IP Packet Switching

Reading: Sect 4.1.1 – 4.1.4, 4.3.5

COS 461: Computer Networks
Spring 2011

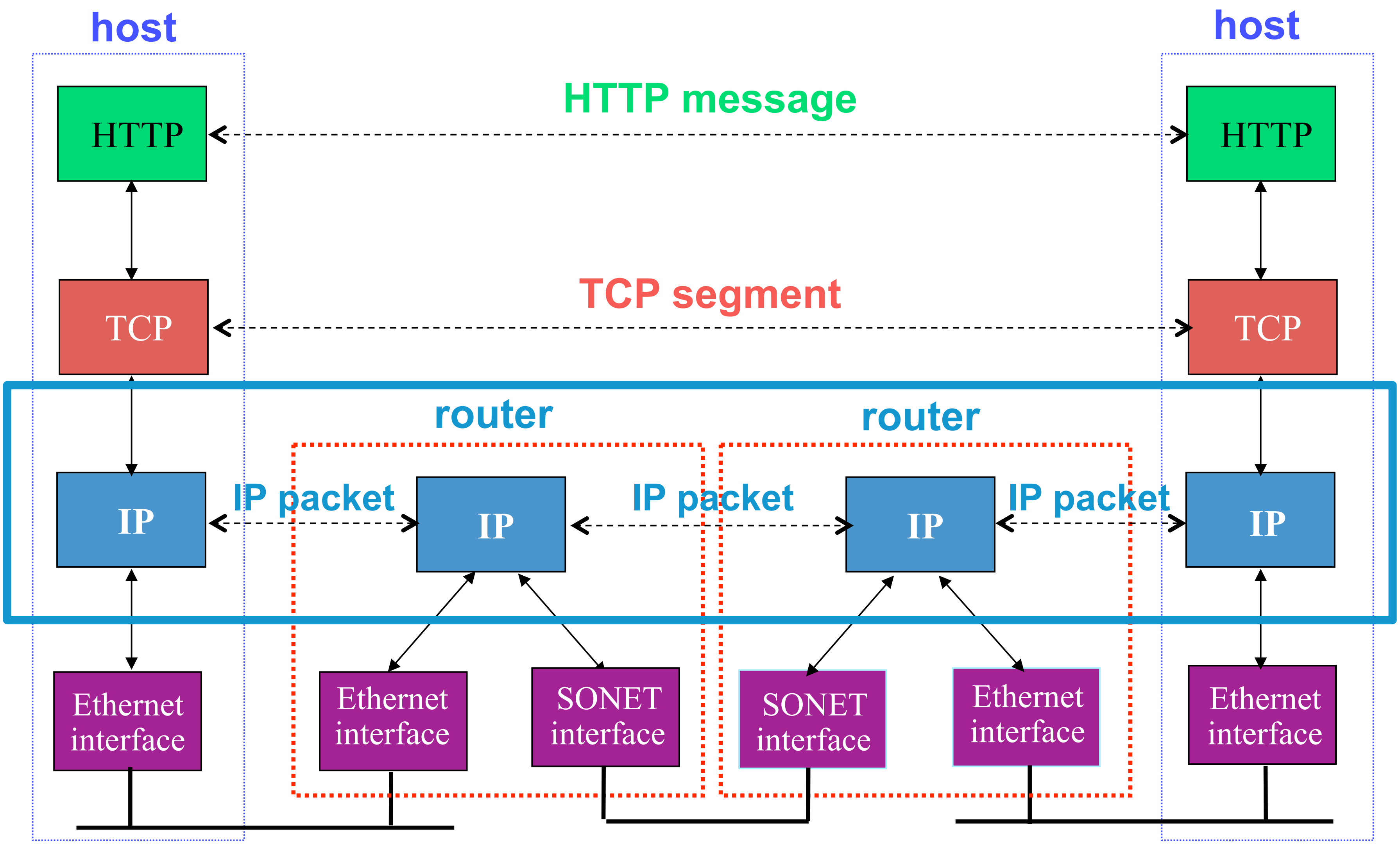
Mike Freedman

http://www.cs.princeton.edu/courses/archive/spring11/cos461/

Goals of Today's Lecture

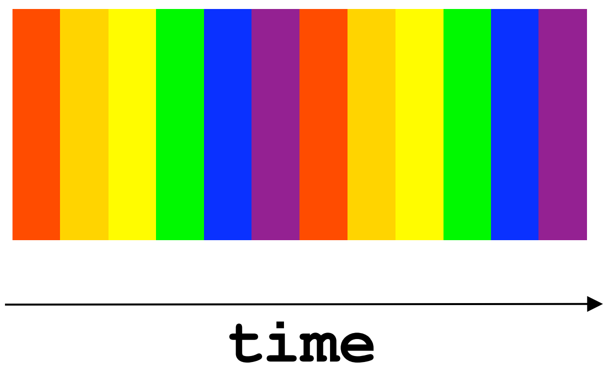
- Connectivity
  - Circuit switching
  - Packet switching
- IP service model
  - Best-effort packet delivery
  - IP as the Internet's "narrow waist"
  - Design philosophy of IP
- IP packet structure
  - Fields in the IP header
  - Traceroute using TTL field
  - Source-address spoofing

