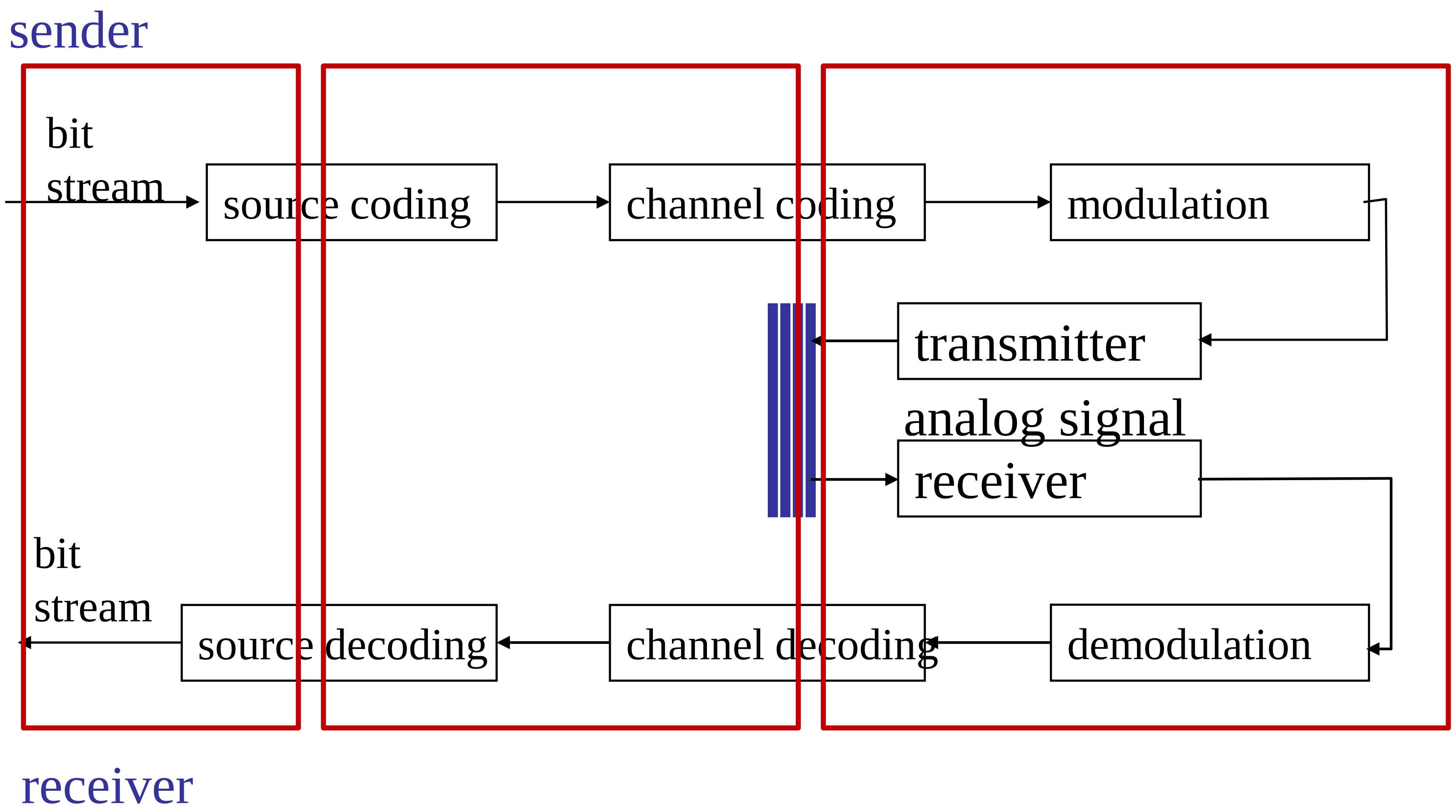
ITH508 컴퓨터망

Introduction & Overview

Hwangnam Kim
hnkim@korea.ac.kr
School of Electrical Engineering
Korea University

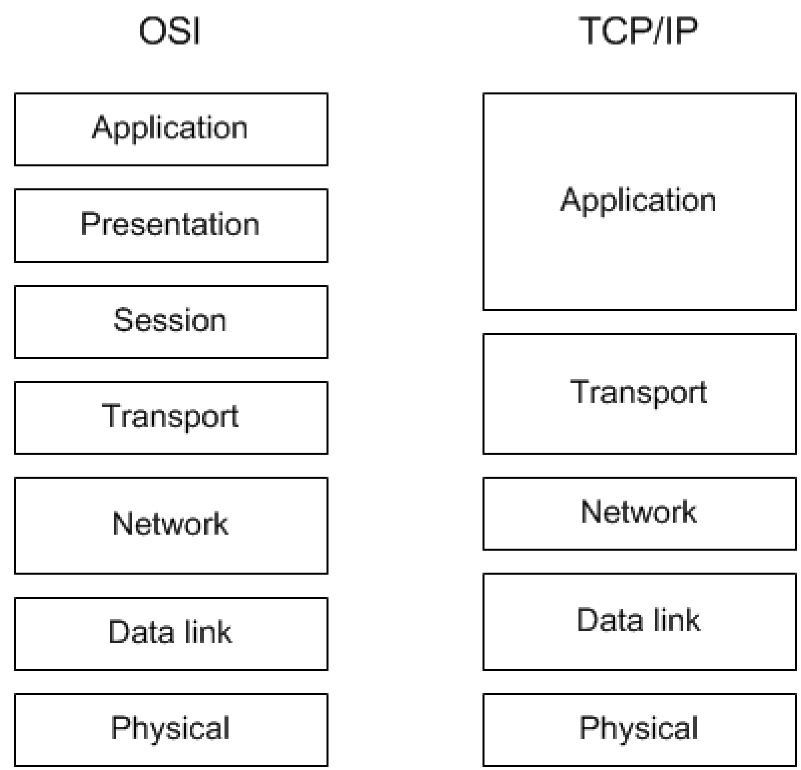
OVERVIEW ON COMPUTER NETWORKS

Network & Communications



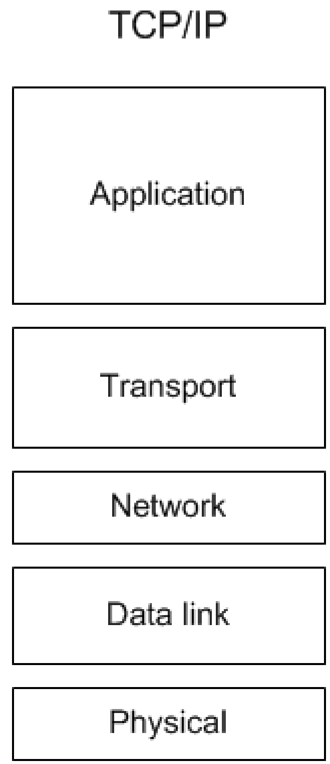
TCP/IP Architecture

- **TCP/IP**
  - Is the **de facto global data communications standard**.
  - Has a 3-layer protocol stack that can be mapped to five of the seven in the OSI model.
  - Can be used with any type of network, even different types of networks within a single session.



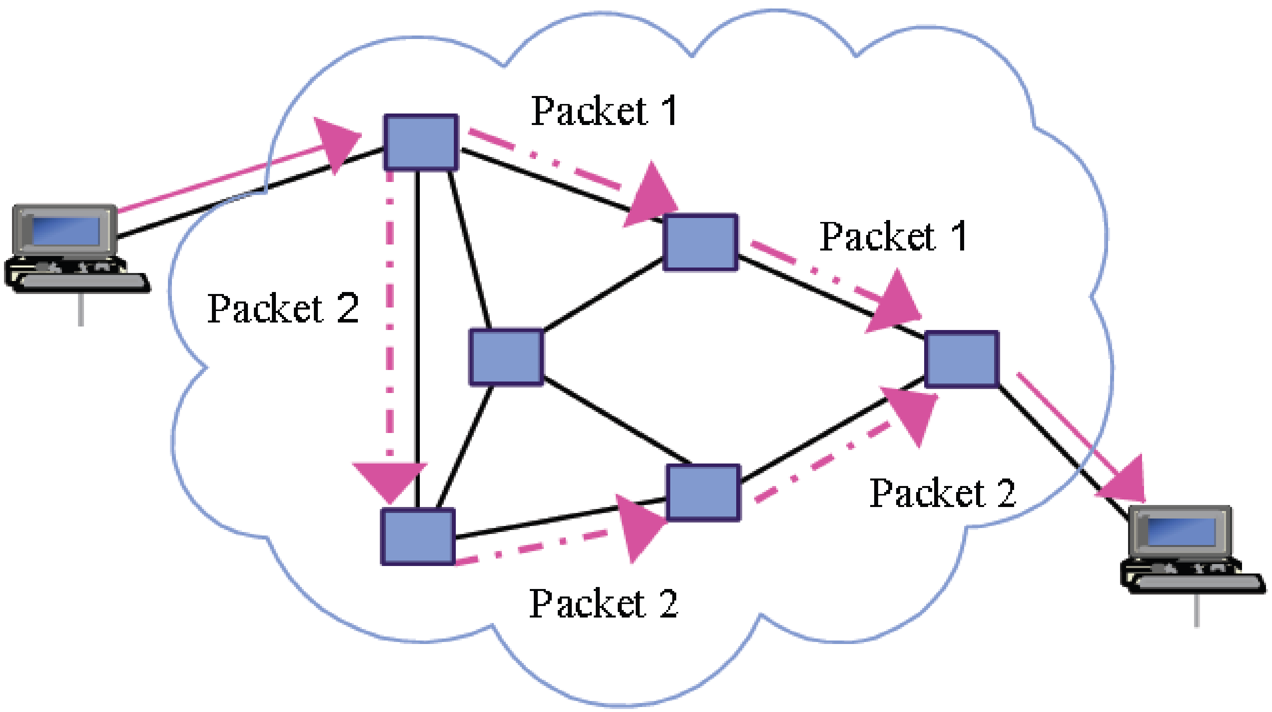
TCP/IP Architecture

- **The IP Layer**
  - Provides essentially the same services as the Network and Data Link layers of the OSI Reference Model.
  - Divides TCP packets into protocol data units called *datagrams*, and then attaches routing information.



TCP/IP Architecture

- The concept of the datagram was fundamental to the robustness of the Internet.
- Datagrams can take any route available to them without human intervention.

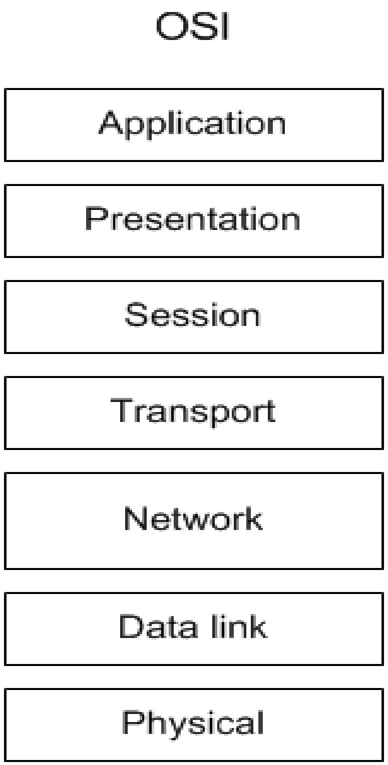


ISO/OSI Reference Model

- **To address incompatible proprietary network protocols, in 1984 the ISO formed a committee to devise a unified protocol standard.**
- **The result is the ISO *Open Systems Interconnect Reference Model* (ISO/OSI RM).**
- **The ISO's work is called a reference model**
  - Virtually no commercial system uses all of the features precisely as specified in the model.
- **The ISO/OSI model make understood the concept of a unified communications architecture.**

ISO/OSI Reference Model

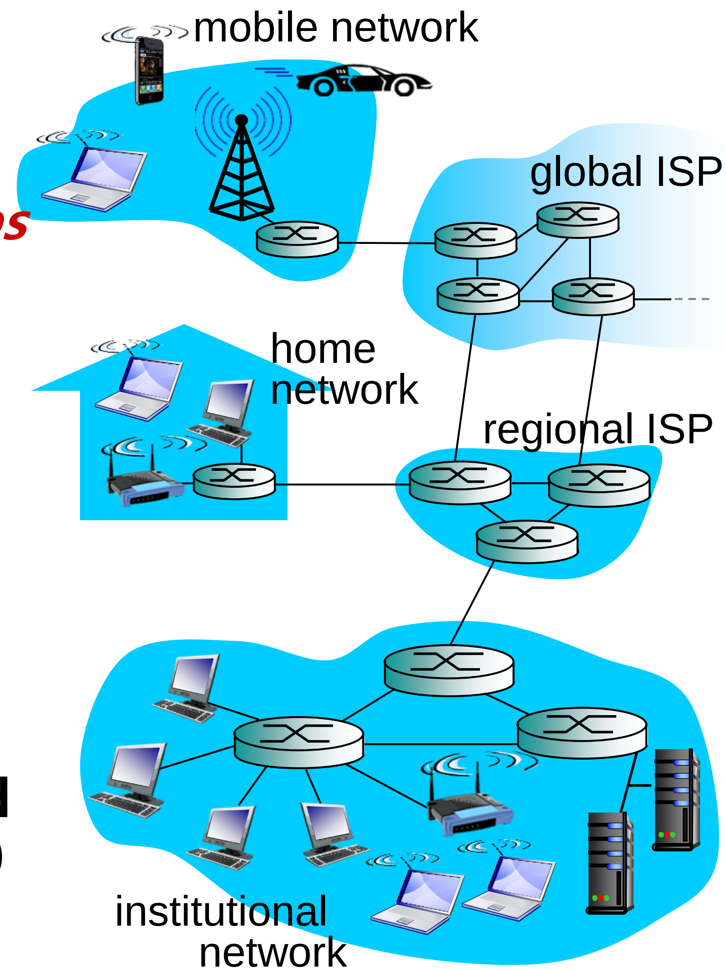
- The OSI RM contains seven protocol layers, starting with physical media interconnections at Layer 1, through applications at Layer 7.
- OSI model defines only the functions of each of the seven layers and the interfaces between them.
- Implementation details are not part of the model.

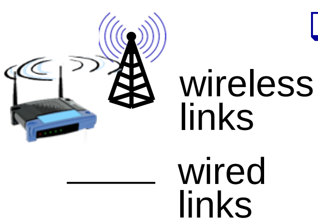


WHAT IS INTERNET?

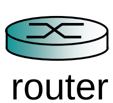
What’s the Internet?

- **Millions of connected computing devices:**
  - ***hosts = end systems***
  - **Running *network apps***





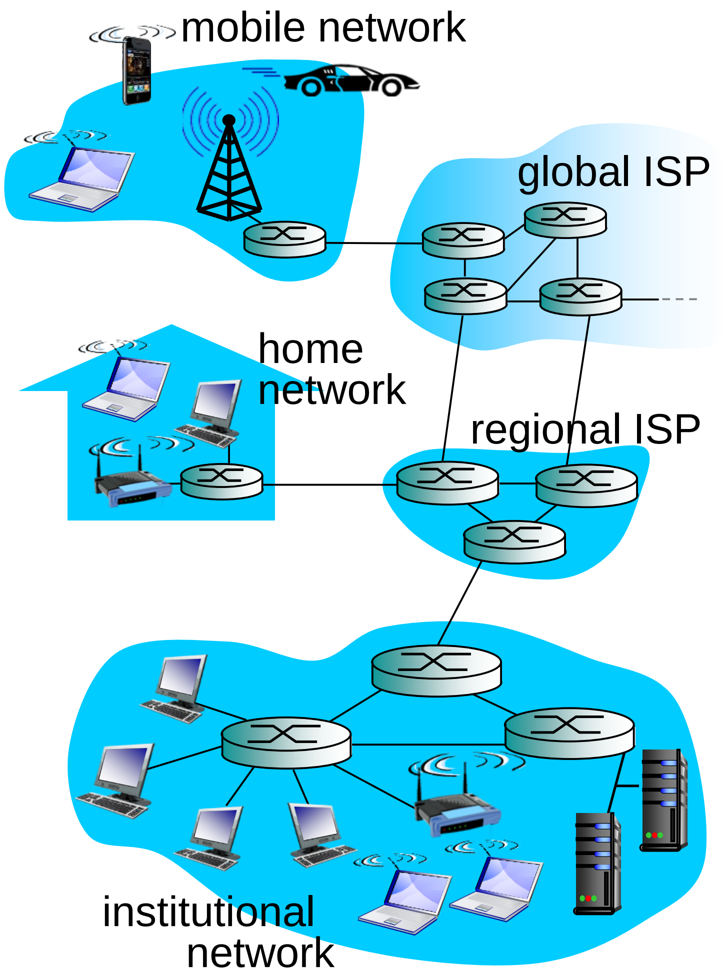
- ***Communication links***
  - **Fiber, copper, radio, satellite**
  - **Transmission rate: *bandwidth***



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

What's the Internet?

