



IP Packet Switching

COS 461: Computer Networks
Spring 2010 (MW 3:00-4:20 in COS 105)

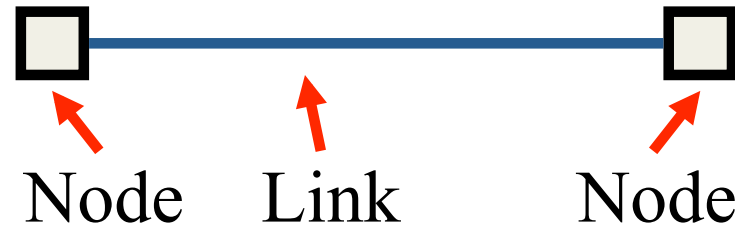
Mike Freedman

<http://www.cs.princeton.edu/courses/archive/spring10/cos461/>

Goals of Today's Lecture

- **Connectivity**
 - Links and nodes
 - 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

Simple Network: Nodes and a Link



- **Node: computer**
 - End host: general-purpose computer, cell phone, PDA
 - Network node: switch or router
- **Link: physical medium connecting nodes**
 - Twisted pair: the wire that connects to telephones
 - Coaxial cable: the wire that connects to TV sets
 - Optical fiber: high-bandwidth long-distance links
 - Space: propagation of radio waves, microwaves, ...

Network Components

Links



Fibers



Coaxial Cable

Interfaces

Ethernet card



Wireless card



Switches/routers

Large router

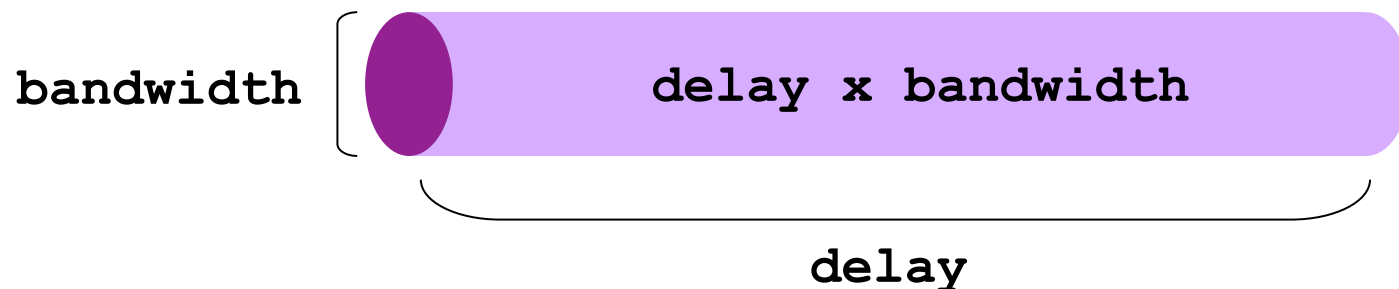


Telephone switch



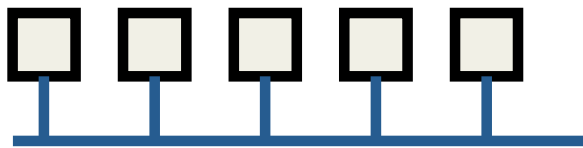
Links: Delay and Bandwidth

- **Delay**
 - Latency for propagating data along the link
 - Corresponds to the “length” of the link
 - Typically measured in seconds
- **Bandwidth**
 - Amount of data sent (or received) per unit time
 - Corresponds to the “width” of the link
 - Typically measured in bits per second

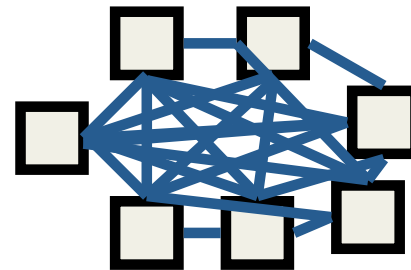


Connecting More Than Two Hosts

- **Multi-access link: Ethernet, wireless**
 - Single physical link, shared by multiple nodes
 - Limitations on distance and number of nodes
- **Point-to-point links: fiber-optic cable**
 - Only two nodes (separate link per pair of nodes)
 - Limitations on the number of adapters per node