Recall the Internet layering model



Review:
Circuit Switching - Multiplexing a Link

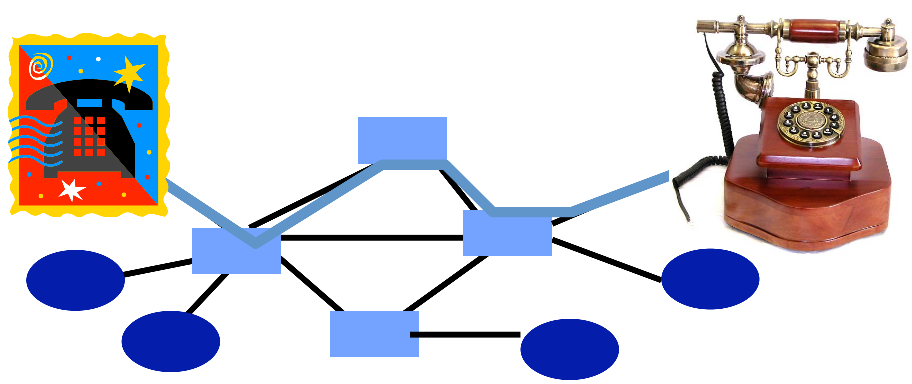
- Time-division
  - Each circuit allocated certain time slots



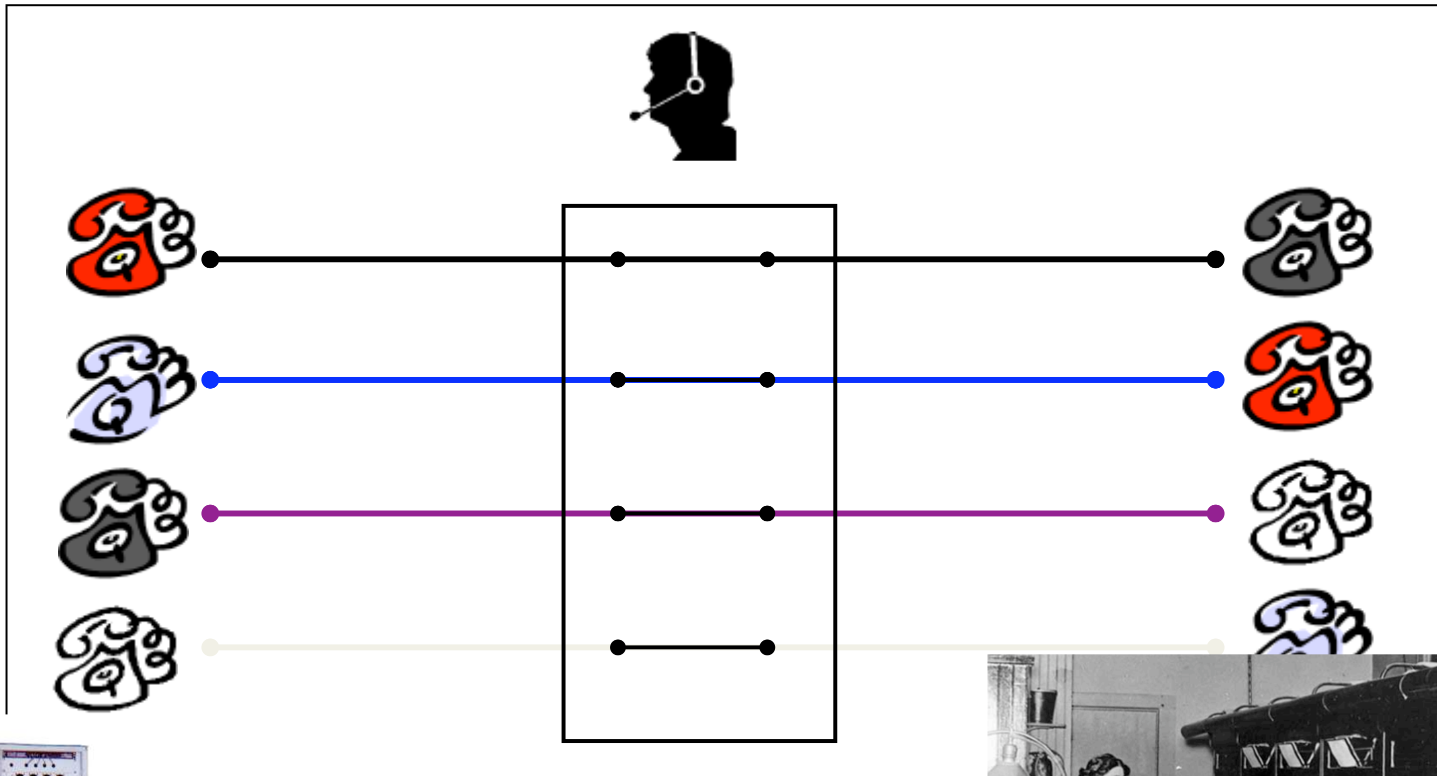
- Frequency-division
  - Each circuit allocated certain frequencies