- *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



Different Perspectives on Network

- **Network users:**
  - Services that their **applications** need
    - Guarantee that each message sent will be delivered without error within a certain amount of time
- **Network designers:**
  - Cost-effective **design**
    - Network resources are efficiently utilized and fairly allocated to different users
- **Network providers:**
  - System that is **easy to administer and manage**
    - Faults can be easily isolated and it is easy to account for usage

Network Role

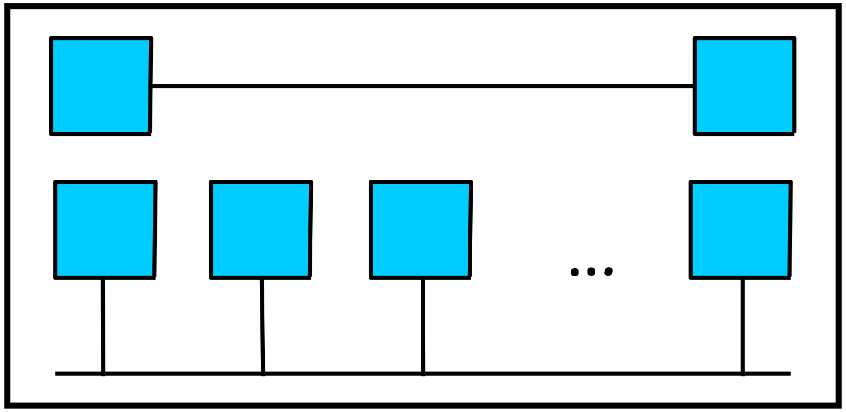
- **What must a network provide?**
  1. **Connectivity**
  2. Cost-effective **Resource Sharing**
  3. **Performance** (in terms of delay and bandwidth)
  4. **Functionality**
- **How are networks designed and built?**
  - Layering
  - Protocols
  - Standards

1. Network Connectivity

NETWORK STRUCTURE: ACCESS AND CORE NETWORK

Connectivity

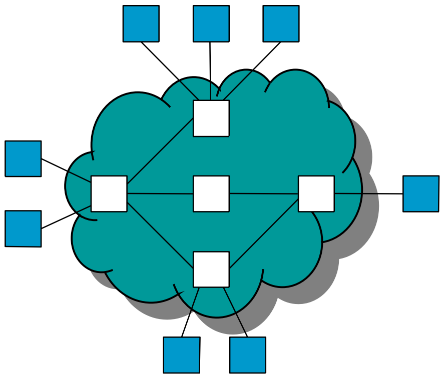
- **Building Physical Block**
  - Links: coax cable, optical fiber, ...
  - Nodes: workstations, routers, ...
- **Links:**
  - Point-to-point
  - Multiple access

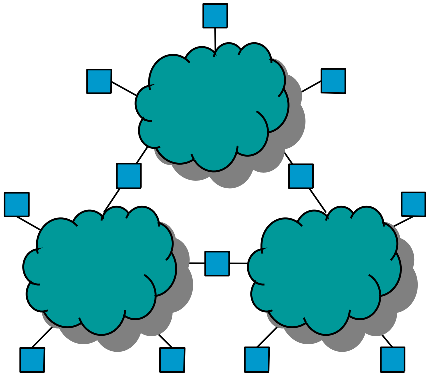


Indirect Connectivity

- **Switched Networks**
- **Internetworks**

- **Recursive definition of a network**
  - Two or more nodes connected by a physical link
  - Two or more networks connected by one or more nodes





Indirect Connectivity

- **Nodes receive data on one link and forward it onto the next**

  → switching network

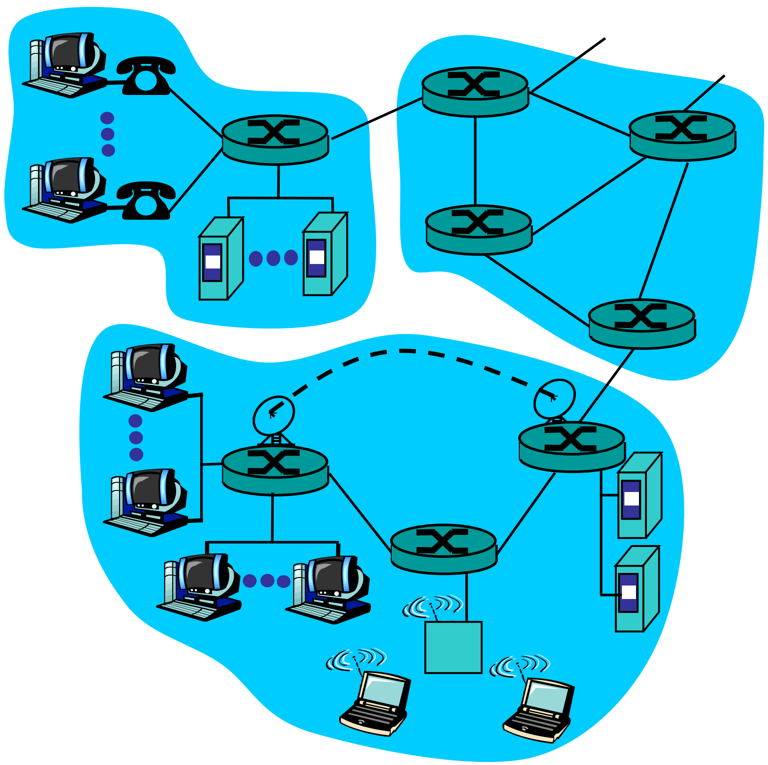
  - **Circuit Switching**
    - Telephone
    - Stream-based (dedicated circuit)
    - Links reserved for use by communication channel
    - Send/receive bit stream at constant rate
  - **Packet Switching**
    - Internet
    - Message-based (store-and-forward)
    - Links used dynamically
    - Admission policies and other traffic determine bandwidth

Addressing for Connectivity

- **Addressing**
  - **Unique byte-string** used to **indicate which node** is the target of communication
- **Routing**
  - The process of d**etermining how to forward messages toward the destination node** based on its address
- **Types of Addresses**
  - Unicast: node-specific
  - Broadcast: all nodes on the network
  - Multicast: subset of nodes on the network

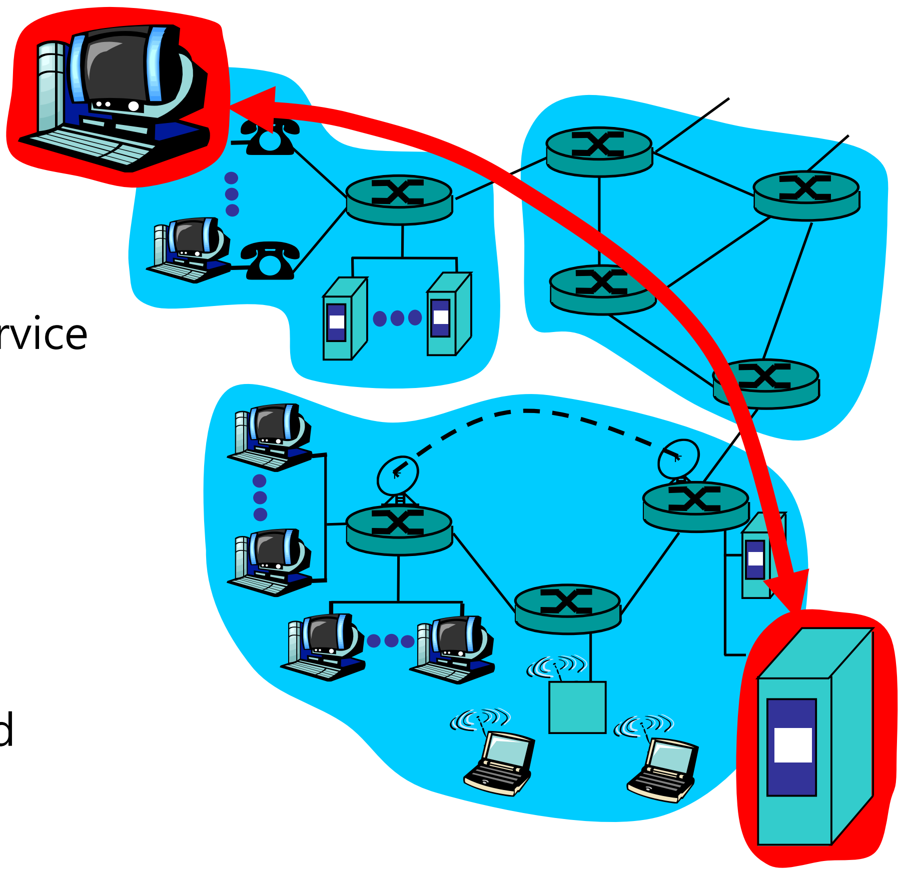
Network Structure

- **Network edge:**
  - Hosts and applications
  - Access networks, physical media: Communication links
- **Network core:**
  - Routers
  - Network of networks



Network Edge

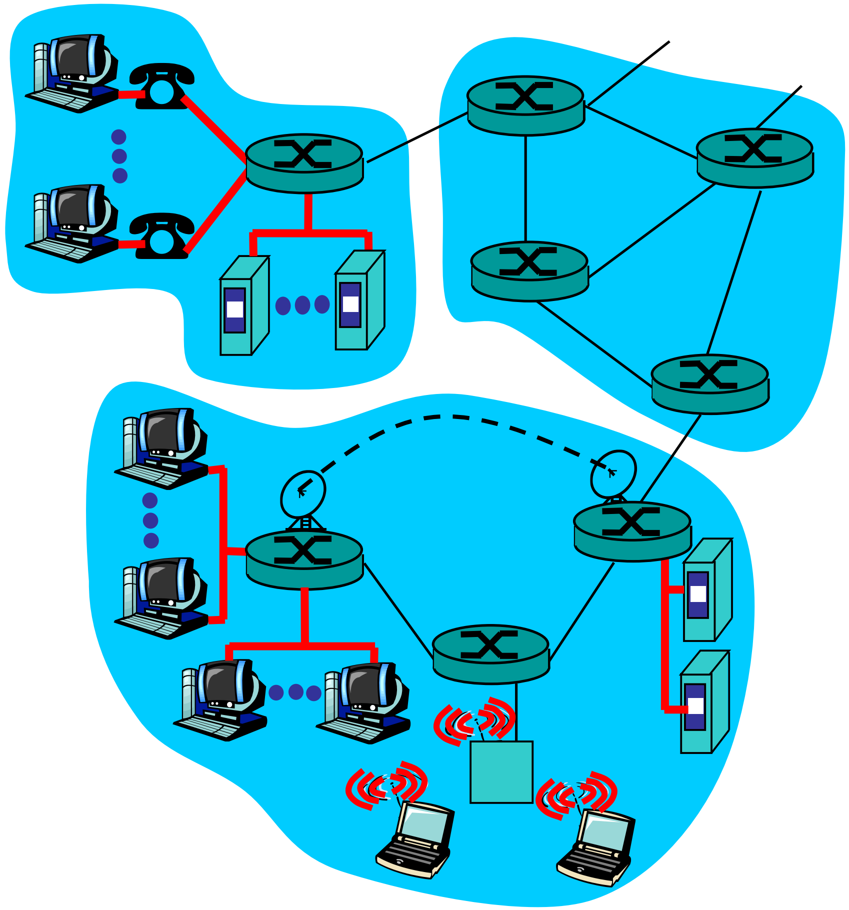
- **End systems (hosts):**
  - Run application programs
  - e.g. Web, email
  - At "edge of network"
- **Client/server model**
  - Client host requests, receives service from always-on server
  - e.g. Web browser/server; email client/server
- **Peer-peer model:**
  - Symmetric client/server
  - Minimal (or no) use of dedicated servers
  - e.g. Skype, BitTorrent, KaZaA



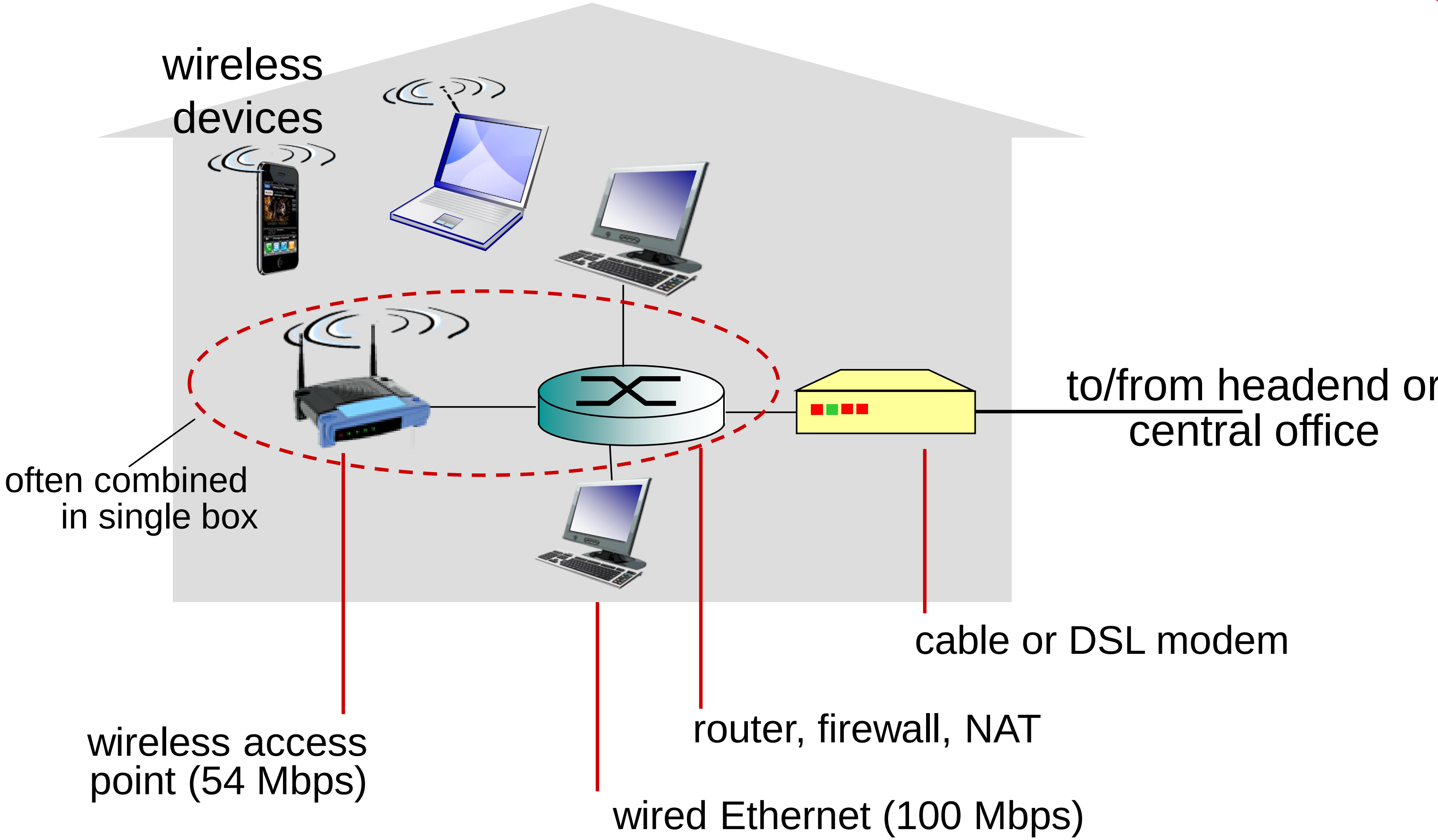
Network Edge: Access Networks

Q: How to connect end systems to edge router?

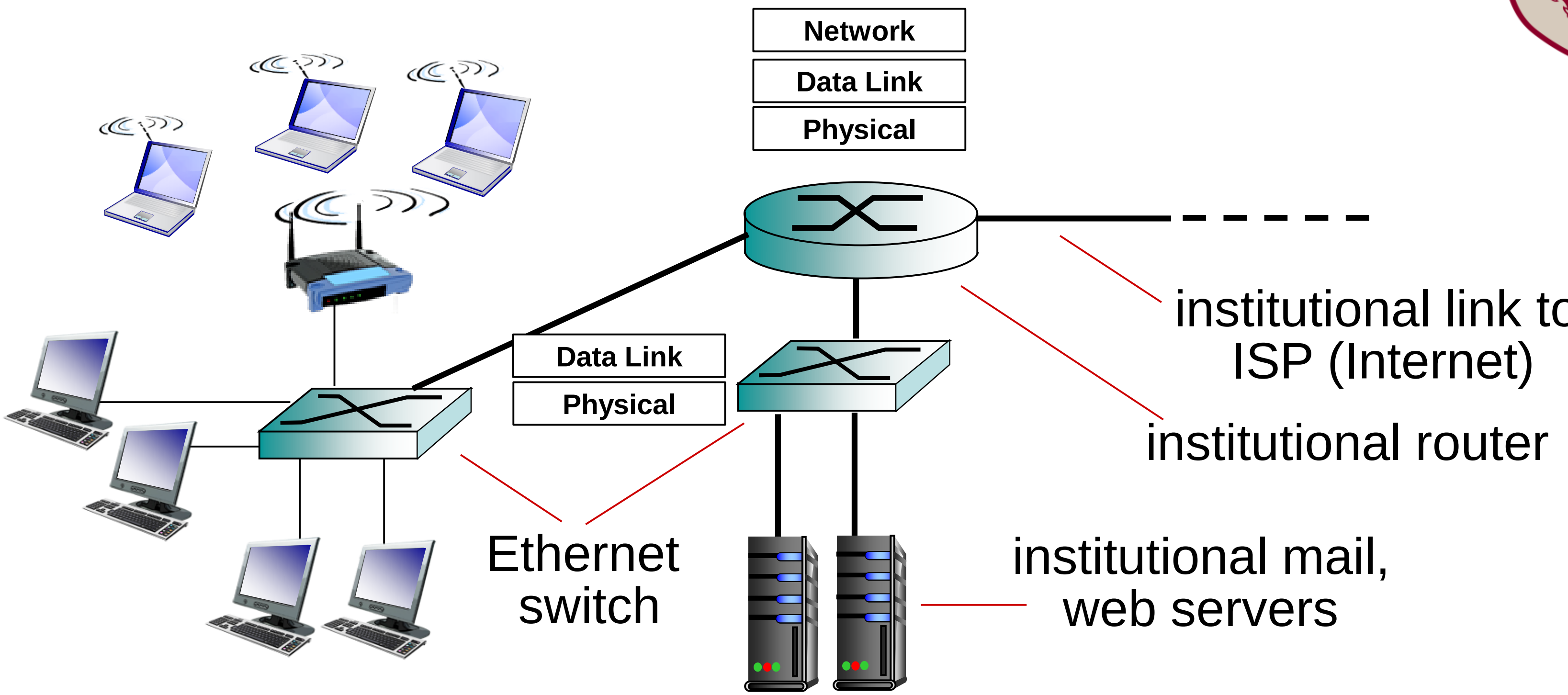
- **Residential** access nets
- **Institutional** access networks (school, company)
- Mobile access networks



Access Networks: Home Network



Access Networks: Enterprise (Ethernet)



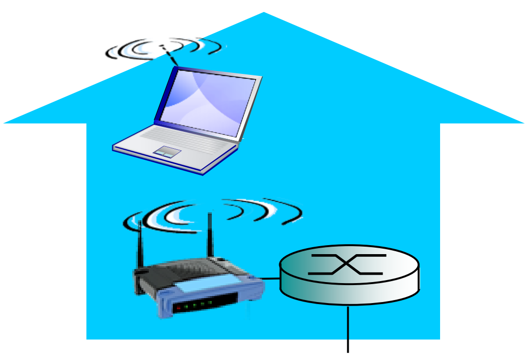
- Typically used in companies, universities, etc
- 10 Mbps, 100Mbps, 1Gbps, 10Gbps transmission rates
- Today, end systems typically connect into Ethernet switch

