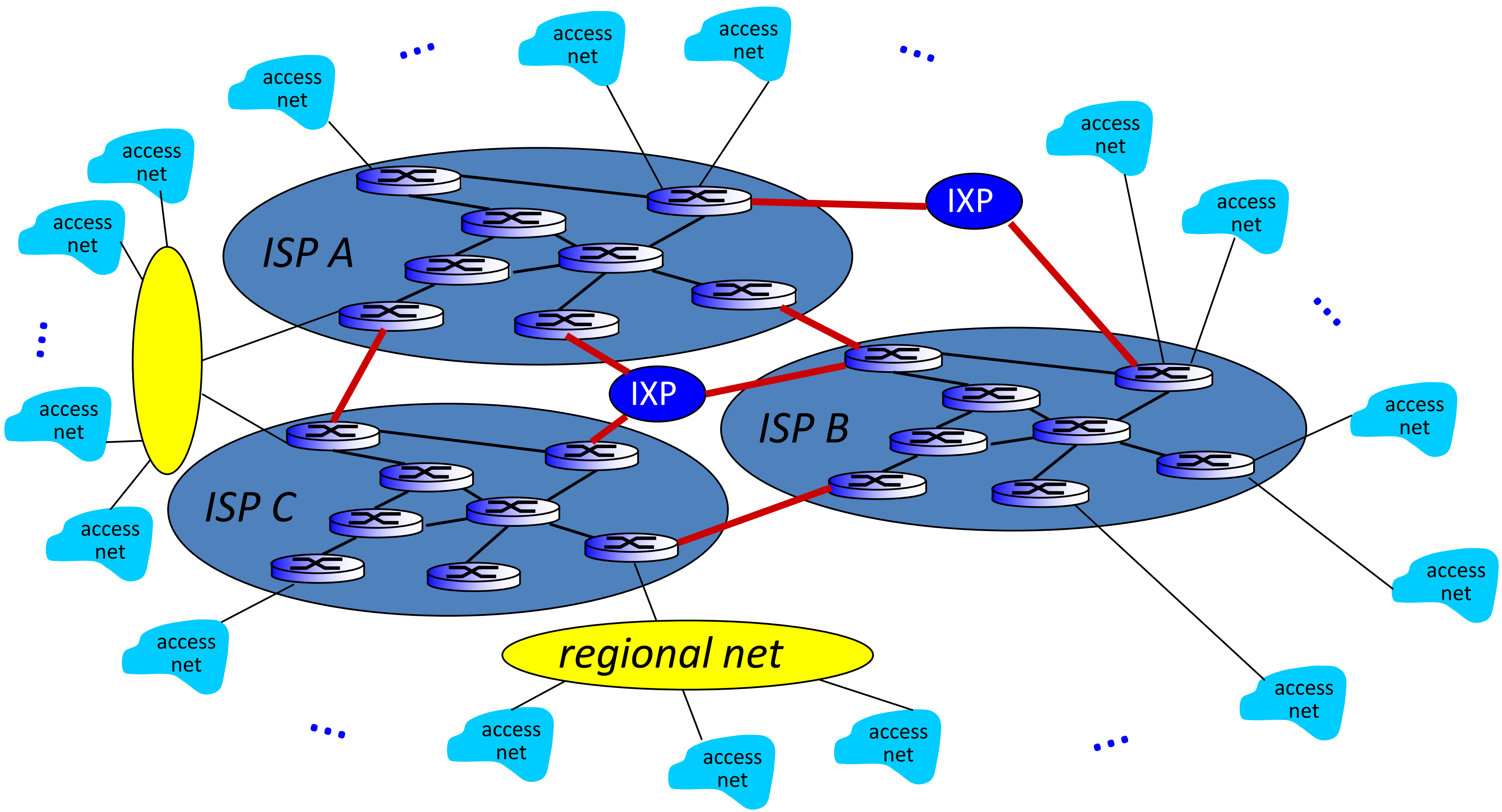
Internet structure: network of networks

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



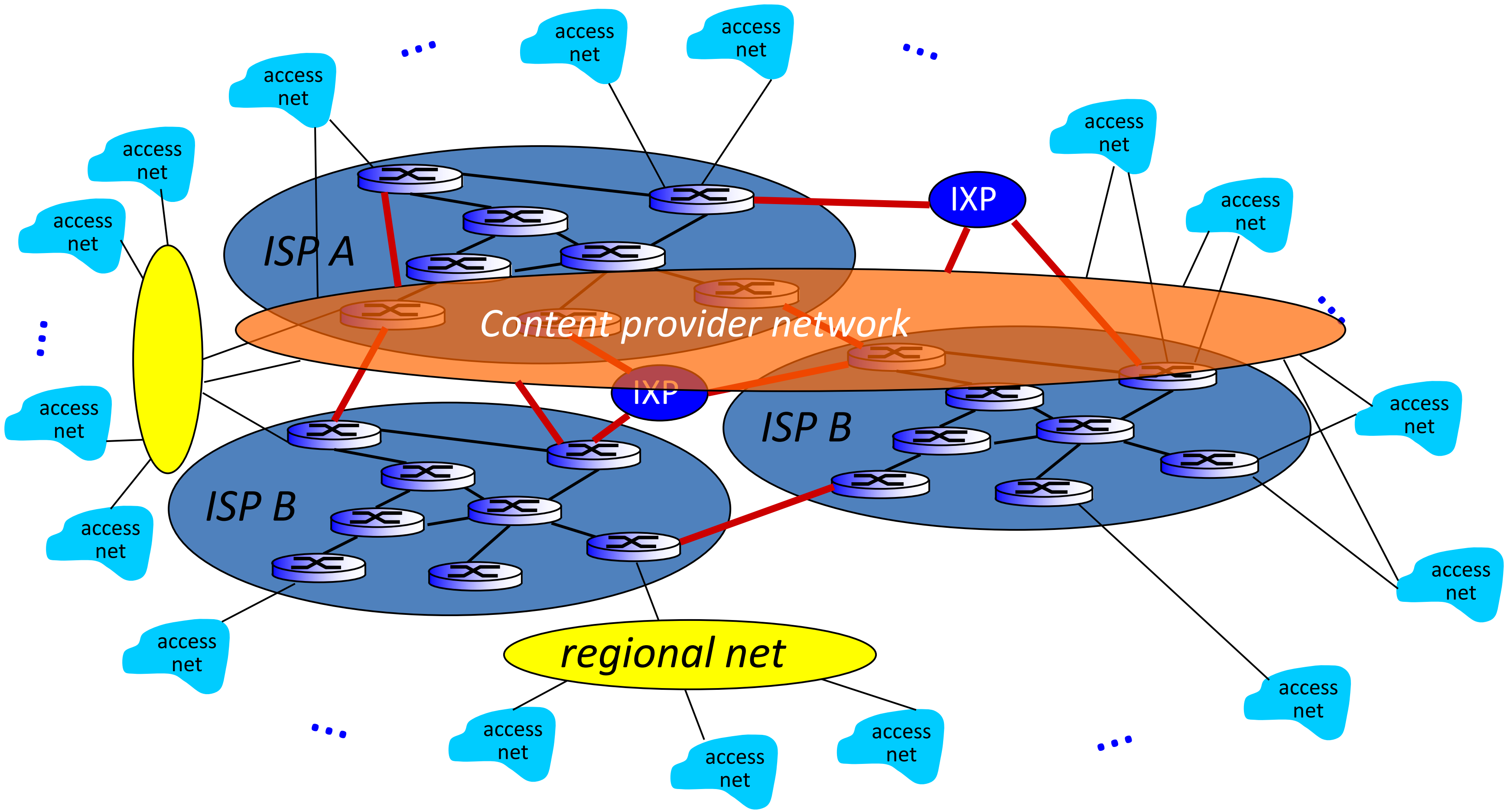
Internet structure: network of networks