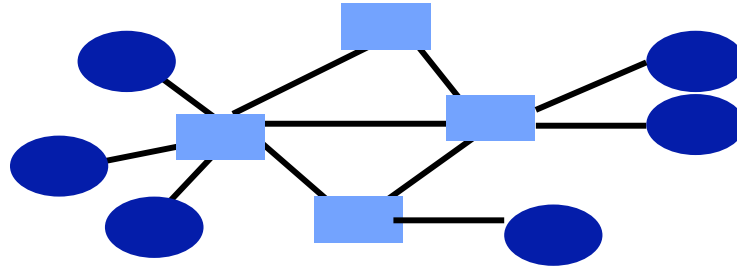


multi-access link



point-to-point links

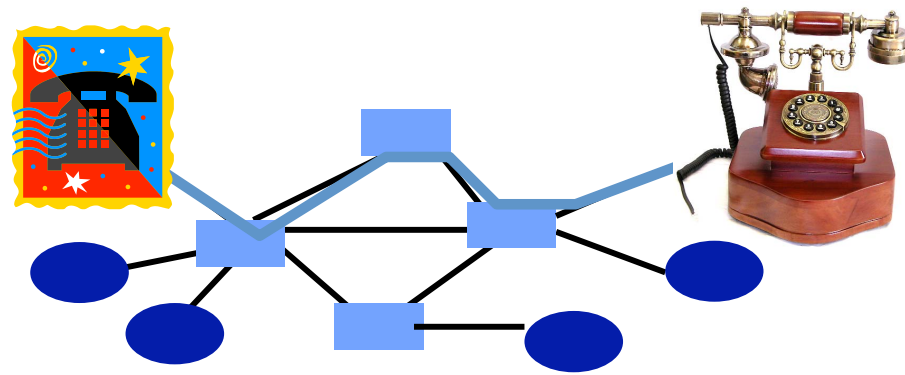
Beyond Directly-Connected Networks



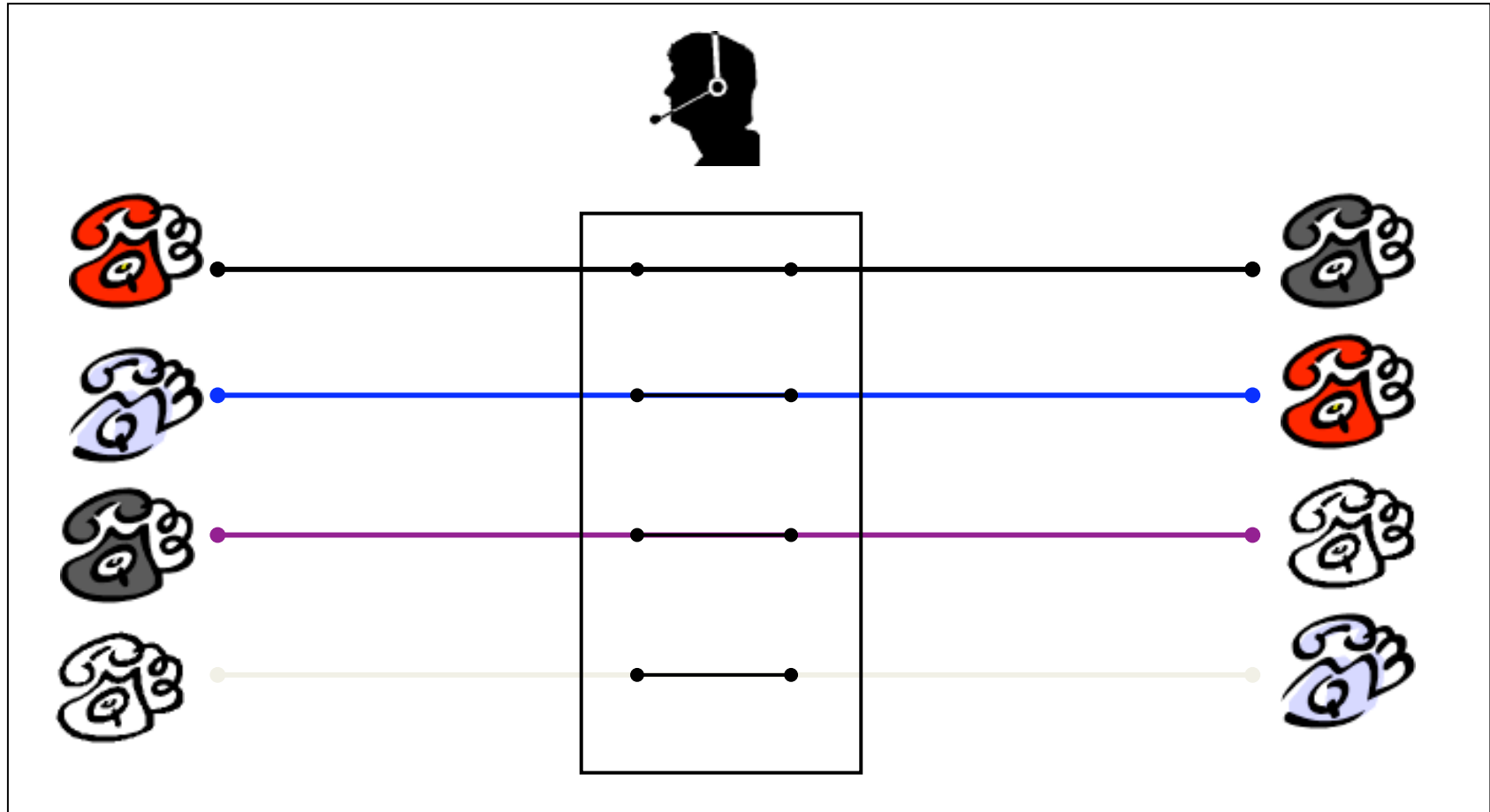
- **Switched network**
 - End hosts at the edge
 - Network nodes that switch traffic
 - Links between the nodes
- **Multiplexing**
 - Many end hosts communicate over the network
 - Traffic shares access to the same links