Access Networks: Wireless Access Network

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

wireless LANs:

- within building (100 ft)
- 802.11b/g/n/ac/ax (WiFi): 11 Mbps, 54 Mbps, 866Mbps, 10Gbps transmission rate



to Internet

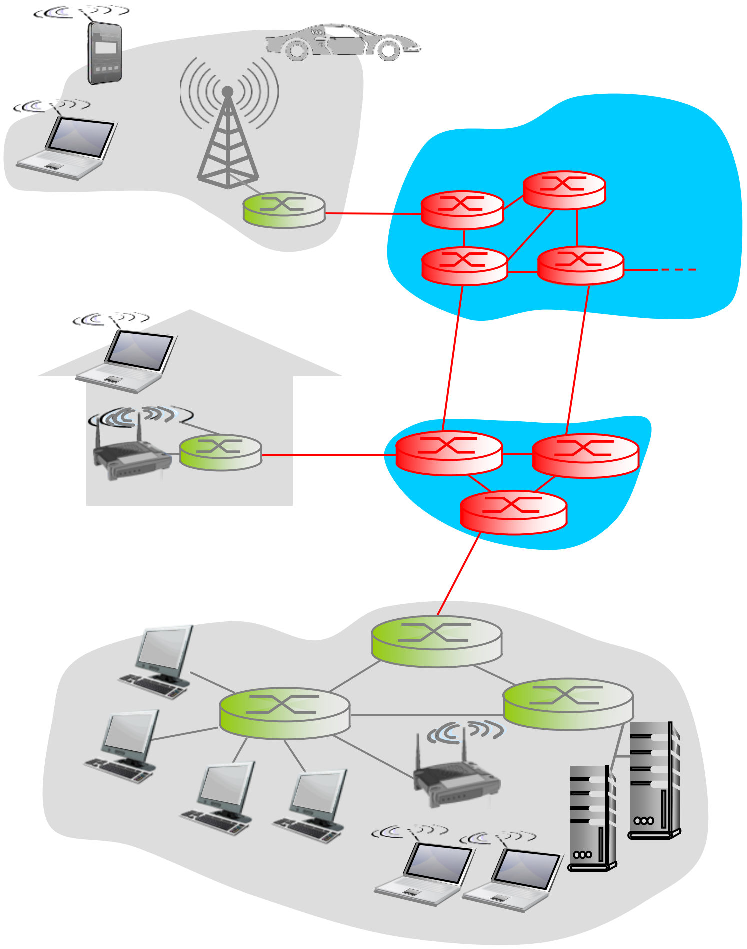
wide-area wireless access

- provided by telco (cellular) operator, 10's km
- between 1 and 10 Mbps
- 3G, 4G, LTE (downlink 30~40, uplink 15~20 Mbps), 5G (downlink 830, uplink 80 Mbps)

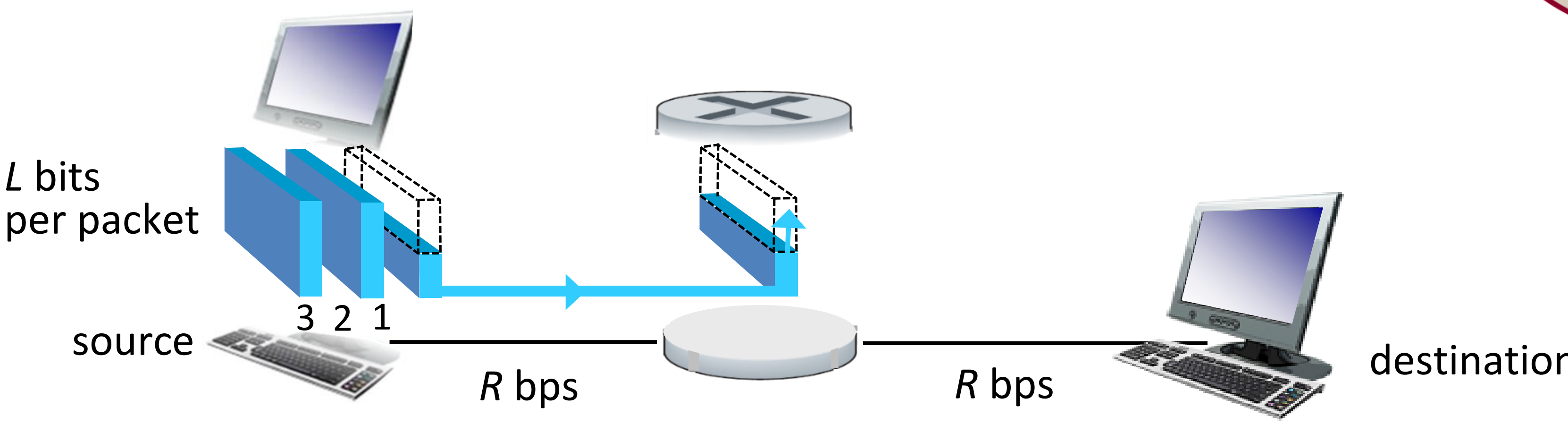
to Internet

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**

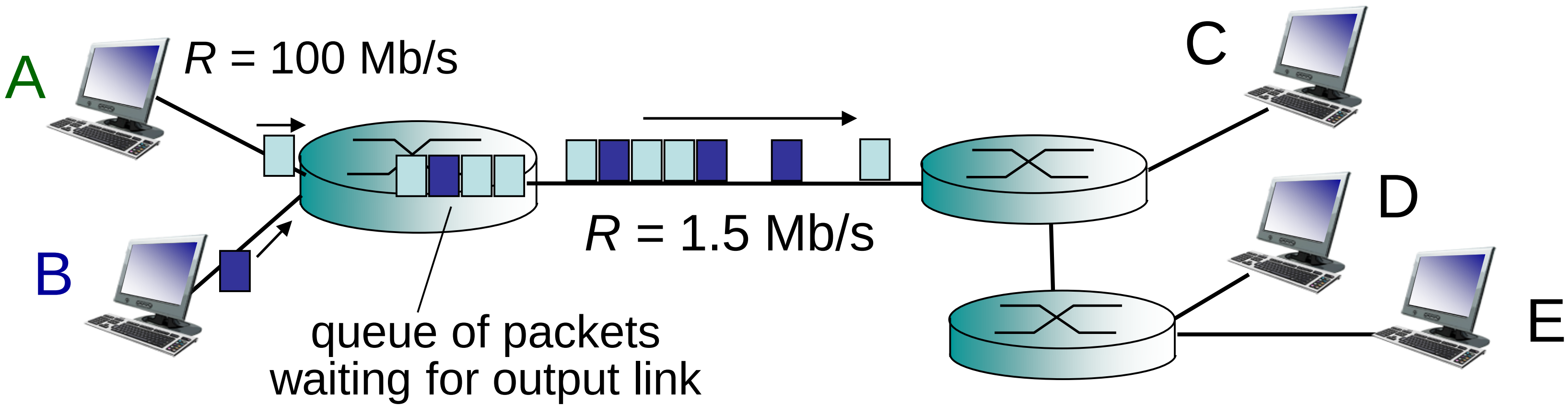
one-hop numerical example:

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

❖ end-end delay = 2*L*/*R* (assuming zero propagation delay)

more on delay shortly …

Packet Switching: Queueing Delay, Loss

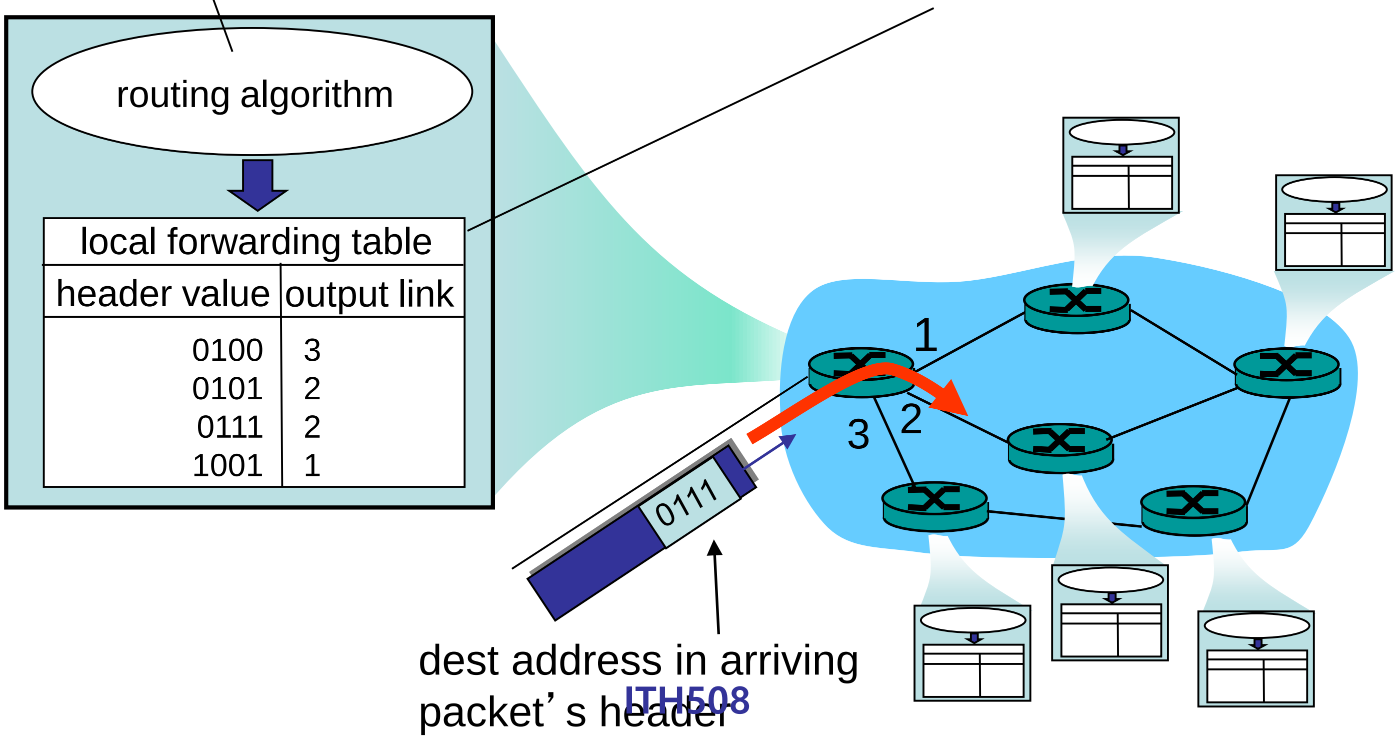


Queuing and Loss:

- **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

Packet Switching: Two Key Network-Core Functions

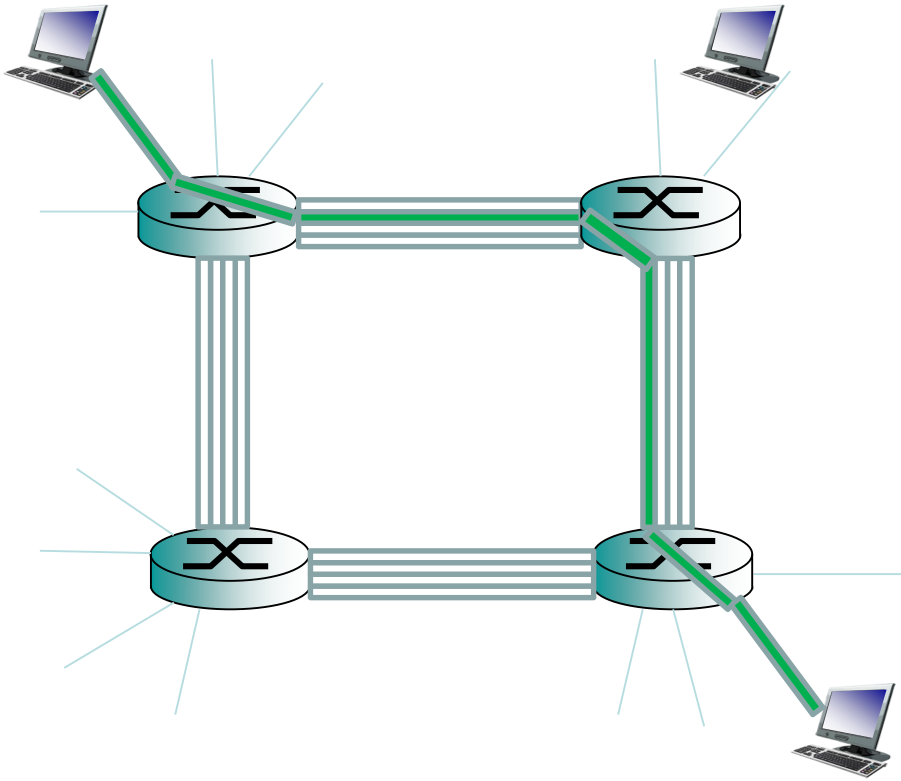
- **Routing: determines source-destination route taken by packets**
  - Routing algorithms
- ***Forwarding:* move packets from router's input to appropriate router output**



Alternative Core Net: Circuit Switching

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

- **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**

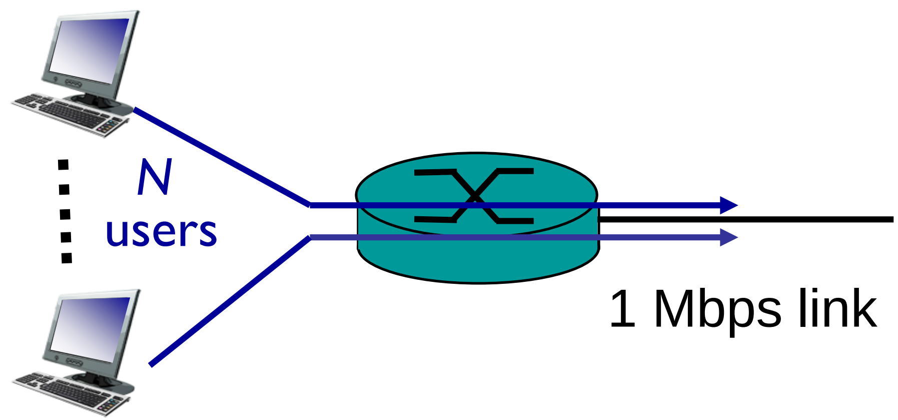


Packet Switching versus Circuit Switching

packet switching allows more users to use network!

Example:

- **1 Mb/s link**
- **Each user:**
  - 100 kb/s when "active"
  - active 10% of time
- ***Circuit-switching:***
  - 10 users
- ***Packet switching:***
  - with 35 users, probability > 10 active at same time is less than .0004.

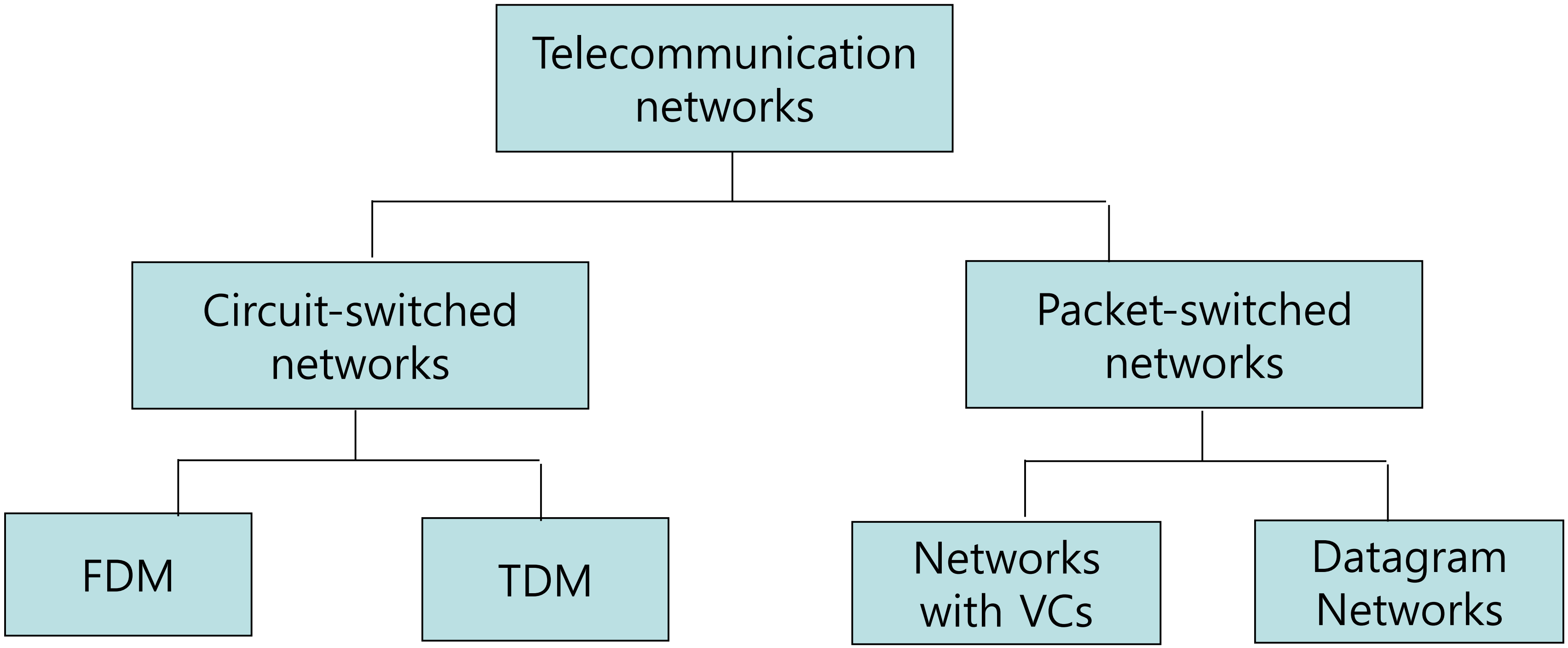


Packet Switching versus Circuit Switching

is packet switching a "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

Network Taxonomy



- Internet provides both connection-oriented (TCP) and connectionless services (UDP) to apps.