Circuit Switching (e.g., Phone Network)

1. Source establishes connection to destination
   – Node along the path store connection info
   – Nodes may reserve resources for the connection
2. Source sends data over the connection
   – No destination address, since nodes know path
3. Source tears down connection when done



Circuit Switching With Human Operator



Telephone switch

"Operator, please connect me to 555-1212"

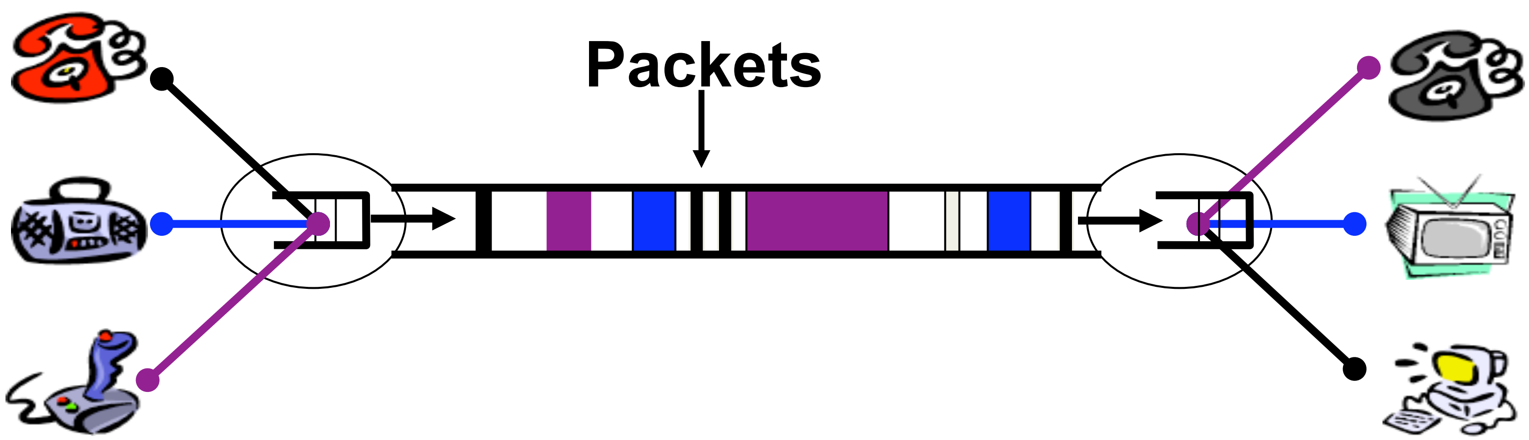
Advantages of Circuit Switching

- Guaranteed bandwidth
  - Predictable performance: not "best effort"
- Simple abstraction
  - Reliable communication channel between hosts
  - No worries about lost or out-of-order packets
- Simple forwarding
  - Forwarding based on time slot or frequency
  - No need to inspect a packet header
- Low per-packet overhead
  - Forwarding based on time slot or frequency
  - No IP (and TCP/UDP) header on each packet

Disadvantages of Circuit Switching

- Wasted bandwidth
  - Bursty traffic leads to idle conn during silent period
- Blocked connections
  - Connection refused when resources are not sufficient
- Connection set-up delay
  - Unable to avoid extra latency for small data transfers
- Network state
  - Network nodes must store per-connection information

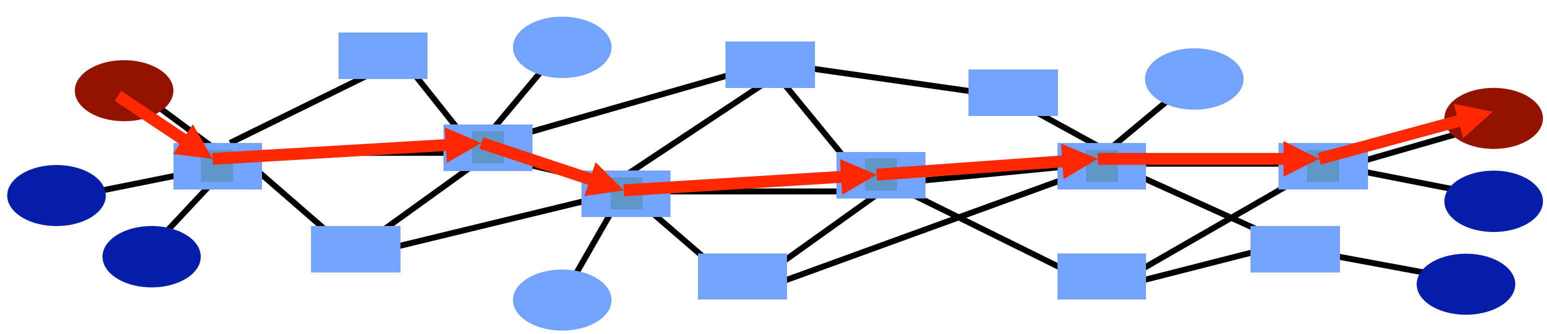
Packet Switching: Statistical (Time Division) Multiplexing