Circuit Switching (e.g., Phone Network)

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



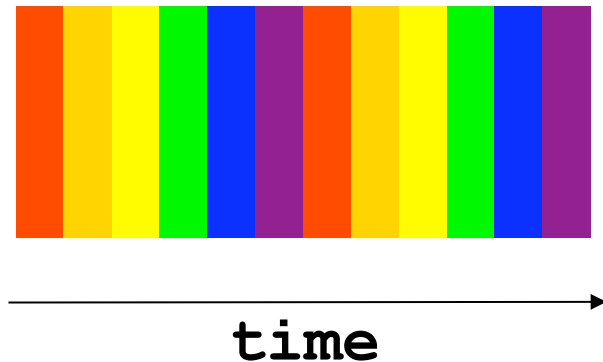
Circuit Switching With Human Operator



Circuit Switching: Multiplexing a Link

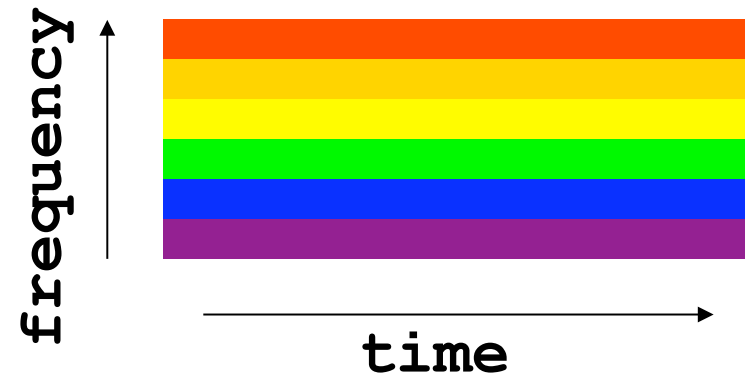
- Time-division

- Each circuit allocated certain time slots



- Frequency-division

- Each circuit allocated certain frequencies



Advantages of Circuit Switching

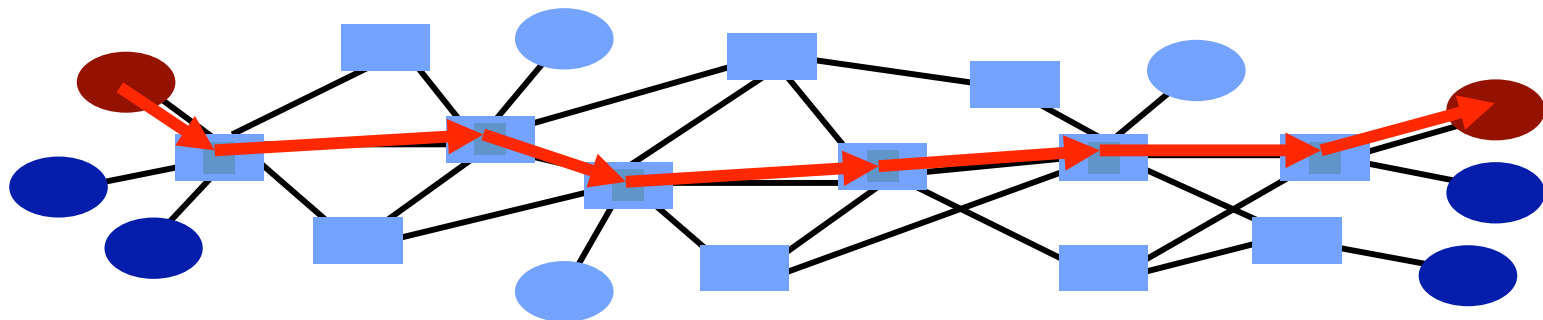
- **Guaranteed bandwidth**
 - Predictable communication performance
 - Not “best-effort” delivery with no real guarantees
- **Simple abstraction**
 - Reliable communication channel between hosts
 - No worries about lost or out-of-order packets
