University of Colorado **Boulder**

CSCI 3403 INTRO TO CYBERSECURITY

Lecture: 12-2

Topic: Networking

Presenter: Matt Niemiec

Announcements

- Project 3 (final project) is posted
- Next homework will be posted Friday or Saturday
- Upcoming guest lecturers
  - From Proofpoint on 4/9
  - From Rule4 on 4/21
  - From Twitter on 4/23 (Andy Sayler)

Final Exam, current picture

- Will probably happen
- If you get a 65% or higher on the final, the highest of your final and midterm will count for both scores
- Will be online Moodle quiz
- Will be open book, open note, open internet
  - NOT open friend
  - Will be different/harder in some ways to compensate
  - May be required to write basic scripts

- Note: This is subject to change

Exam Extra Credit (Current picture. Will be finalized before next Tuesday)

- Research a topic that won't be discussed in class
- Get credit in one of the following ways
  - Record a 5-minute video explanation of your topic for up to 10%. Details to come soon
  - Present your topic for 5 minutes in recitation for up to 15%. Depending on demand, this may be first-come-first-serve. If you got an 85% or better on the midterm, please leave this for others
  - If you're selected as an outstanding project from recitation, give a 10-minute presentation for up to 25% (total)
- Percentages apply to your higher exam score

Exam Extra Credit Criteria

- Will be graded on at least the following:
  - Interesting topic/information relevant to cybersecurity
  - Quality, professional preparation and presentation
  - Inspires the listener to want to learn more and provides resources to do so
  - Shows insight and depth in research presented in an appropriate manner for the given timeframe

Some Extra Credit Potential Topics

- Network security
  - Wireless security, honeypots, cloud security, SIEM, Tor
- Applied security
  - OWASP Top 10, reverse engineering, penetration testing
- Crypto
  - Common crypto libraries, homomorphic encryption
- Windows
  - Windows/AD security, Windows CLI
- Miscellaneous
  - Ethics in security, auditing, data provenance
- Or anything else! Just run it by Matt or your TA

Technology Recap 3/17 (Old stuff)

- Piazza is used for content-related questions
- Feedback: https://forms.gle/WRUUbPkmFNsa6q3D6
- Instructor/TA email is used for individual circumstances
- cyber@Colorado.edu is used for accommodations/logistical questions
- Moodle is used for assignments, slides, and additional resources

Technology Recap 3/17 (New stuff)

- Calendar is used for holding all Zoom meetings, instructions, and meeting IDs
  - May contain due dates, but not guaranteed
- Lecture Zoom ID: https://cuboulder.zoom.us/j/633893668
  - This and others found in Google Calendar
- Lecture capture folder: https://drive.google.com/drive/folders/1VMrHEigP4AgDwRnRPTsgQS35EAozc19-?usp=sharing

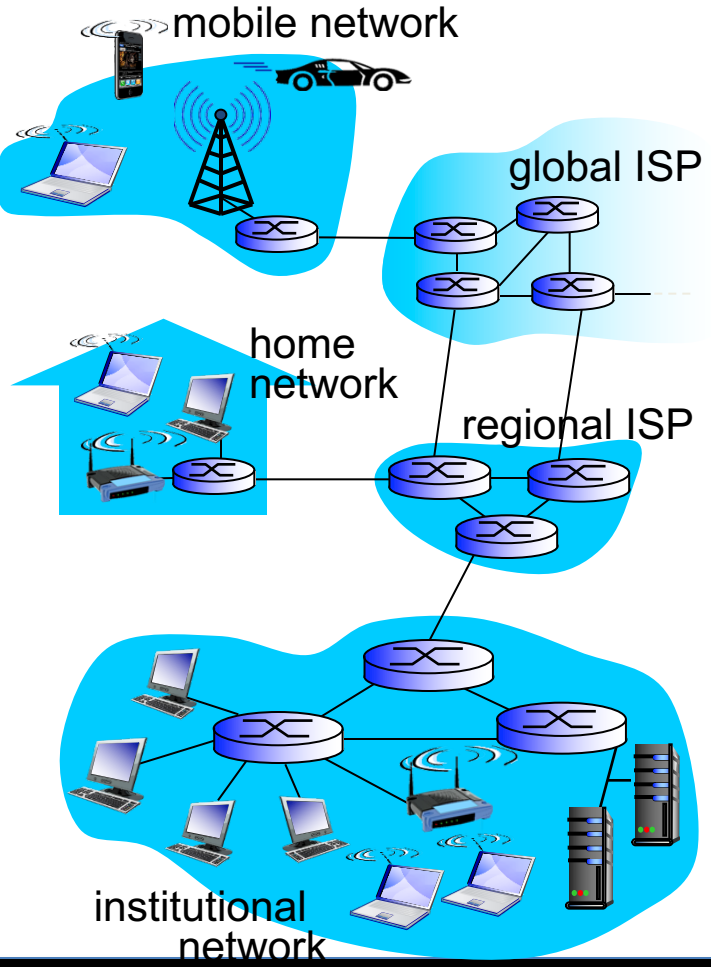
Networking Recap

Networking

- Computers talk to each other on the Internet
- We need to look at the following:
  - What's a computer network?
- We're looking at a computer network in terms of layers

A 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



What's a Protocol?

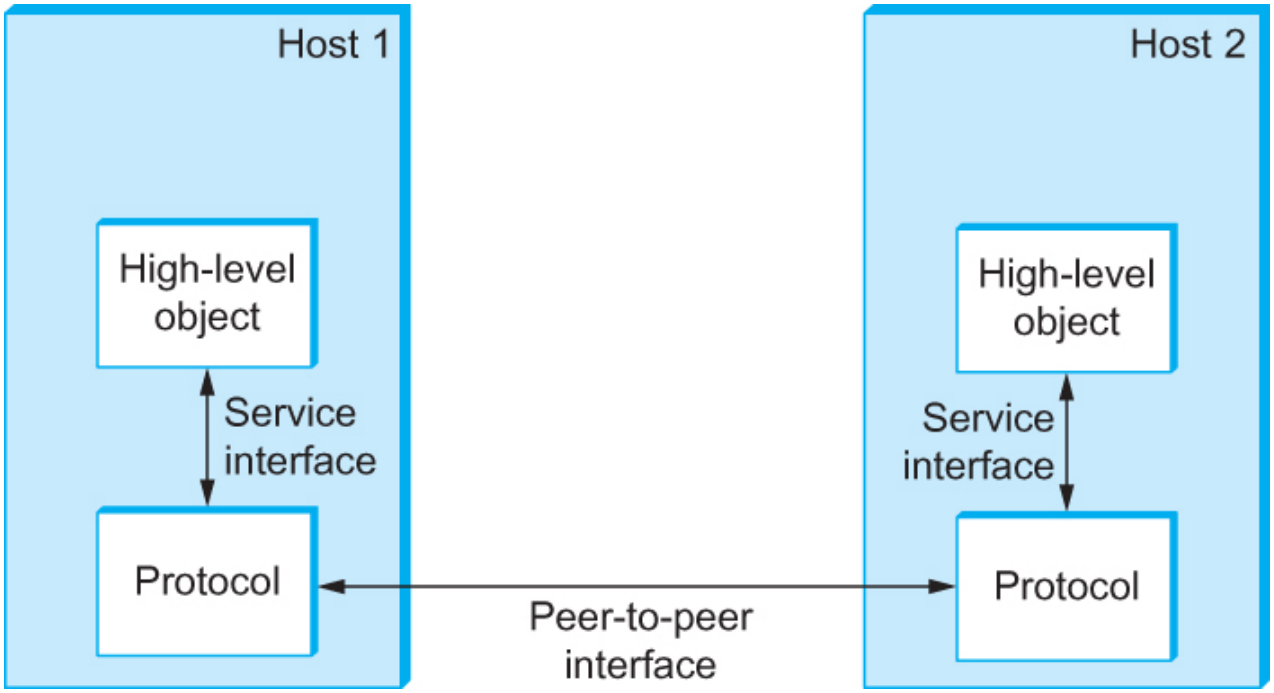
- Human protocols:
  - "What's the time?"
  - "I have a question"
  - Introductions

- … Specific messages sent
- … Specific actions taken when messages received, or other events

- Network protocols:
  - Machines rather than humans
  - All communication activity in Internet governed by protocols

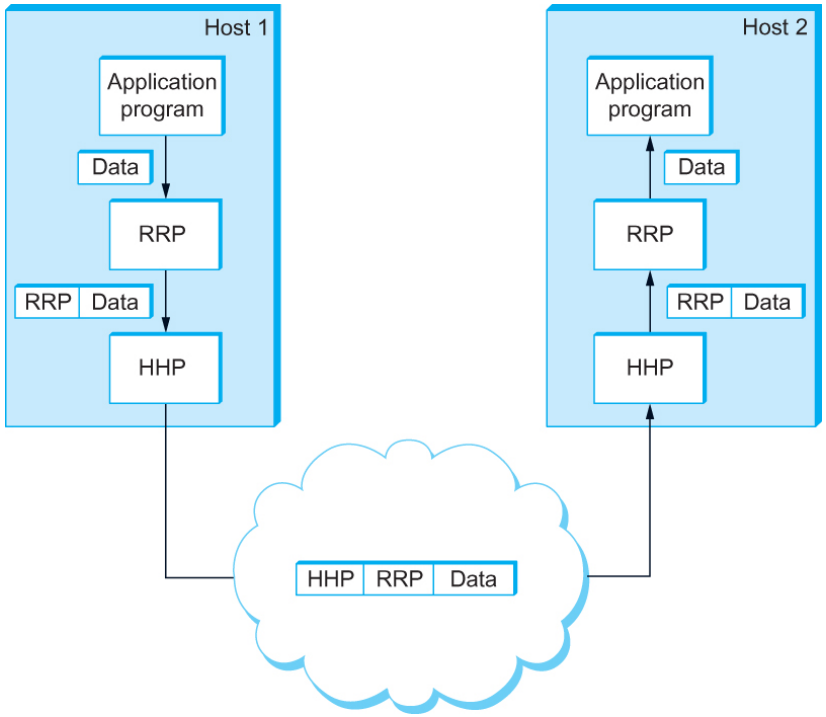
Protocols define format, order of messages sent and received among network entities, and actions taken on msg transmission, receipt

Interfaces

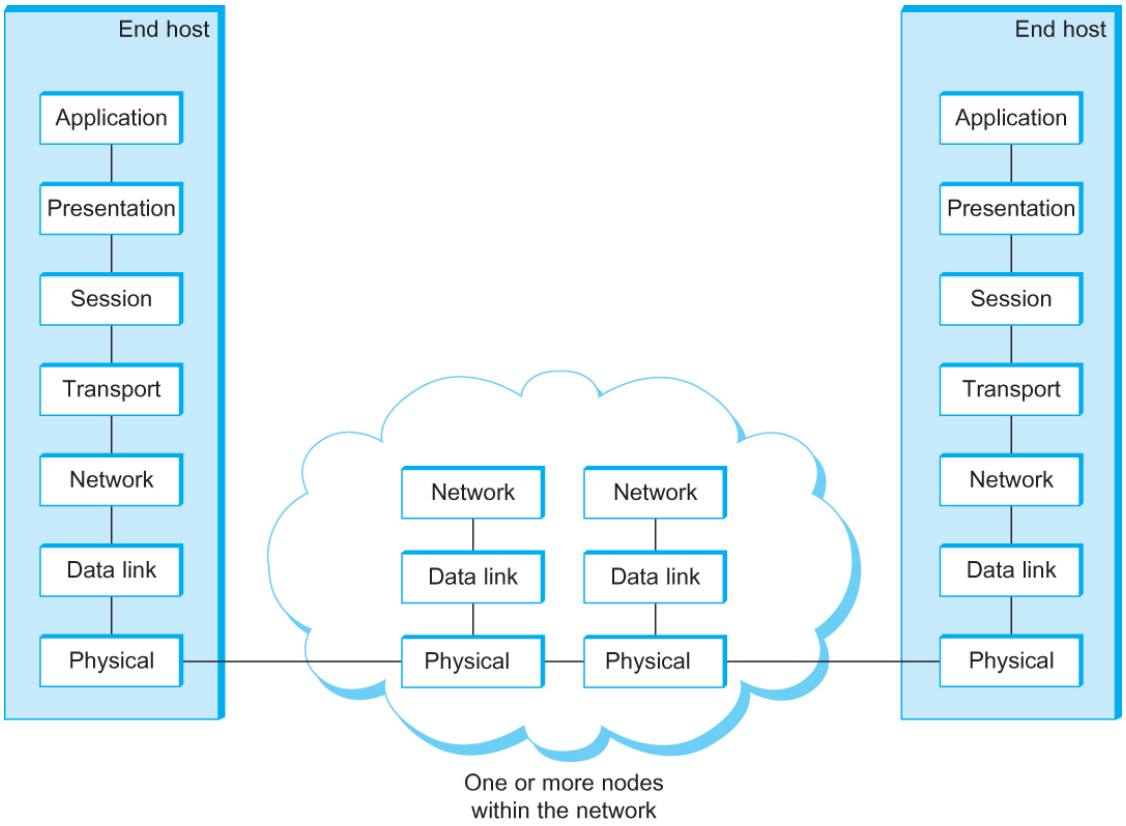


Service and Peer Interfaces

Encapsulation



High-level messages are encapsulated inside of low-level messages



The OSI 7-layer Model

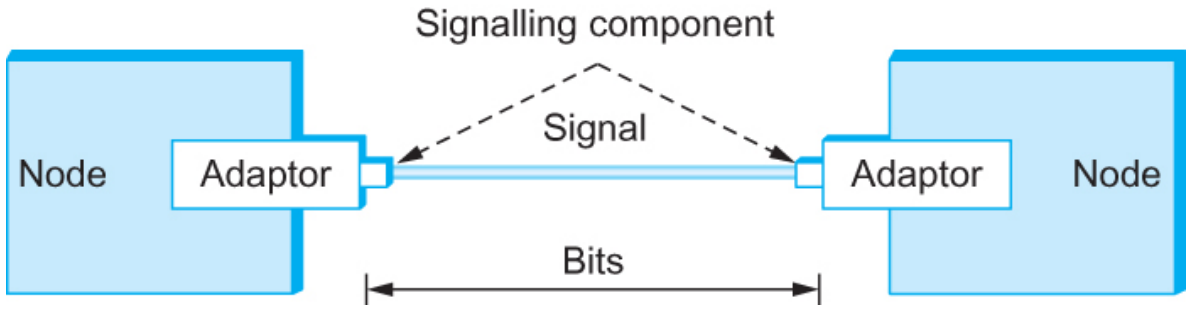
OSI – Open Systems Interconnection

Physical Layer

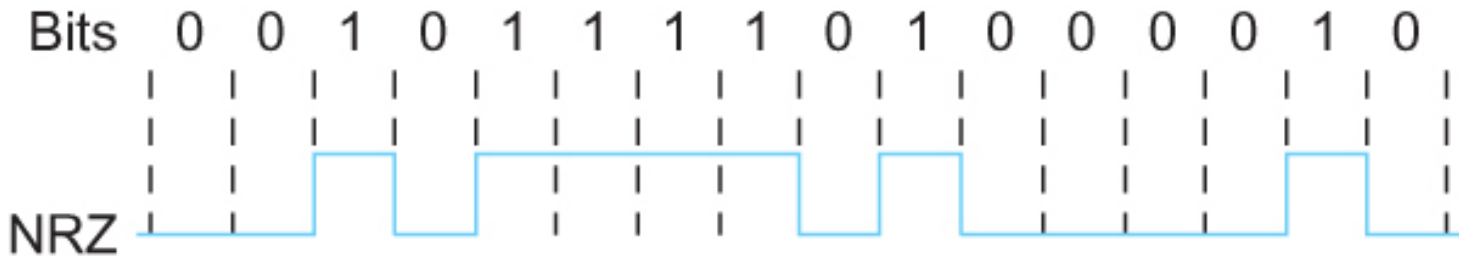
- Sends 1s and 0s across a physical/wireless link
- Ensure reliability
- Placing binary data on a link is called *encoding*
  - We will know that encoding exists, but not deal with it directly
- Provides framing
  - Gives metadata in order to better transmit (e.g. checksums, error detection/correction, number of bytes being transmitted in frame)

- Example: I want to send you 60 bytes of data. Here

Encoding: Most Basic



Signals travel between signaling components; bits flow between adaptors



NRZ encoding of a bit stream

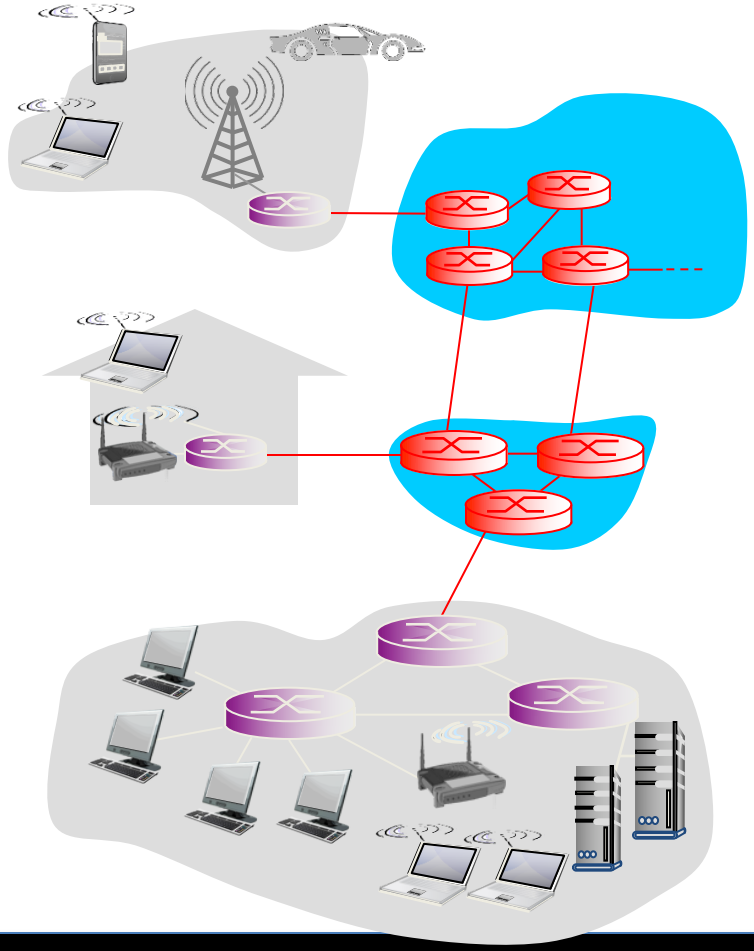
Data Link Layer

- Provides some basic frame distribution
- Introduces *switching* and *bridging*, but not yet routing
  - We won't get caught up with these
- Begins providing scalability
- Transfers are performed based on the physical address of the recipient
  - Media Access Control (MAC)

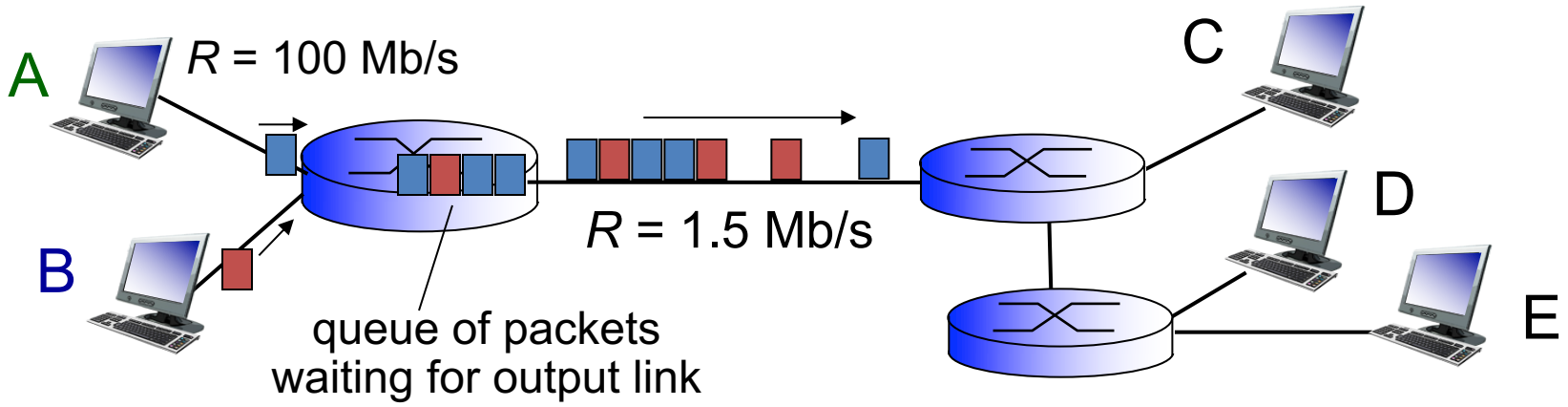
- Example: Hello! You are directly connected to Bob, or know exactly how to get to him. Please deliver these 60 bytes of data to him