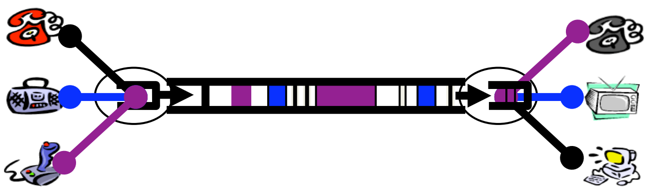
- Intuition: Traffic by computer end-points is bursty!
  - Versus: Telephone traffic not bursty (e.g., constant 56 kbps)
- Nodes differ in network demand
  - Peak data rate (e.g., Mbps)
  - Duty cycle (how much time spetn sending/receiving)
- Packet switching: Packets queue, handled in FIFO order
  - Each sender gets # time slots ~ demand

Packet Switching (e.g., Internet)

1. Data traffic divided into packets
   – Each packet contains header (with src and dst addr)
2. Packets travel separately through network
   – Packet forwarding based on the header
   – Network nodes may store packets temporarily
   – Best effort: Packets may be loss, corrupted, reordered
3. Destination reconstructs the message



IP Service Model: Why Packets?



- Data traffic is bursty
  - Web surfing, email, etc.
- Don't want to waste bandwidth
  - No traffic exchanged during idle periods
- Better to allow multiplexing
  - Different transfers share access to same links
- Don't want complex, stateful routers
  - Don't need to reserve bandwidth/memory,
  - Don't need to remember from one pkt to next
- Packets can be delivered by most anything
  - RFC 1149: IP Datagrams over Avian Carriers