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

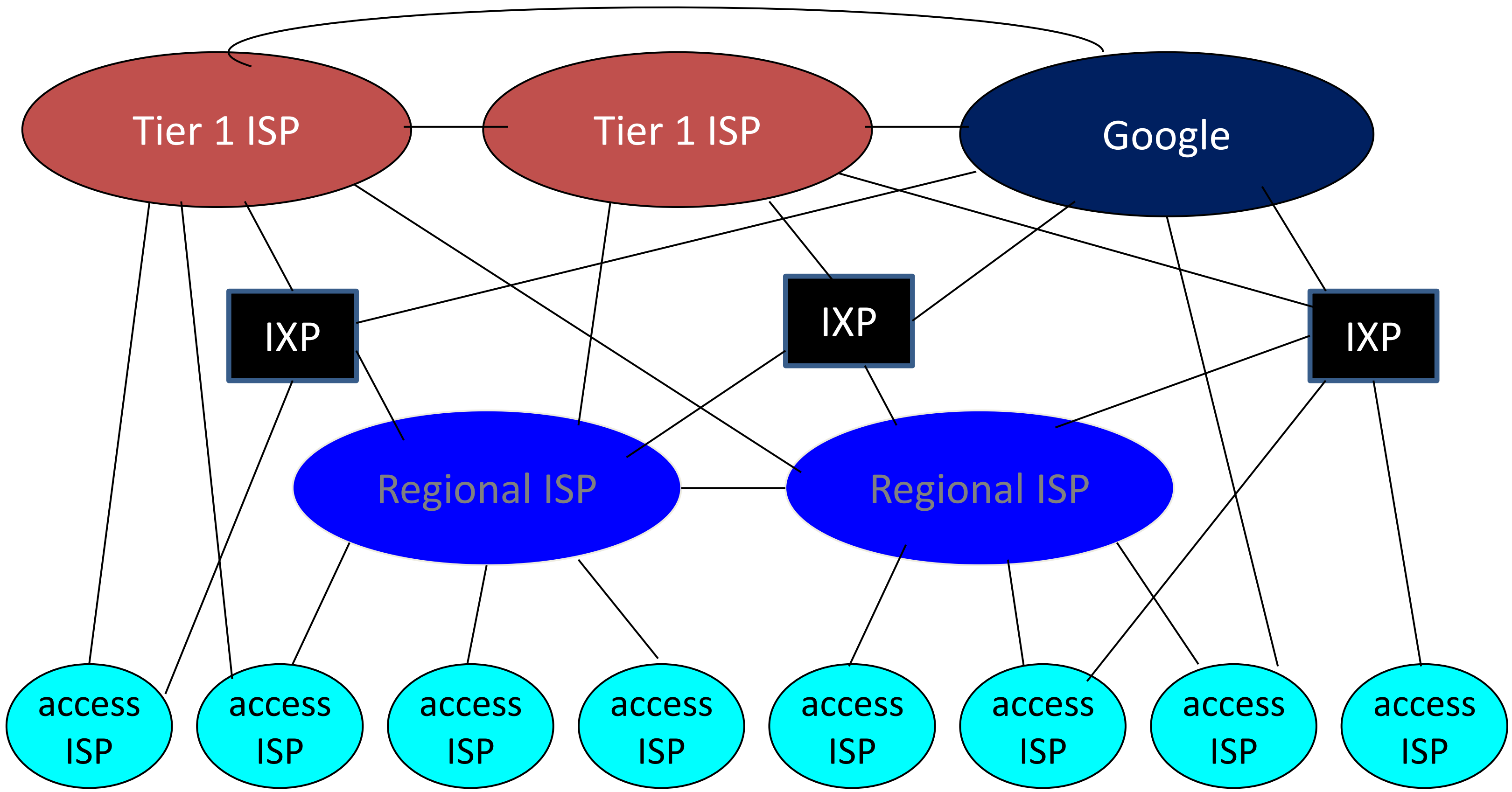


Internet structure: network of networks

… 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 it data centers to Internet, often bypassing tier-1, regional ISPs