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



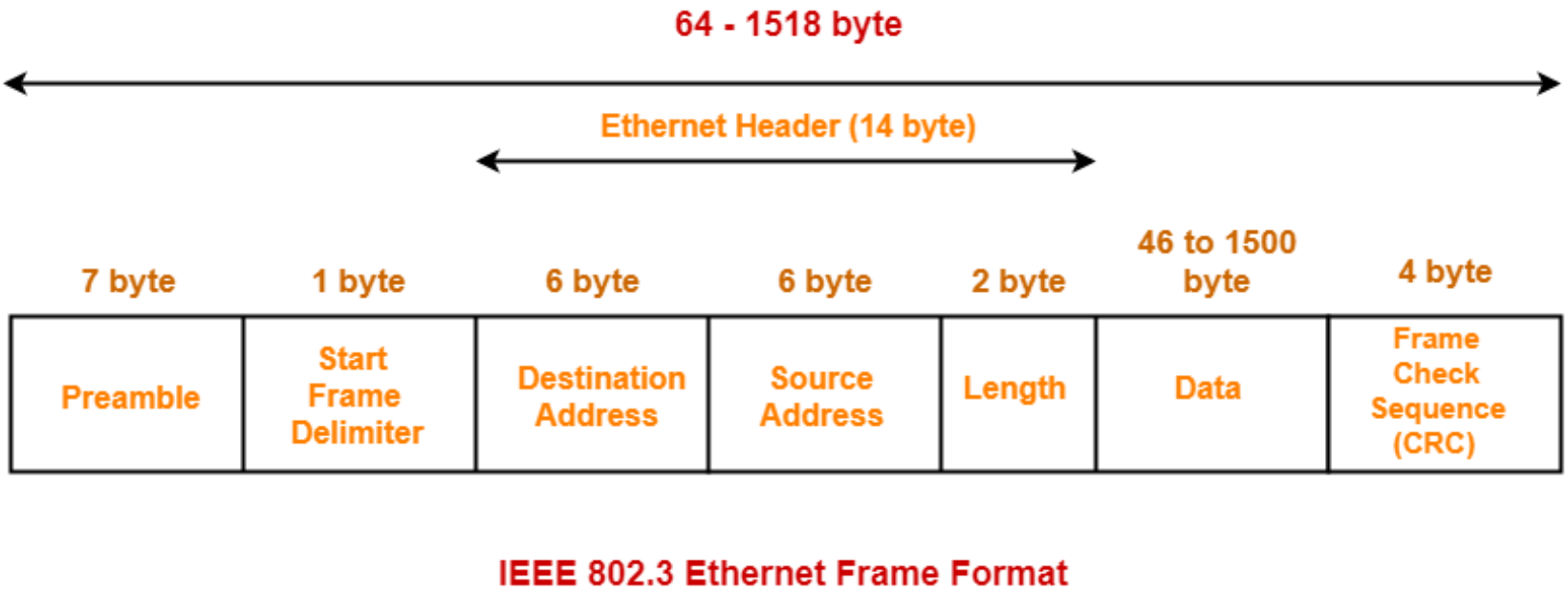
Packet Switching: Queueing Delay, Loss



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

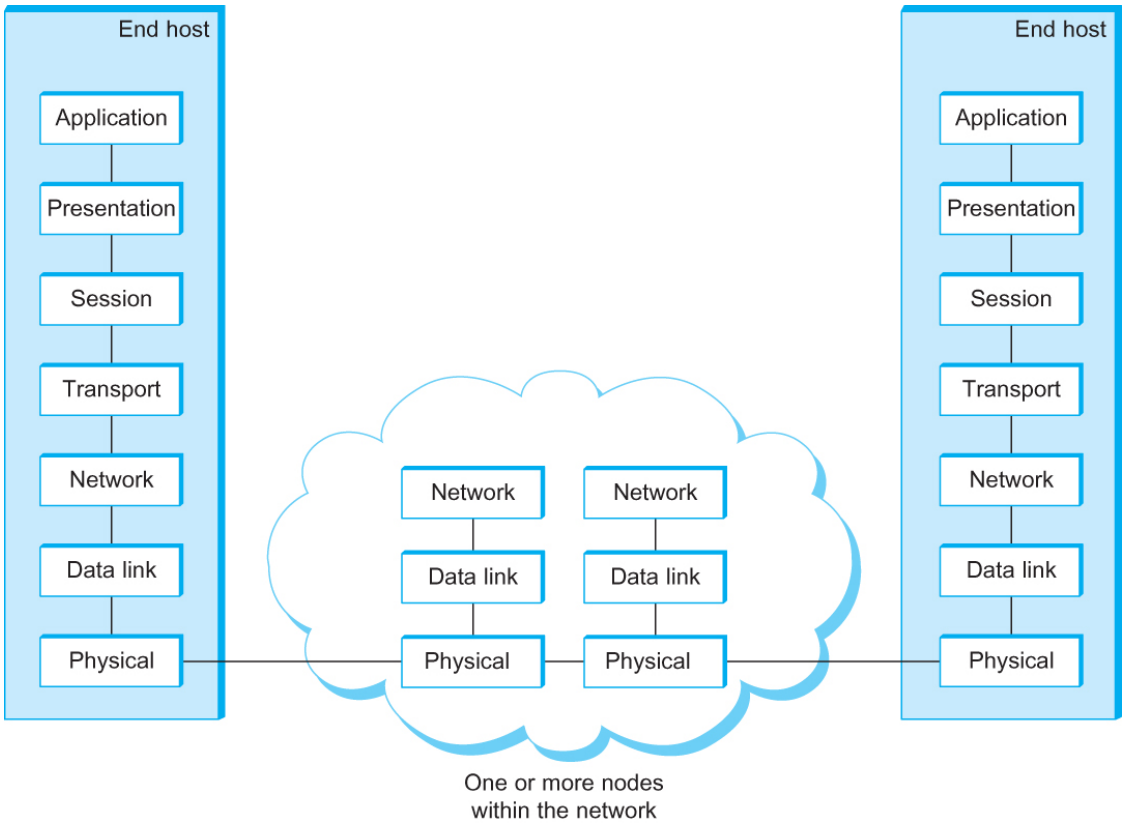
Current Transmission Example



IEEE 802.3 Ethernet Frame Format

Layer 3: Network Layer

*Slides adapted from Computer Networking: A top-Down Approach



The OSI 7-layer Model

OSI – Open Systems Interconnection

Network Layer

- This is the “Internet of Internets”
- In the second layer, we see
- Protocols in this layer: **IP**
- Programs such as *ping* and *traceroute* operate here
- Transfers can no longer be done with physical address
  - Must use IP address

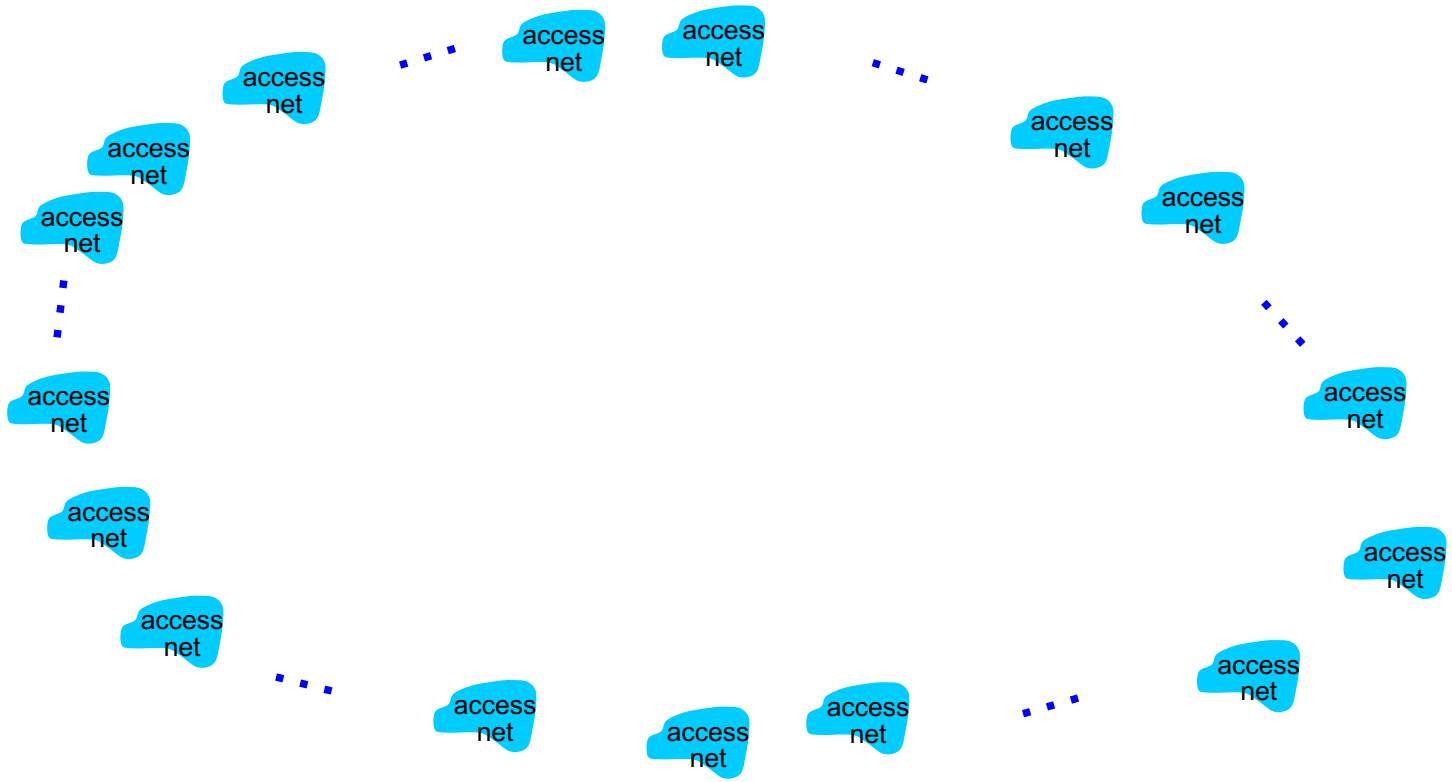
- Example: Here’s an address for Bob and 60B of data. Discuss with your neighbors to find a good route to Bob, then pass it along until it reaches him