- Still, can be inefficient: header bits in every packets

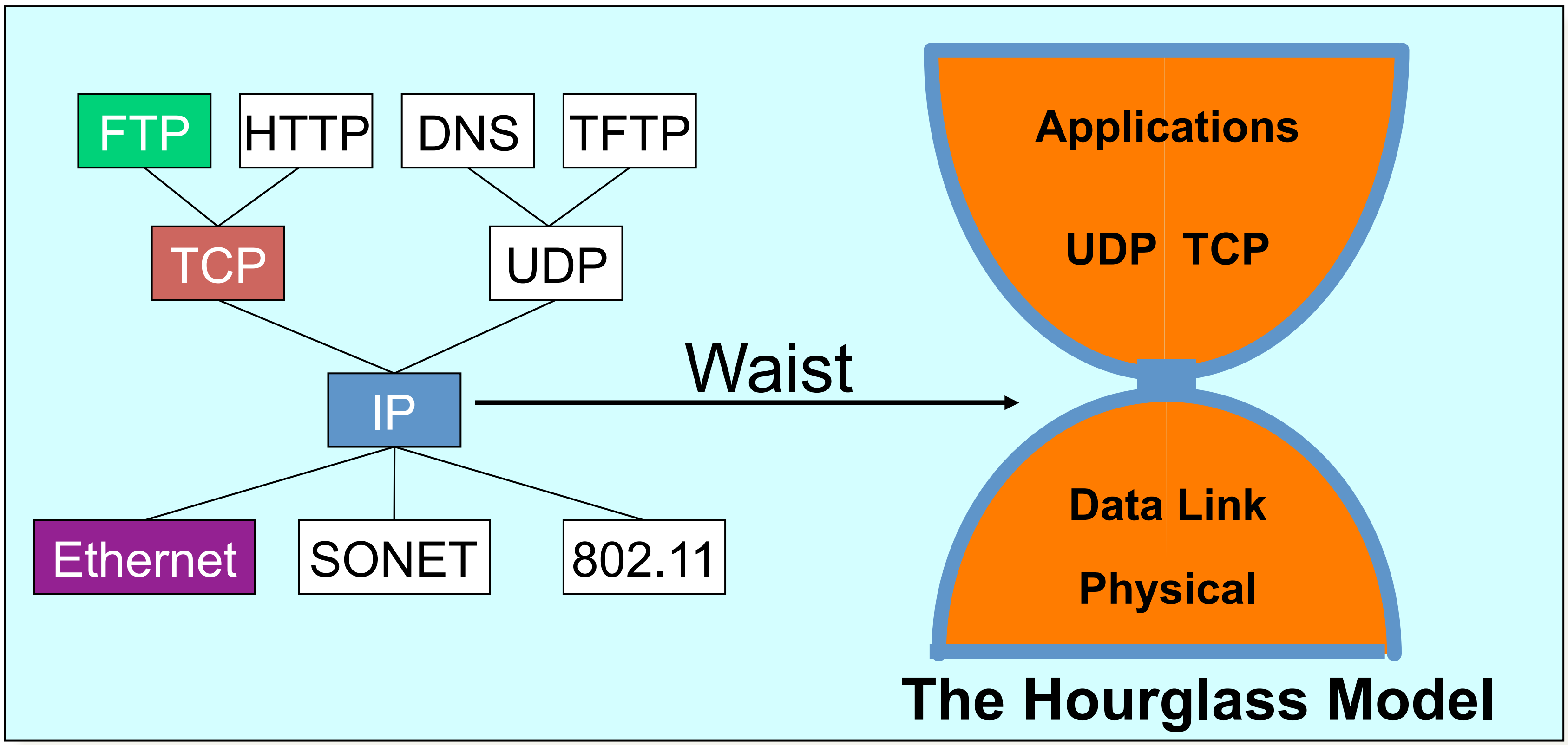
IP Service: Best-Effort is Enough

- No error detection or correction
  - Higher-level protocol can provide error checking
- Successive packets may not follow the same path
  - Not a problem as long as packets reach the destination
- Packets can be delivered out-of-order
  - Receiver can put packets back in order (if necessary)
- Packets may be lost or arbitrarily delayed
  - Sender can send the packets again (if desired)
- No network congestion control (beyond "drop")
  - Sender can slow down in response to loss or delay

END-TO-END ARGUMENTS IN SYSTEM DESIGN

J.H. Saltzer, D.P. Reed and D.D. Clark*

The Internet Protocol Suite



The waist facilitates interoperability

History: Why IP Packets?

- IP proposed in the early 1970s
  - Defense Advanced Research Project Agency (DARPA)
- Goal: connect existing networks
  - Multiplexed utilization of existing networks
  - E.g., connect packet radio networks to the ARPAnet
- Motivating applications
  - Remote login to server machines
  - Inherently bursty traffic with long silent periods
- Prior ARPAnet experience with packet switching
  - Previously showed store-and-forward packet switching

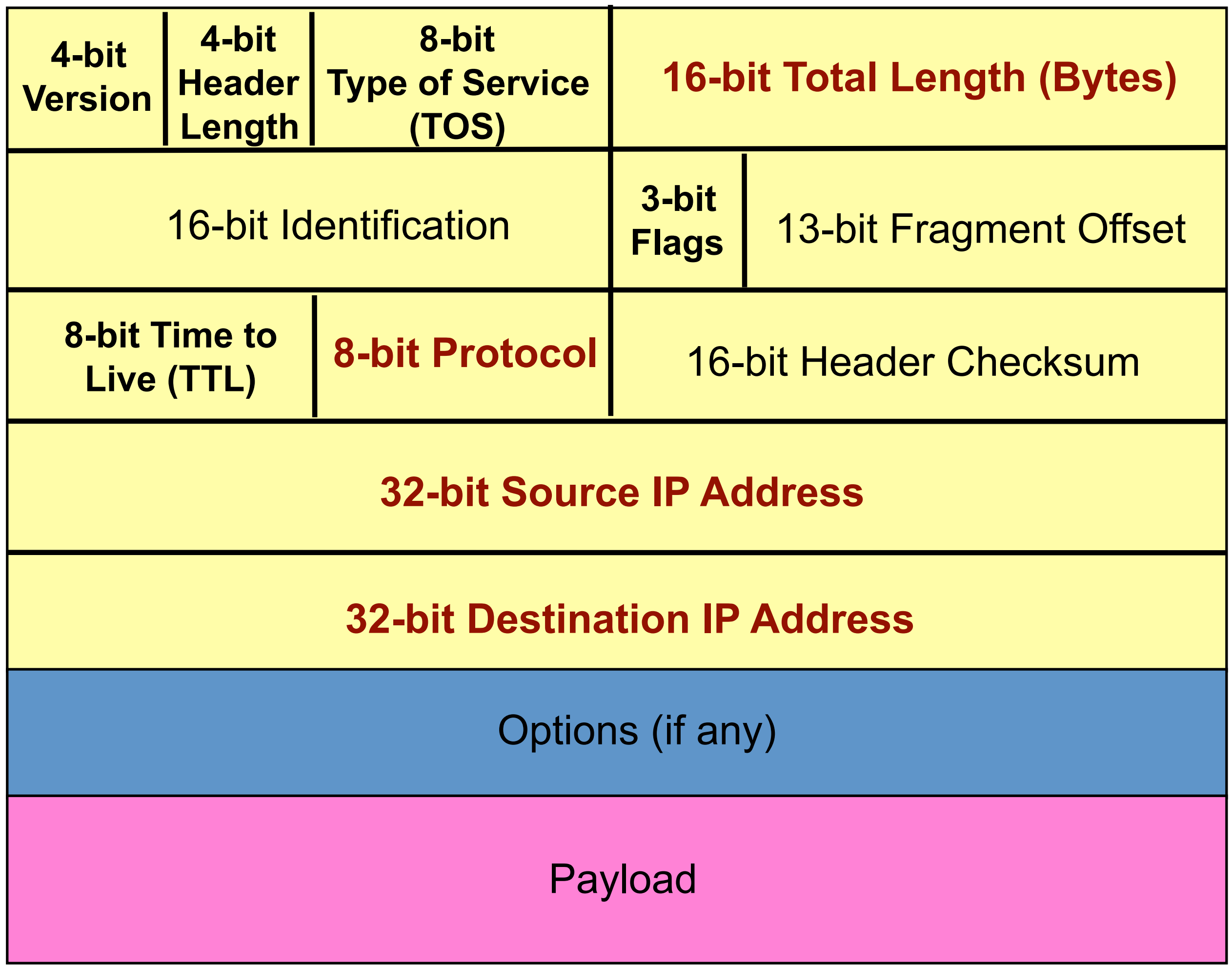
Other Main Driving Goals (In Order)