- **Simple forwarding**
 - Forwarding based on time slot 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

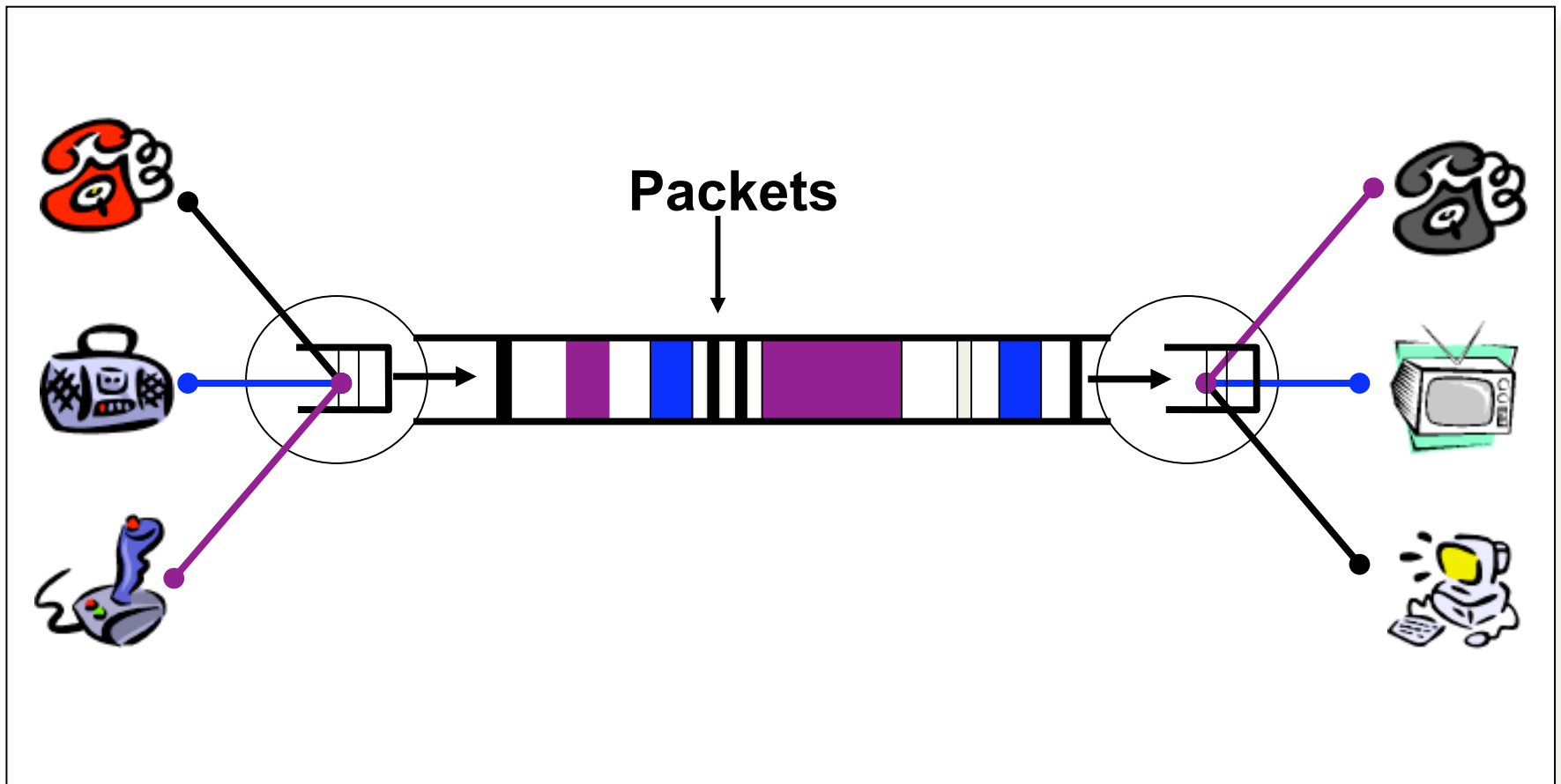
- **Wasted bandwidth**
 - Bursty traffic leads to idle conn during silent period
 - Unable to achieve gains from statistical multiplexing
- **Blocked connections**
 - Connection refused when resources are not sufficient
 - Unable to offer “okay” service to everybody
- **Connection set-up delay**
 - No communication until the connection is set up
 - Unable to avoid extra latency for small data transfers
- **Network state**
 - Network nodes must store per-connection information
 - Unable to avoid per-connection storage and state