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

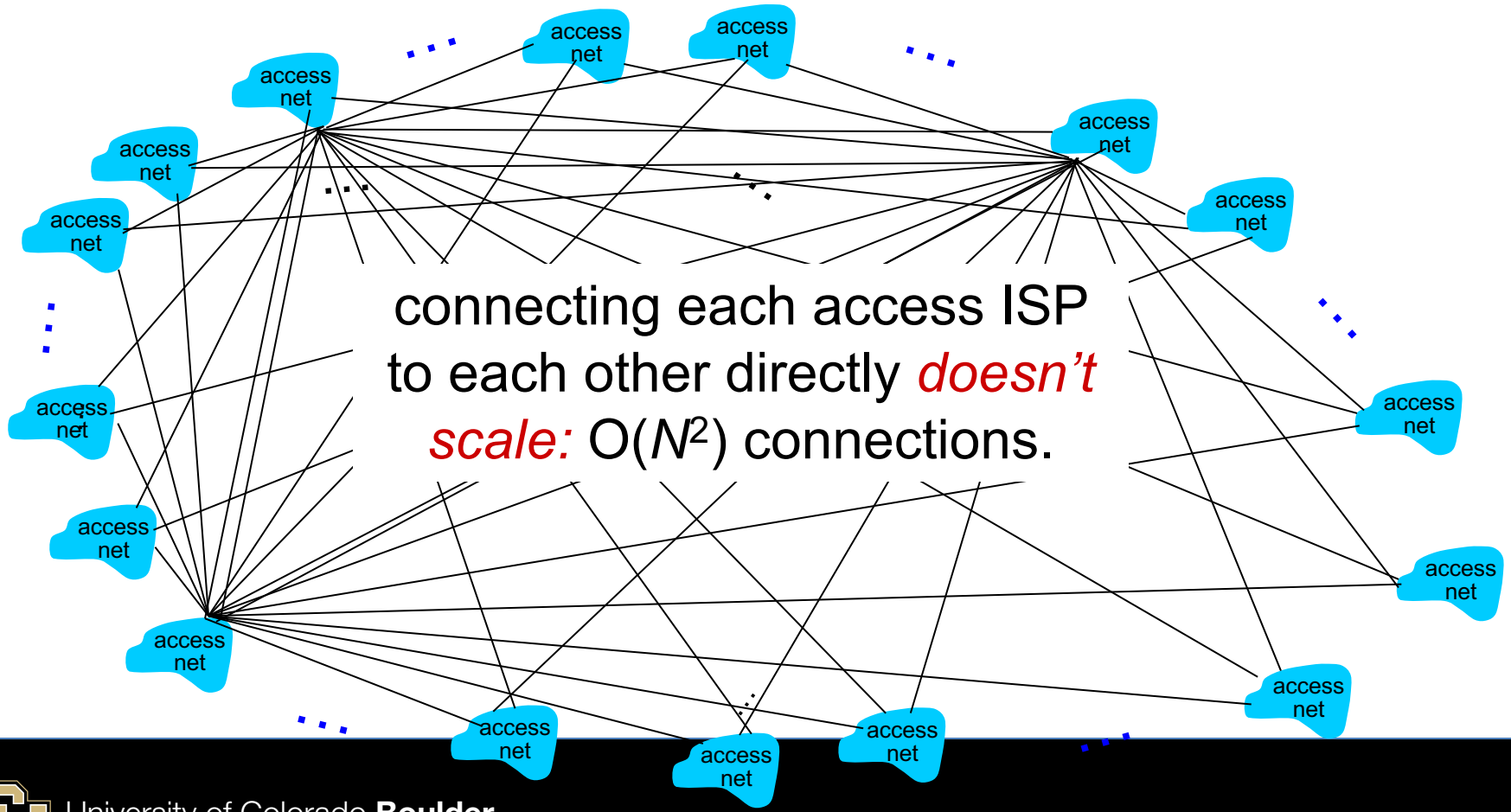
Internet Structure: Network of Networks

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



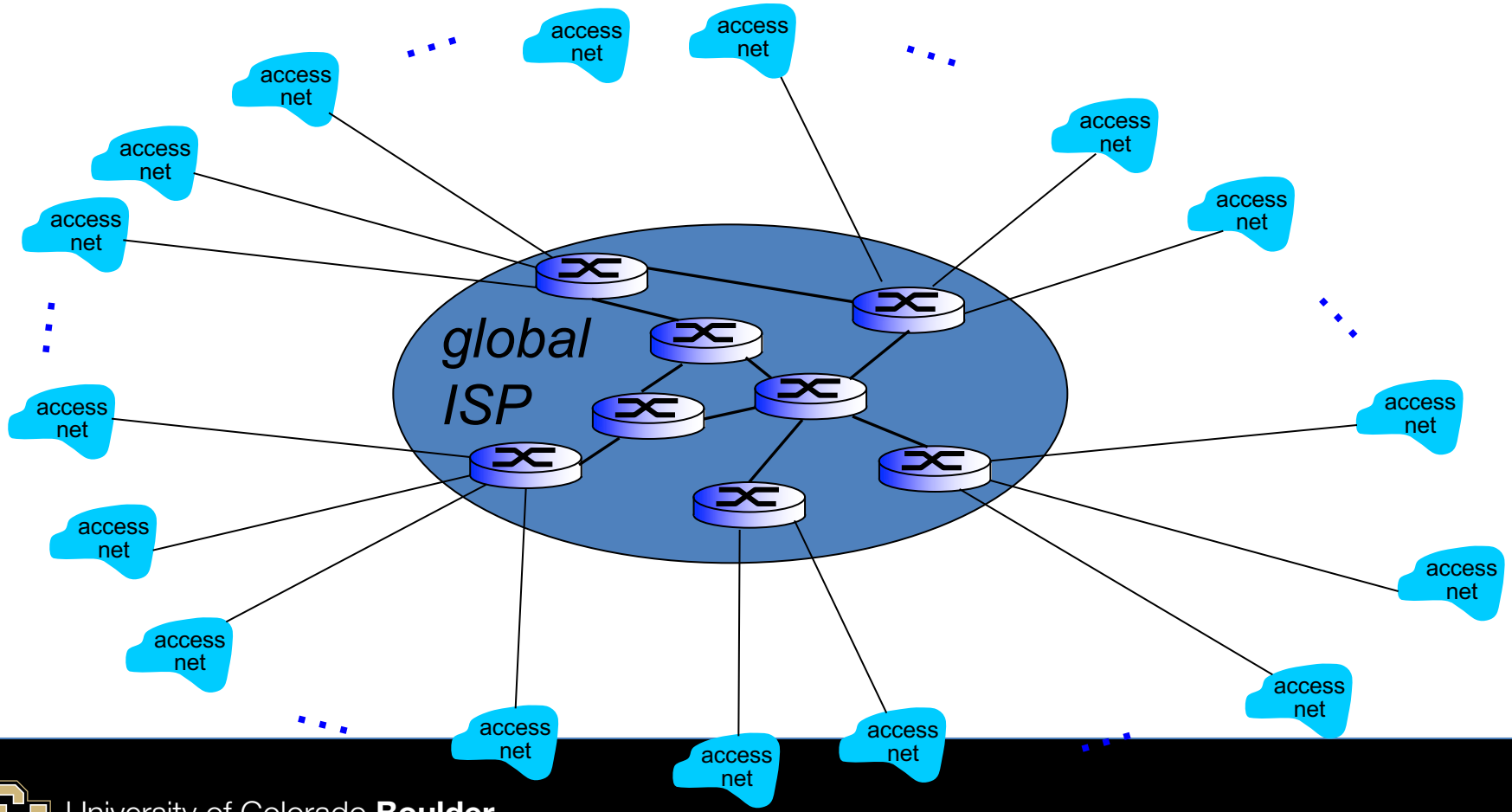
Internet Structure: Network of Networks

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



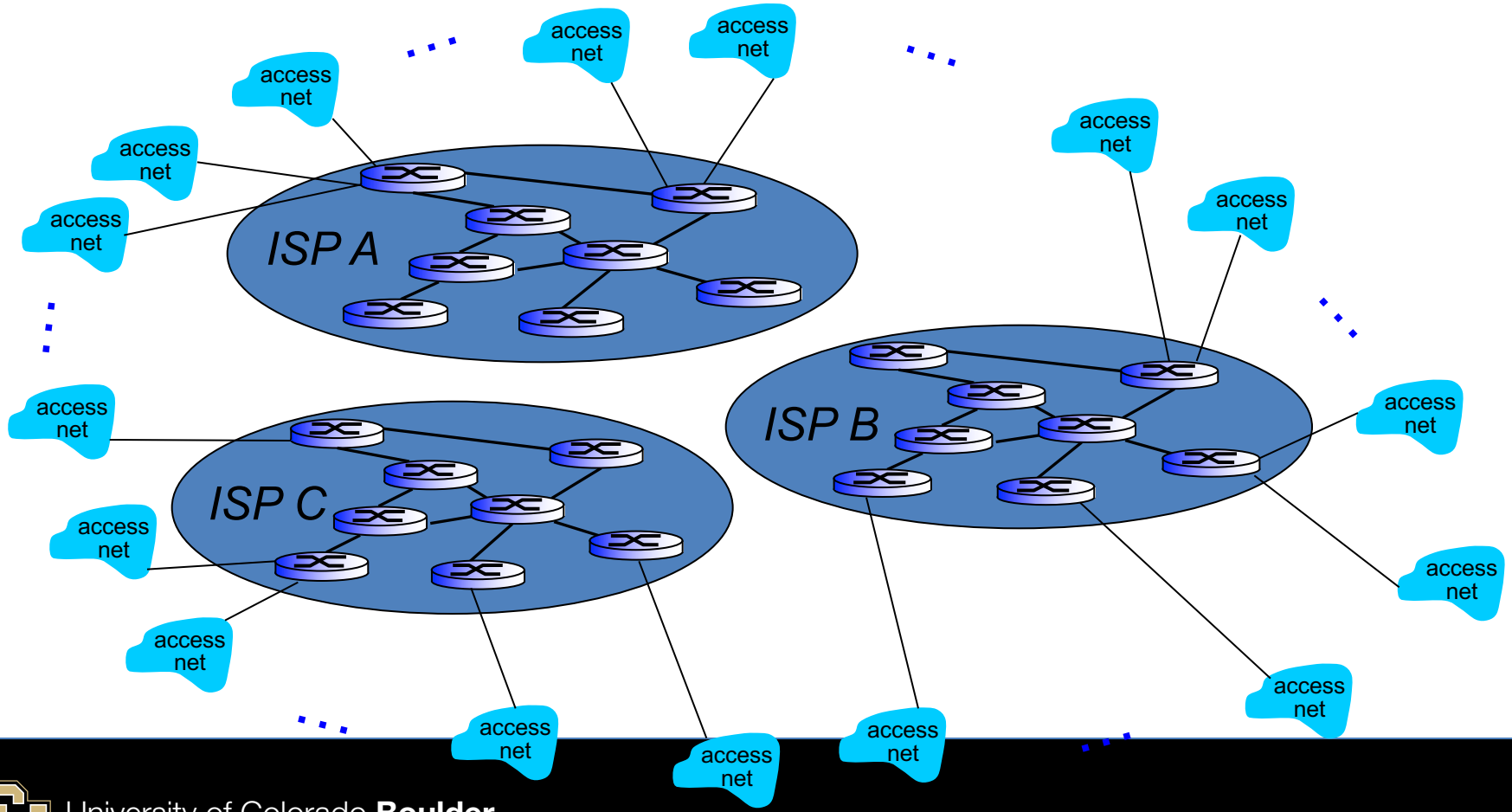
Internet Structure: Network of Networks

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



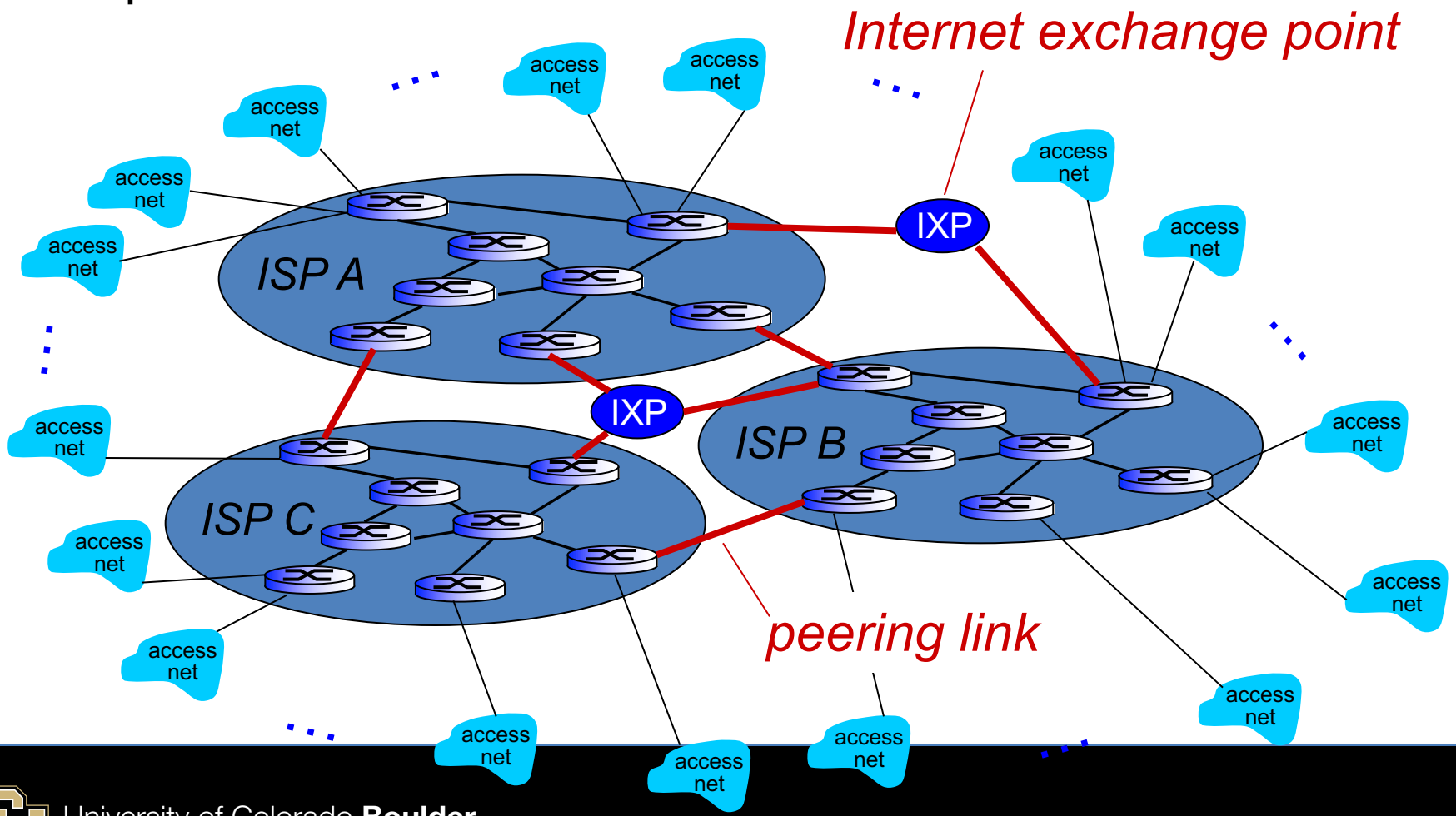
Internet Structure: Network of Networks

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



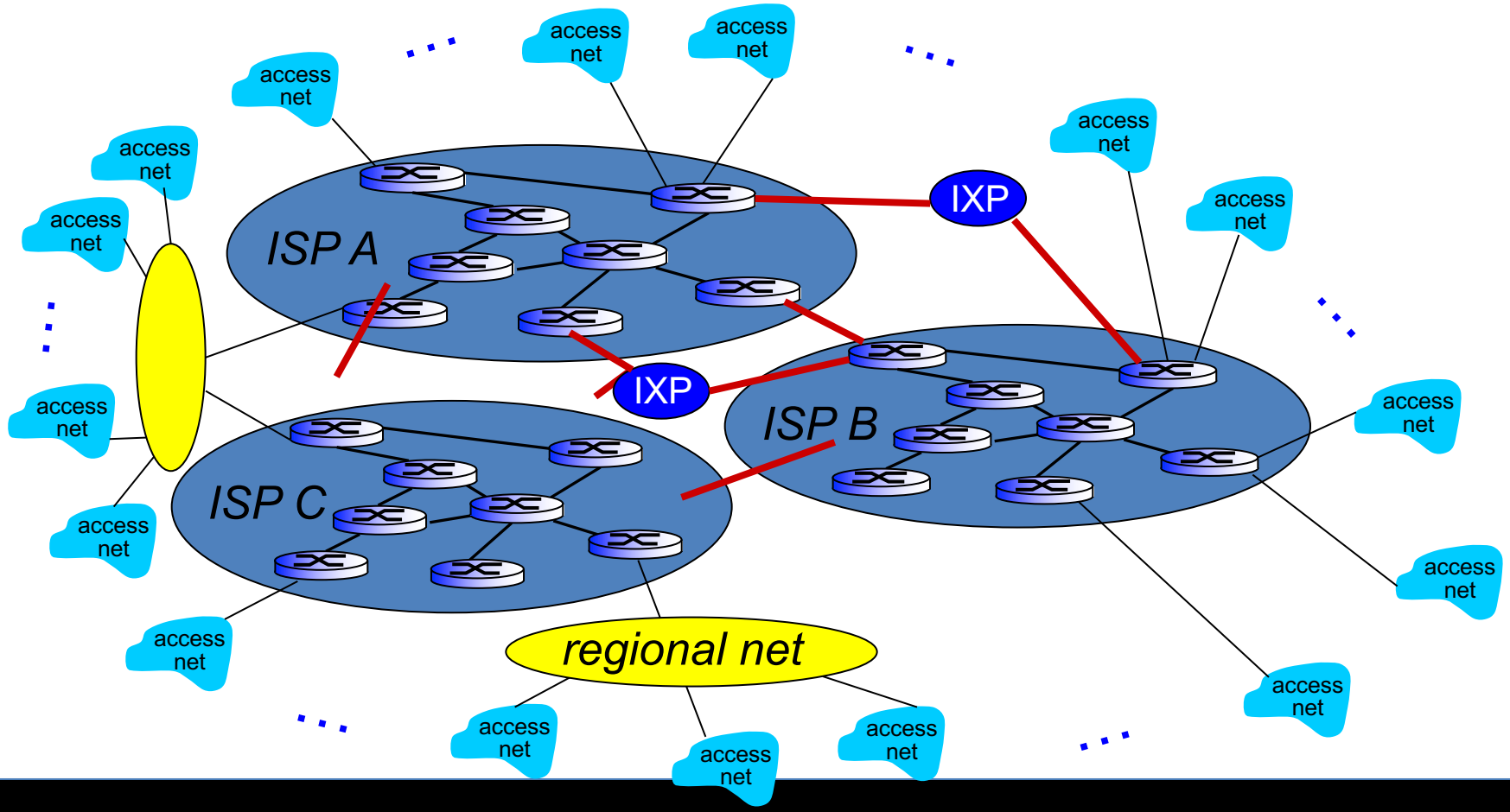
Internet Structure: Network of Networks

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

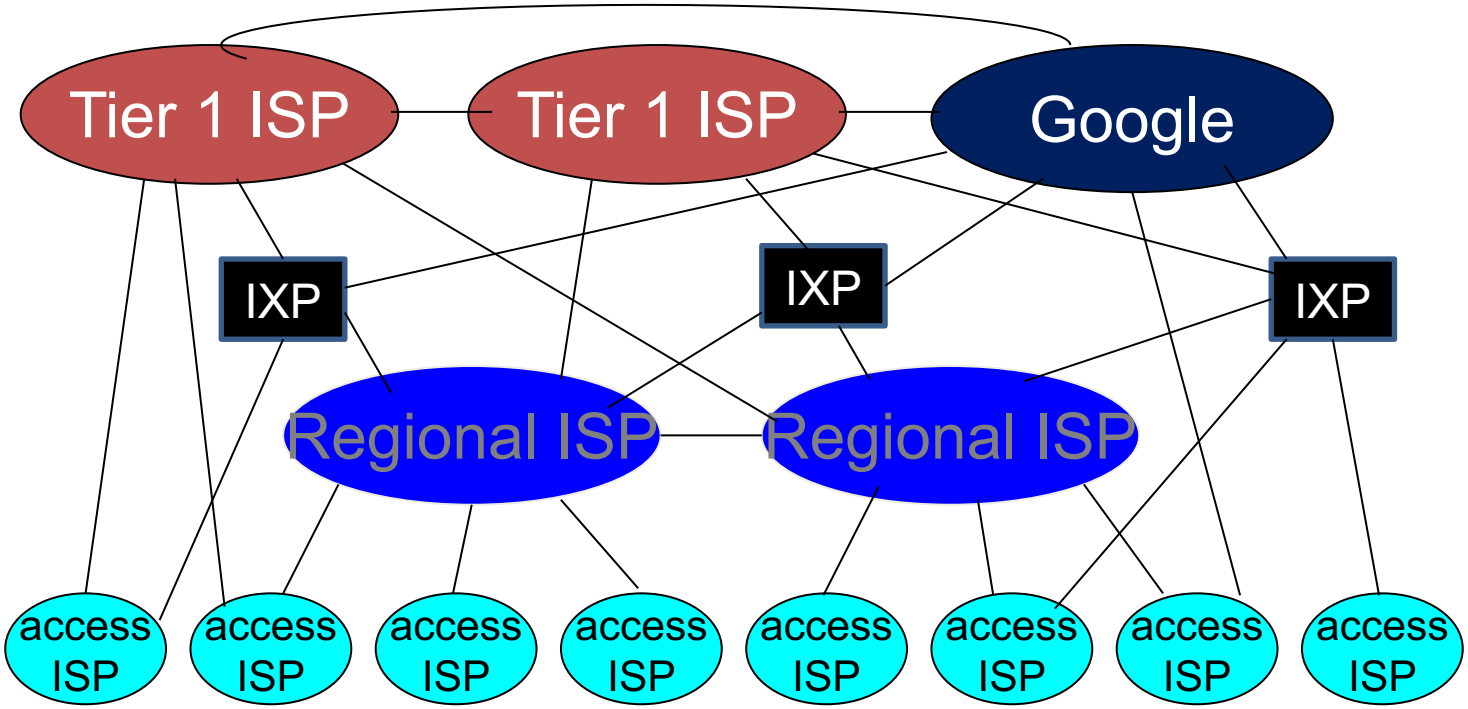


Internet Structure: Network of Networks

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



Internet Structure: Network of Networks



- At center: small # of well-connected large networks
  - "Tier-1" commercial ISPs (e.g., Level 3, Sprint, AT&T, NTT), national & international coverage

Interdomain Routing (BGP)

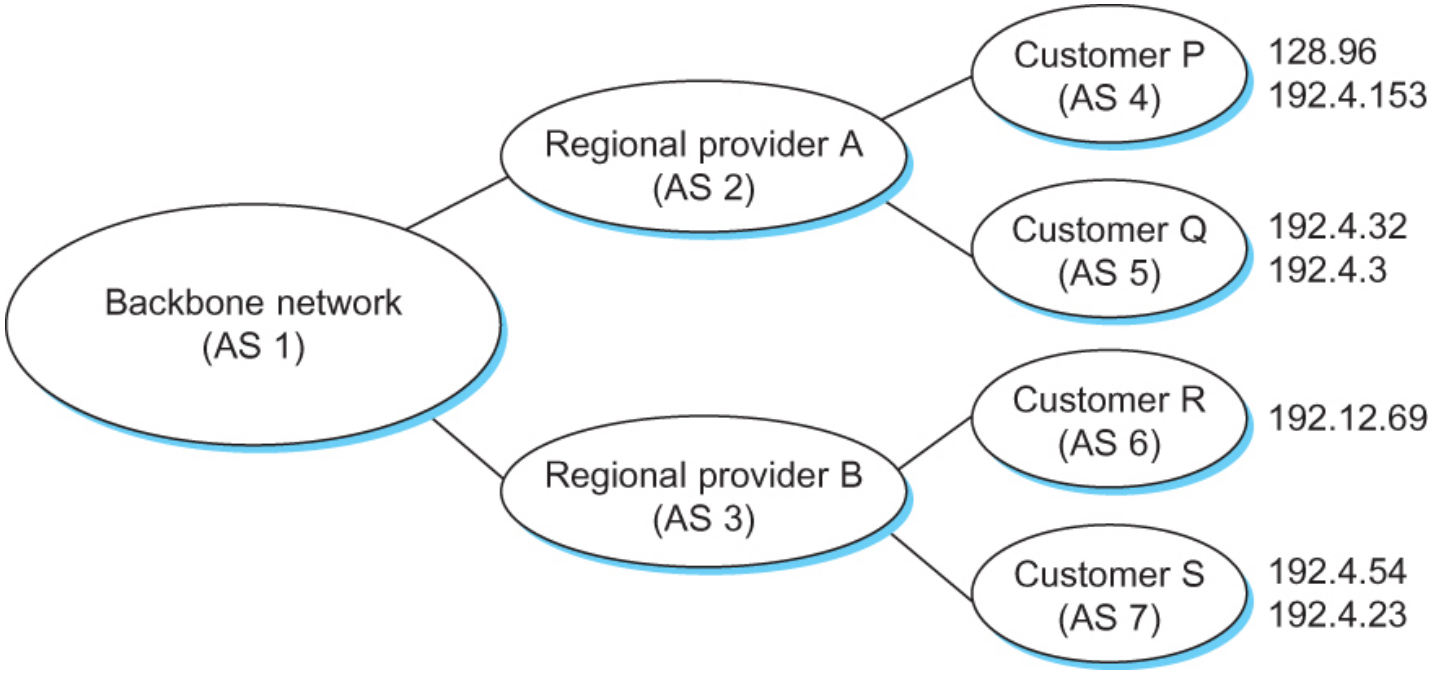
- Internet is organized as autonomous systems (AS) each of which is under the control of a single administrative entity
- Autonomous System (AS)
  - Corresponds to an administrative domain
  - Examples: University, company, backbone network
  - “Here’s some info, do what you want”
- A corporation’s internal network might be a single AS, as may the network of a single Internet service provider

Border Gateway Protocol (BGP)

Each AS has:

- One BGP speaker that advertises:
  - Local networks
  - Other reachable networks (transit AS only)
  - Gives path information
- In addition to the BGP speakers, the AS has one or more border “gateways” which need not be the same as the speakers
- The border gateways are the routers through which packets enter and leave the AS

BGP Example



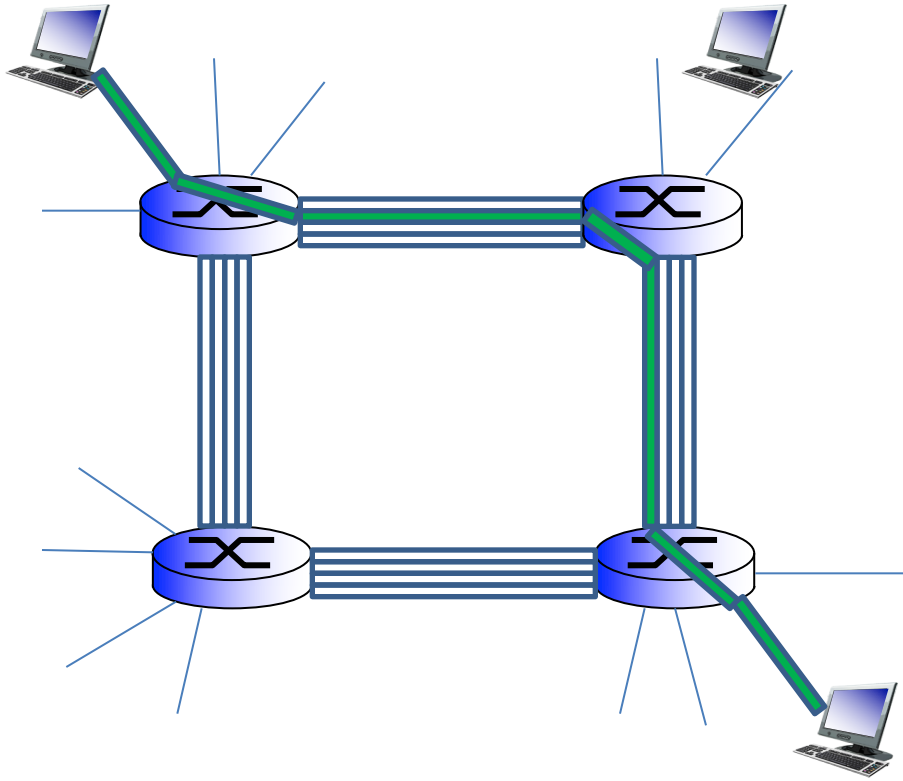
Example of a network running BGP

BGP Example

- Speaker for AS 2 advertises reachability to P and Q
  - Network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS 2.
- Speaker for backbone network then advertises
  - Networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path <AS 1, AS 2>.
- Speaker can also cancel previously advertised paths

Alternative core: 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

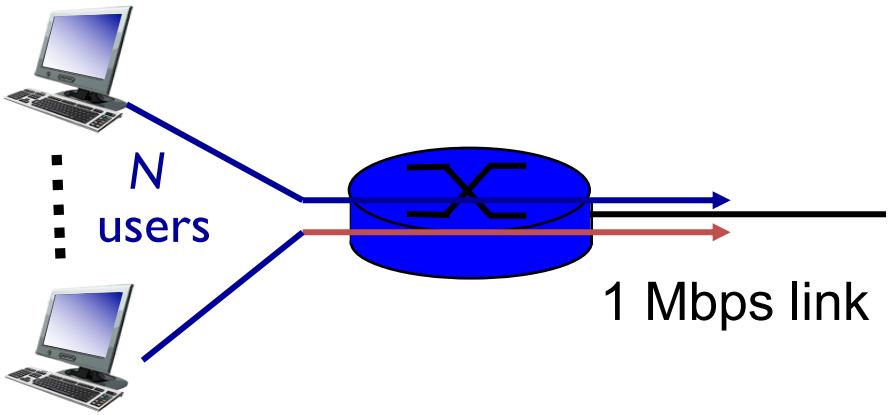


Packet Switching Vs Circuit Switching

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 Vs Circuit Switching

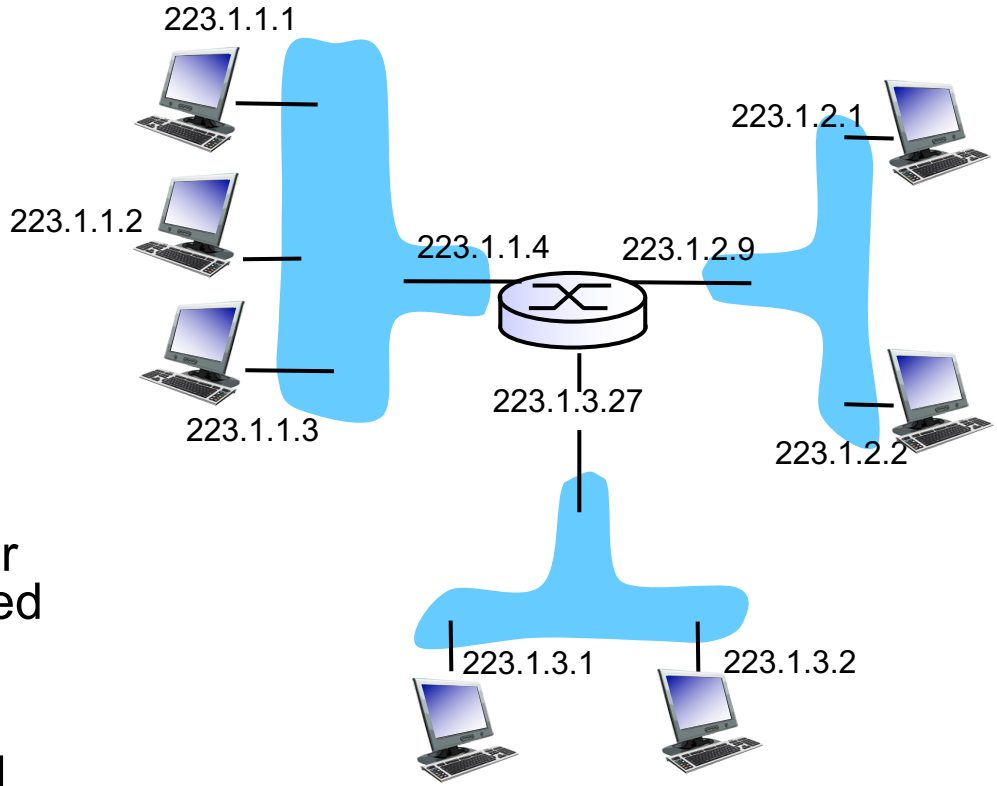
Is packet switching a "slam dunk winner?"