- Communication should continue despite failures
  - Survive equipment failure or physical attack
  - Traffic between two hosts continue on another path
- Support multiple types of communication services
  - Differing requirements for speed, latency, & reliability
  - Bidirectional reliable delivery vs. message service
- Accommodate a variety of networks
  - Both military and commercial facilities
  - Minimize assumptions about the underlying network

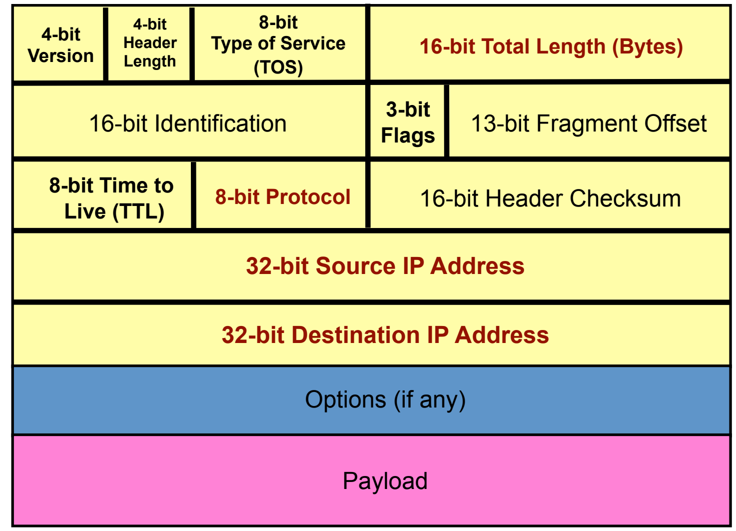
Other Driving Goals, Somewhat Met

- Permit distributed management of resources
  - Nodes managed by different institutions
  - … though this is still rather challenging
- Cost-effectiveness
  - Statistical multiplexing through packet switching
  - … though packet headers and retransmissions wasteful
- Ease of attaching new hosts
  - Standard implementations of end-host protocols
  - … though still need a fair amount of end-host software
- Accountability for use of resources
  - Monitoring functions in the nodes
  - … though this is still fairly limited and immature

IP Packet Structure

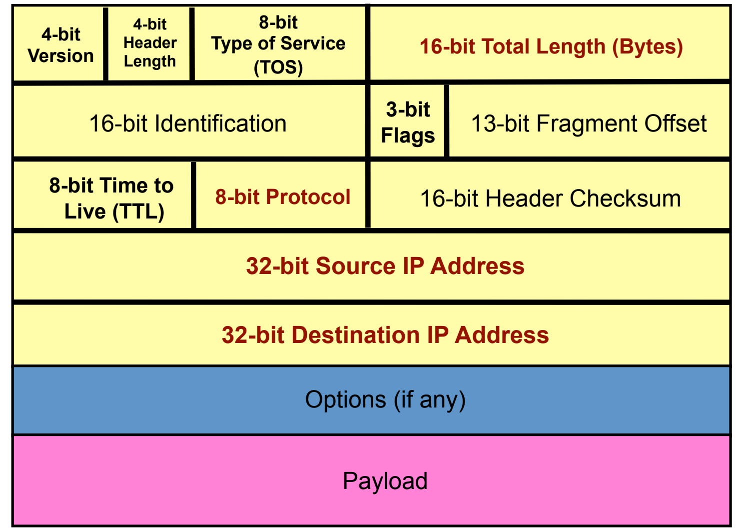


IP Header: Version, Length, ToS



- IP Version number (4 bits)
  - Necessary to know what other fields to expect: how to parse?
  - "4" (for IPv4), "6" (for IPv6)
- Header length (4 bits)
  - # of 32-bit words in header
  - Typically "5" for 20-byte IPv4 header, more if "IP options"
- Type-of-Service (8 bits)
  - Allow packets to be treated differently based on needs
  - E.g., low delay for audio, high b/w for bulk transfer
  - (We'll discuss more during "Quality of Service" lecture)