Packet Switching (e.g., Internet)

- Data traffic divided into packets
 - Each packet contains a header (with address)
- Packets travel separately through network
 - Packet forwarding based on the header
 - Network nodes may store packets temporarily
- Destination reconstructs the message

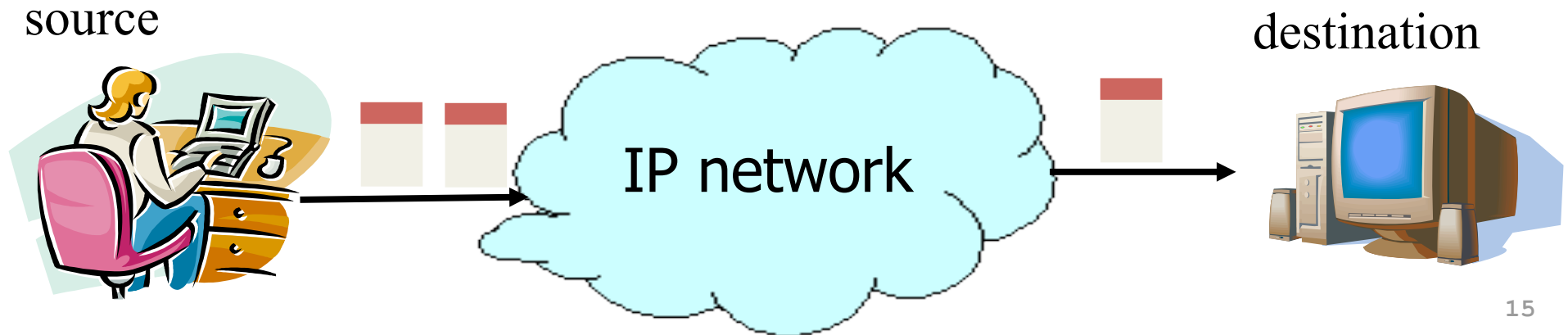


Packet Switching: Statistical Multiplexing



IP Service: Best-Effort Packet Delivery

- **Packet switching**
 - Divide messages into a sequence of packets
 - Headers with source and destination address
- **Best-effort delivery**
 - Packets may be lost
 - Packets may be corrupted
 - Packets may be delivered out of order



IP Service Model: Why Packets?