- Great for data that comes in bursts
  - 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 (chapter 7)

Question: Can you think of human analogies of reserved resources (circuit switching) versus on-demand allocation (packet-switching)?

IP Addressing: Introduction

- IP address: 32-bit identifier for host, router interface
- Interface: connection between host/router and physical link
  - Router's typically have multiple interfaces
  - Host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- IP addresses associated with each interface

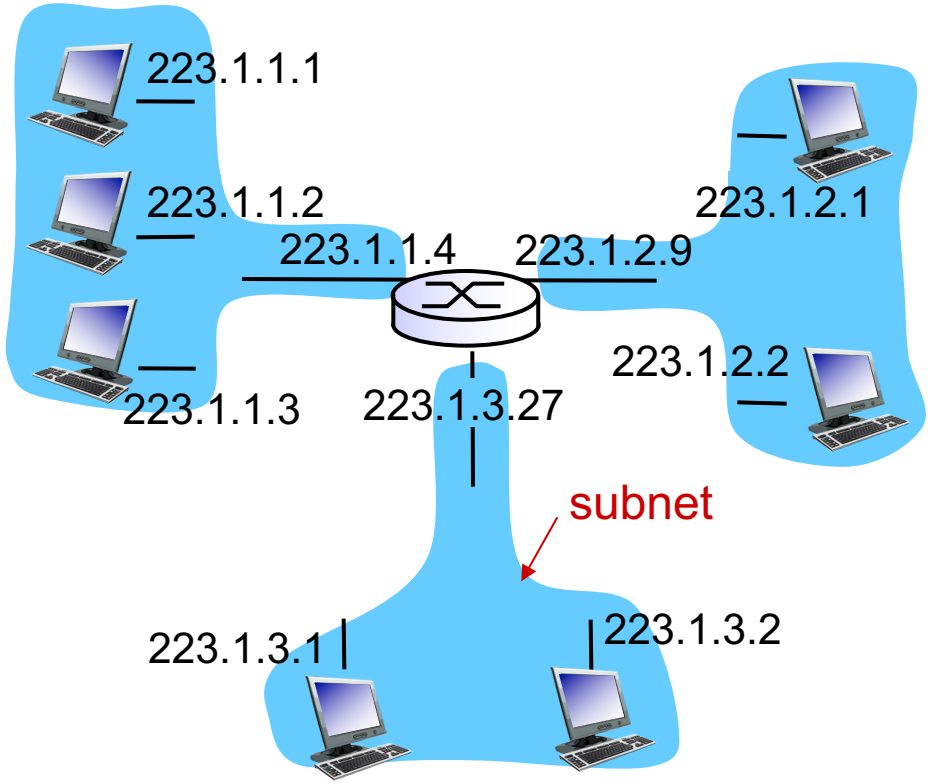


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Subnets

- IP address:
  - Subnet part - high order bits
  - Host part - low order bits
- What's a subnet ?
  - Device interfaces with same subnet part of IP address
  - Can physically reach each other without intervening router

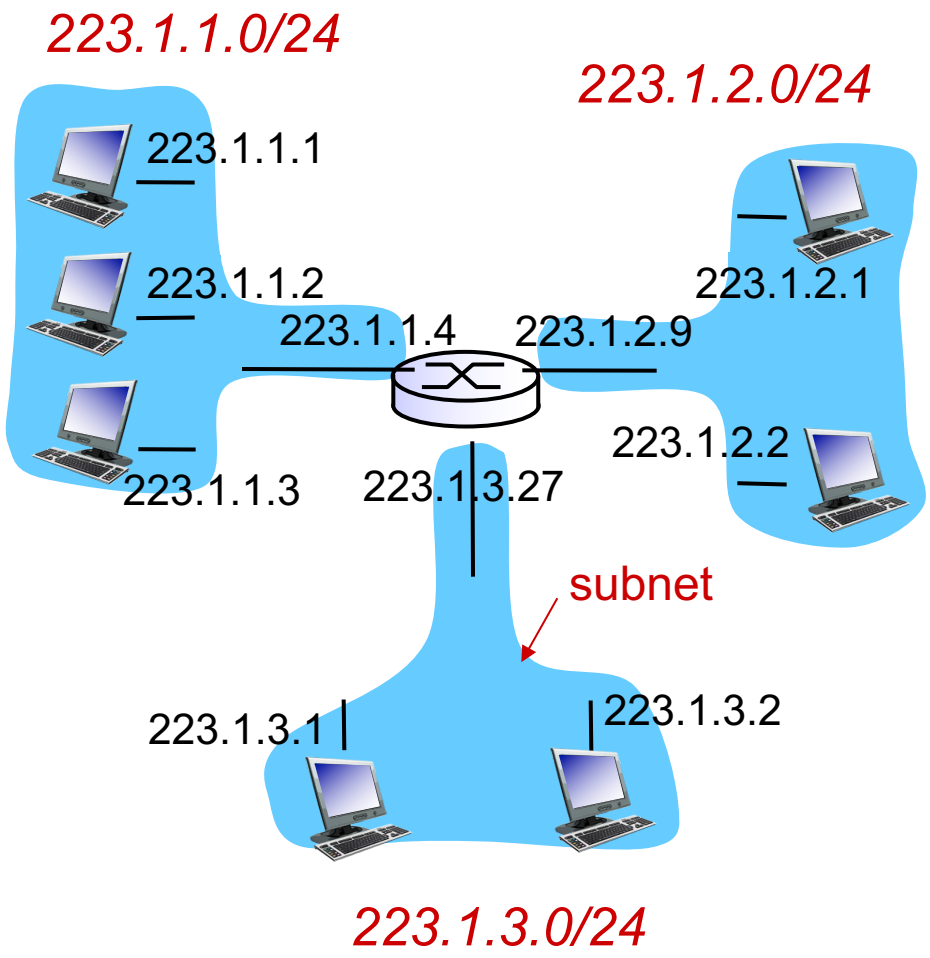


network consisting of 3 subnets

Subnets

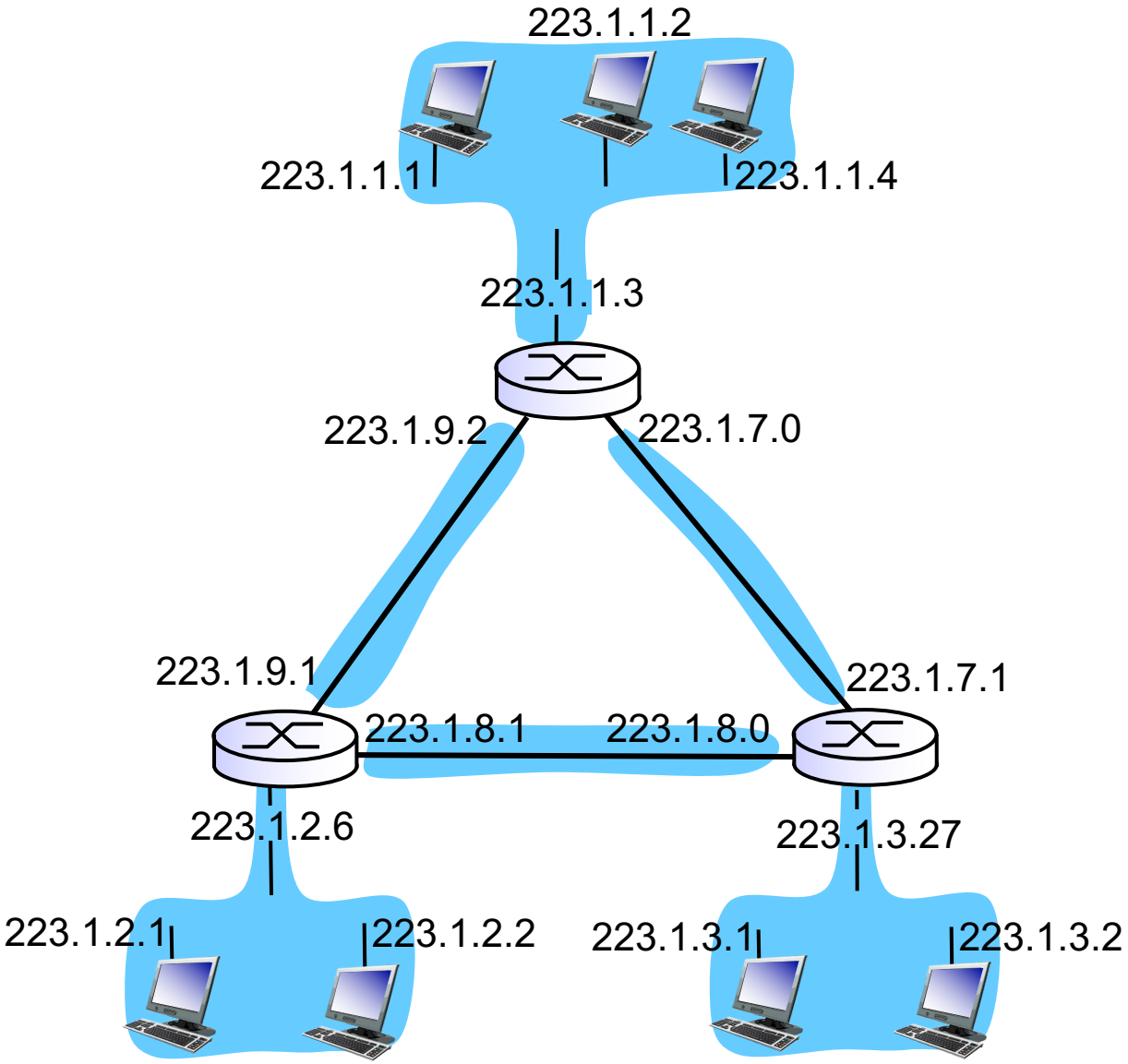
Recipe

- To determine the subnets, detach each interface from its host or router, creating islands of isolated networks
  - Each isolated network is called a subnet



subnet mask: /24

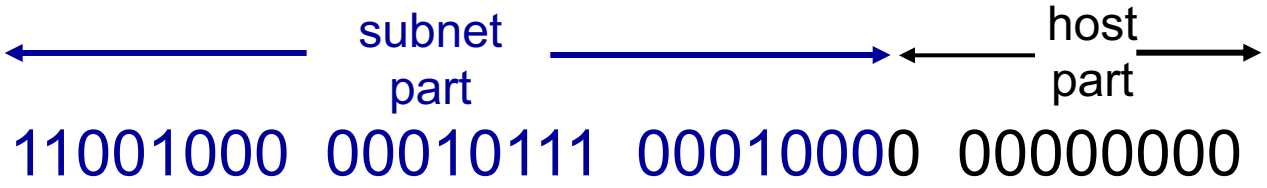
Subnets


223.1.1.2
223.1.1.1
223.1.1.4
223.1.1.3
223.1.9.2
223.1.7.0
223.1.9.1
223.1.7.1
223.1.8.1
223.1.8.0
223.1.2.6
223.1.3.27
223.1.2.1
223.1.2.2
223.1.3.1
223.1.3.2

IPv4 Addressing: CIDR

CIDR: Classless Inter-Domain Routing

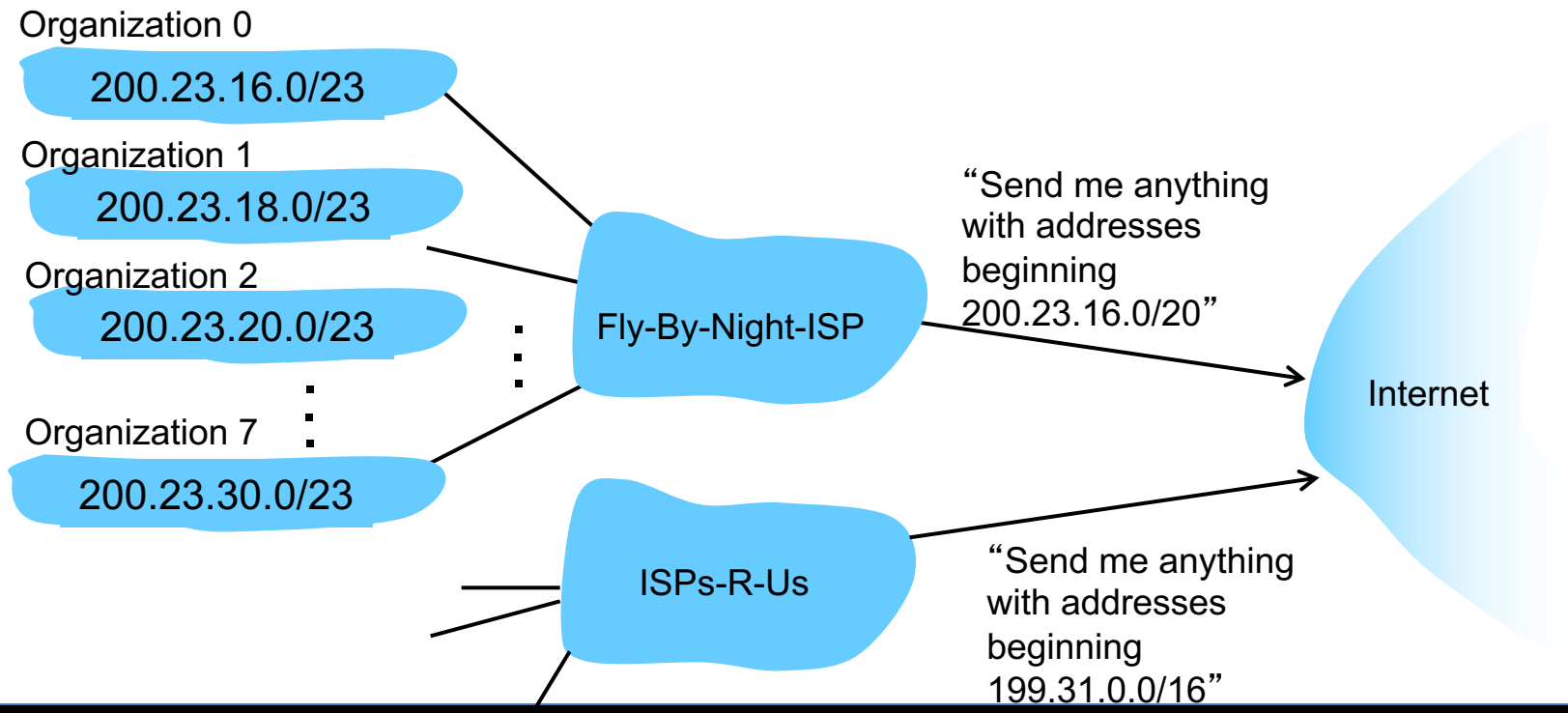
- Subnet portion of address of arbitrary length
- Address format: a.b.c.d/x, where x is # bits in subnet portion of address



200.23.16.0/23

Hierarchical Addressing: Route Aggregation

Hierarchical addressing allows efficient advertisement of routing information:

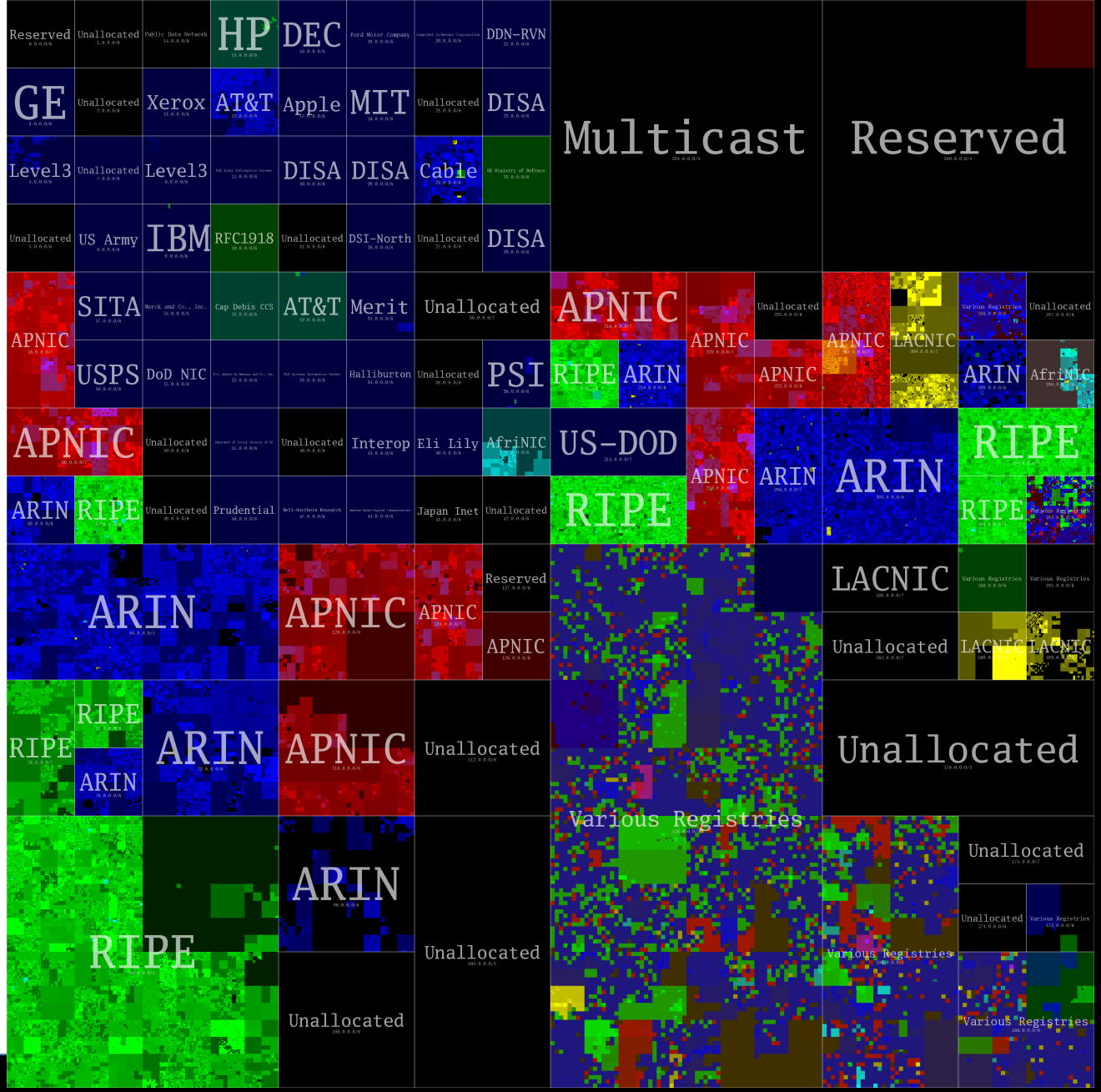


IP Addressing: The Authority

- **Q**: How does an ISP get block of addresses?
- **A**: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/
  - Allocates addresses
  - Manages DNS
  - Assigns domain names, resolves disputes

IPv4 vs IPv6

- The original internet only had a few hosts
- The address space was large at the time
- There are no more IPv4 addresses for use
- IPv6 was created to solve this problem
  - And it does!
  - # 32-bit IPv4 addresses: 4,294,967,692
  - # 128-bit IPv6 addresses: 340,282,366,920,938,463,463,374,607,431,768,211,456
- Problems?



RIRs IPv4 Whois Map

October 2007

/8 /16 /24 /32

- AFRINIC
- APNIC
- AUNIC
- JPNIC
- TWNIC
- ARIN
- LACNIC
- RIPE

This map shows the characteristics of IPv4 "whois" data from the Regional Internet Registries.

Each pixel in the full-size image represents a single /24 network containing up to 256 hosts. The color of each pixel corresponds to the RIR(s) where the address space is listed.

Areas are shaded with varying levels of transparency. Larger blocks of address space are more transparent, which also makes them darker. Thus, varying shades of a color indicate different-sized blocks of addresses appearing in the whois data.