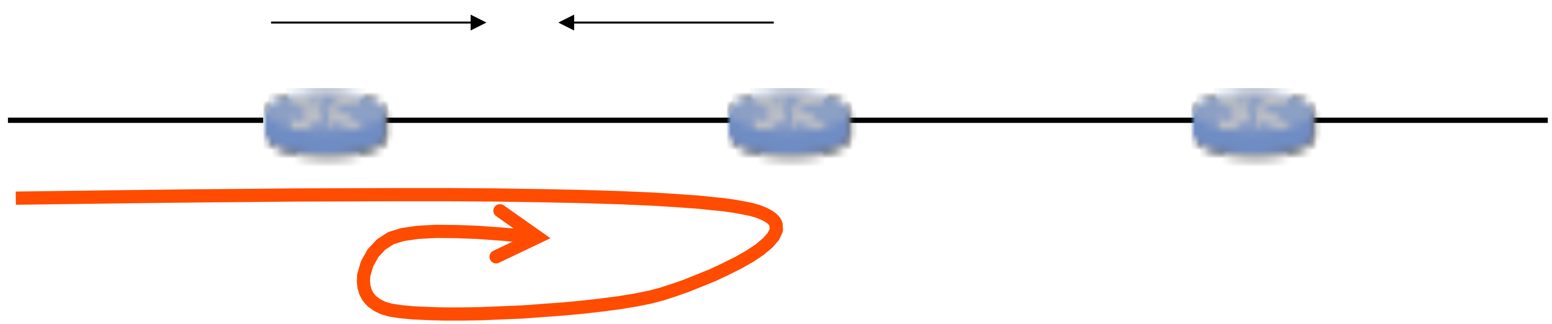
IP Header: Length, Fragments, TTL



- Total length (16 bits)
  - # of bytes in the packet
  - Max size is 63,535 bytes (2^{16} -1)
  - Links may have harder limits: Ethernet "Max Transmission Unit" (MTU) commonly 1500 bytes
- Fragmentation information (32 bits)
  - Packet identifier, flags, and fragment offset
  - Split large IP packet into fragments if link cannot handle size
  - … so why typically send max MTU packets?
- Time-To-Live (8 bits)
  - Helps identify packets stuck in forwarding loops
  - … and eventually discard from network

IP Header: More on Time-to-Live (TTL)

- Potential robustness problem
  - Forwarding loops can cause packets to cycle forever
  - Confusing if the packet arrives much later



- Time-to-live field in packet header
  - TTL field decremented by each router on path
  - Packet is discarded when TTL field reaches 0…
  - …and “time exceeded” message (ICMP) sent to source

Aside: Traceroute as network tool

- Common uses of traceroute
  - Discover the topology of the Internet
  - Debug performance and reachability problems
- On UNIX machine
  - "traceroute cnn.com" or "traceroute 12.1.1.1"
- On Windows machine
  - "tracert cnn.com" or "tracert 12.1.1.1"