Real Connectivity

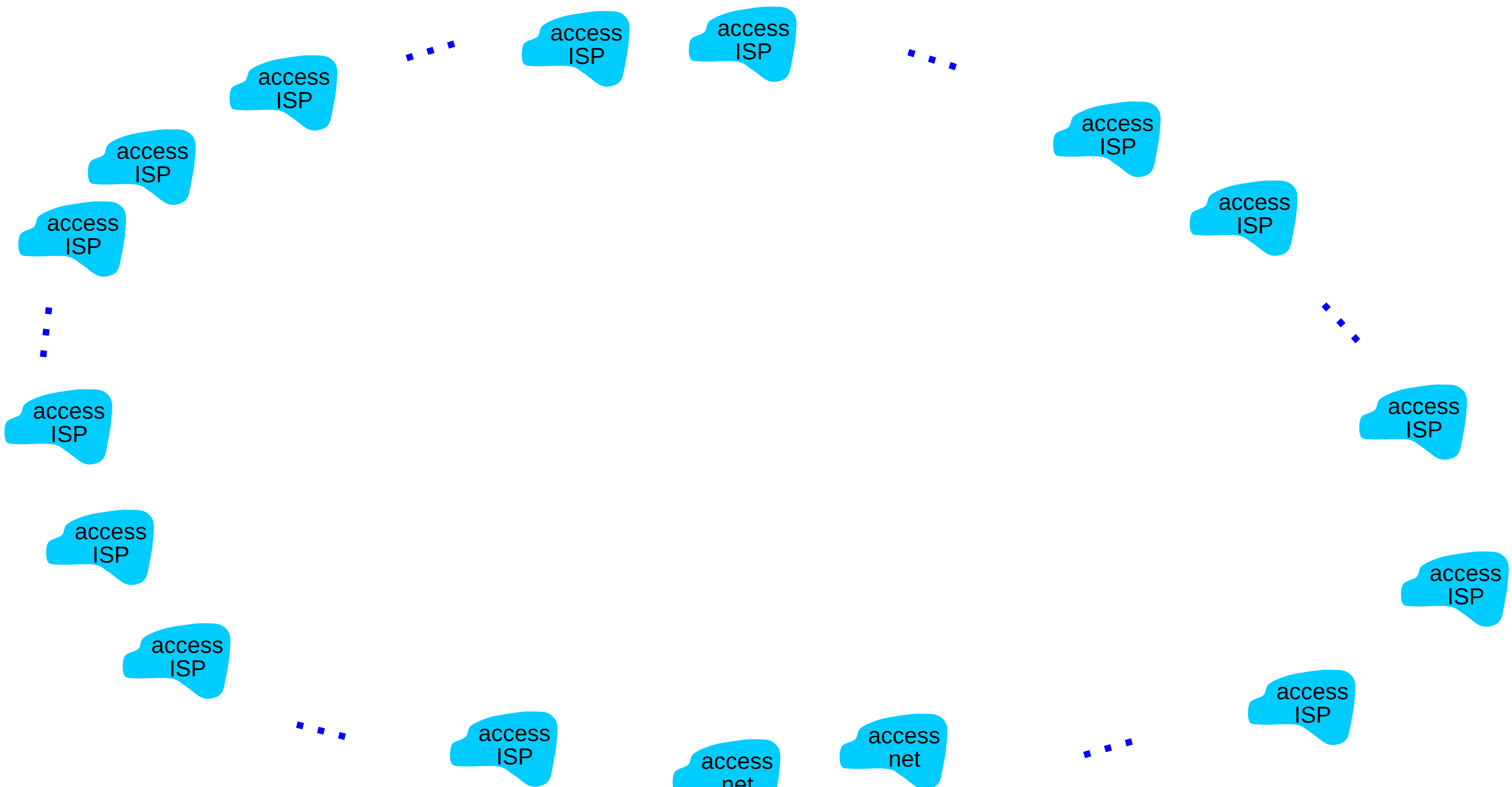
INTERNET

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**

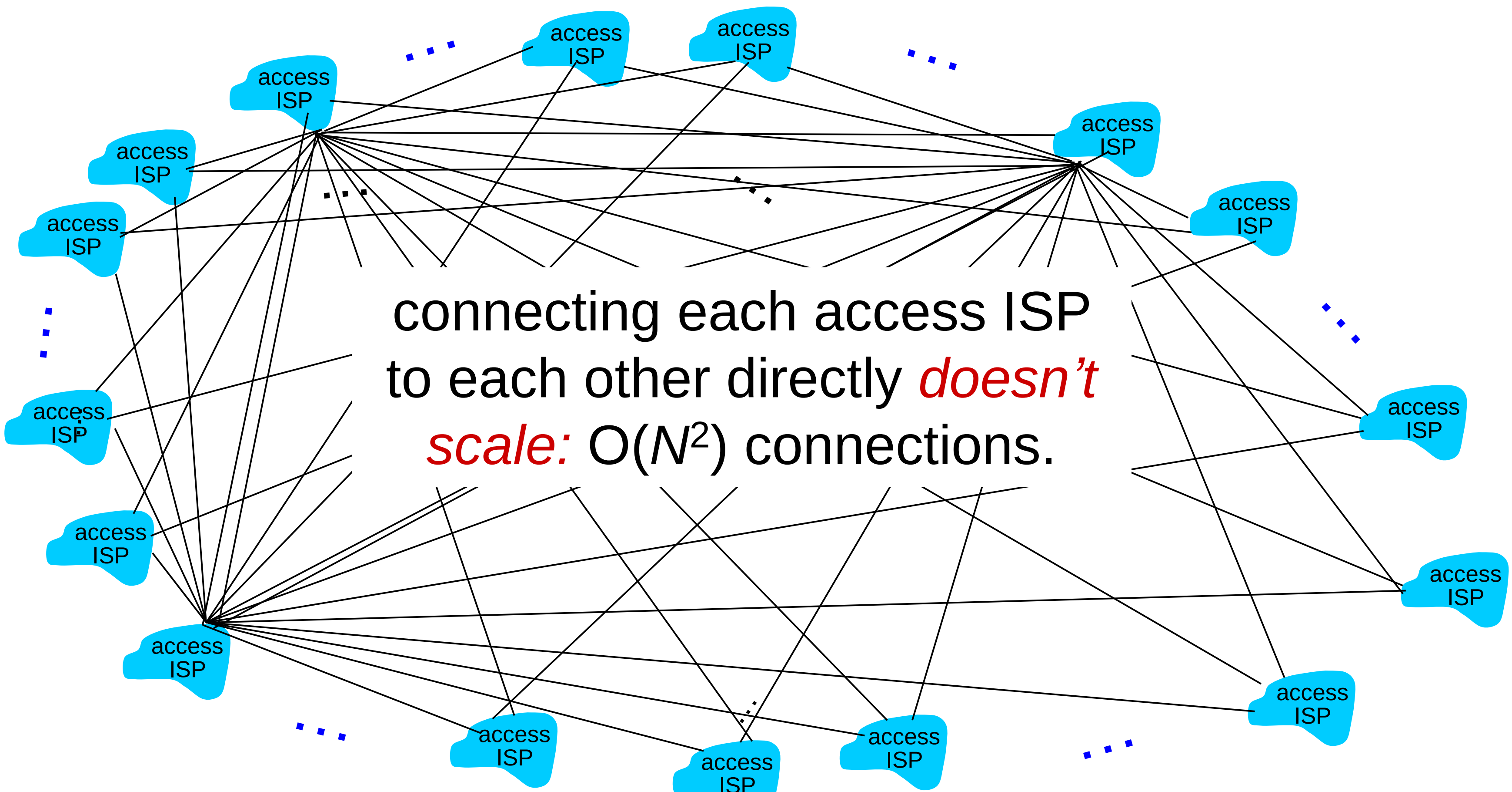
Internet Structure: network of networks

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



Internet Structure: network of networks

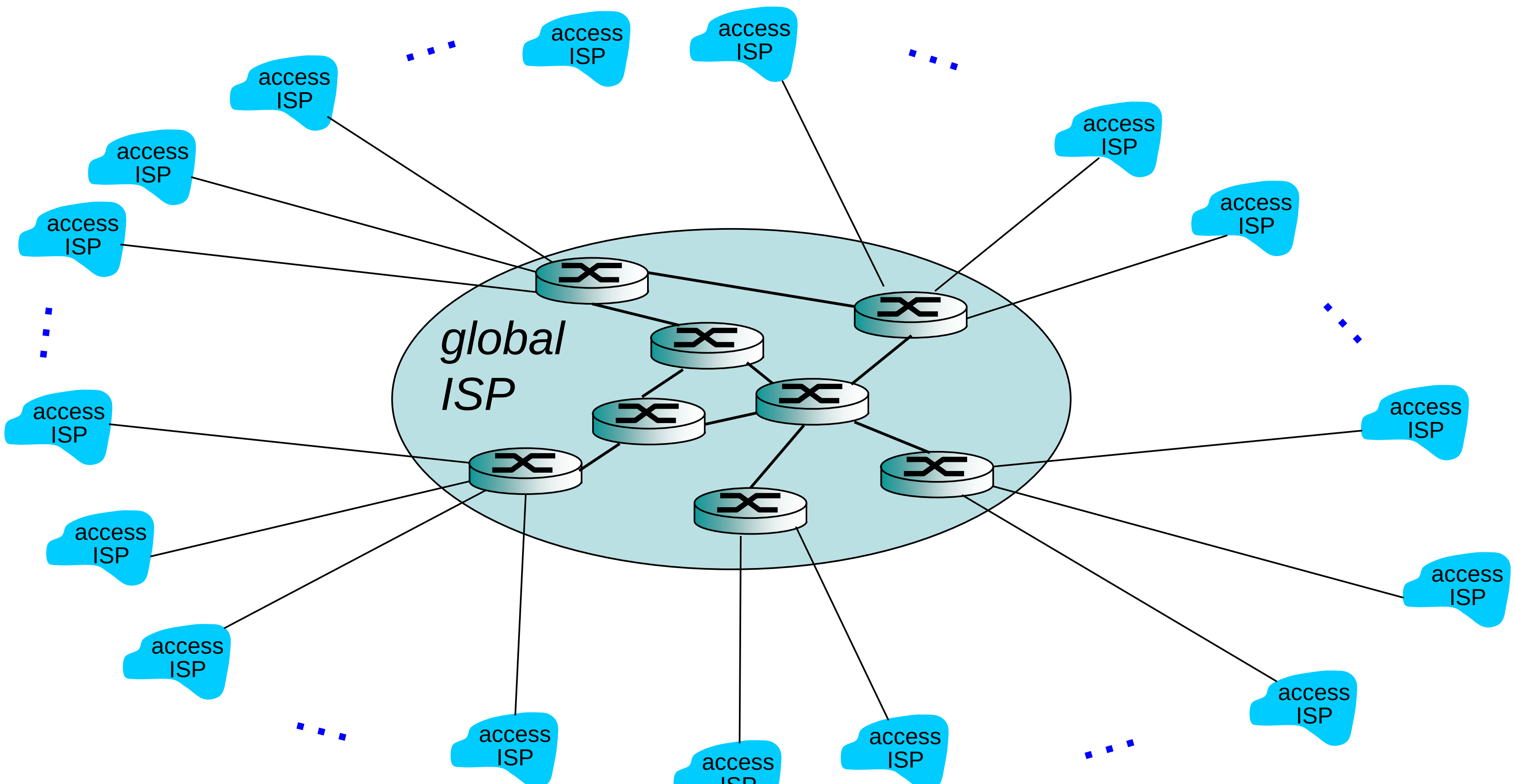
***Option:* connect each access ISP to every other access ISP?**