Furthermore, some parts of the address space are listed in multiple RIRs, for various reasons. When this happens, new colors may be created. This is especially evident in the areas labeled "Various Registries" where you can see large areas of brown, created by combining red (ARIN) and green (RIPE) together.

caida

The Measurement Factory

University of Colorado **Boulder**

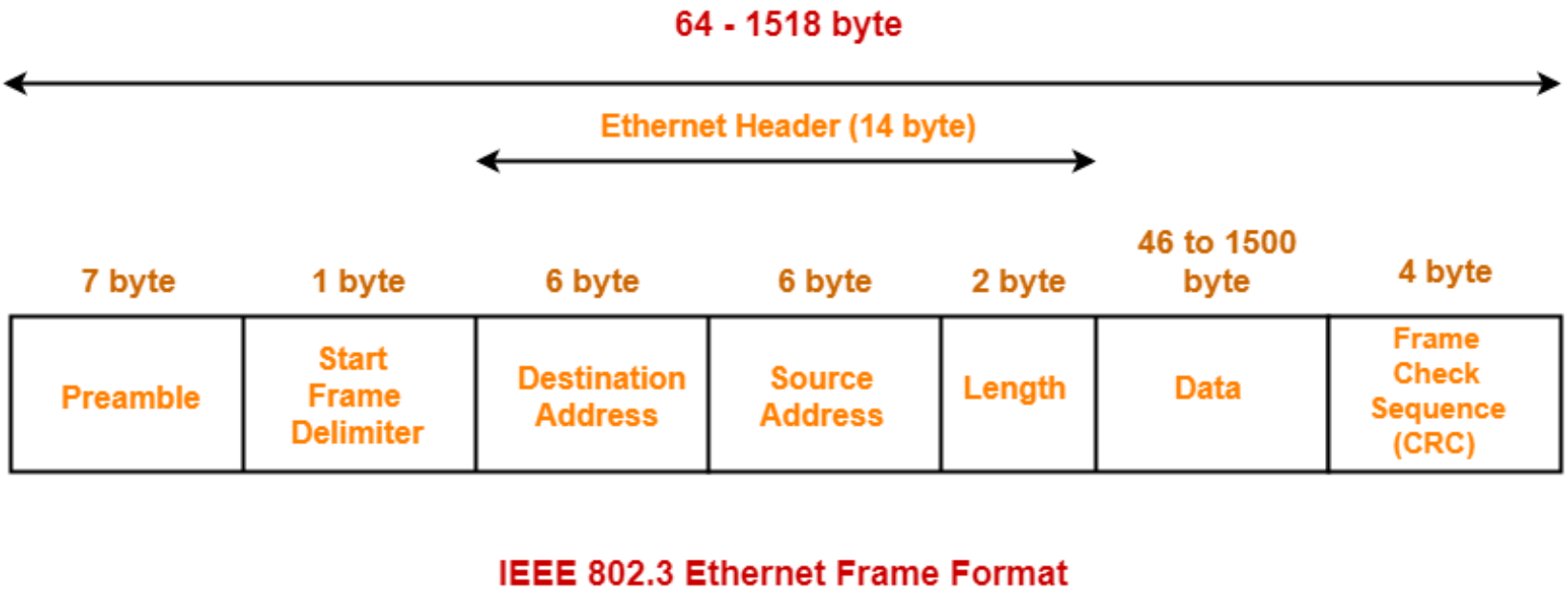
IPv4 vs IPv6

- The original internet only had a few hosts
- The address space was large at the time
- There are no more IPv4 addresses for use
- IPv6 was created to solve this problem
  - And it does!
  - # 32-bit IPv4 addresses: 4,294,967,692
  - # 128-bit IPv6 addresses: 340,282,366,920,938,463,463,374,607,431,768,211,456
- Problems?
  - No “launch day”
  - People aren’t incentivized to switch since NATs came around

Network Address Translators (NAT)

- A patch for lack of IP space
- Each residence has 20 devices, all have same IP
- Discussed more in recitation
- But first...

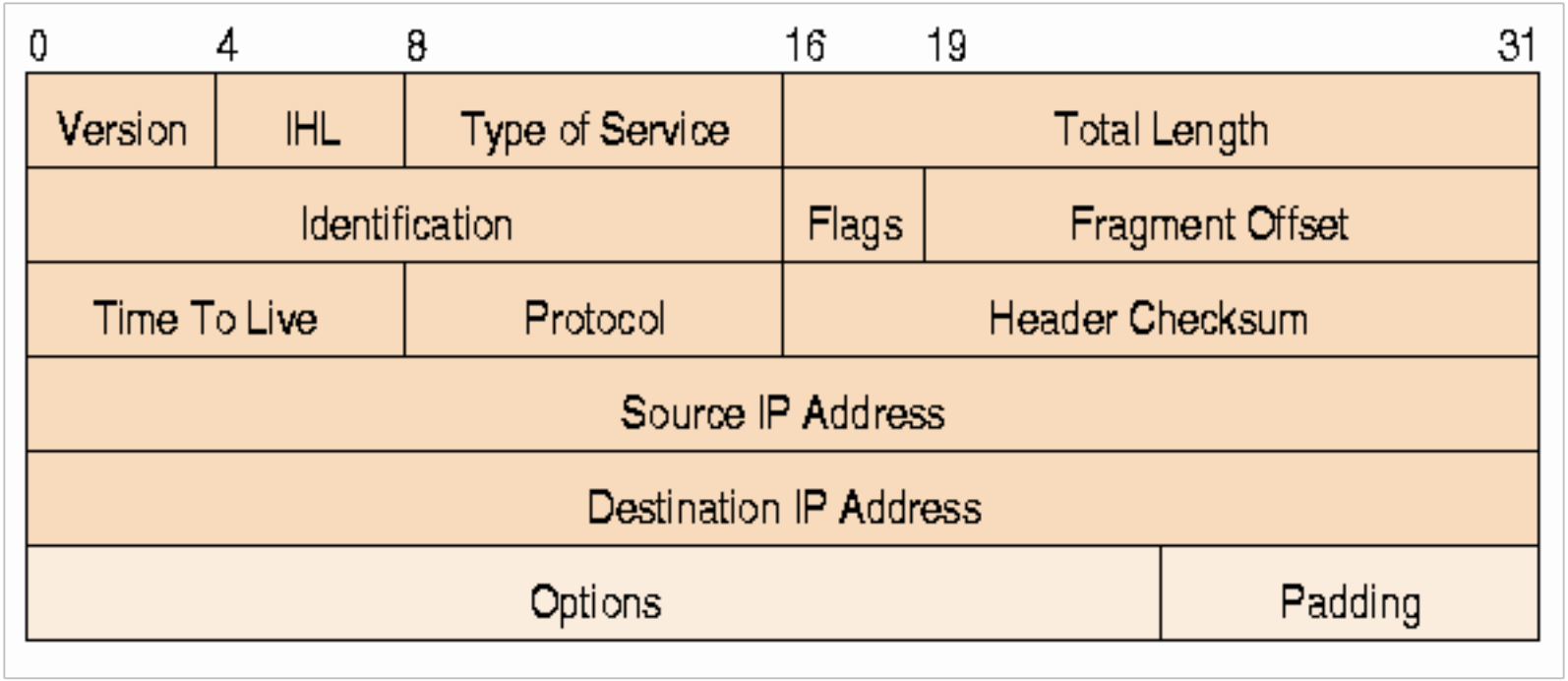
Previous Packet Structure



IEEE 802.3 Ethernet Frame Format

*Source: https://www.gatevidyalay.com

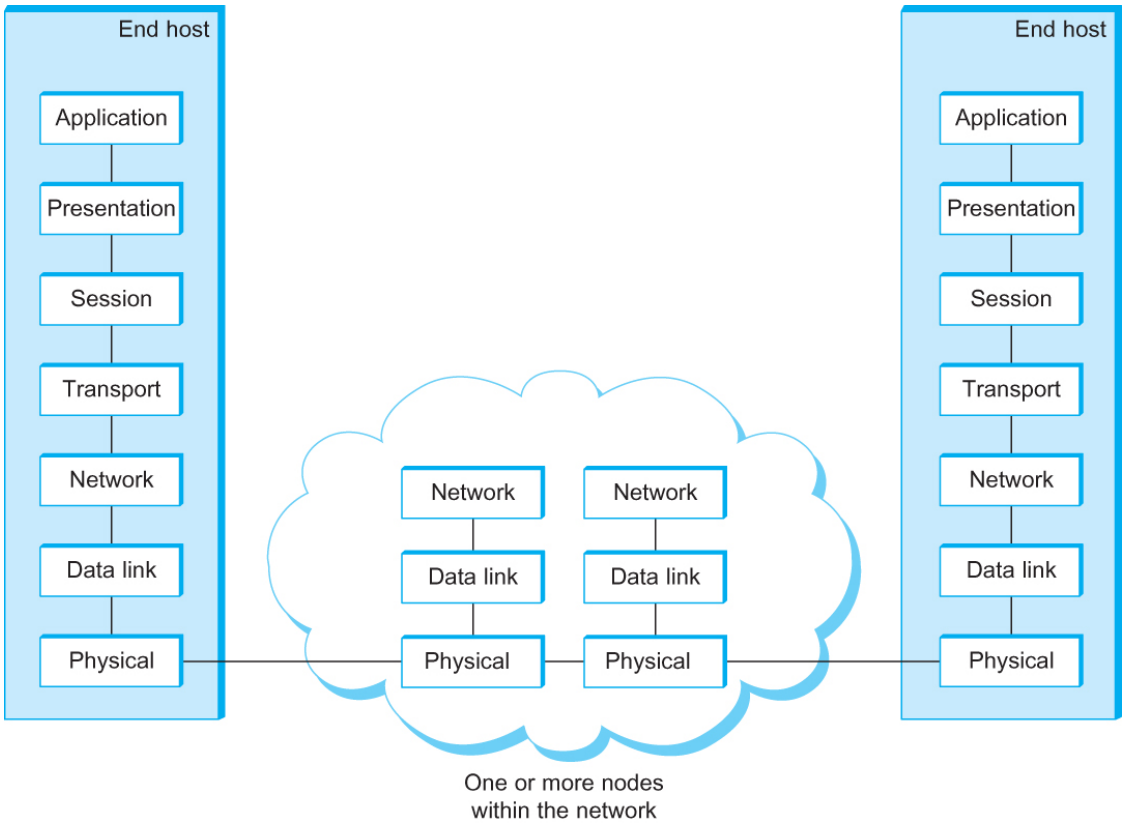
Current Packet Structure



*Source: https://freesoft.org

Layer 4: Transport Layer

*Slides adapted from Computer Networking: A top-Down Approach



The OSI 7-layer Model

OSI – Open Systems Interconnection

So Far...

- We can send packets anywhere
- We can manage and scale the internet to hold billions of hosts
- We can guarantee a route from one person to another
- Sounds great!

So Far...

- We can send packets anywhere
- We can manage and scale the internet to hold billions of hosts
- We can guarantee a route from one person to another
- Sounds great!
- Problems?

So Far...

- We can send packets anywhere
- We can manage and scale the internet to hold billions of hosts
- We can guarantee a route from one person to another
- Sounds great!
- Problems?
  - Our connections are unreliable
  - We can only direct info to a machine, not to different processes on that machine!

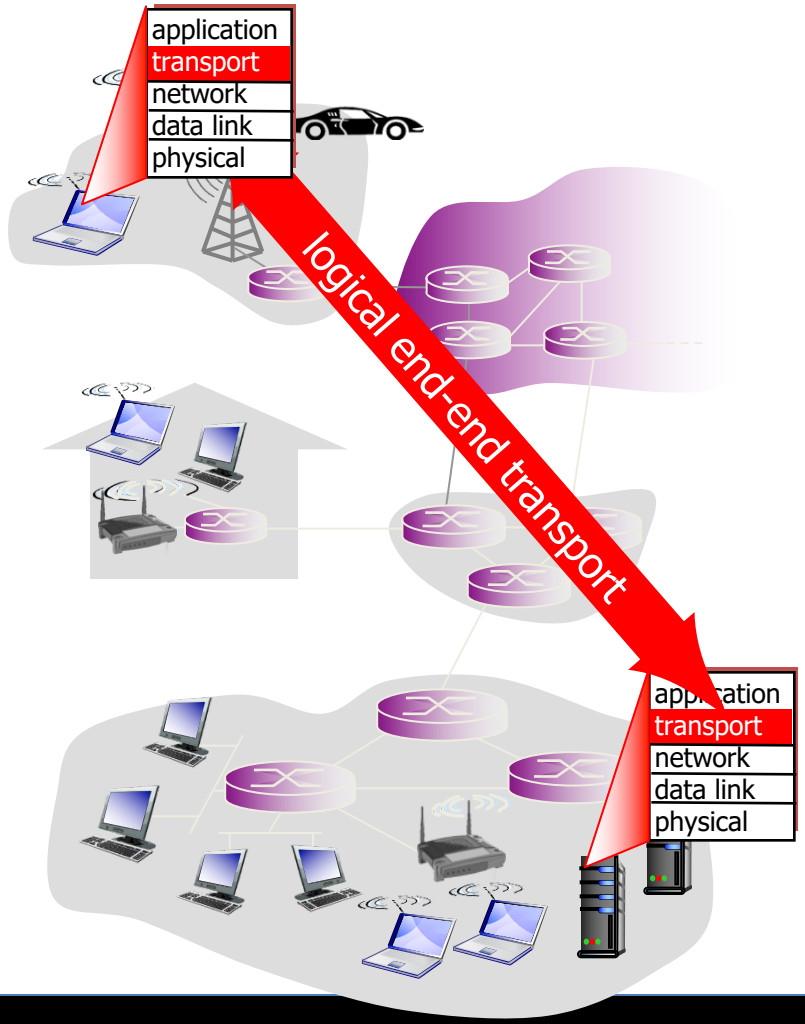
Transport Layer

- Uses IP addressing to create services
  - Rather than simply delivering to a machine, we can run multiple services per machine
- Introduces the concept of ports for “process-to-process” communication
- Protocols in this layer: **TCP, UDP**

- Example: Bob has a service on port 80. Use the internet’s knowledge to deliver my 60B to him and wait for his confirmation

Transport Services and Protocols

- Provide logical communication between app processes running on different hosts
- Transport protocols run in end systems
  - Send side: breaks app messages into segments, passes to network layer
  - Receiver side: reassembles segments into messages, passes to app layer
- More than one transport protocol available to apps
  - Internet: TCP and UDP



Transport vs. Network Layer

- Network layer: logical communication between hosts
- Transport layer: logical communication between processes
  - Relies on, enhances, network layer services

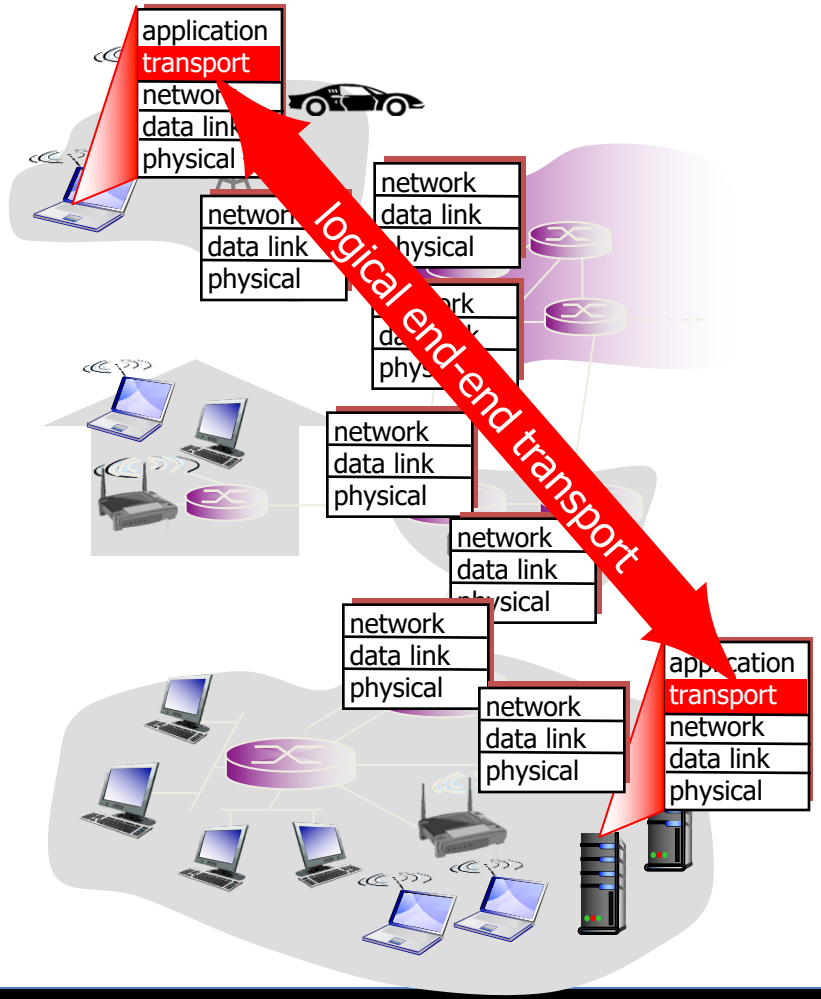
Household Analogy:

12 kids in Ann's house sending letters to 12 kids in Bill's house:

- hosts = houses
- processes = kids
- app messages = letters in envelopes
- transport protocol = Ann and Bill who demux to in-house siblings
- network-layer protocol = postal service

Internet Transport-Layer Protocols

- Reliable, in-order delivery (TCP)
  - Congestion control
  - Flow control
  - Connection setup
- Unreliable, unordered delivery: UDP
  - No-frills extension of “best-effort” IP
- Services not available:
  - Delay guarantees
  - Bandwidth guarantees



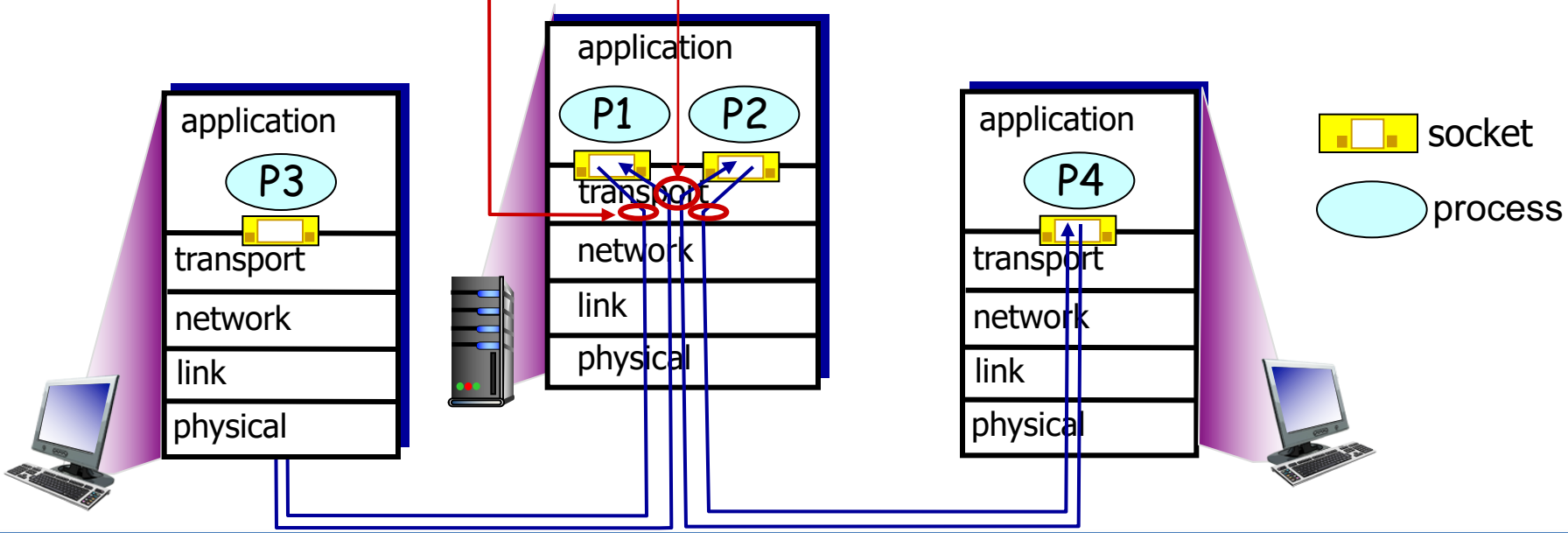
Multiplexing and Demultiplexing

Multiplexing at Sender:

Handle data from multiple sockets, add transport header (later used for demultiplexing)

Demultiplexing at Receiver:

Use header info to deliver received segments to correct socket



Socket

- What is a socket?
  - The point where a local application process attaches to the network
  - An interface between an application and the network
  - An application creates the socket
- The interface defines operations for
  - Creating a socket
  - Attaching a socket to the network
  - Sending and receiving messages through the socket
  - Closing the socket

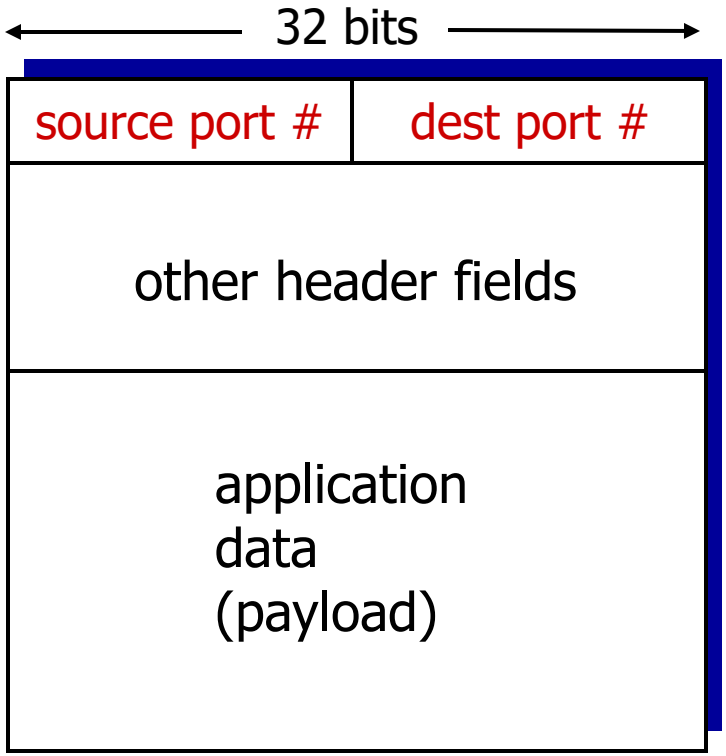
Socket

- Socket Family
  - PF_INET denotes the Internet family
  - PF_UNIX denotes the Unix pipe facility
  - PF_PACKET denotes direct access to the network interface (i.e., it bypasses the TCP/IP protocol stack)

- Socket Type
  - SOCK_STREAM is used to denote a byte stream
  - SOCK_DGRAM is an alternative that denotes a message-oriented service, such as that provided by UDP

Connectionless Demultiplexing

- Host receives IP datagrams
  - Each datagram has source IP address, destination IP address
  - Each datagram carries one transport-layer segment
  - Each segment has source, destination port number
- Host uses IP addresses & port numbers to direct segment to appropriate socket



TCP/UDP segment format

Connection-Oriented Demultiplexing

- TCP socket identified by 4-tuple:
  - Source IP address
  - Source port number
  - Dest IP address
  - Dest port number
- Demultiplexing: receiver uses all four values to direct segment to appropriate socket
- Server host may support many simultaneous TCP sockets:
  - Each socket identified by its own 4-tuple
- Web servers have different sockets for each connecting client
  - Non-persistent HTTP will have different socket for each request

UDP: User Datagram Protocol

- "No frills," "bare bones" Internet transport protocol
- "Best effort" service, UDP segments may be:
  - Lost
  - Delivered out-of-order to app
- Connectionless:
  - No handshaking between UDP sender, receiver
  - Each UDP segment handled independently of others

- UDP use:
  - Streaming multimedia apps (loss tolerant, rate sensitive)
  - DNS
  - SNMP
- Reliable transfer over UDP:
  - Add reliability at application layer
  - Application-specific error recovery!