Chapter 1: roadmap

1.1 what is the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

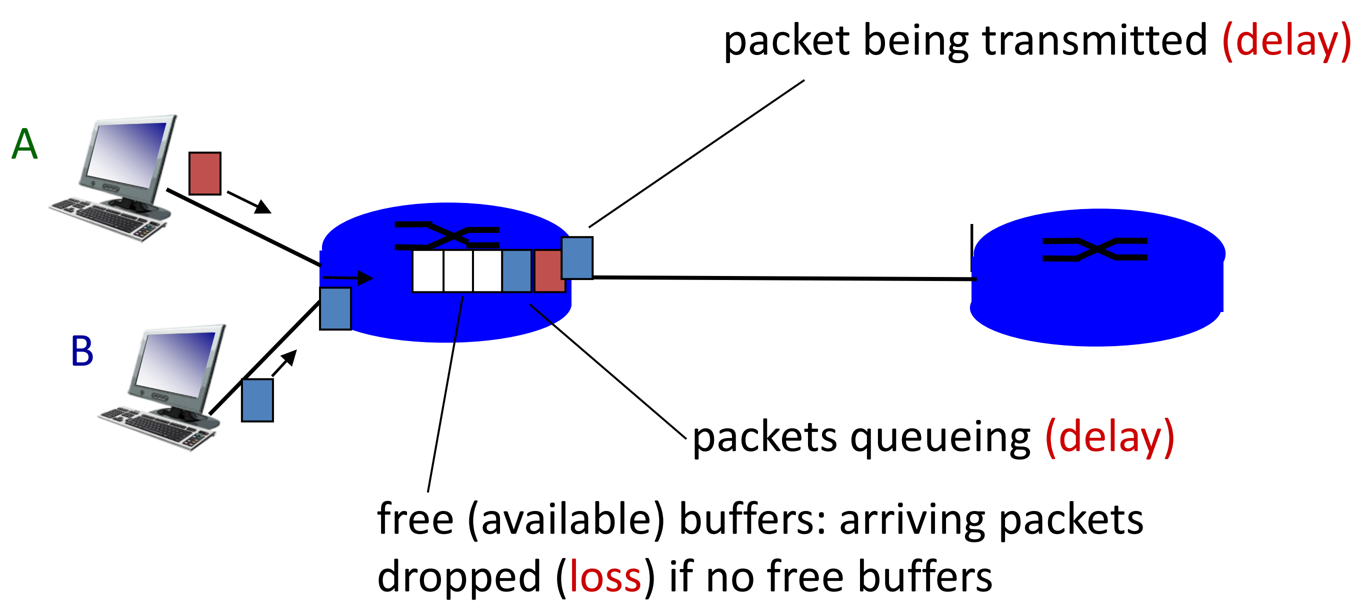
1.5 protocol layers, service models

1.6 networks under attack: security

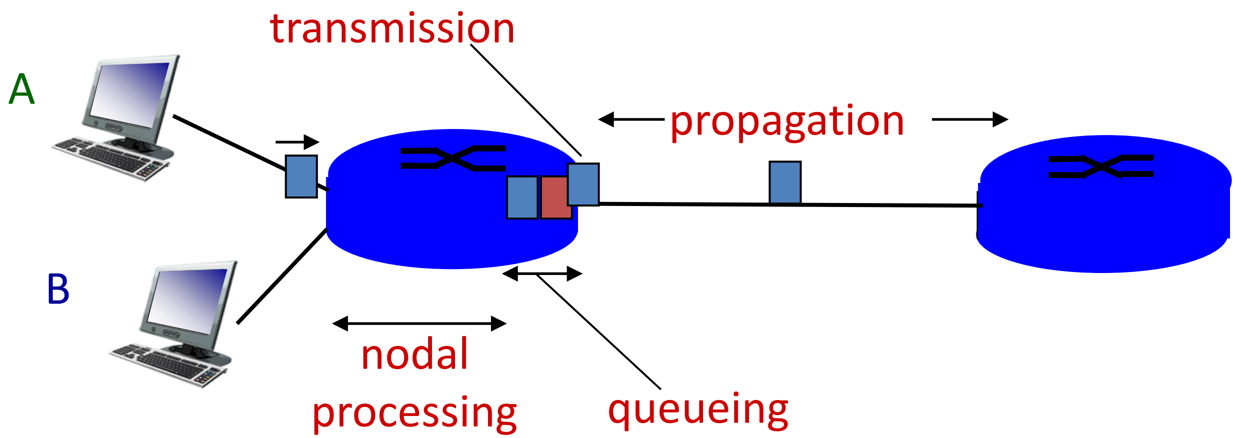
How do loss and delay occur?