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

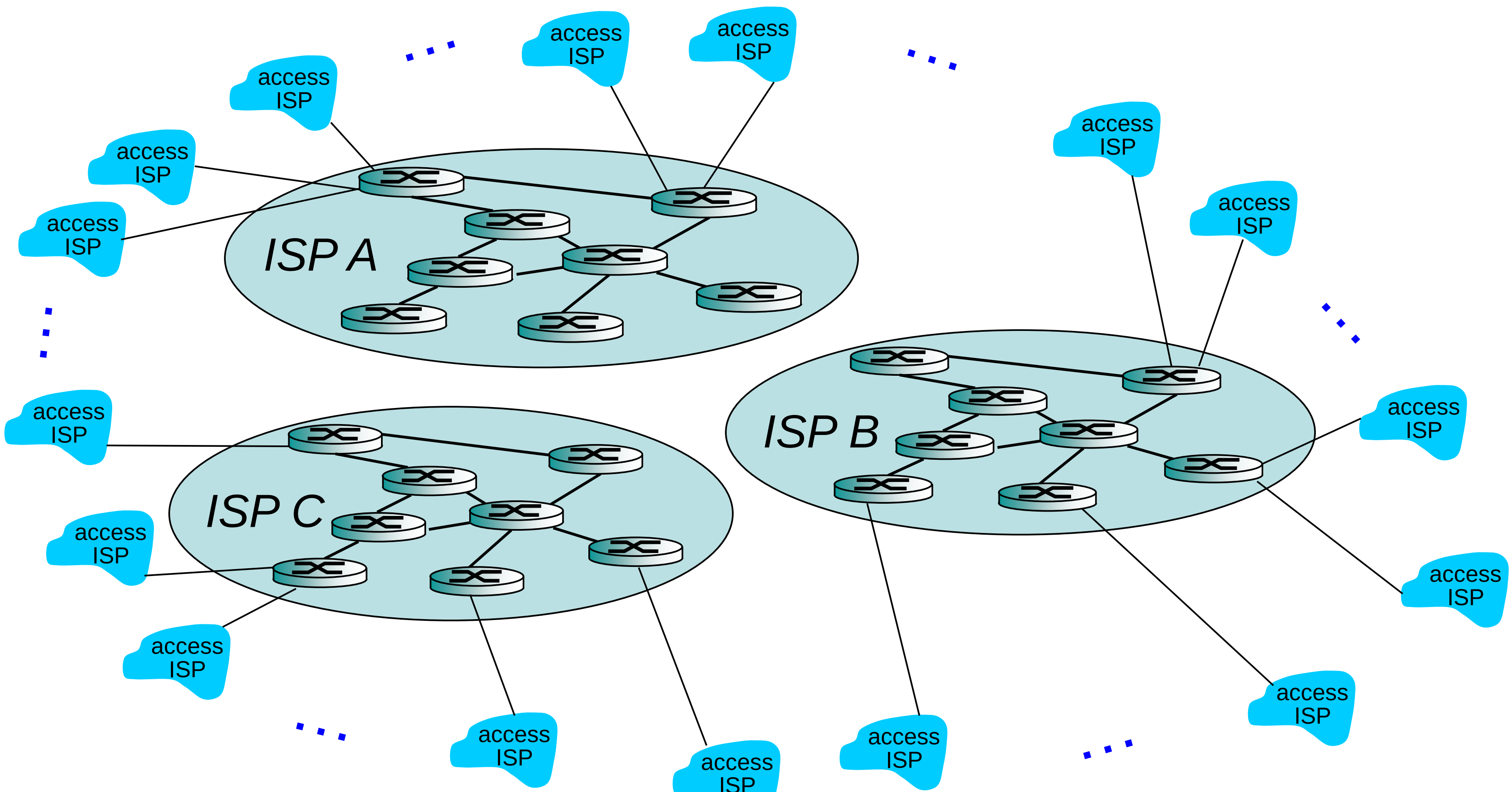
Internet Structure: network of networks

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



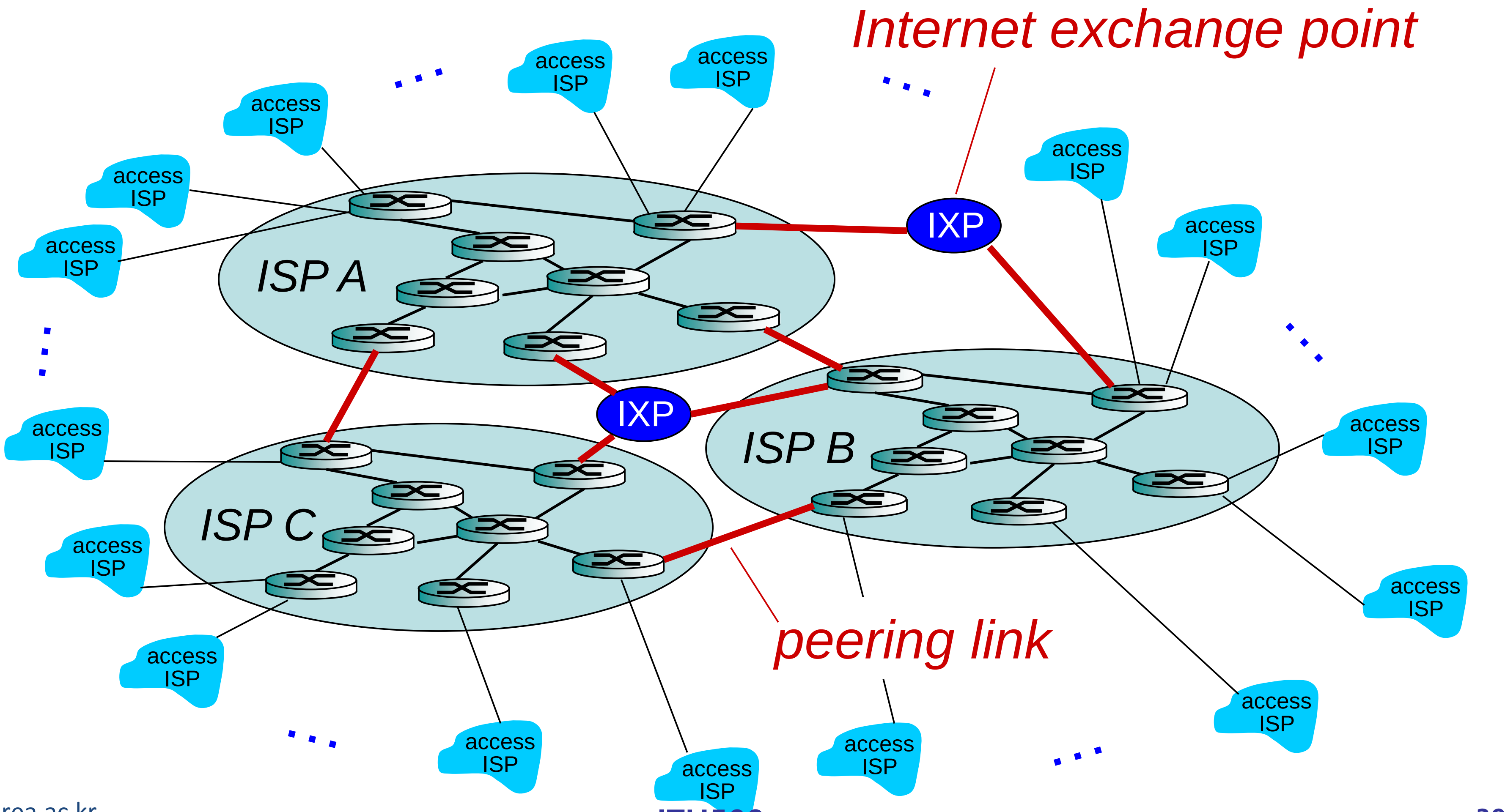
Internet Structure: network of networks

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



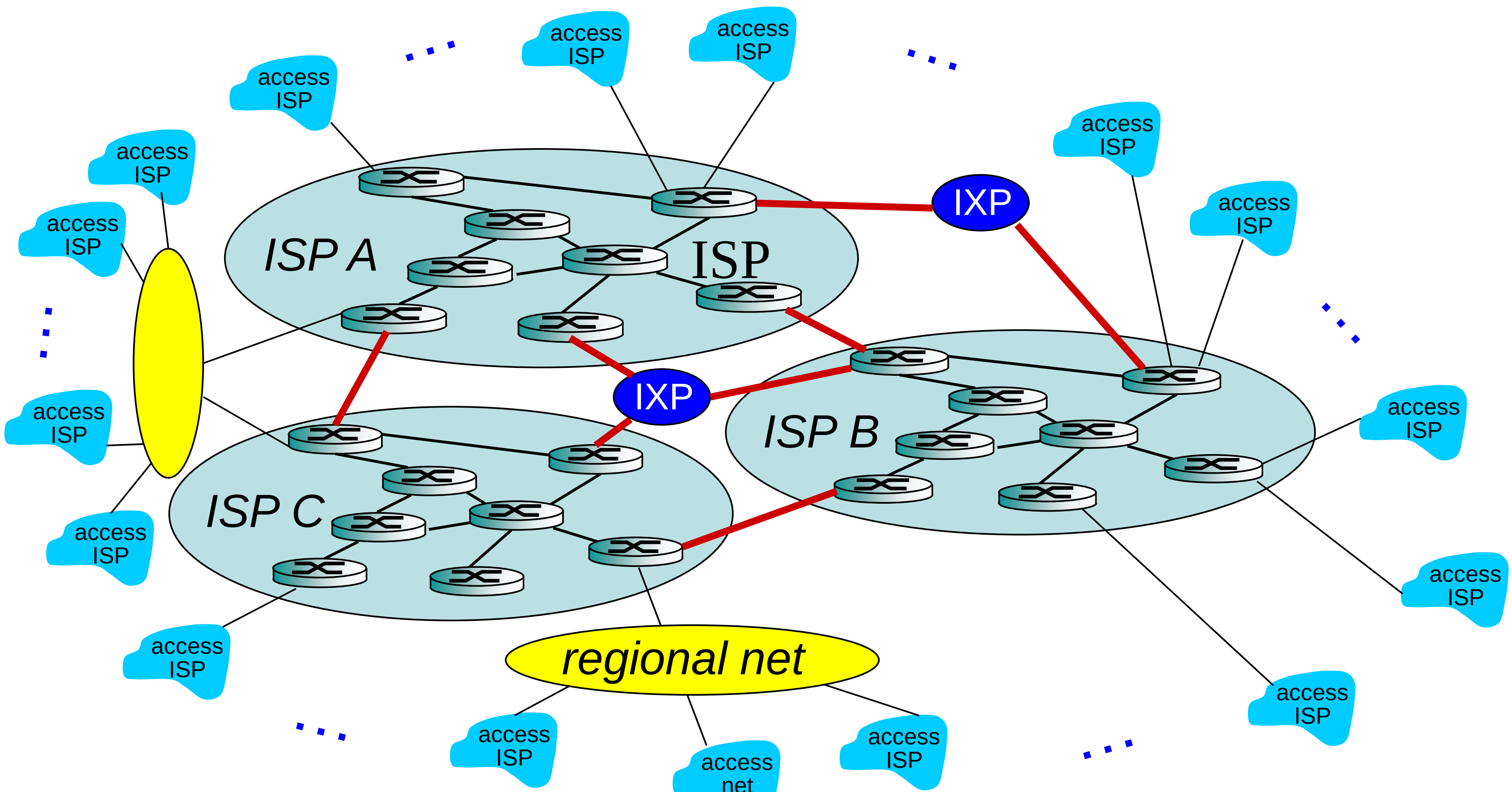
Internet Structure: network of networks

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



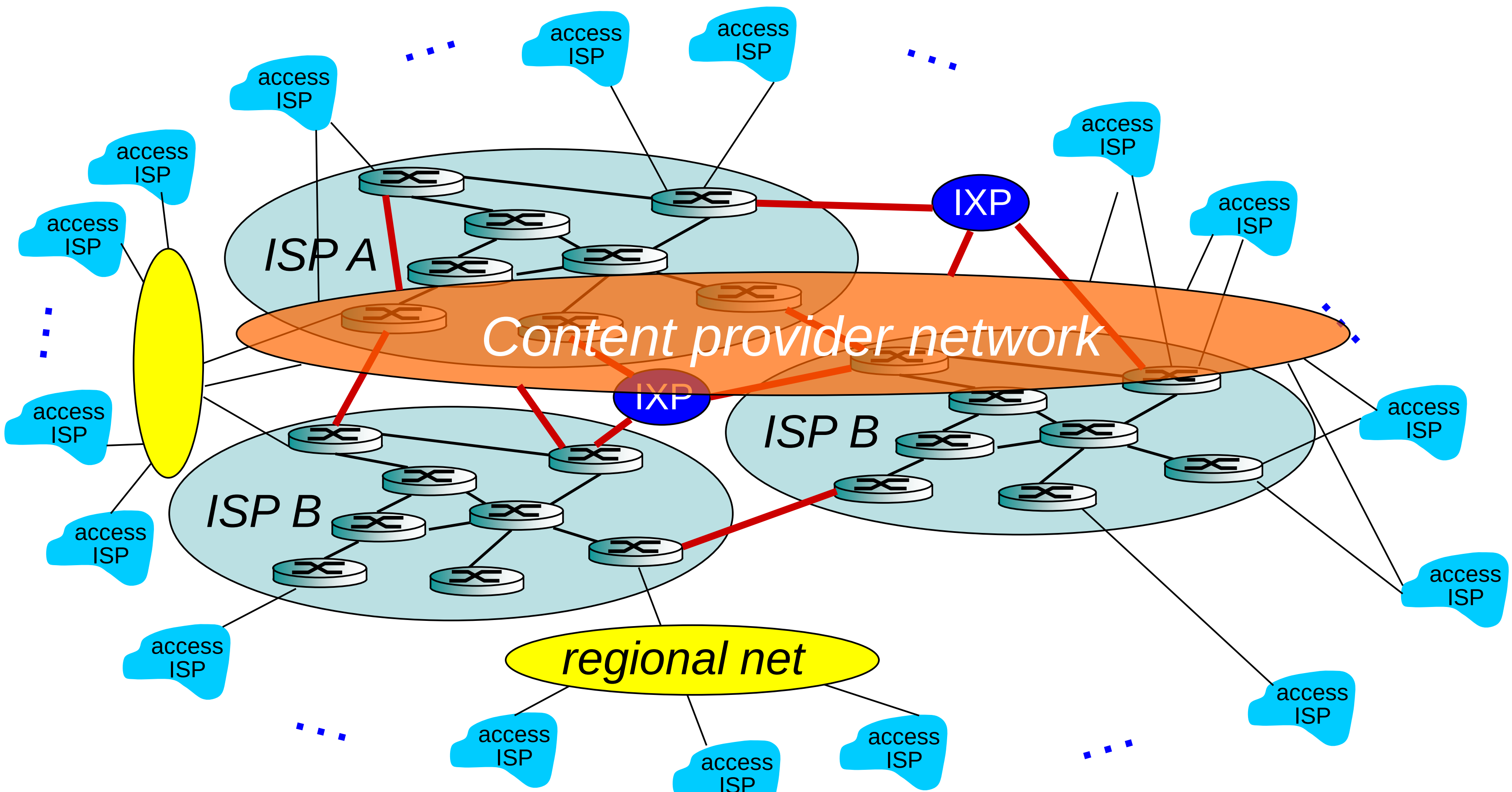
Internet Structure: network of networks

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

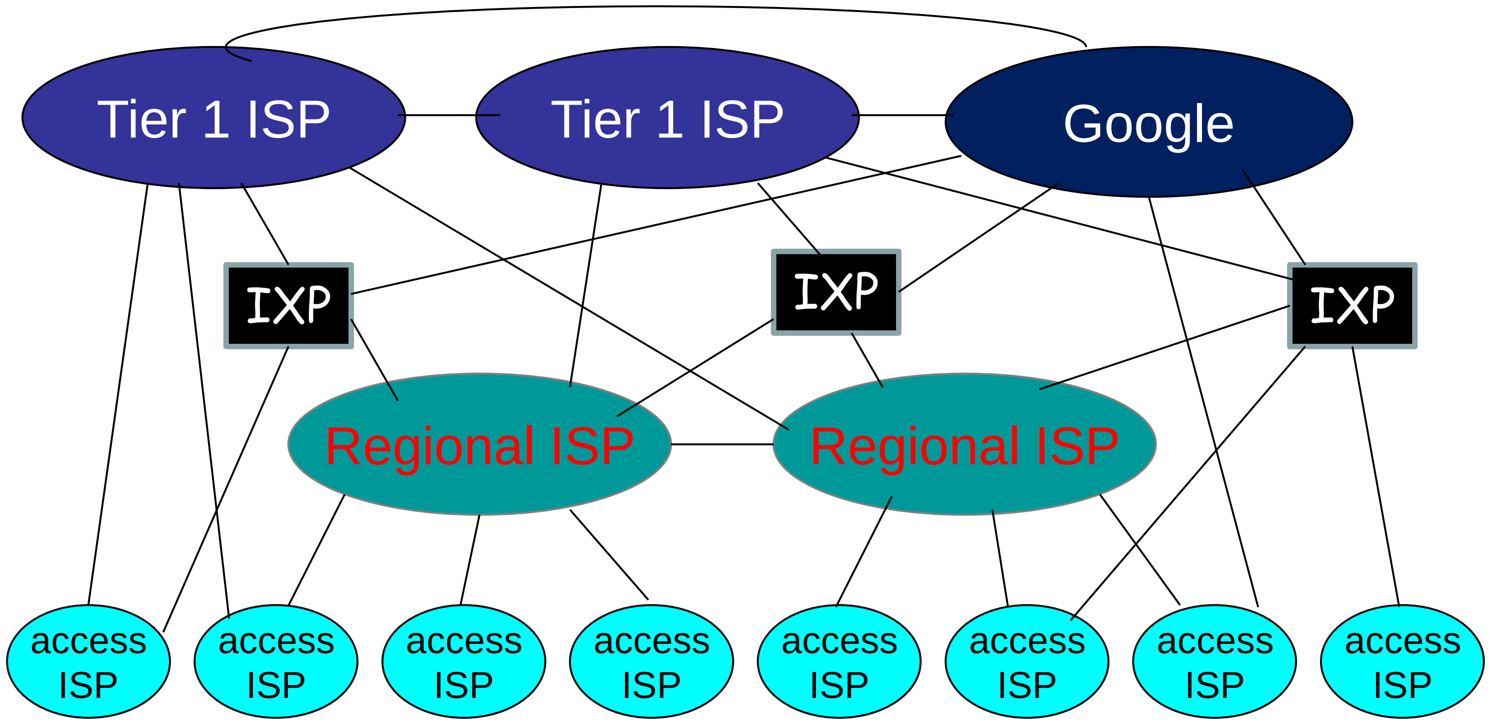


Internet Structure: network of networks

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

2. Cost-effective resource sharing

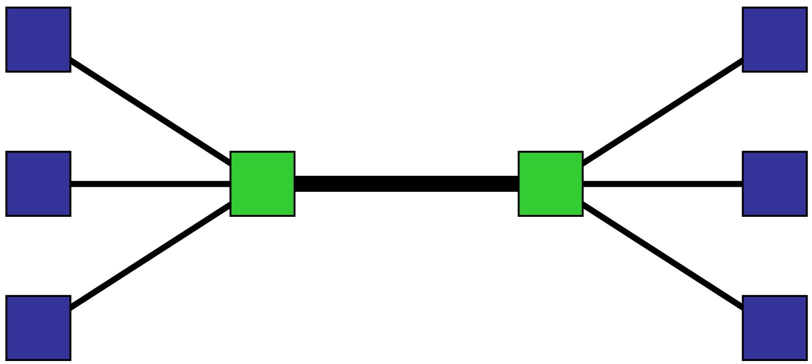
RESOURCE SHARING

Recall Role of Computer Networks

- Connectivity
- Cost-effective resource sharing
- Performance (in terms of delay and bandwidth)
- Functionality

Cost-Effective Sharing of Resources

- **Physical links and switches must be shared among many users**



- **Common multiplexing strategies**
  - (Synchronous) time-division multiplexing (TDM)
  - Frequency-division multiplexing (FDM)

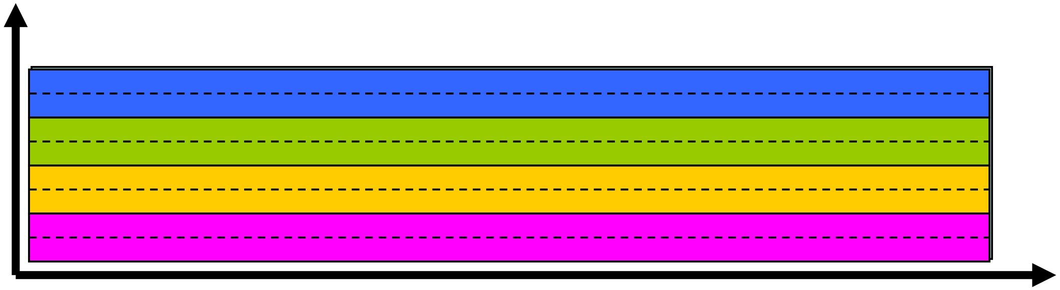
Circuit Switching: FDM versus TDM

Example:

4 users

FDM

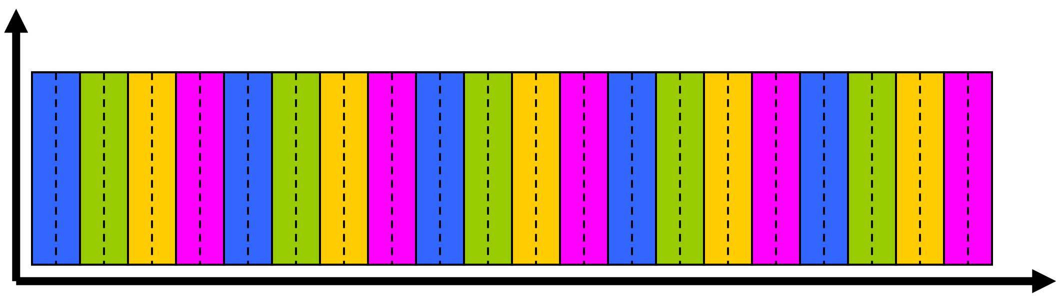
frequency



time

TDM

frequency



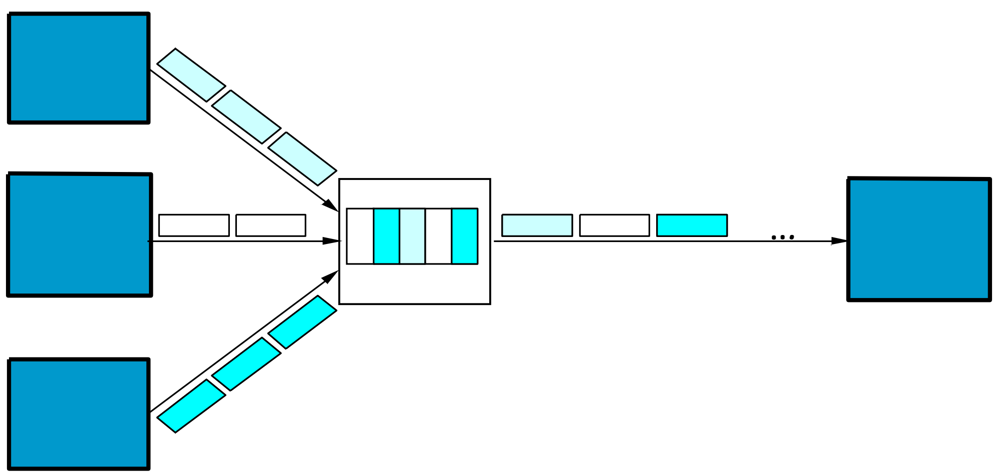
time

Statistical Multiplexing

- **Statistical Multiplexing (SM)**
  - On-demand time-division multiplexing
  - Scheduled on a **per-packet basis**
  - Packets from different sources **are interleaved**
  - Uses upper bounds to limit transmission
    - Queue size determines capacity per source

Statistical Multiplexing in a Switch

- **Packets buffered in switch until forwarded**
- **Selection of next packet depends on policy**
  - How do we make these decisions in a fair manner? Round Robin? FIFO?
  - How should the switch handle congestion?