UDP Checksum

Goal: detect “errors” (e.g., flipped bits) in transmitted segment

Sender:

- Treat segment contents, including header fields, as sequence of 16-bit integers
- Checksum: addition (one’s complement sum) of segment contents
- Sender puts checksum value into UDP checksum field

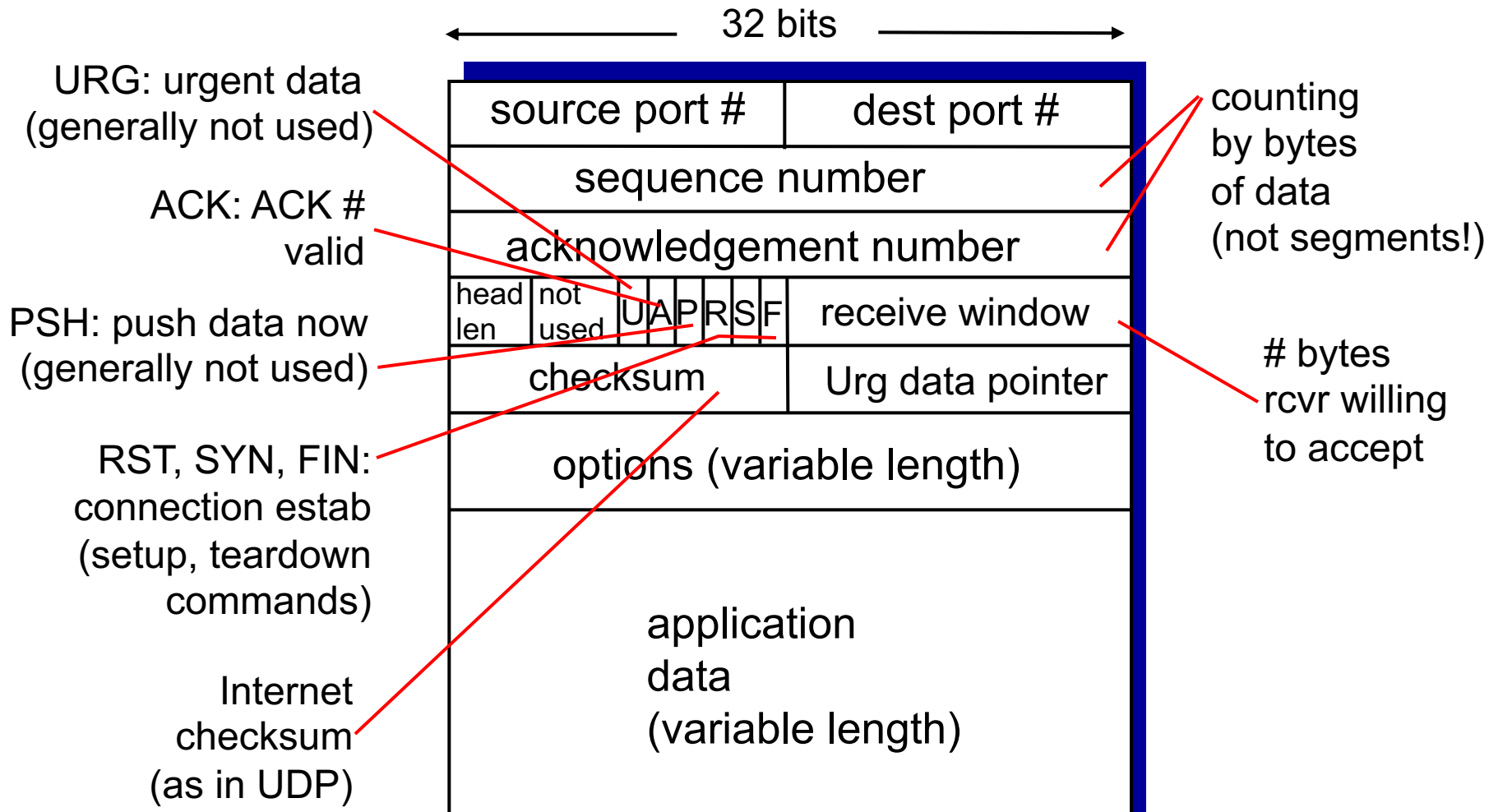
Receiver:

- Compute checksum of received segment
- Check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. But maybe errors nonetheless?

TCP: Overview

- Point-to-point:
  - One sender, one receiver
- Reliable, in-order byte steam:
  - No “message boundaries”
- Pipelined:
  - TCP congestion and flow control set window size
- Full duplex data:
  - Bi-directional data flow in same connection
  - MSS: maximum segment size
- Connection-oriented:
  - Handshaking (exchange of control msgs) inits sender, receiver state before data exchange
- Flow controlled:
  - Sender will not overwhelm receiver

TCP Segment Structure



TCP Sequence Numbers, ACKs

Sequence numbers:

- Byte stream “number” of first byte in segment’s data

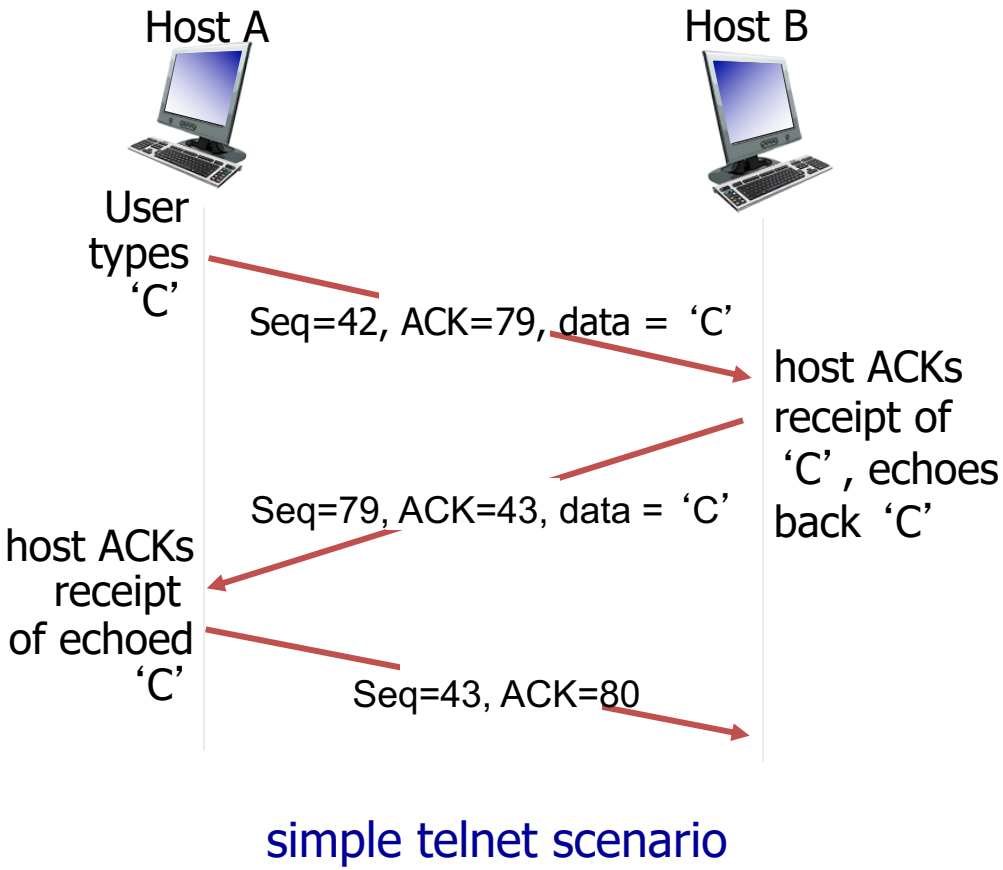
Acknowledgements:

- Seq # of next byte expected from other side
- Cumulative ACK

Q: How receiver handles out-of-order segments

A: TCP spec doesn’t say, - up to implementor

TCP Sequence Numbers, ACKs



simple telnet scenario

TCP Round-Trip Time (RTT), Timeout

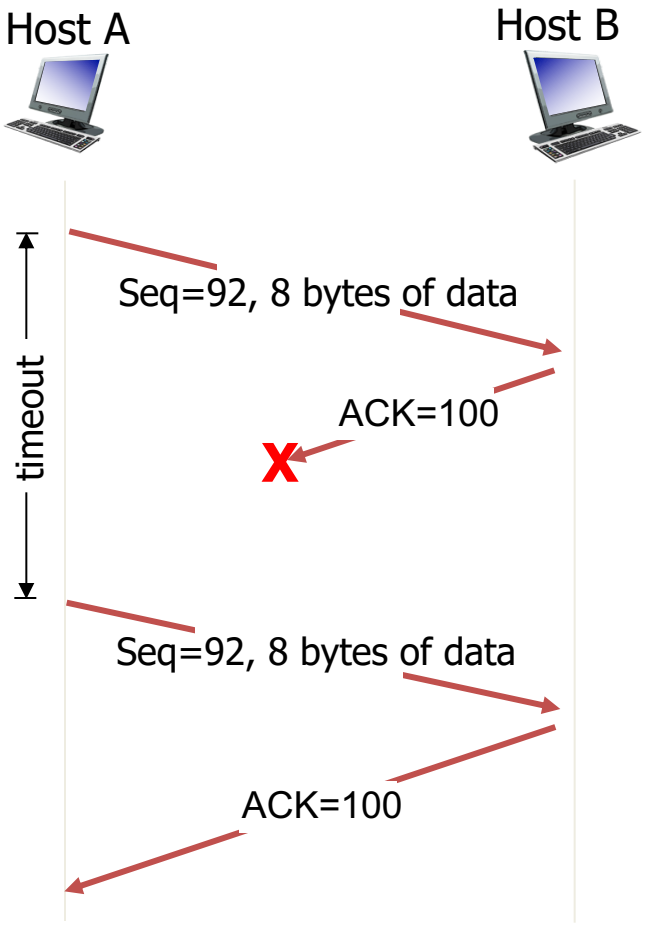
Q: How to set TCP timeout value?

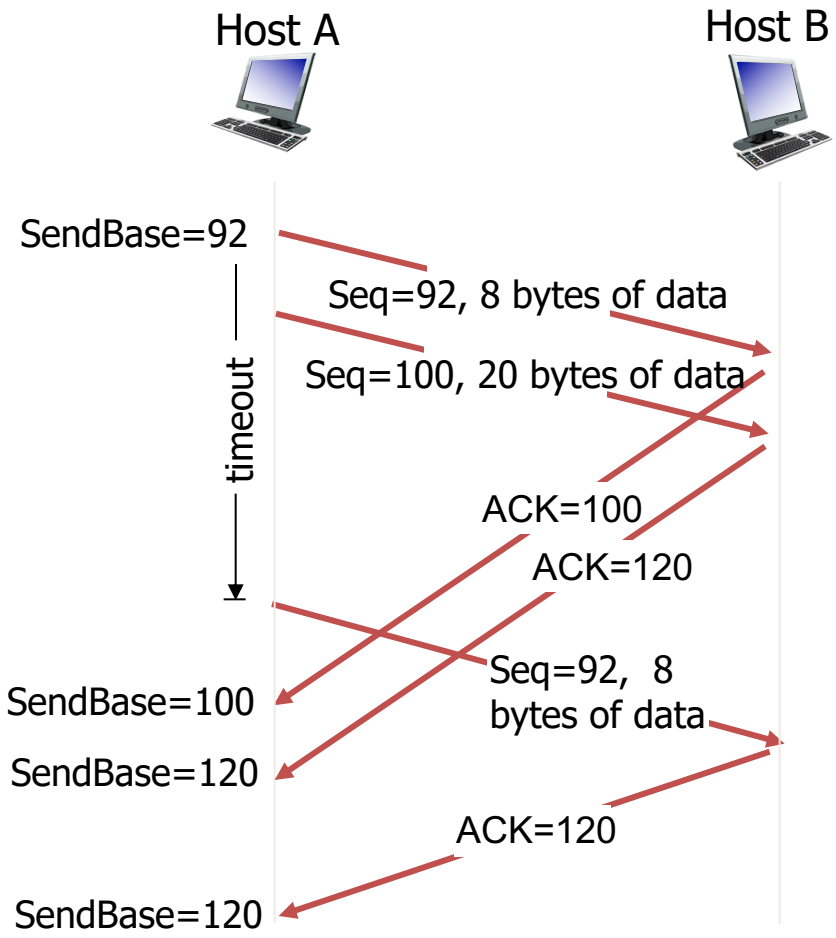
- Longer than RTT
  - But RTT varies
- Too short: premature timeout, unnecessary retransmissions
- Too long: slow reaction to segment loss

Q: How to estimate RTT?

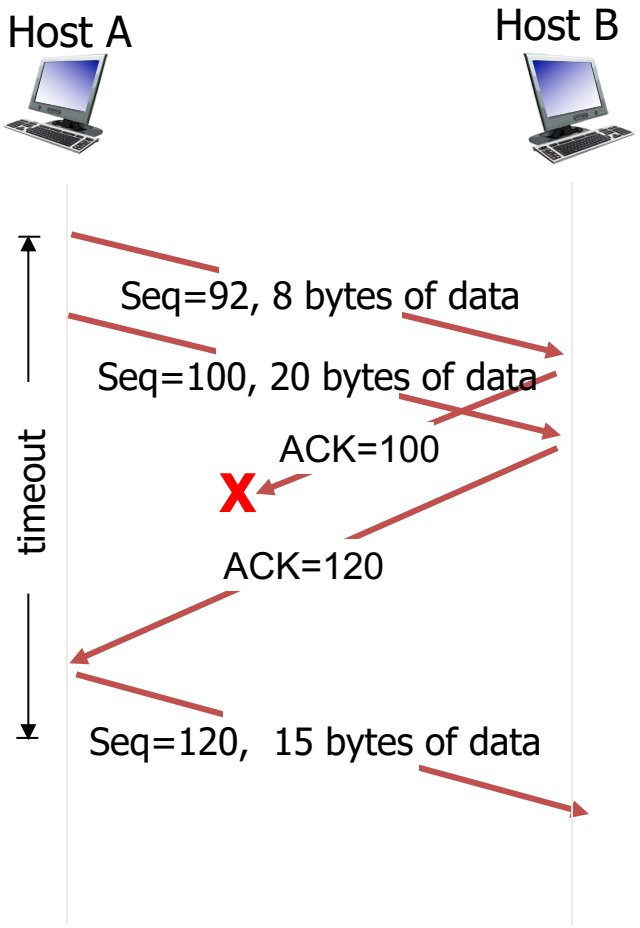
- SampleRTT: measured time from segment transmission until ACK receipt
  - Ignore retransmissions
- SampleRTT will vary, want estimated RTT “smoother”
  - Average several recent measurements, not just current SampleRTT

TCP: Retransmission Scenarios



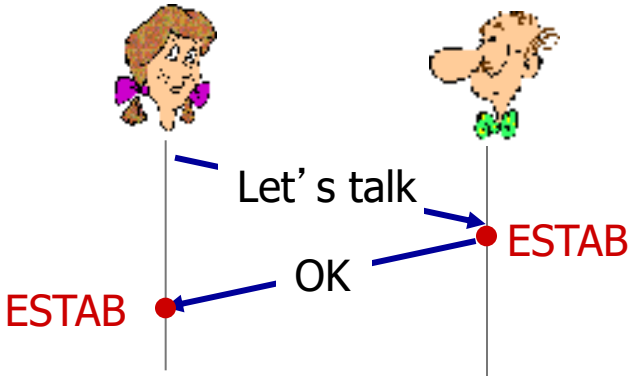


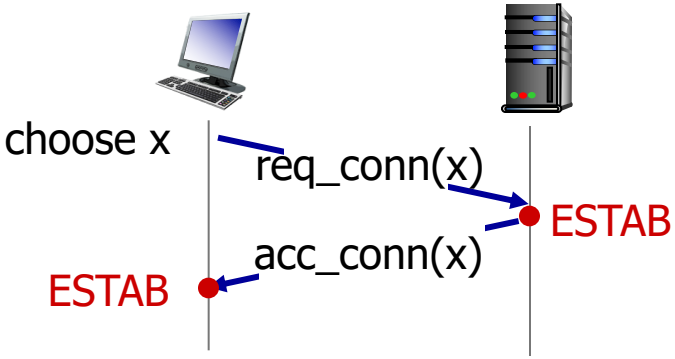
TCP: Retransmission Scenarios



Agreeing to Establish a Connection

2-way handshake:

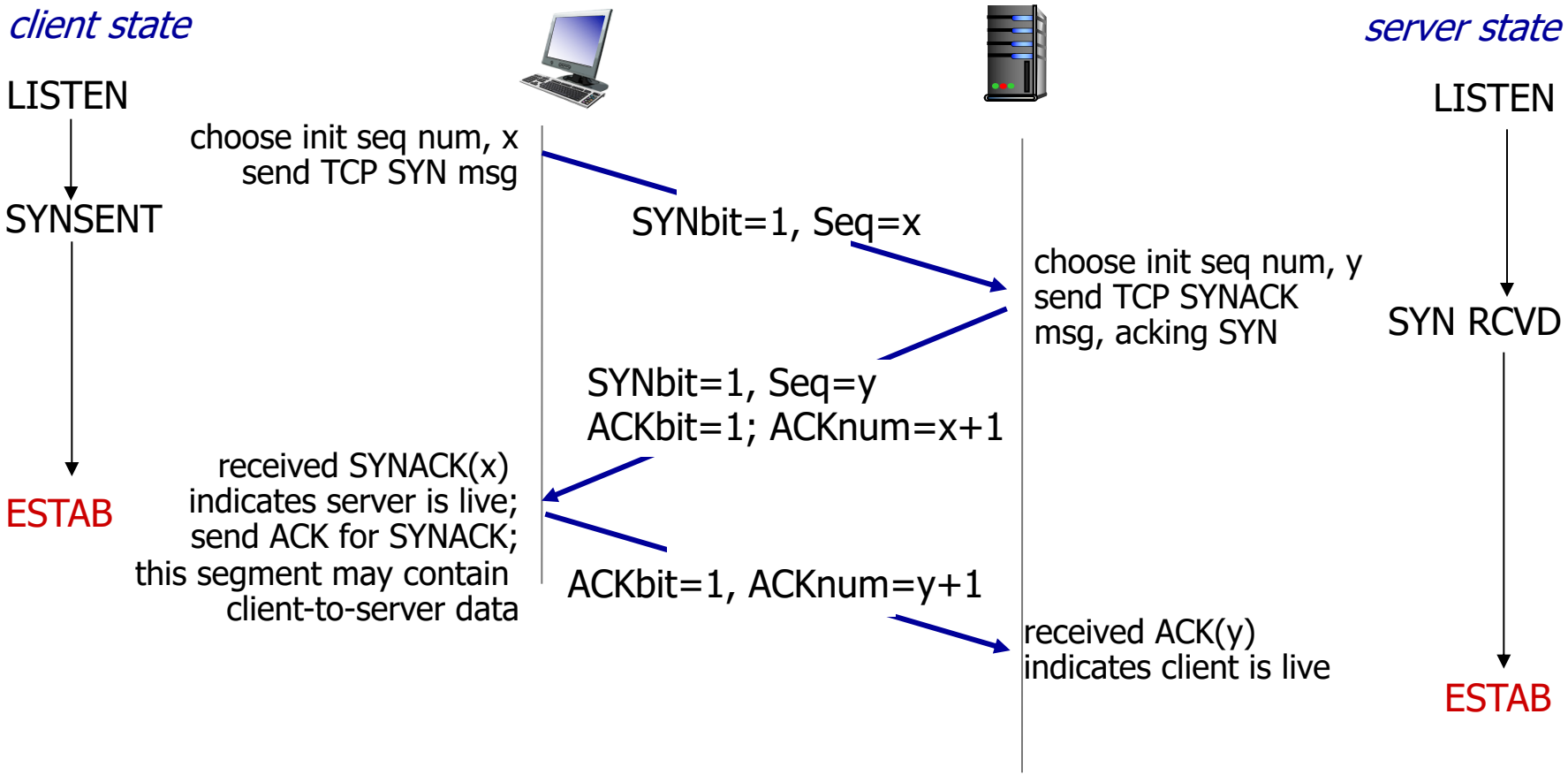




Q: Will 2-way handshake always work in network?