packets *queue* in router buffers

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



Four sources of packet delay



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

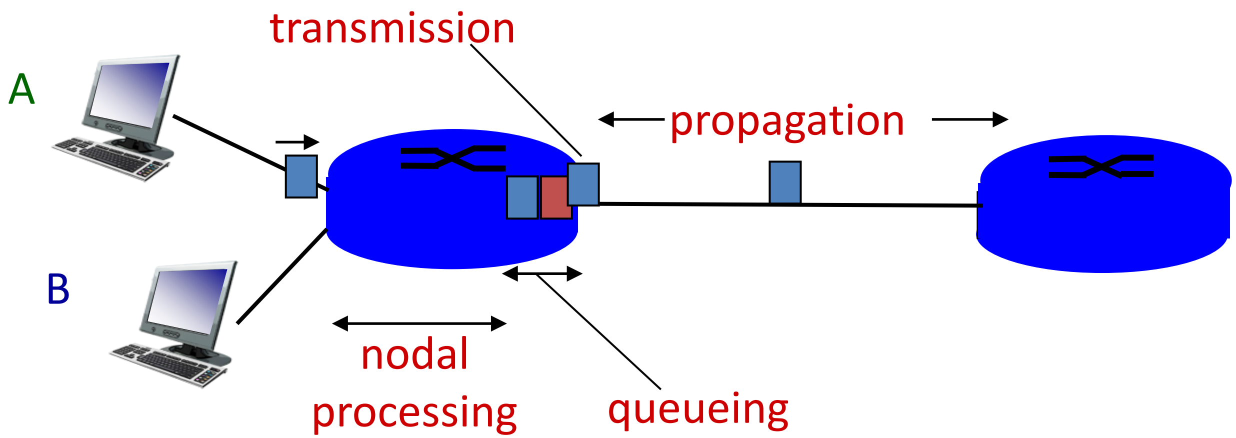
d_{proc}: nodal processing

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

d_{queue}: *queueing delay*

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

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)
- $d_{trans} = L/R$
- constant

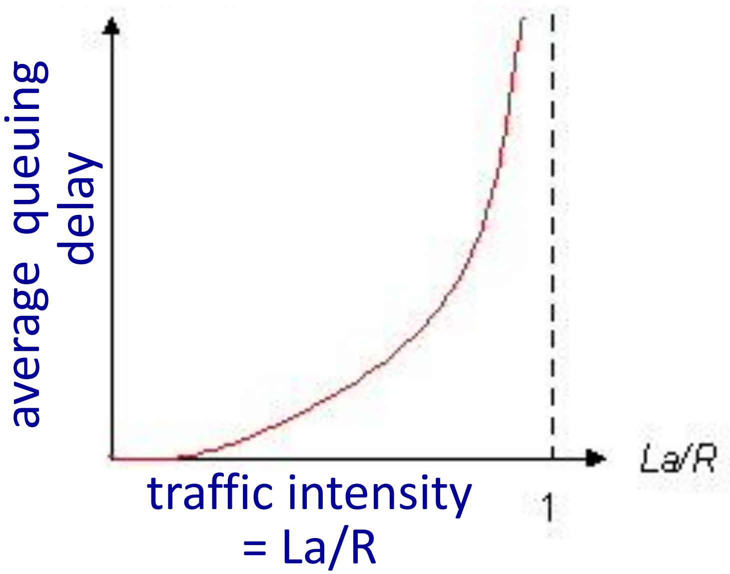
d_{trans} and d_{prop} *very* different

d_{prop}: propagation delay:

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

Queuing delay (revisited)

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



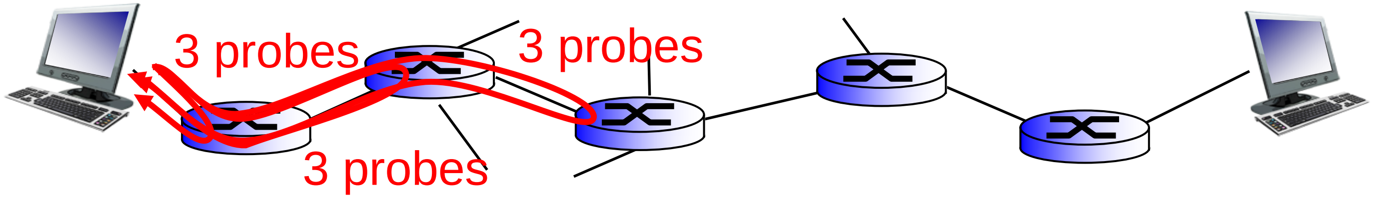
- *La/R* ~ 0: avg. queuing delay small
- *La/R* -> 1: avg. queuing delay large
- *La/R* > 1: more "work" arriving than can be serviced, average delay infinite!

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 routers along end-to-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: olemiss campus wifi to www.eurecom.fr