3. Performance (in terms of delay and bandwidth)

PERFORMANCE EVALUATION

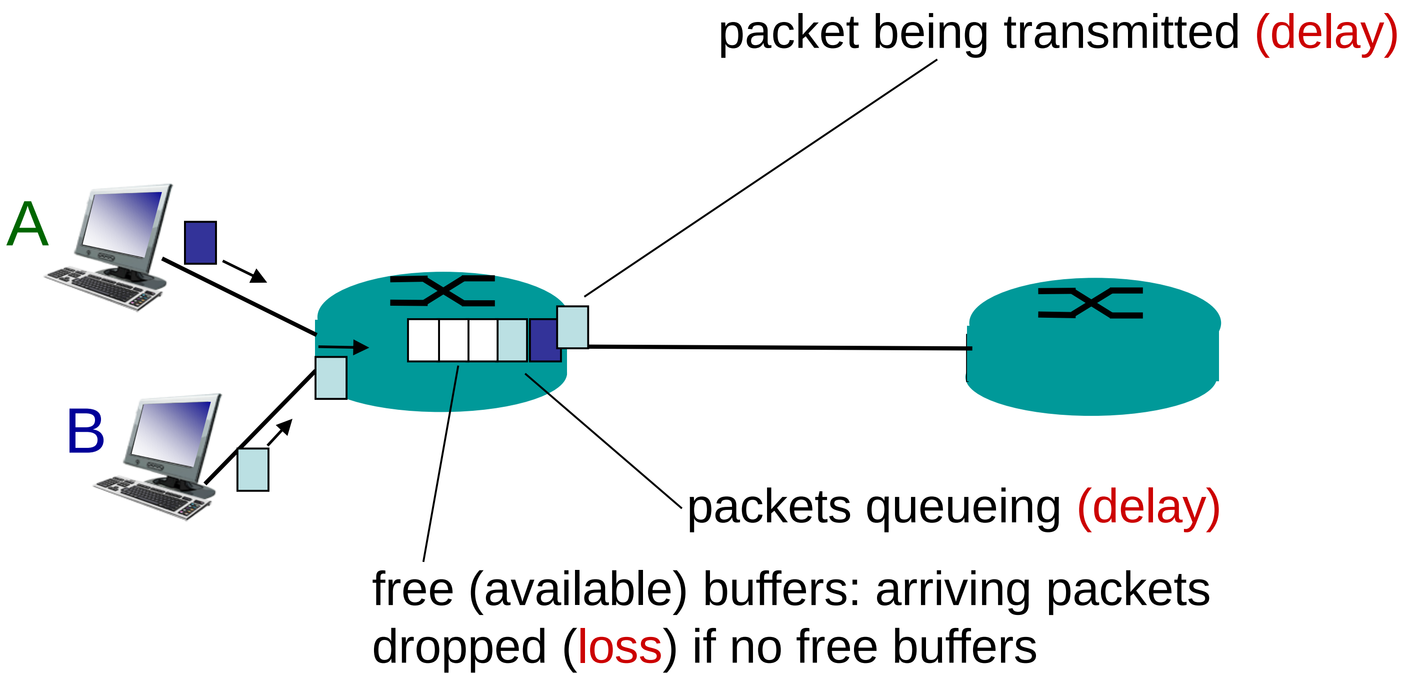
Recall Role of Computer Networks

- Connectivity
- Cost-effective resource sharing
- Performance (in terms of delay and bandwidth)
- Functionality

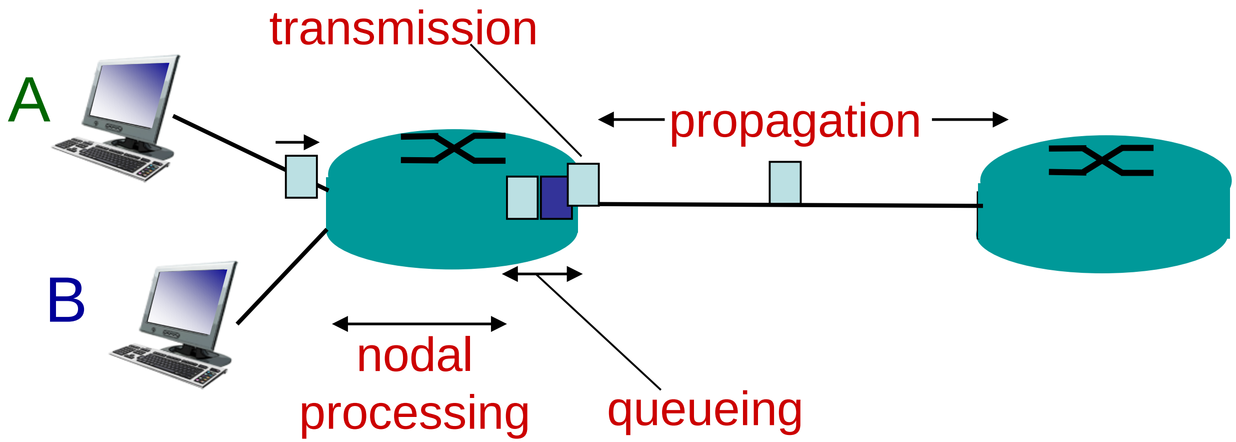
How do Loss and Delay occur?

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

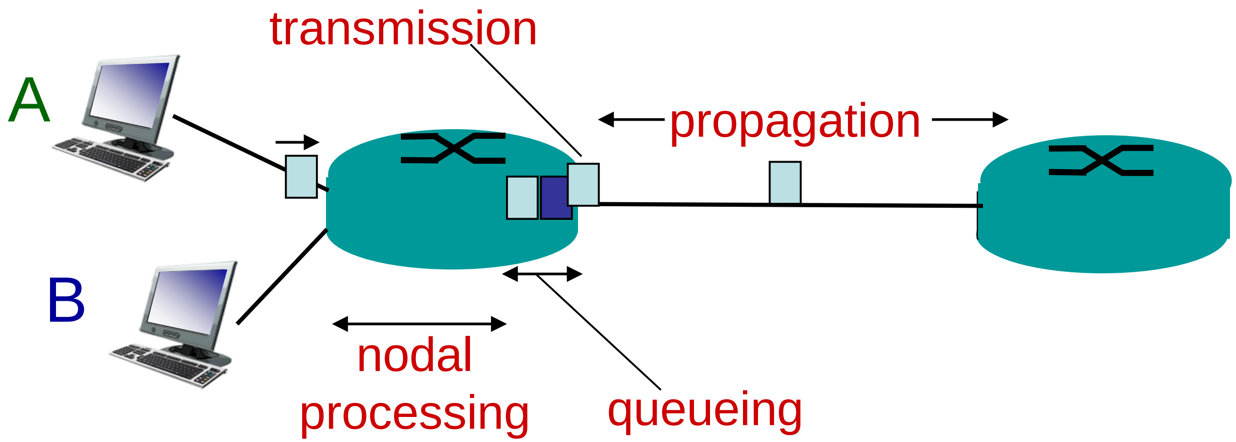


$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{proc}: nodal processing
- check bit errors
- determine output link
- typically < msec

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

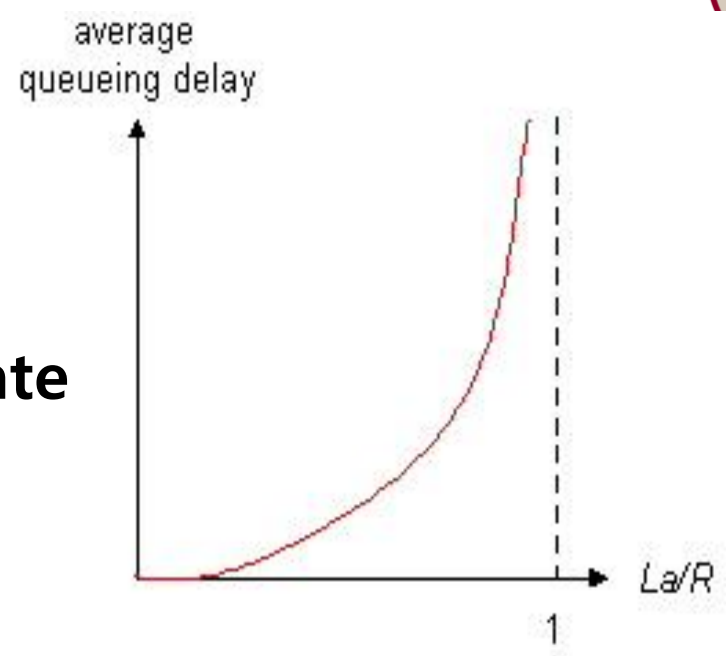
d_{trans} and d_{prop} *very* different

d_{prop}: propagation delay:

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

Queueing Delay

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

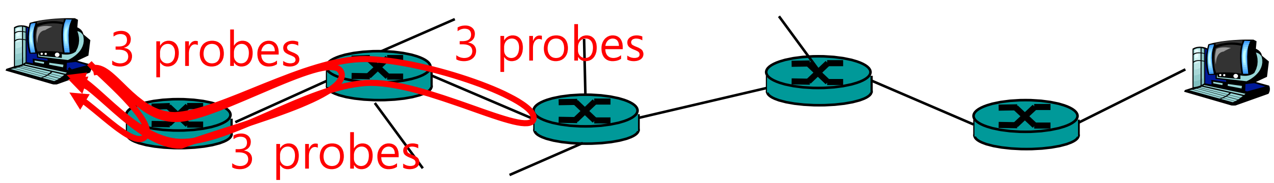


Traffic intensity = La/R

- La/R ~ 0: average queueing delay small
- La/R → 1: delays become large
- La/R → 1: more "work" arriving than can be serviced, average delay infinite!

Real Internet Delays and Routes

- What do "real" Internet delay & loss look like?
- **Traceroute program:** provides delay measurement from source to **router along end-end Internet path** towards destination. For all *i*:
  - Sends three packets that will reach router *i* on path towards destination
  - Router *i* will return packets to sender
  - Sender times interval between transmission and reply.