- Variable delays
- Retransmitted messages (e.g. req_conn(x)) due to message loss
- Message reordering
- Can't "see" other side

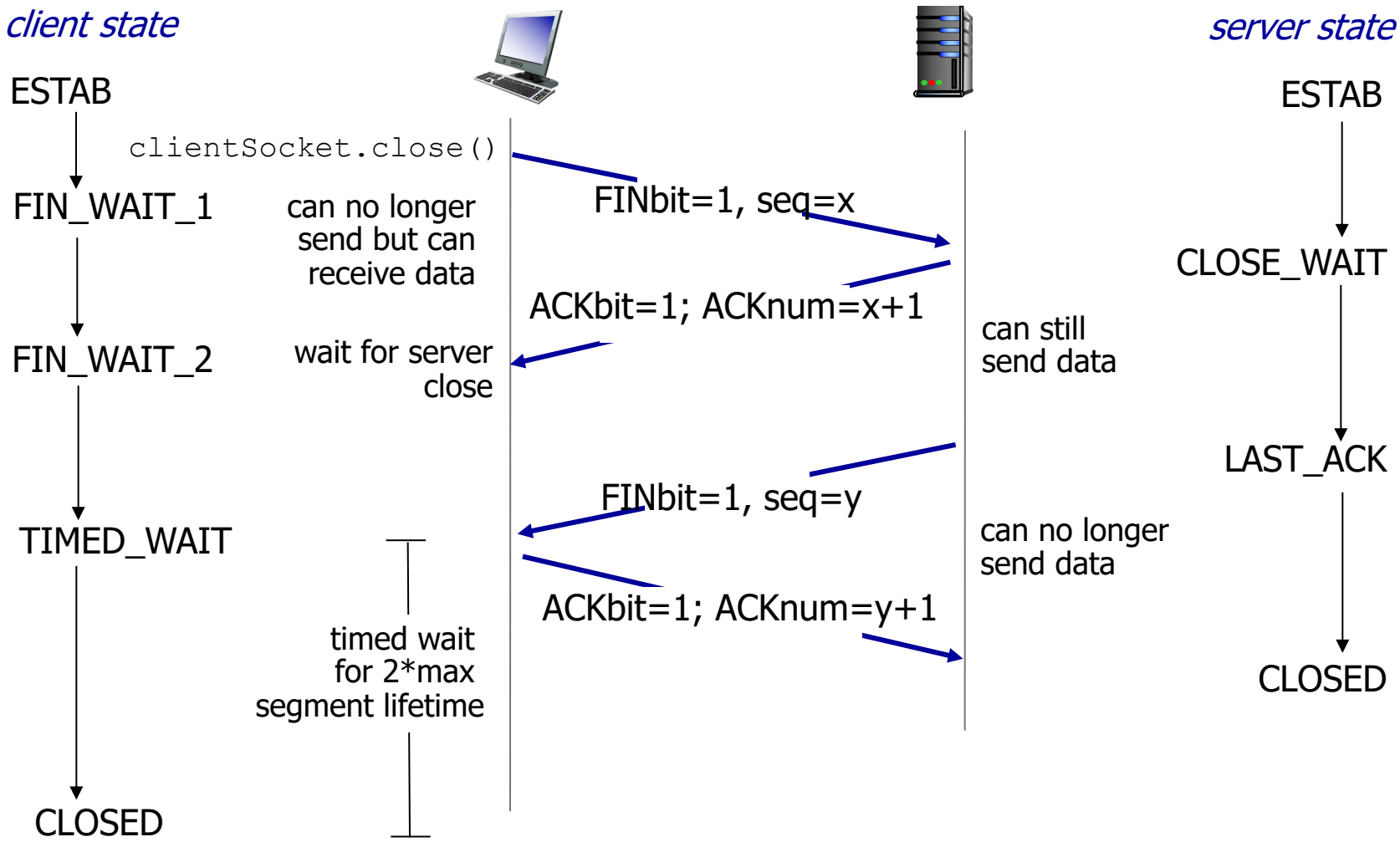
TCP 3-Way Handshake



TCP: Closing a Connection

- Client, server each close their side of connection
  - Send TCP segment with FIN bit = 1
- Respond to received FIN with ACK
  - On receiving FIN, ACK can be combined with own FIN
- Simultaneous FIN exchanges can be handled

TCP: Closing a Connection



Congestion Control

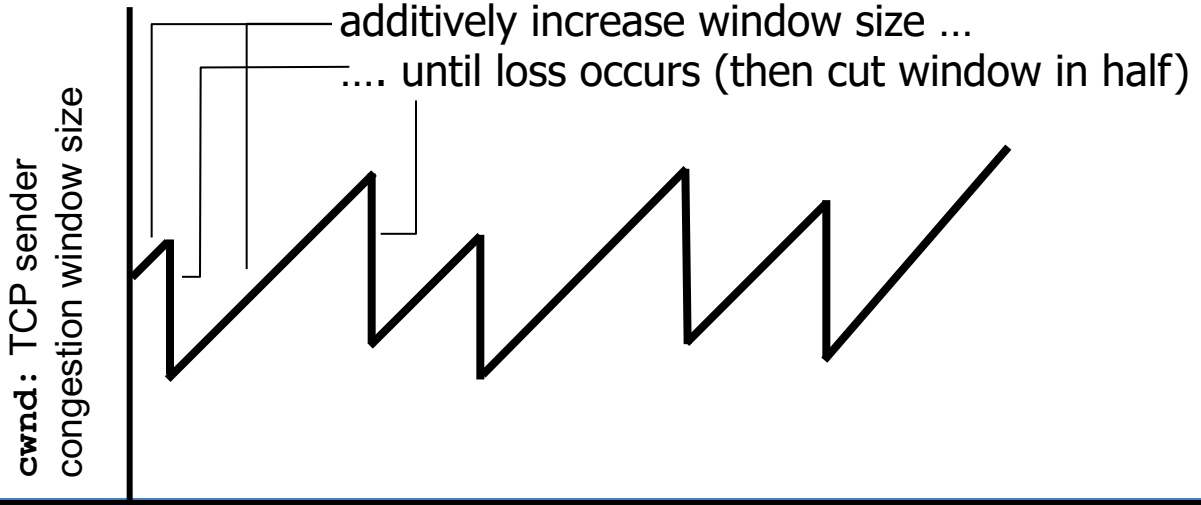
Congestion:

- Informally: “too many sources sending too much data too fast for network to handle”
- Different from flow control!
- Manifestations:
  - Lost packets (buffer overflow at routers)
  - Long delays (queueing in router buffers)

TCP Congestion Control: Additive Increase Multiplicative Decrease

- Approach: sender increases transmission rate (window size), probing for usable bandwidth, until loss occurs
- Additive increase: increase current window by 1 MSS every RTT until loss detected
- Multiplicative decrease: cut current window in half after loss

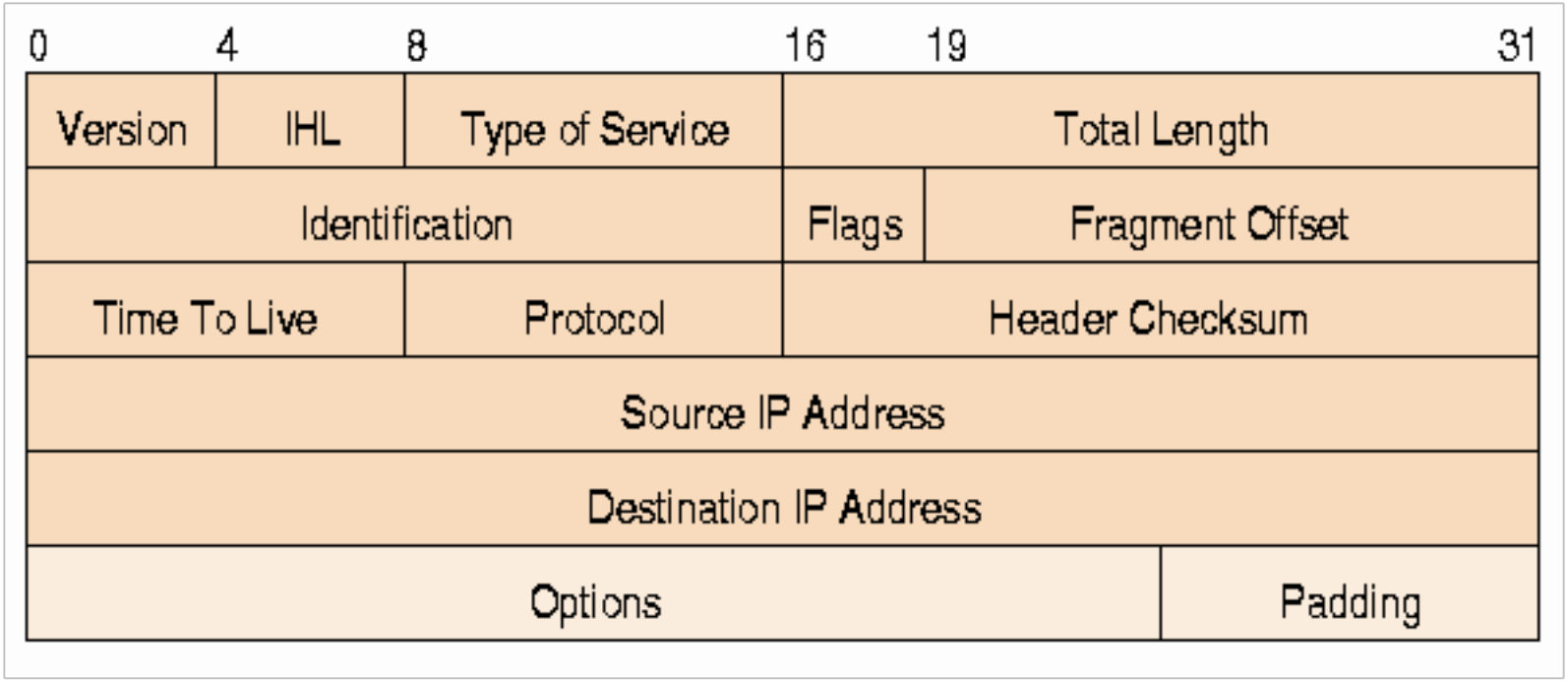
AIMD saw tooth behavior: probing for bandwidth



TCP Reliable Data Transfer

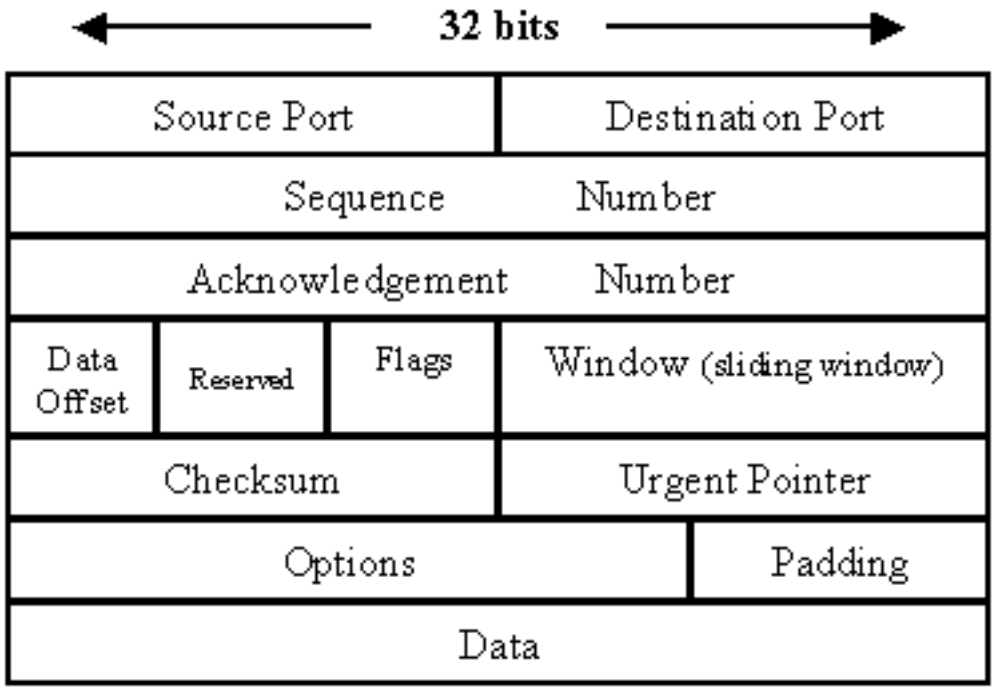
- TCP creates reliable data transfer service on top of IP's unreliable service
  - Pipelined segments
  - Cumulative acks
  - Single retransmission timer
- Retransmissions triggered by:
  - Timeout events
  - Duplicate acks
- Regulates data transfer rates based on feedback from network to avoid excessive congestion

Current Packet Structure



*Source: https://freesoft.org

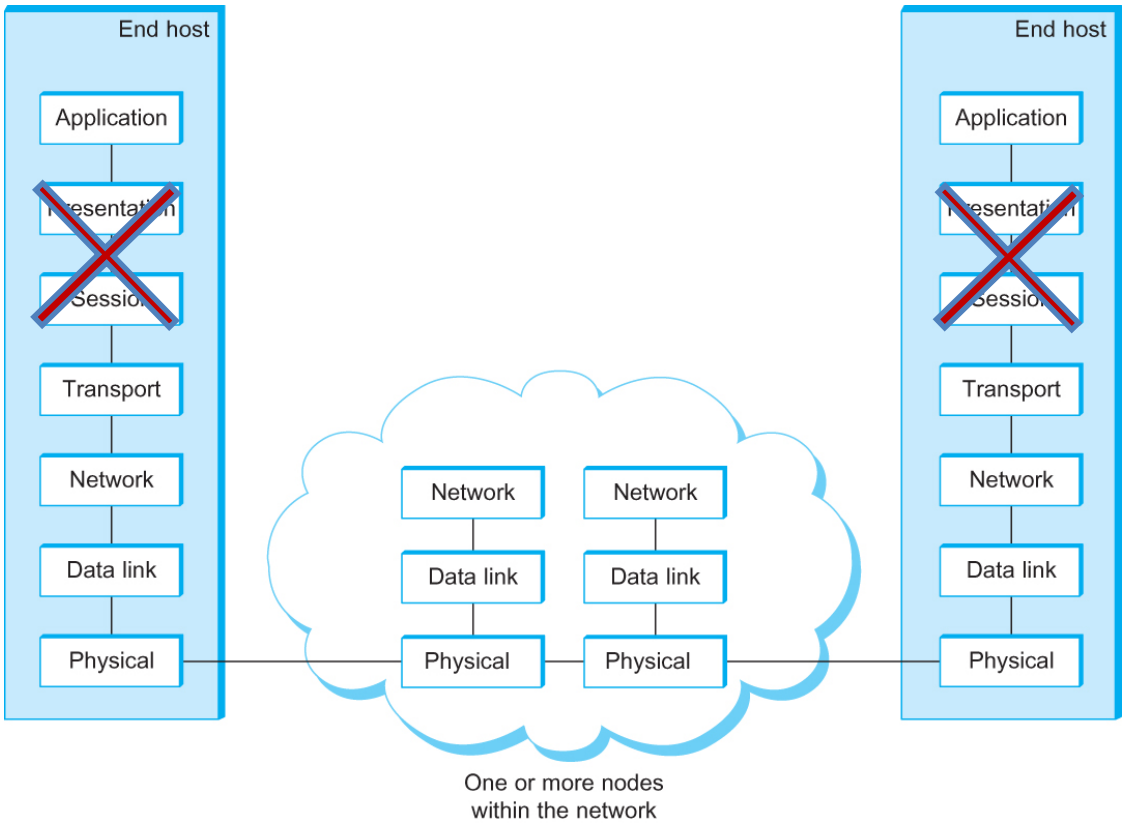
Current Packet Structure



*Source: https://techrepublic.com

Layer 5: Application Layer

*Slides adapted from Computer Networking: A top-Down Approach



The OSI 7-layer Model

OSI – Open Systems Interconnection

Application Layer

- Applications that use TCP/UDP to communicate
  - Anything that uses these is in the application layer
- Protocols in this layer: **DNS, HTTP, FTP, SMTP, POP, all their secure versions, and many, many more**

- Example: Bob is running an HTTP server on port 80. Here is my (hopefully encrypted) cookie and headers: I'd like to send a GET request to retrieve the homepage

Protocol Hourglass

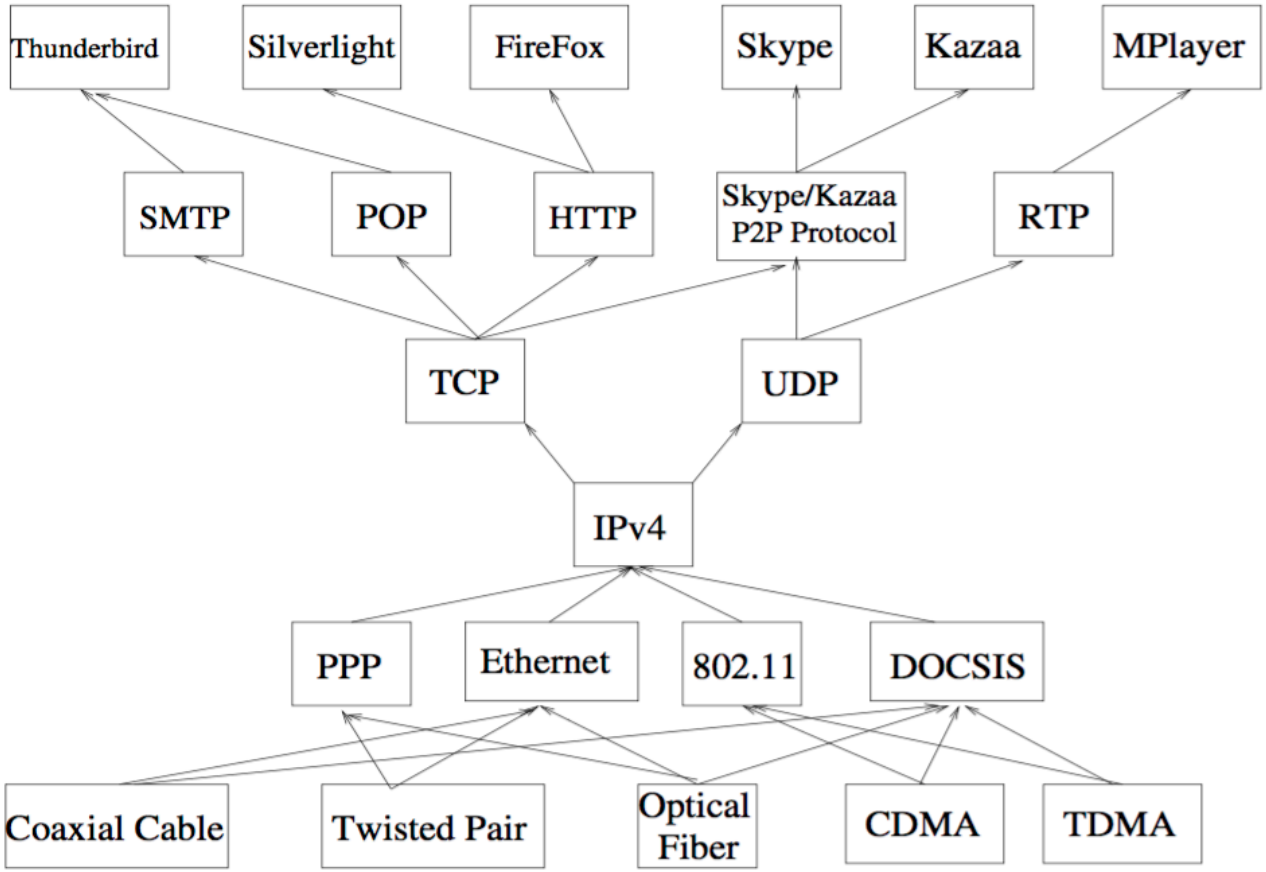


Figure 1: An (incomplete) illustration of the hourglass Internet architecture.