3 delay measurements from host to Campus WiFi AP

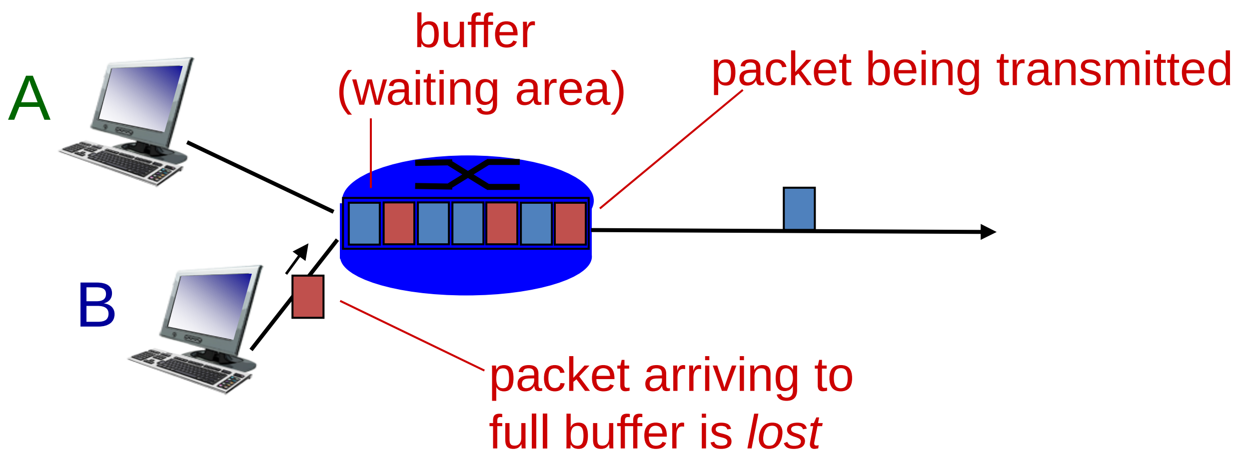
```
1    51 ms    13 ms     4 ms  rtr954.nat.olemiss.edu [172.24.248.1]
2     3 ms     4 ms     3 ms  um-dat--um-wifi.backbone.olemiss.edu [130.74.8.61]
3     9 ms     3 ms     3 ms  rtr922.backbone.olemiss.edu [130.74.4.65]
4    12 ms    12 ms    12 ms  205.233.255.100
5    19 ms    20 ms   117 ms  205.233.255.33
6    34 ms    34 ms    34 ms  et-10-0-0.105.rtr.atla.net.internet2.edu [198.71.45.12]
7    48 ms    47 ms    47 ms  et-9-0-0.104.rtr.wash.net.internet2.edu [198.71.45.7]
8   140 ms   121 ms   122 ms  internet2-gw.mx1.lon.uk.geant.net [62.40.124.44]
9   127 ms   127 ms   127 ms  ae0.mx1.par.fr.geant.net [62.40.98.77]
10  132 ms   129 ms   157 ms  renater-lb1-gw.mx1.par.fr.geant.net [62.40.124.70]
11  149 ms   149 ms   149 ms  te0-6-0-4-lyon1-rtr-001.noc.renater.fr [193.51.177.219]
12  143 ms   142 ms   142 ms  te2-7-marseille1-rtr-021.noc.renater.fr [193.51.177.222]
13  143 ms   144 ms   145 ms  te1-2-sophia-rtr-021.noc.renater.fr [193.51.177.21]
14  157 ms   157 ms   156 ms  eurocom-valbonne-gi9-7-sophia-rtr-021.noc.renater.fr
    [193.51.187.17]
15    *        *        *     Request timed out.
16    *        *        *     Request timed out.
17    *        *        *     Request timed out.
18    *        *        *     Request timed out.
19    *        *        *     Request timed out.
```