Real Internet Delays and Routes

traceroute: mailserver.cs.uiuc.edu to portal.korea.ac.kr

```
traceroute to portal.korea.ac.kr (163.152.6.19), 30 hops max, 40 byte packets
 1  dcsgw-dept (128.174.252.129)  1.253 ms  4.447 ms  0.731 ms
 2  uiuc-node1siebel-net.gw.uiuc.edu (128.174.1.185)  1.912 ms  0.642 ms  0.675 ms
 3  * * *
 4  130.126.0.14 (130.126.0.14)  1.444 ms  1.066 ms  0.942 ms
 5  130.126.0.30 (130.126.0.30)  0.969 ms  1.063 ms  0.933 ms
 6  t-dmzo.gw.uiuc.edu (130.126.0.70)  1.004 ms  1.020 ms  1.374 ms
 7  192.17.10.46 (192.17.10.46)  4.334 ms  4.548 ms  3.804 ms
 8  206.220.240.166 (206.220.240.166)  4.125 ms  10.570 ms  3.867 ms
 9  192.203.116.9 (192.203.116.9)  195.091 ms  192.615 ms  196.139 ms
10  apii-juniper-ge1-0-0-1036.jp.apan.net (203.181.248.226)  190.498 ms  191.076 ms  190.460 ms
11  203.181.249.161 (203.181.249.161)  192.779 ms  192.425 ms  192.874 ms
12  192.168.99.1 (192.168.99.1)  193.692 ms  193.713 ms  193.740 ms
13  192.168.66.2 (192.168.66.2)  193.756 ms  193.755 ms  193.673 ms
14  sgp-tgp.koren21.net (203.255.248.161)  193.683 ms  193.878 ms  201.024 ms
15  ku-sgp.koren21.net (203.255.248.205)  211.900 ms  203.850 ms  202.772 ms
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Three delay measurements

trans-oceaniclink

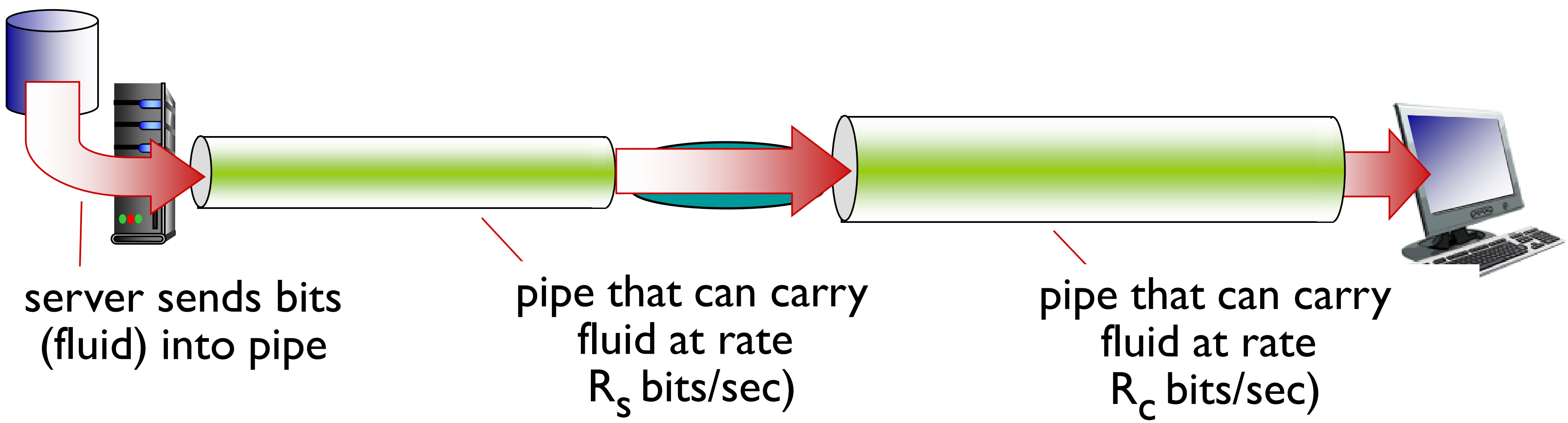
* means no response (probe lost, router not replying)

Packet Loss

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

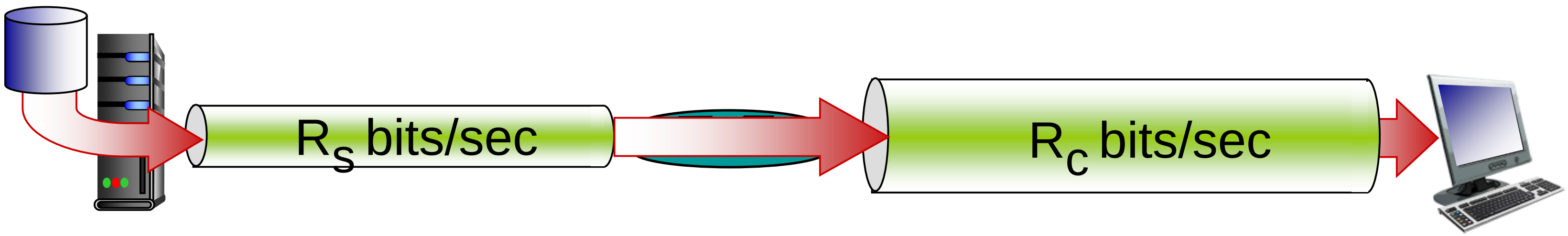
Throughput

- ***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

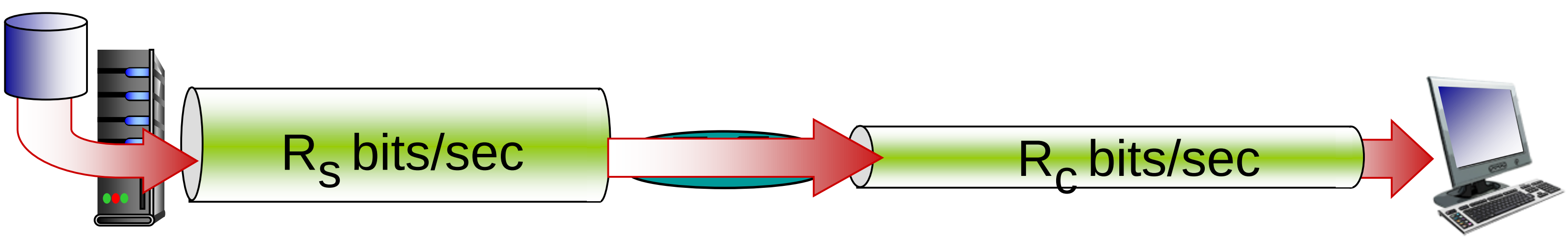


Throughput (more)

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



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

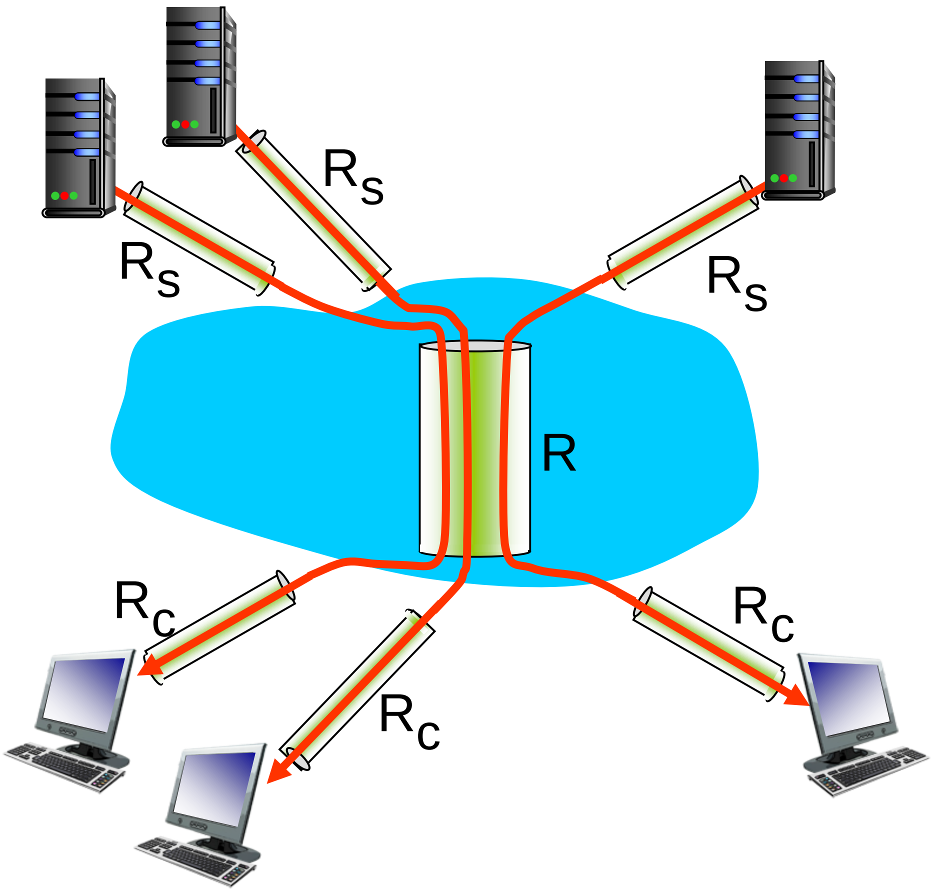


bottleneck link

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

4. Abstracting Common functionalities.

PROTOCOLS

Recall Role of Computer Networks

- Connectivity
- Cost-effective resource sharing
- Performance (in terms of delay and bandwidth)
- Functionality

Functionality

- **Support For Common Services**
  - Goal
    - Meaningful communication between hosts on a network
  - Idea
    - Common services simplify the role of applications
    - **Hide the complexity of the network** without overly constraining the application designer
  - Semantics and interface **depend on applications**
    - Request/reply: FTP, HTTP, Digital Library
    - Message stream: video-on-demand, video conferencing

Abstraction through Layering

Networks are complex!

- **Many "pieces":**
  - Hosts
  - Routers
  - Links of various media
  - Hardware, software
  - Applications
  - Protocols
- **How to organize this pieces?**

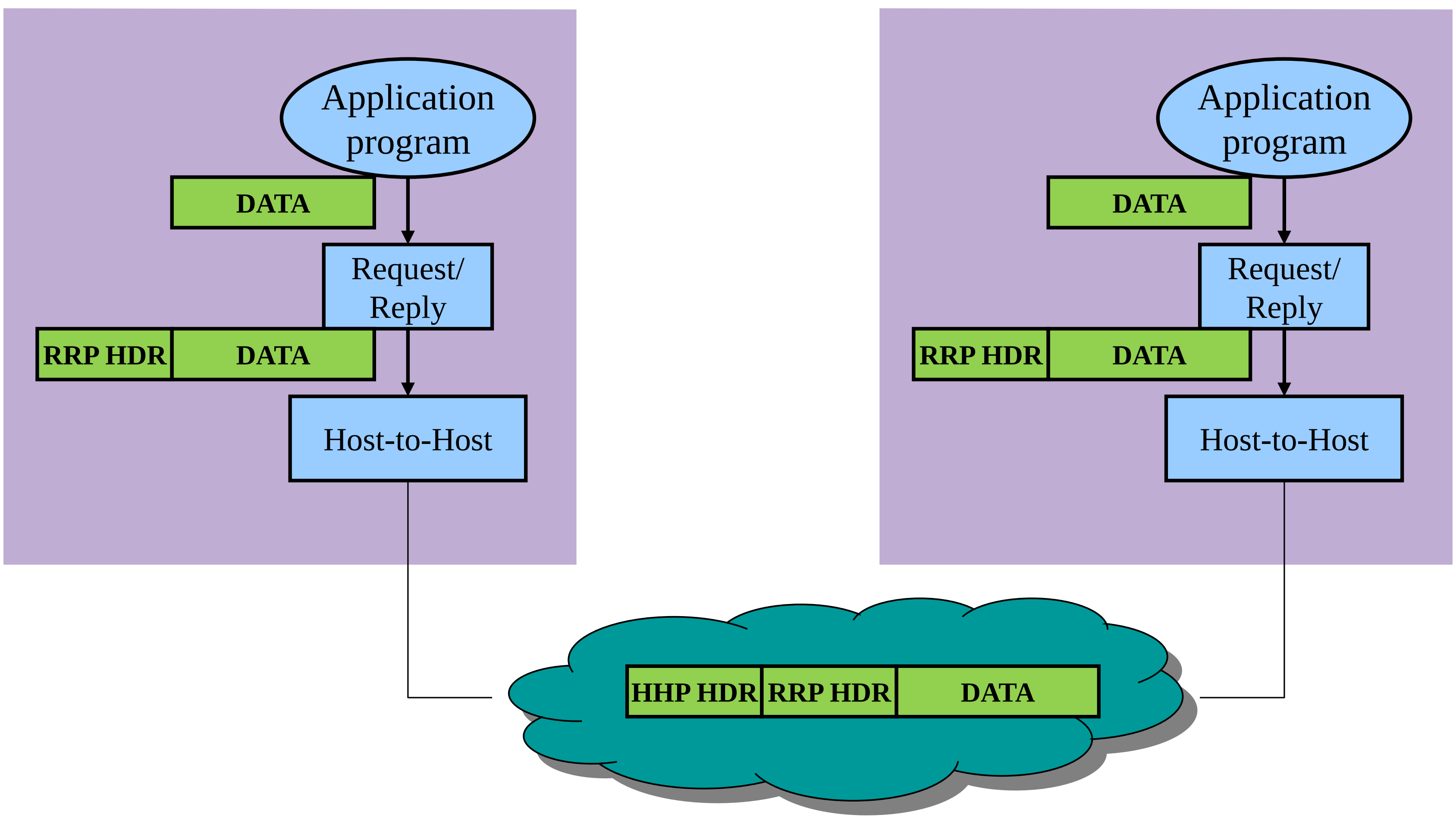
Abstraction through Layering

- **Abstract system into layers:**
  - Decompose the problem of building a network into manageable components
    - Each layer provides some functionality
  - Modular design provides flexibility
    - Modify layer independently
    - Allows alternative abstractions

Layering Concepts

- **Encapsulation**
  - **Higher layer protocols create messages** and send them **via the lower layer protocols**
  - These messages are treated as data by the lower-level protocol
  - Higher-layer protocol adds its own control information in the form of headers or trailers
- **Multiplexing and Demultiplexing**
  - Use protocol keys in the header to determine correct upper-layer protocol

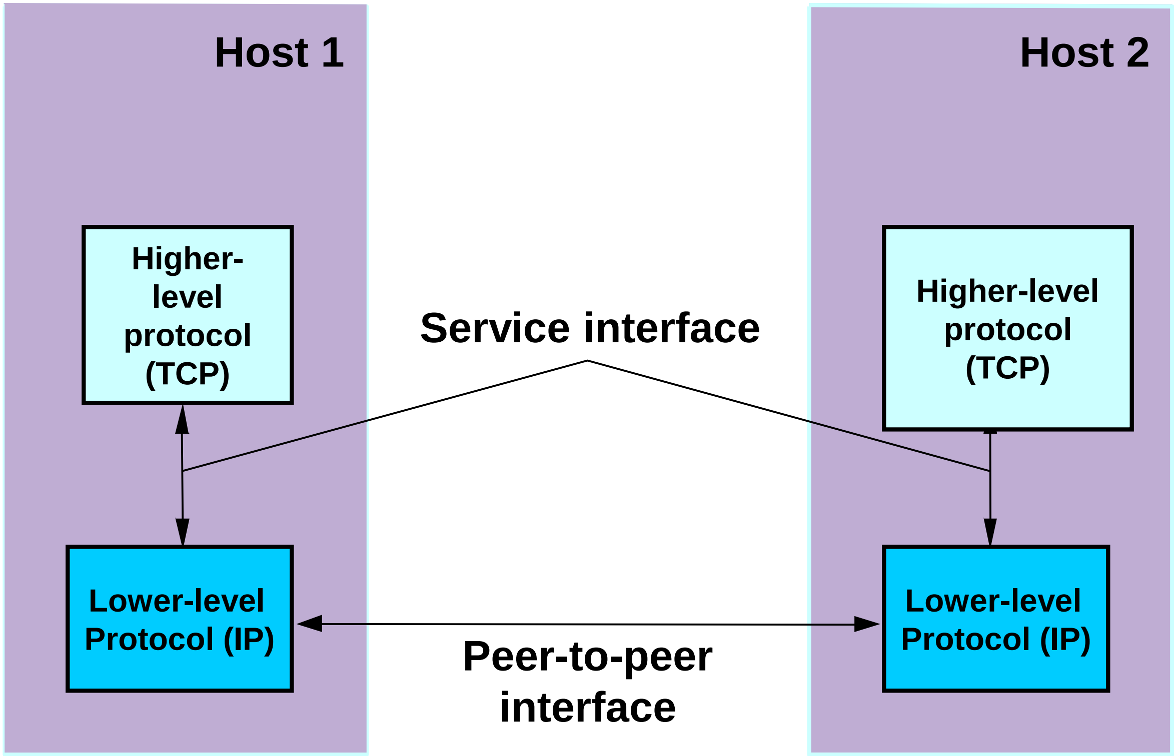
Encapsulation



Protocols

- **Definition**
  - A protocol is an abstract object that **makes up the layers** of a network system
  - A protocol **provides a communication service** that higher-layer objects use to exchange messages
    - **Service interface**:
      - To objects **on the same computer** that want to use its communication services
    - **Peer interface**:
      - To its counterpart on **a different machine**. Peers communicate using the services of lower-level protocols

Protocols: Interfaces

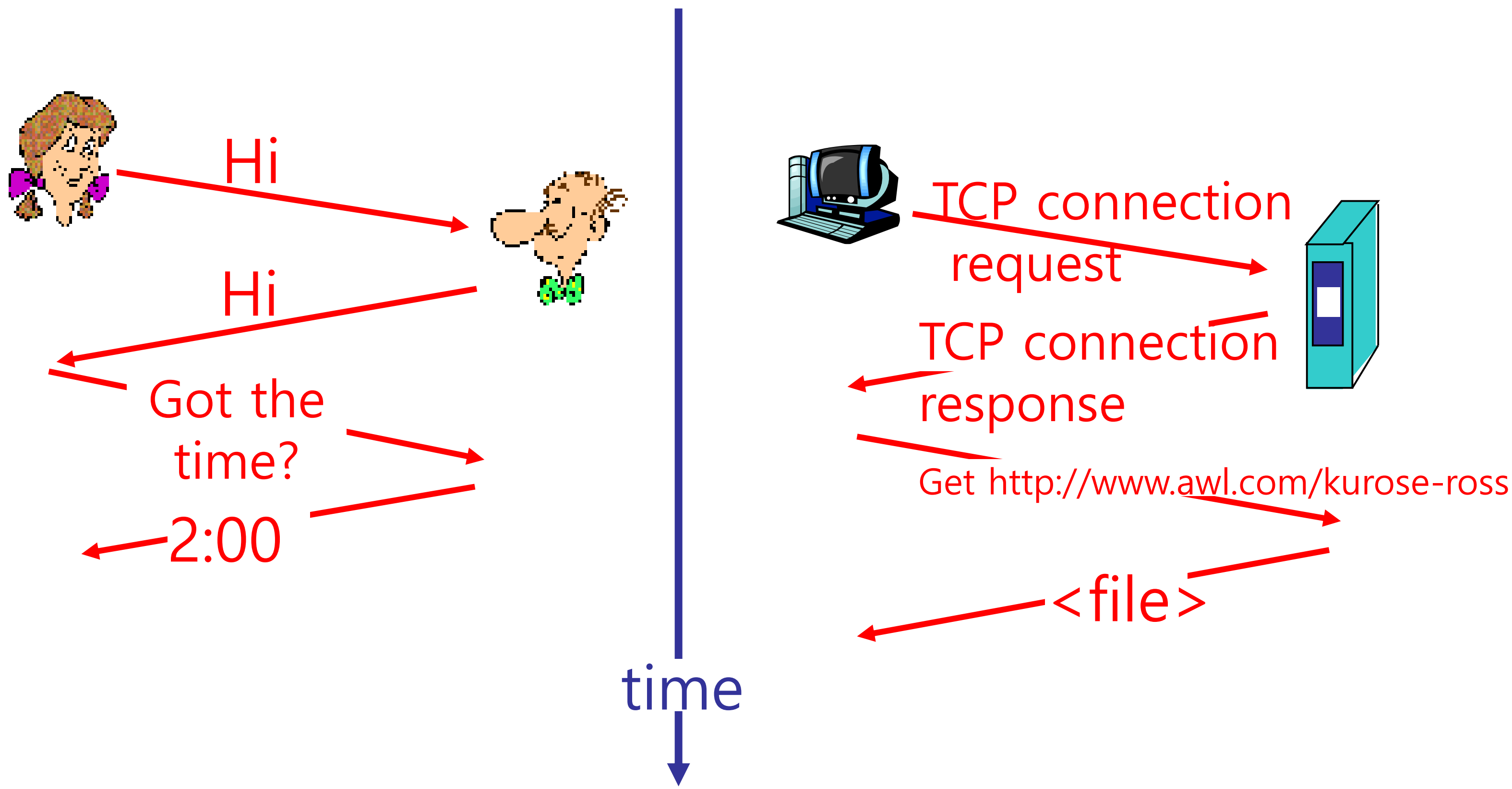


Protocols: Terminology

- **Term "protocol" is overloaded**
  - **Specification** of **peer-to-peer interface**
  - **Module that implements** this interface
- **Protocol**
  - Specifies **the format of data** in messages.
  - Specifies **the sequence of messages** to be exchanged and the action to be taken upon receipt of messages.

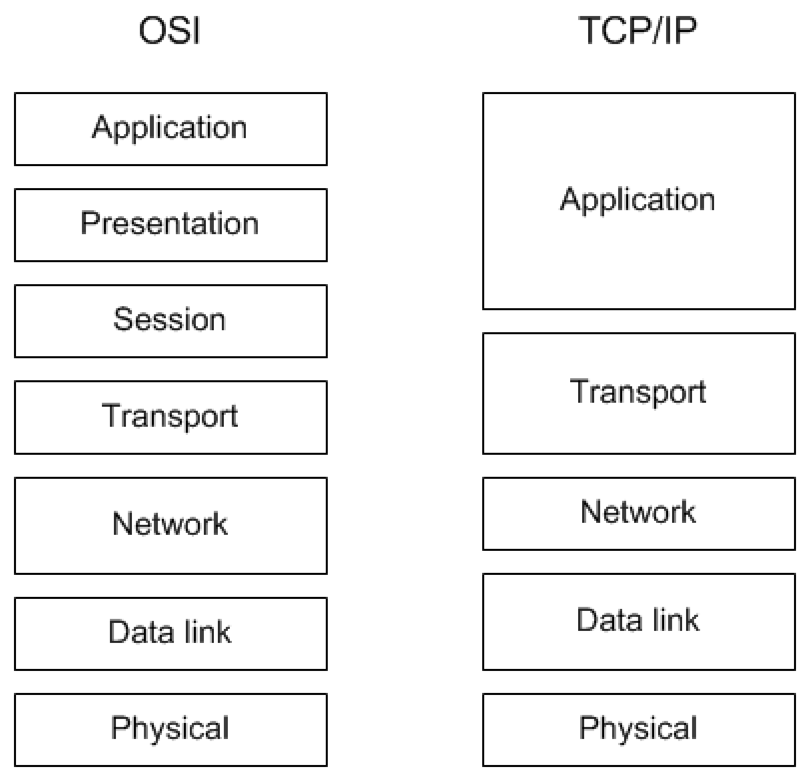
Protocols: Sequence