First Look at Security

*Slides adapted from Computer Networking: A top-Down Approach

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”
  - security considerations in all layers!

Bad guys: Put Malware Into Hosts Via Internet

- Malware can get on 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

There Are Bad People Out There!

Q: What can a “bad guy (or girl)” do?

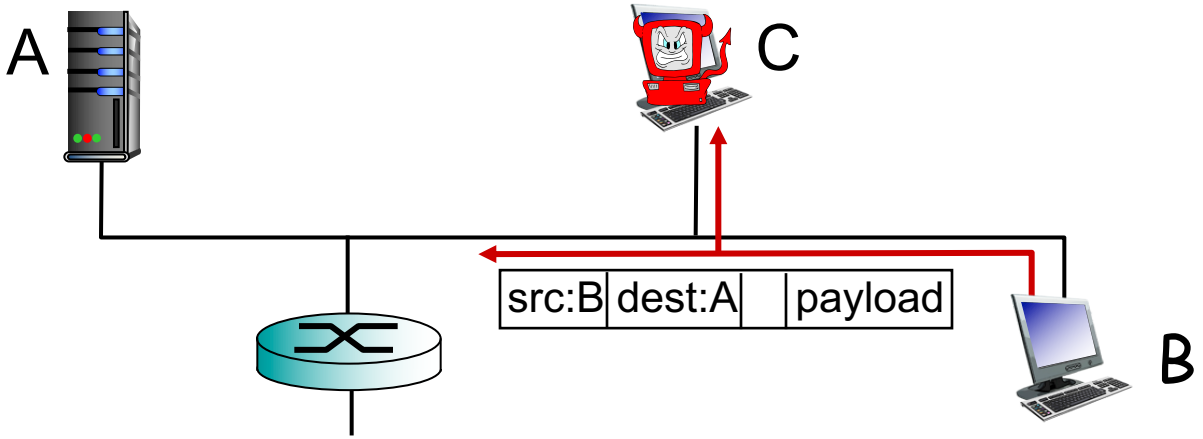
A lot!

- Eavesdrop: intercept messages
- Actively insert messages into connection
- Impersonation: can fake (spoof) source address in packet (or any field in packet)
- Hijacking: “take over” ongoing connection by removing sender or receiver, inserting himself in place
- Denial of service: prevent service from being used by others (e.g., by overloading resources)

Bad Guys Can Sniff Packets

Packet “sniffing”:

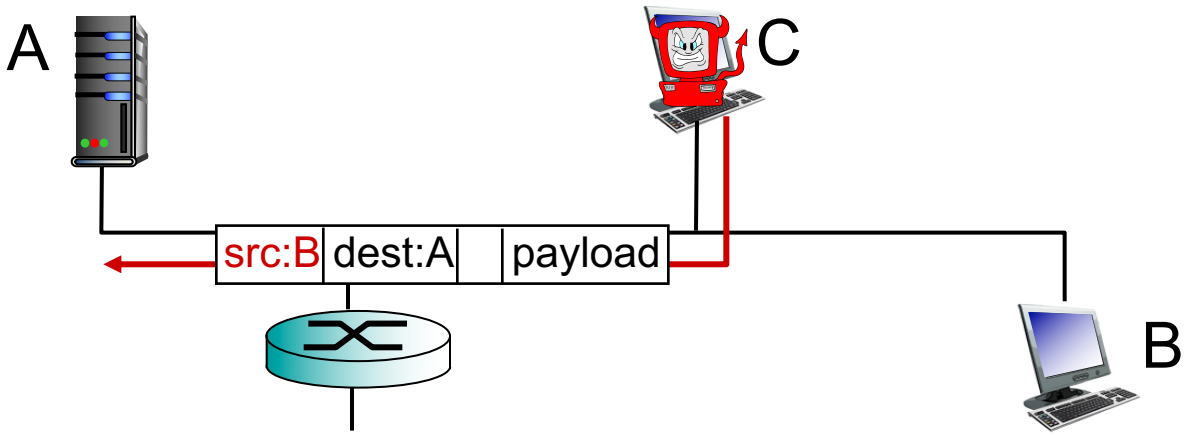
- Broadcast media (shared ethernet, wireless)
- Promiscuous network interface reads/records all packets (e.g., including passwords!) passing by



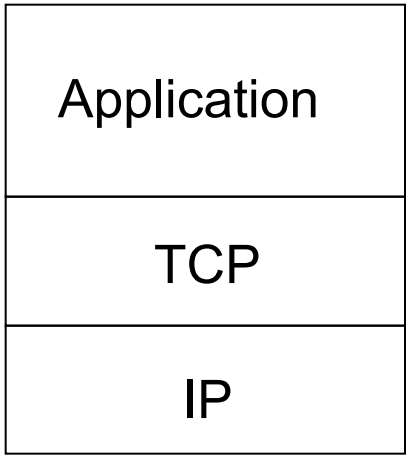
- Wireshark software is a (free) packet-sniffer

Adversaries Can Use Fake Addresses

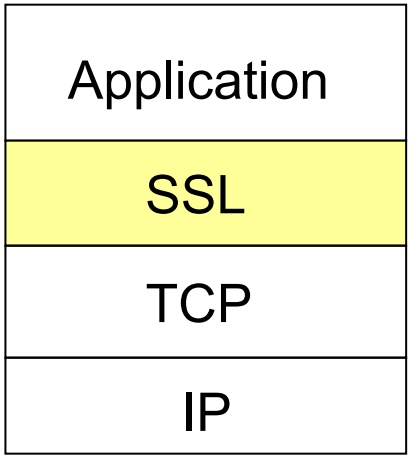
IP spoofing: send packet with false source address



SSL And TCP/IP



normal application



application with SSL

- SSL provides application programming interface (API) to applications
- C and Java SSL libraries/classes readily available

What Is Network-Layer Confidentiality?

Between two network entities:

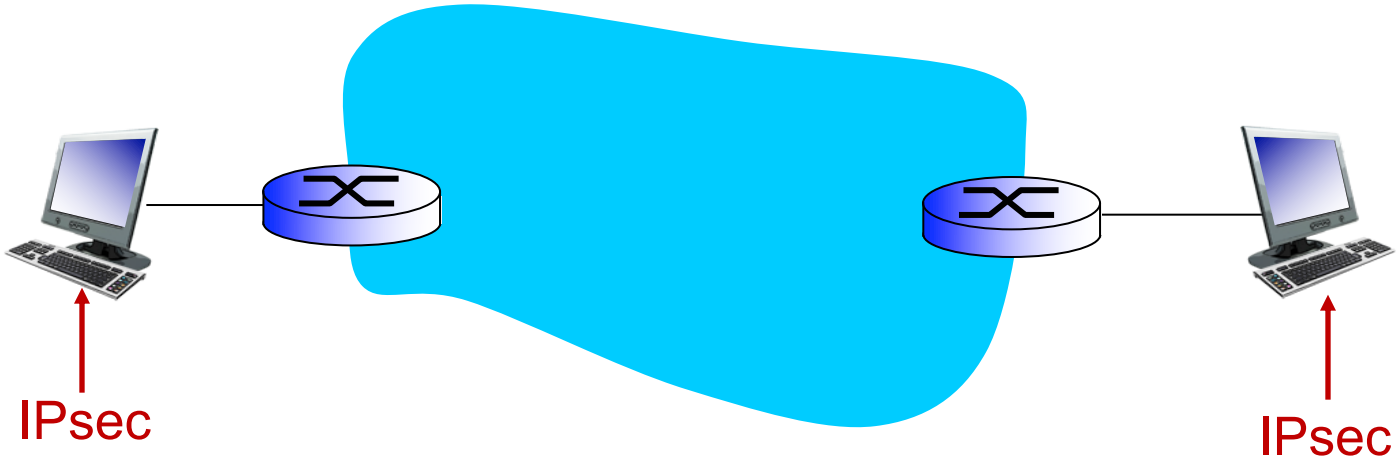
- Sending entity encrypts datagram payload, payload could be:
  - TCP or UDP segment, ICMP message, OSPF message ….
- All data sent from one entity to other would be hidden:
  - Web pages, e-mail, P2P file transfers, TCP SYN packets …
- “Blanket coverage”

Virtual Private Networks (VPNs)

Motivation:

- Institutions often want private networks for security.
  - Costly: separate routers, links, DNS infrastructure.
- VPN: institution's inter-office traffic is sent over public Internet instead
  - Encrypted before entering public Internet
  - Logically separate from other traffic

IPsec Transport Mode



- IPsec datagram emitted and received by end-system
- Protects upper level protocols

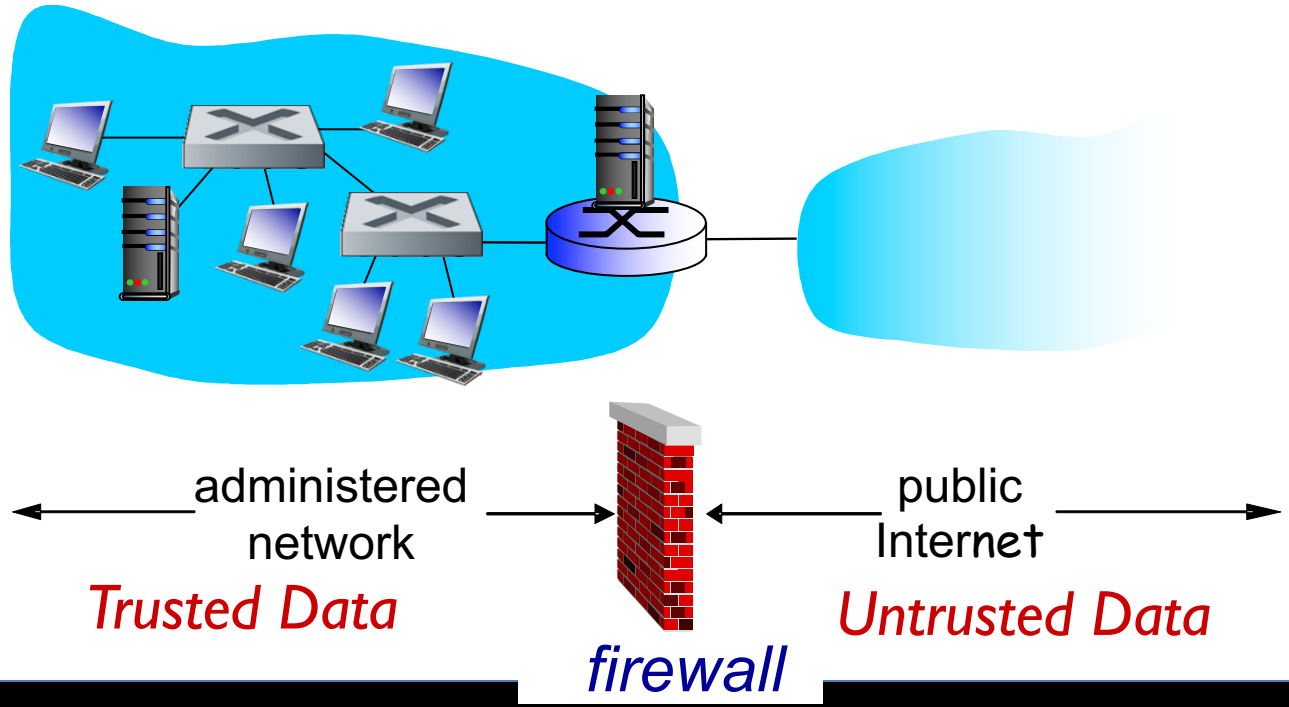
Wireless Security

- Establish encryption to prevent traffic sniffing
- We won’t have time to cover here
  - May make a good research project!
- The gist: WEP is bad; WPA is better; use WPA2

Firewalls

Firewall

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others



Firewalls: Why

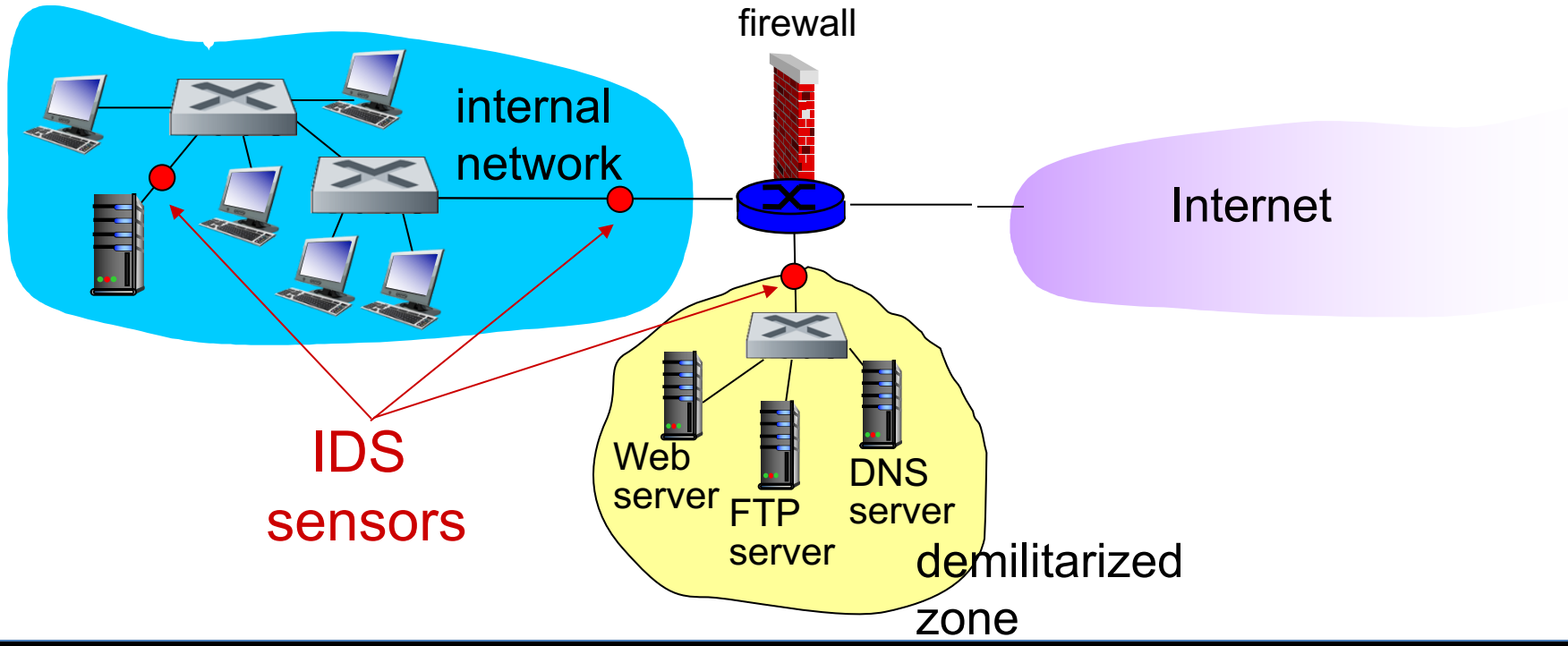
- Prevent denial of service attacks:
  - SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections
- Prevent illegal modification/access of internal data
  - E.g., attacker replaces CIA’s homepage with something else
- Allow only authorized access to inside network
  - Set of authenticated users/hosts
- Three types of firewalls:
  - Stateless packet filters
  - Stateful packet filters
  - Application gateways

Intrusion Detection Systems

- Packet filtering:
  - Operates on TCP/IP headers only
  - No correlation check among sessions
- IDS: Intrusion Detection System
  - Deep Packet Inspection (DPI): look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
  - Examine correlation among multiple packets
    - Port scanning
    - Network mapping
    - DoS attack

Intrusion Detection Systems

- Multiple IDSs: different types of checking at different locations



Memes and Jokes

Networking Joke (1)

- “Knock knock”
- “Who’s there?”
- “SYN flood”
- “SYN flood who?”
- “Knock knock”

Networking Joke (2)

- “Did you hear about the chaos at the networking manufacturer?”

Networking Joke (2)

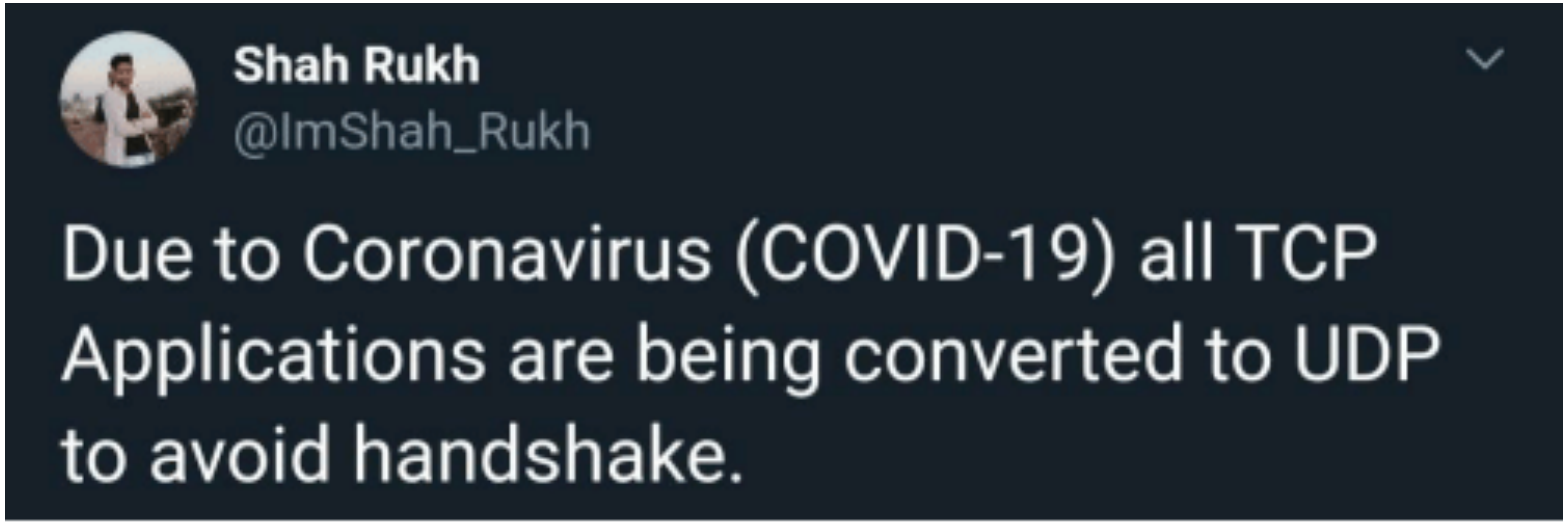
- “Did you hear about the chaos at the networking manufacturer?”
- It was Panic at the Cisco

"Hi, I'd like to hear a TCP joke."
"Hello, would you like to hear a TCP joke?"
"Yes, I'd like to hear a TCP joke."
"OK, I'll tell you a TCP joke."
"Ok, I will hear a TCP joke."
"Are you ready to hear a TCP joke?"
"Yes, I am ready to hear a TCP joke."
"Ok, I am about to send the TCP joke. It will last 10 seconds, it has two characters, it does not have a setting, it ends with a punchline."
"Ok, I am ready to get your TCP joke that will last 10 seconds, has two characters, does not have an explicit setting, and ends with a punchline."
"I'm sorry, your connection has timed out.
...Hello, would you like to hear a TCP joke?"


Kirk Bater
@KirkBater
Follow
This image is a TCP/IP Joke. This tweet is a UDP joke. I don't care if you get it.
Thread
iamkirkbater and jkjustjoshing
iamkirkbater Aug 23rd, 2017 at 9:37 AM
in #www
Do you want to hear a joke about TCP/IP?
7
7 replies
jkjustjoshing 5 months ago
Yes, I'd like to hear a joke about TCP/IP
iamkirkbater 5 months ago
Are you ready to hear the joke about TCP/IP?
jkjustjoshing 5 months ago
I am ready to hear the joke about TCP/IP
iamkirkbater 5 months ago
Here is a joke about TCP/IP.
iamkirkbater 5 months ago
Did you receive the joke about TCP/IP?
jkjustjoshing 5 months ago
I have received the joke about TCP/IP.
iamkirkbater 5 months ago
Excellent. You have received the joke about TCP/IP. Goodbye.

You vs the guy she tells you not to worry about.

UDP	TCP
Unreliable	Reliable
Connectionless	Connection-oriented
No windowing or retransmission	Segment retransmission and flow control through windowing
No sequencing	Segment sequencing
No acknowledgement	Acknowledge segments



UDP is the way to go