trans-oceanic link

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

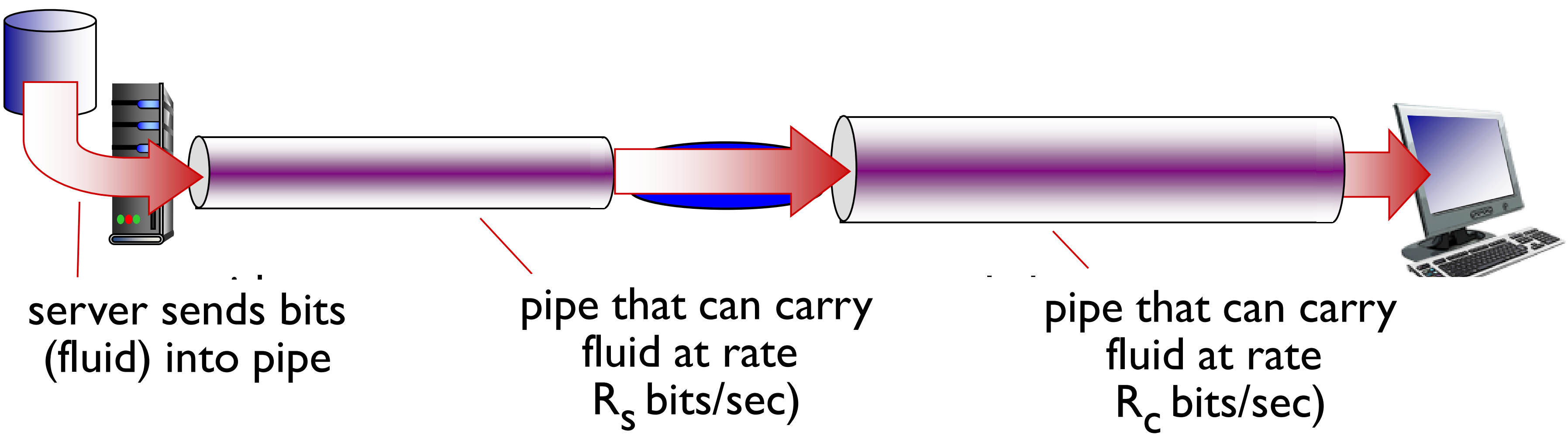
Packet loss

- 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



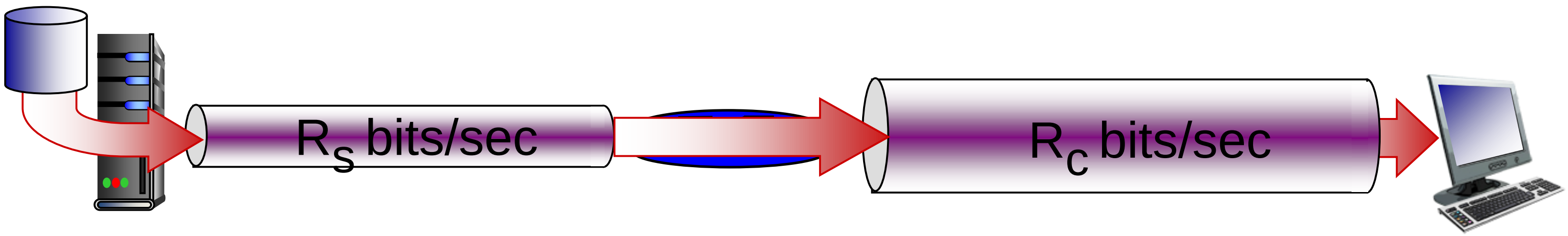
Throughput

- *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

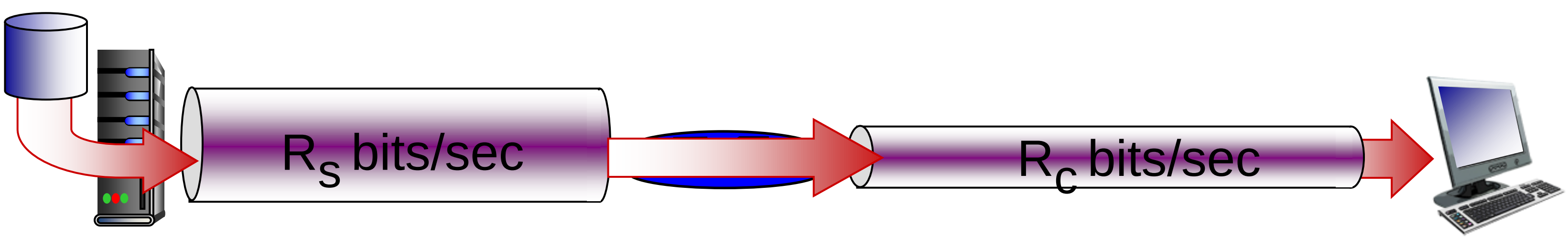


Throughput (more)

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



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

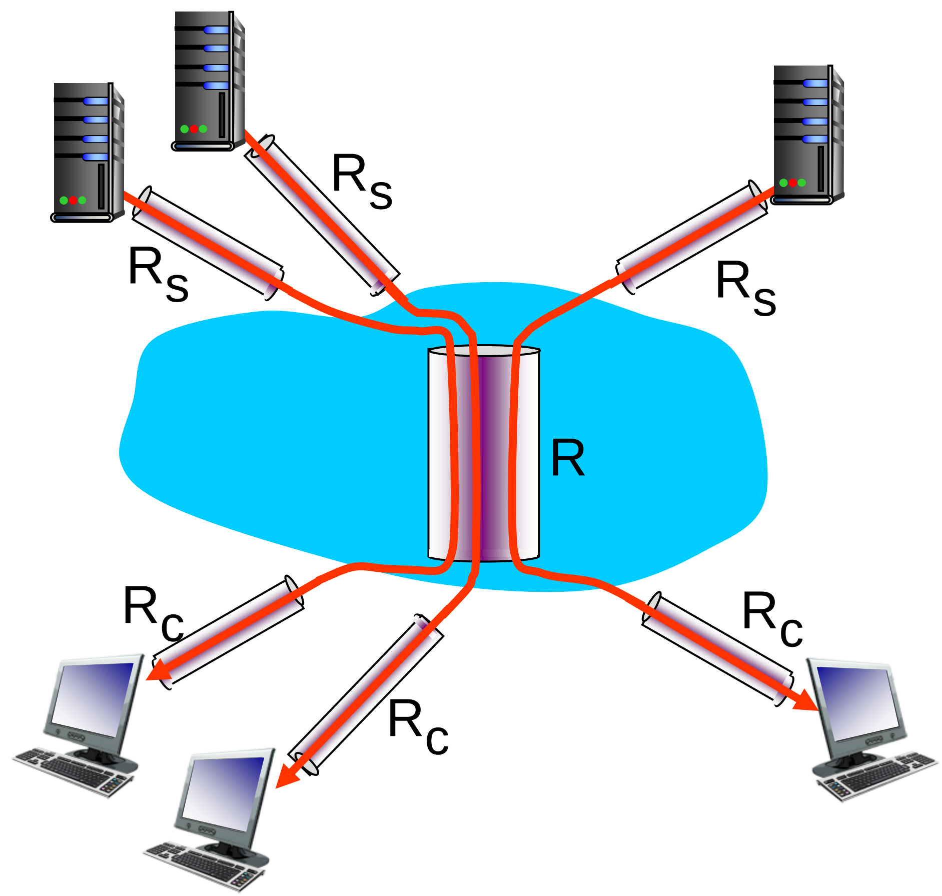


bottleneck link

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 link R bits/sec

Chapter 1: roadmap

1.1 what is the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

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?

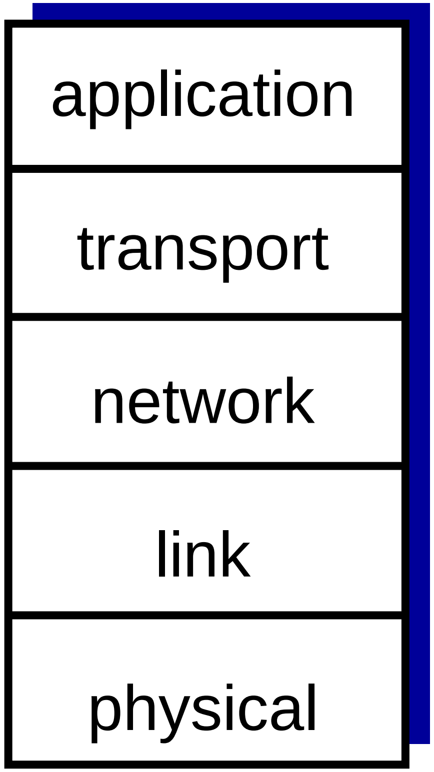
Why layering?

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?