a human protocol and a computer network protocol:



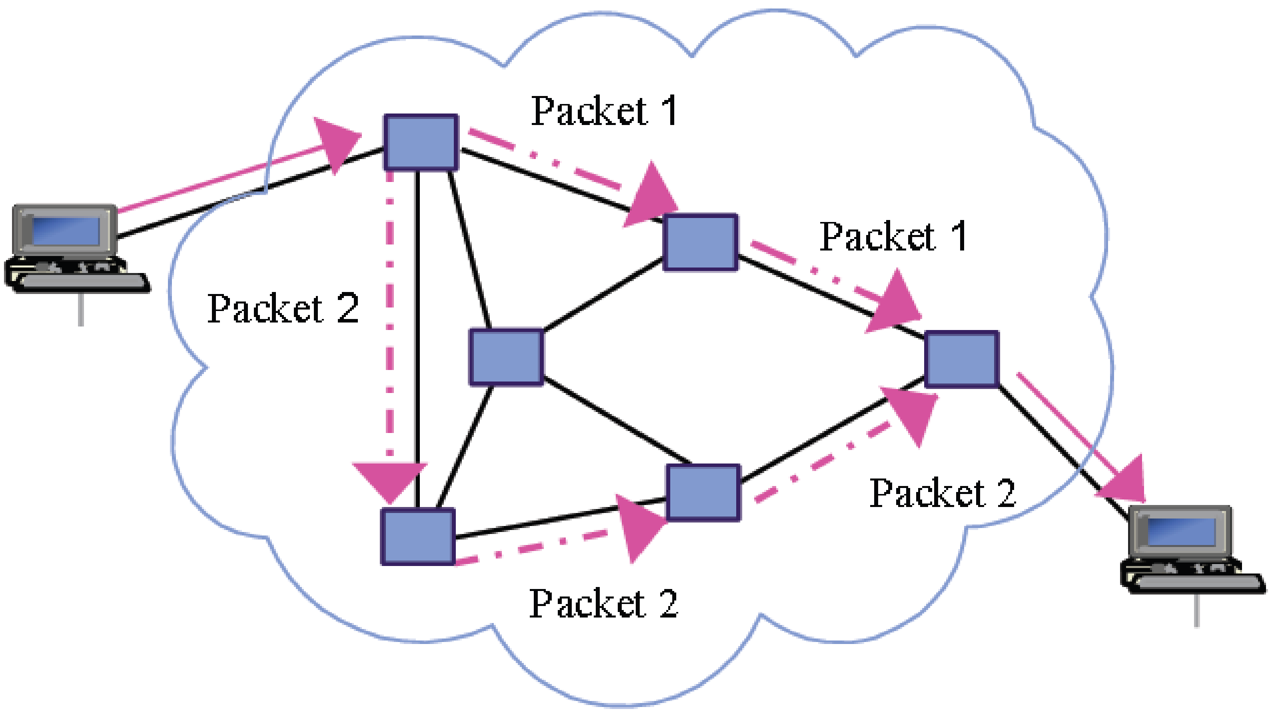
Internet Protocol Stack

- **Application:** **supporting network applications**
  - FTP, SMTP, HTTP
- **Transport:** **host-host data transfer**
  - TCP, UDP
- **Network:** **routing of datagrams from source to destination**
  - IP, routing protocols
- **Link:** **data transfer between neighboring network elements**
  - PPP, Ethernet
- **Physical:** **bits "on the wire"**



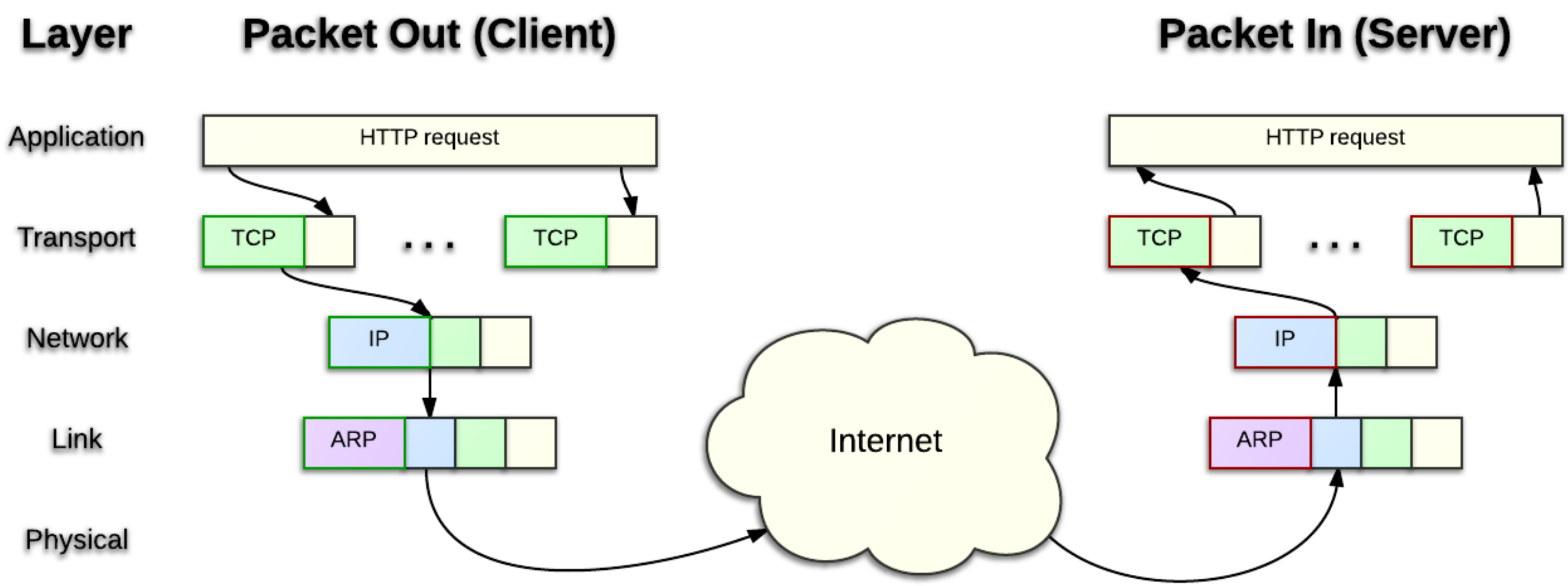
Recall Internet Protocol Architecture

- *The concept of the datagram* was fundamental to the robustness of the Internet.
- *Datagrams* can take any route available to them without human intervention.

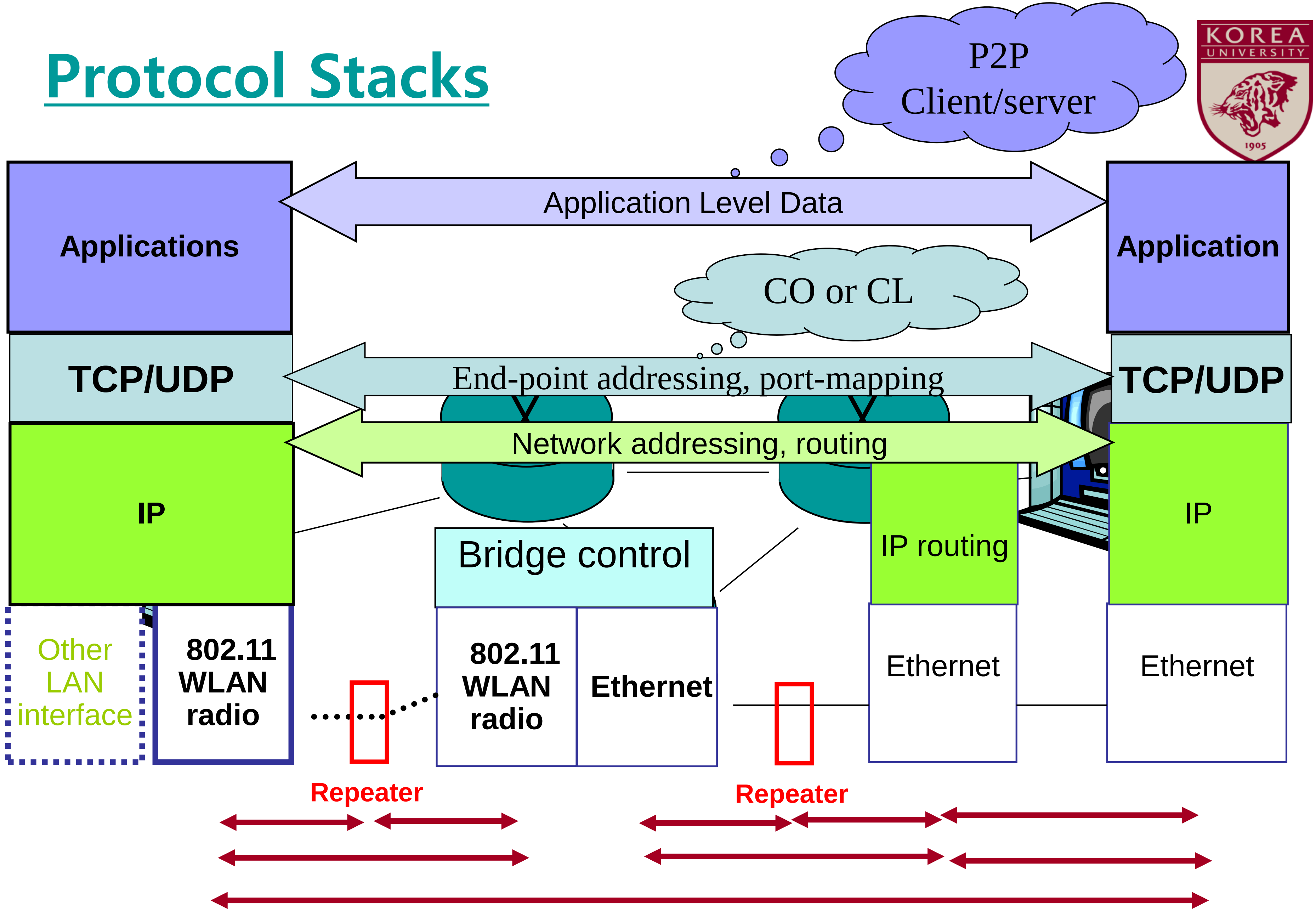


Layering & Headers

- Each layer needs to add some control information to the data to do it's job.
- This information is typically pre-appended to the data before being given to the lower layer.
- Once the lower layers deliver the data and control information - the peer layer uses the control information.



Protocol Stacks



NETWORK SECURITY

Network Security

- **Field of network security:**
  - How bad guys can attack computer networks
  - How we can defend networks against attacks
  - How to design architectures that are immune to attacks
- **Internet not originally designed with (much) security in mind**
  - Original vision: "a group of mutually trusting users attached to a transparent network" ☺
  - Internet protocol designers playing "catch-up"
  - We need security considerations in all layers!

Attackers can inject malware

- **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**

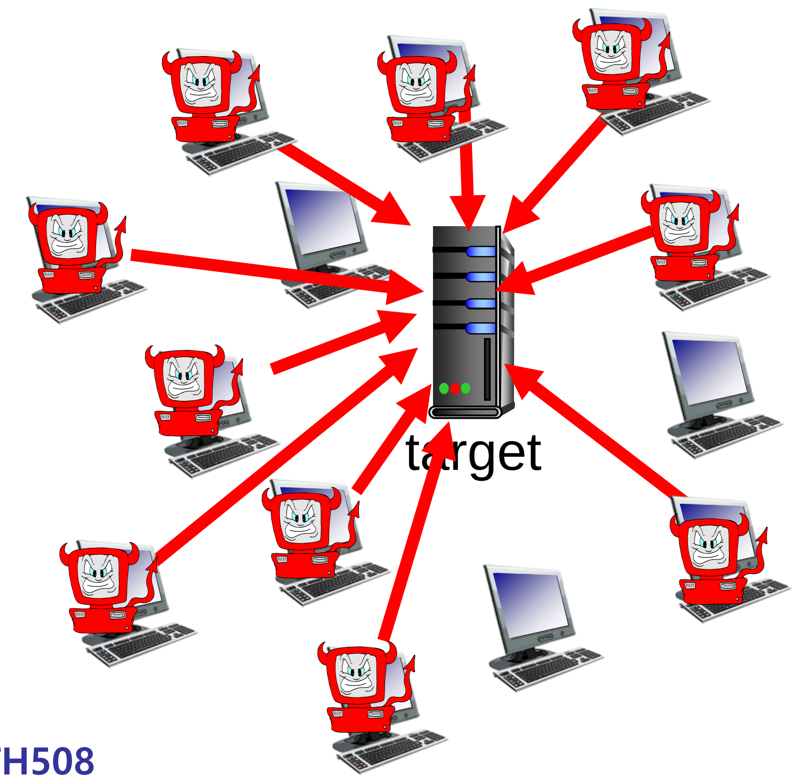
Attackers can degrade services

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

1. select target

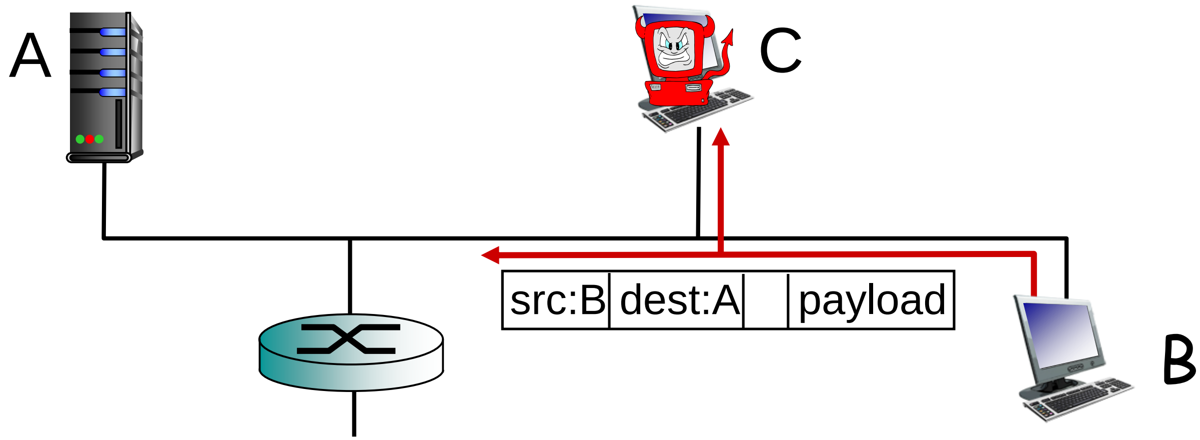
2. break into hosts around the network (see botnet)

3. send packets to target from compromised hosts



Attackers can sniff packets

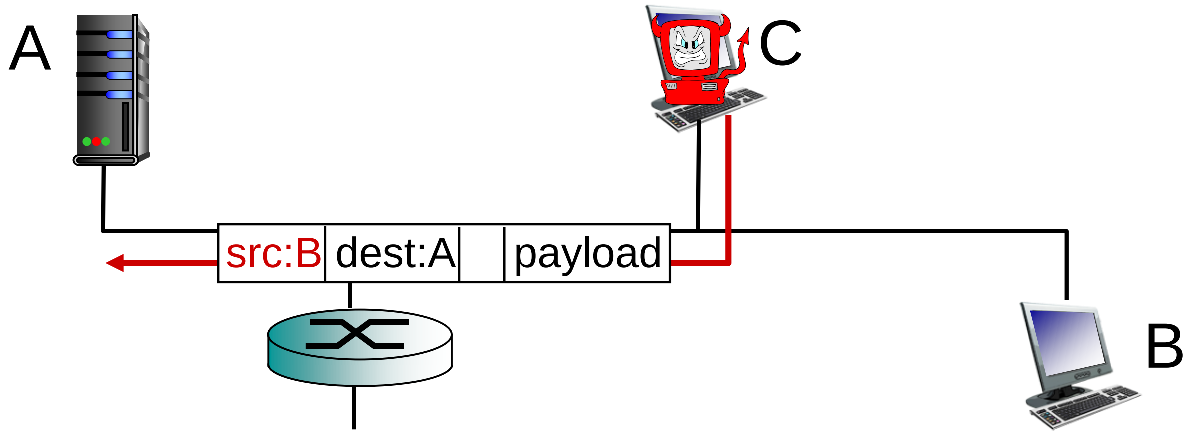
- **Packet “sniffing”:**
  - Broadcast media (shared ethernet, wireless)
  - Promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



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

Attackers can use fake addresses

- **IP spoofing: send packet with false source address**



… lots more on security

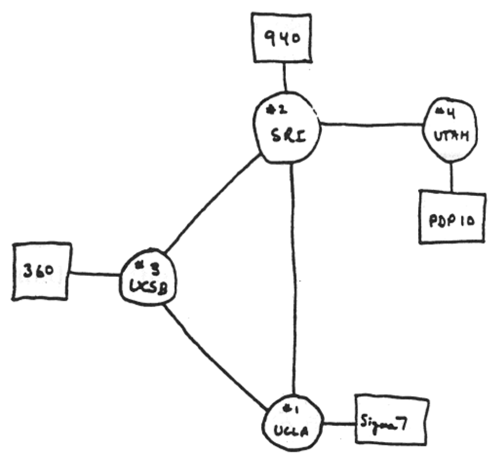
INTERNET HISTORY

Internet History

1961-1972: Early packet-switching principles

- **1961:** **Kleinrock - queueing theory shows effectiveness of packet-switching**
- **1964:** **Baran - packet-switching in military nets**
- **1967:** **ARPAnet conceived by Advanced Research Projects Agency**
- **1969:** **first ARPAnet node operational**

- **1972:**
  - ARPAnet public demonstration
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes



THE ARPA NETWORK

Internet History

1972-1980: Internetworking, new and proprietary nets

- 1970: **ALOHAnet satellite network in Hawaii**
- 1974: **Cerf and Kahn - architecture for interconnecting networks**
- 1976: **Ethernet at Xerox PARC**
- late70's: **proprietary architectures: DECnet, SNA, XNA**
- late 70's: **switching fixed length packets (ATM precursor)**
- 1979: **ARPAnet has 200 nodes**

Cerf and Kahn's internetworking principles:

- minimalism, autonomy - no internal changes required to interconnect networks
- best effort service model
- stateless routers
- decentralized control

define today's Internet architecture

Internet History

1980-1990: new protocols, a proliferation of networks

- 1983: deployment of TCP/IP
- 1982: SMTP e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: FTP protocol defined
- 1988: TCP congestion control

- new national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

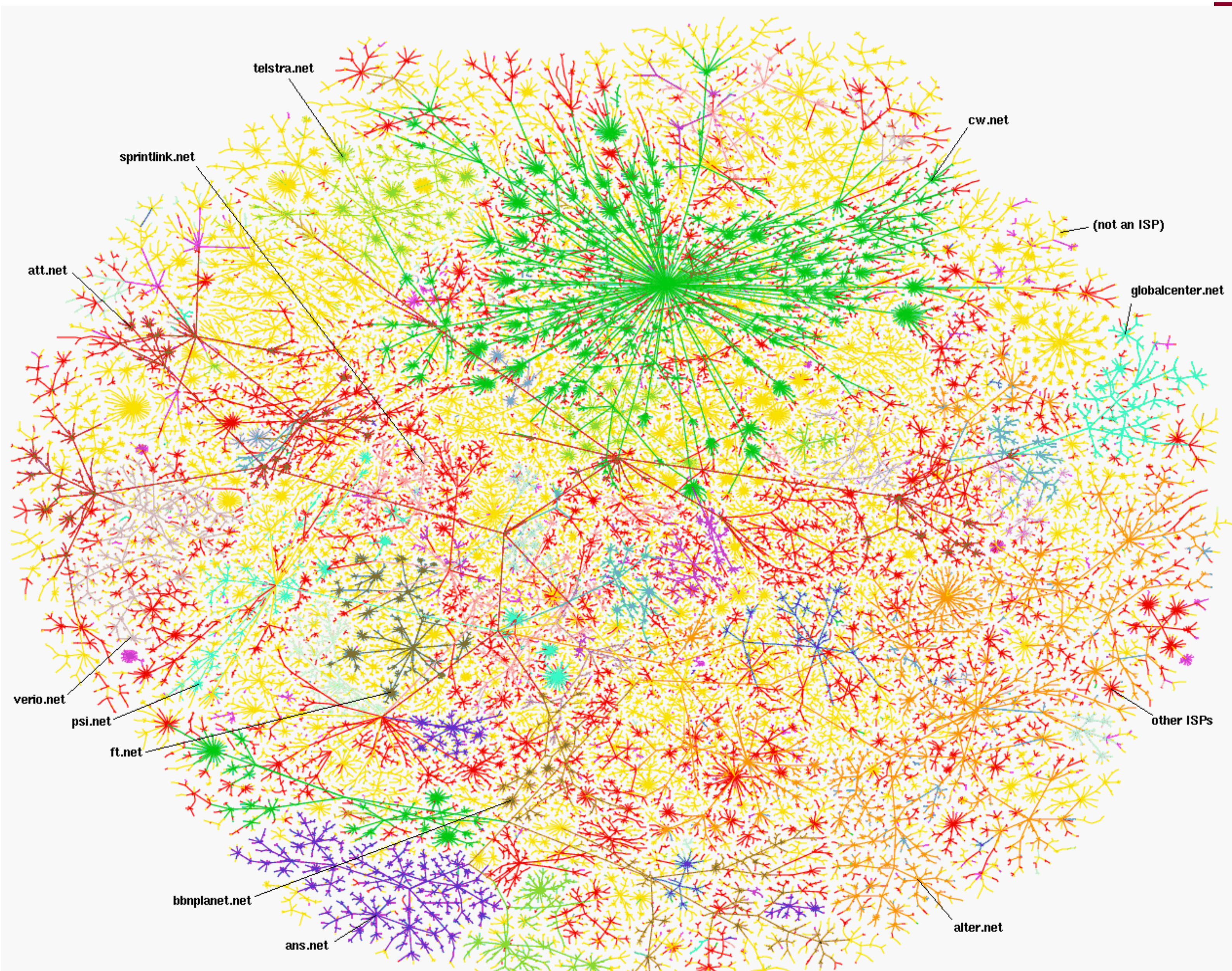
Internet History

1990, 2000's: commercialization, the Web, new apps

- **Early 1990's: ARPAnet decommissioned**
- **1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)**
- **early 1990s: Web**
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
  - late 1990's: commercialization of the Web

Late 1990's – 2000's:

- **more killer apps: instant messaging, P2P file sharing**
- **network security to forefront**
- **est. 50 million host, 100 million+ users**
- **backbone links running at Gbps**



Burch/Cheswick map of the Internet
showing the major ISPs. Data collected 28 June 1999

http://www.cheswick.com/map/index.html
Copyright (C) 1999, Lucent Technologies