Example Traceroute: Berkeley to CNN

Hop number, IP address, DNS name

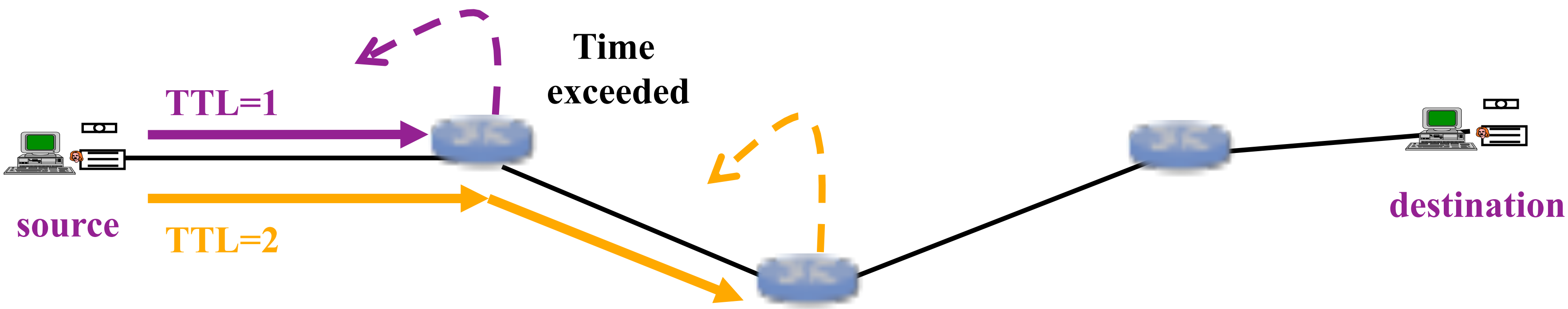
No response from router

1	169.229.62.1	inr-daedalus-0.CS.Berkeley.EDU
2	169.229.59.225	soda-cr-1-1-soda-br-6-2
3	128.32.255.169	vlan242.inr-202-doecev.Berkeley.EDU
4	128.32.0.249	gigE6-0-0.inr-666-doecev.Berkeley.EDU
5	128.32.0.66	qsv-juniper--ucb-gw.calren2.net
6	209.247.159.109	POS1-0.hsipaccess1.SanJose1.Level3.net
7	*	?
8	64.159.1.46	?
9	209.247.9.170	pos8-0.hsa2.Atlanta2.Level3.net
10	66.185.138.33	pop2-atm-P0-2.atdn.net
11	*	?
12	66.185.136.17	pop1-atl-P4-0.atdn.net
13	64.236.16.52	www4.cnn.com

No name resolution

IP Header: Use of TTL in Traceroute

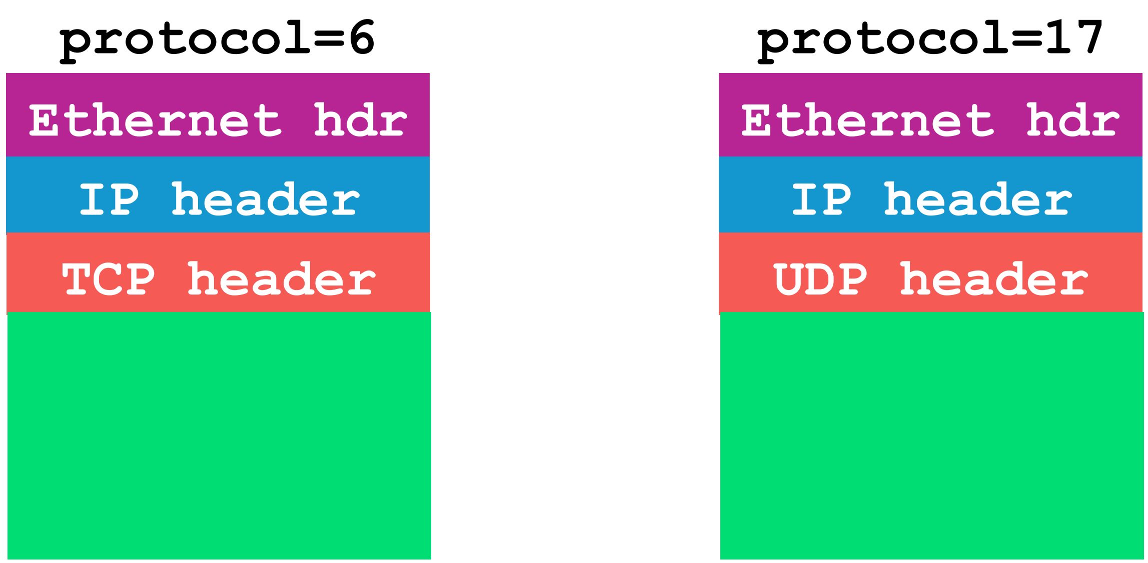
- Time-To-Live field in IP packet header
  - Source sends a packet with a TTL of *n*
  - Each router along the path decrements the TTL
  - “TTL exceeded” sent when TTL reaches *0*
- Traceroute tool exploits this TTL behavior



Send packets with TTL=1, 2, …
and record source of “time exceeded” message

IP Header Fields: Transport Protocol

- Protocol (8 bits)
  - Identifies the higher-level protocol
    - E.g., "6" for TCP, "17" for UDP
  - Important for demultiplexing at receiving host
    - Indicates what kind of header to expect next



IP Header: Checksum on Header

- Checksum (16 bits)
  - Sum of all 16-bit words in IP header
  - If any bits of header are corrupted in transit, checksum won't match at receiving host
  - Receiving host discards corrupted packets
    - Sending host will retransmit the packet, if needed

```
  134            134
+ 212    →     + 216
-----          -----
= 346          = 350
```

Mismatch!

IP Header: To and From Addresses

- Two IP addresses
  - Source and destination (32 bits each)
- Destination address
  - Unique identifier for receiving host
  - Allows each node to make forwarding decisions
- Source address
  - Unique identifier for sending host
  - Enables recipient to send a reply back to source

Source Address: What if Source Lies?

- Source address should be the sending host
  - But, who's checking? You can "spoof" any address!

- Why would someone want to do this?
  - Launch a denial-of-service attack
    - Send excessive packets to destination
    - … to overload node, or links leading to it
  - Evade detection by "spoofing"
    - But, victim could identify you by source addr, so lie!
  - Also, an attack against the spoofed host
    - Spoofed host is wrongly blamed
    - Spoofed host may receive return traffic from receiver