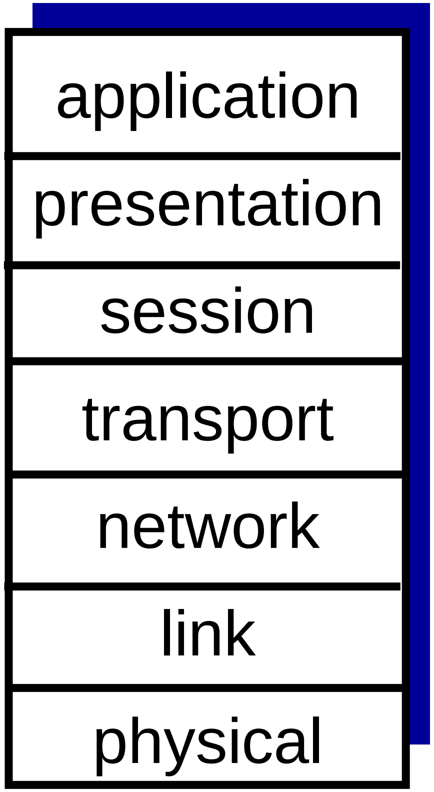
Internet protocol stack

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

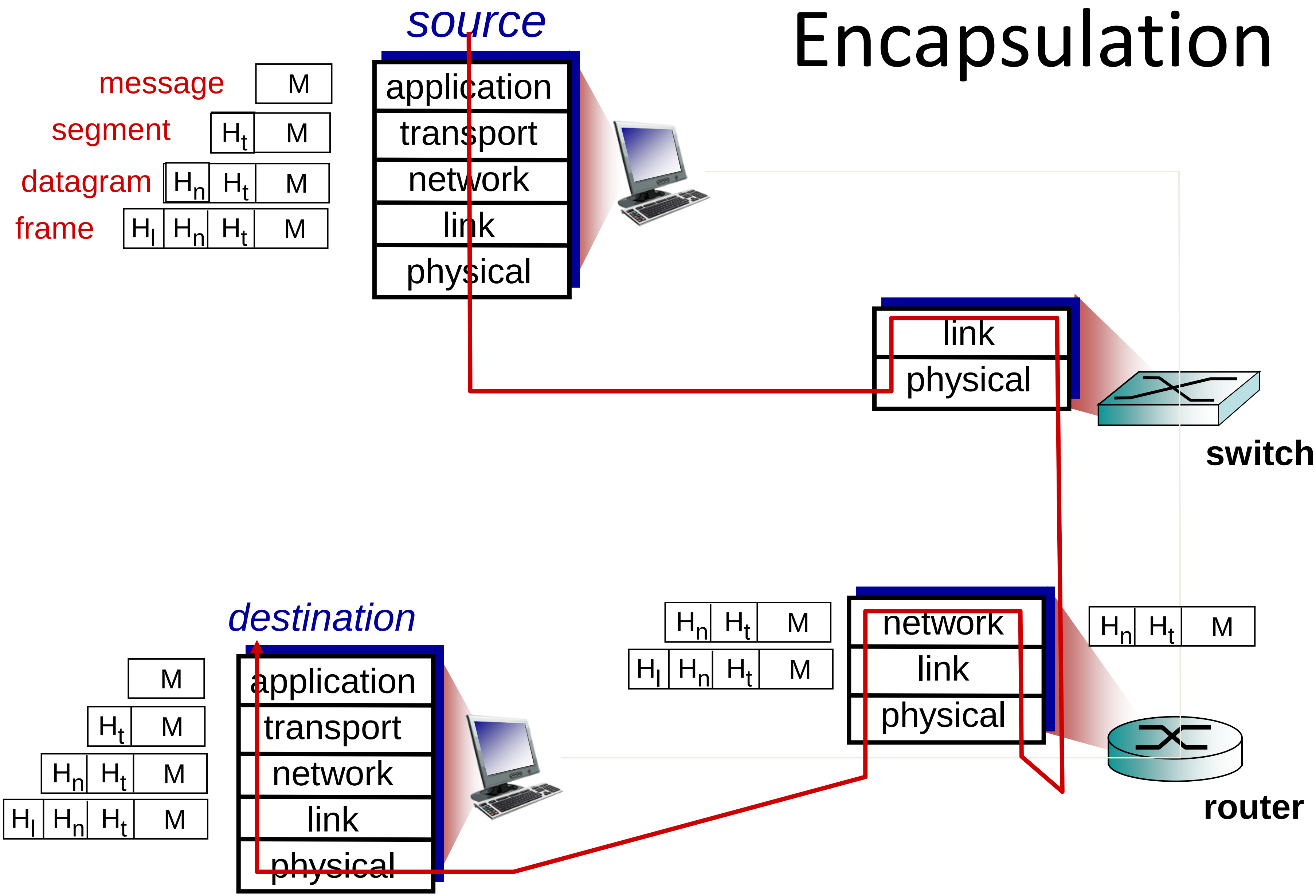


ISO/OSI reference model

- *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



Chapter 1: roadmap

1.1 what is the Internet?

1.2 network edge

- end systems, access networks, links

1.3 network core

- packet switching, circuit switching, network structure

1.4 delay, loss, throughput in networks

1.5 protocol layers, service models

1.6 networks under attack: security

Network security

- 3 major topic field of network security about:
  - 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!

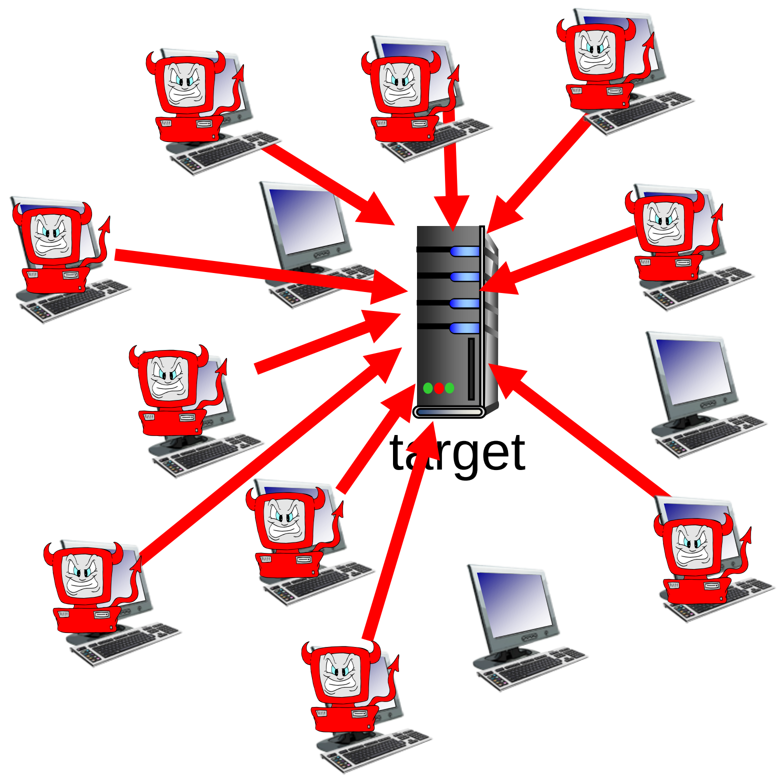
Bad guys: put malware into hosts via Internet

- 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

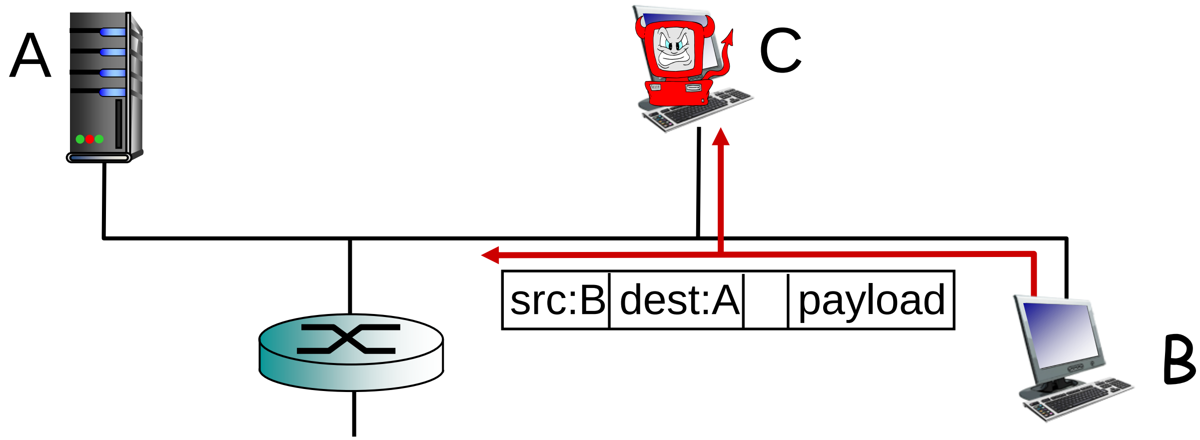
1. select target
2. break into hosts around the network (see botnet)
3. send packets to target from compromised hosts



Bad guys can sniff packets

packet "sniffing":

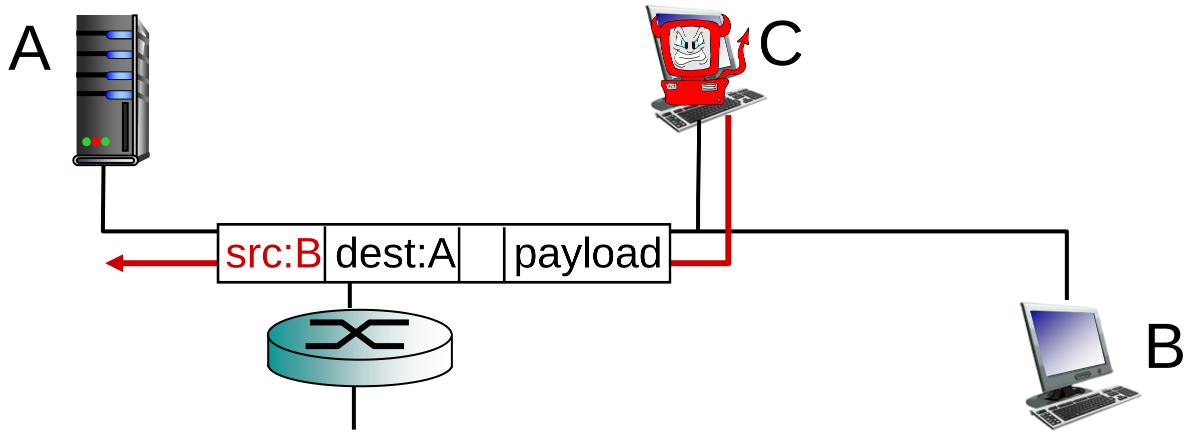
- broadcast media (shared Ethernet, wireless)
- promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



- Wireshark software is a (free) packet-sniffer

Bad guys can use fake addresses

IP spoofing: send packet with false source address



Introduction: summary

covered a "ton" of material!

- Internet overview
- what's a protocol?
- network edge, core, access network
  - packet-switching versus circuit-switching
  - Internet structure
- performance: loss, delay, throughput
- layering, service models
- security