- **Data traffic is bursty**
 - Logging in to remote machines
 - Exchanging e-mail messages
- **Don't want to waste bandwidth**
 - No traffic exchanged during idle periods
- **Better to allow multiplexing**
 - Different transfers share access to same links
- **Packets can be delivered by most anything**
 - RFC 1149: IP Datagrams over Avian Carriers
- **... still, packet switching can be inefficient**
 - Extra header bits on every packet



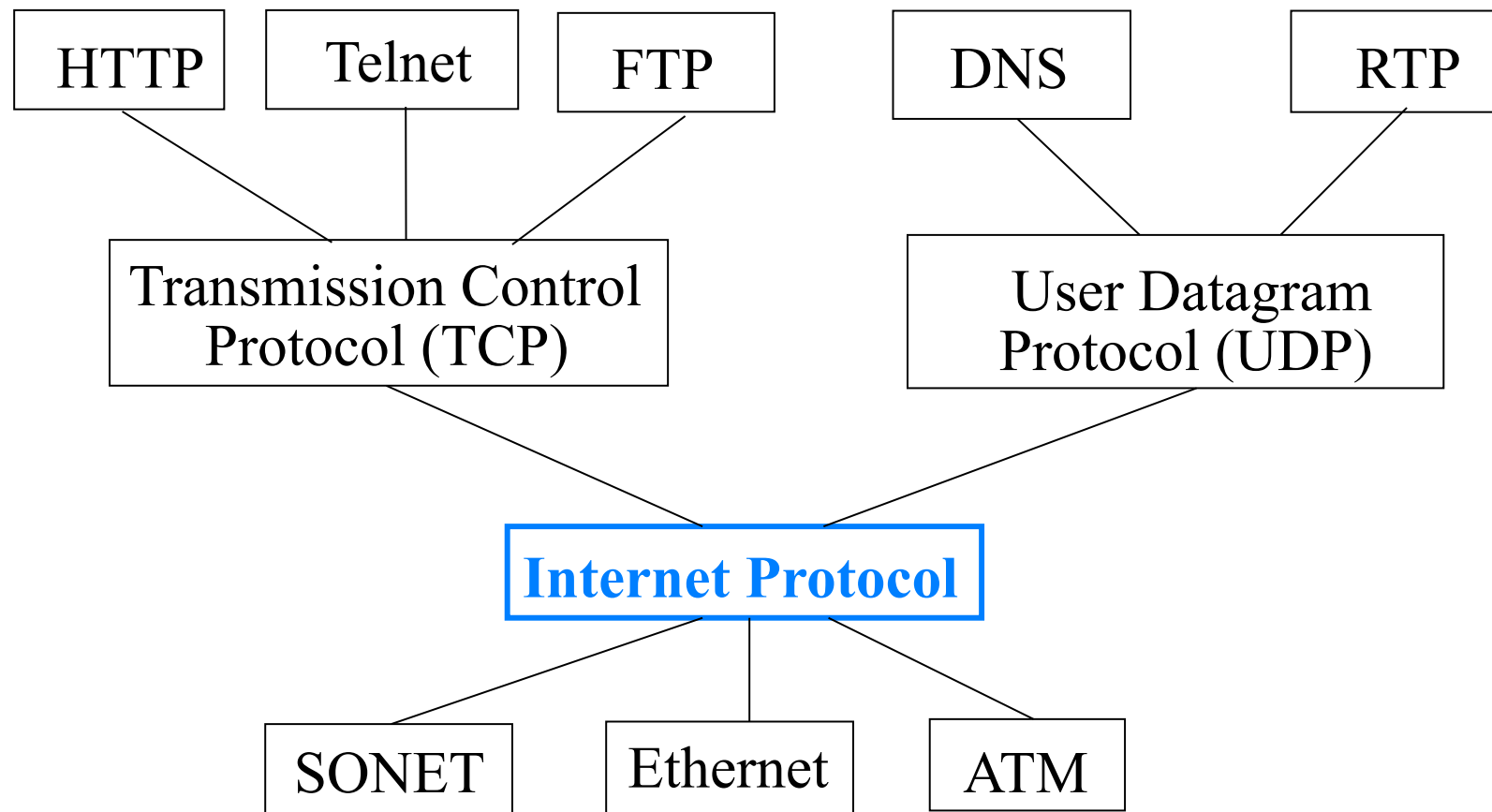
IP Service Model: Why Best-Effort?

- IP means never having to say you're sorry...
 - Don't need to reserve bandwidth and memory
 - Don't need to do error detection & correction
 - Don't need to remember from one packet to next
- Easier to survive failures
 - Transient disruptions are okay during failover
- ... but, applications *do* want efficient, accurate transfer of data in order, in a timely fashion

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

Layering in the IP Protocols



History: Why IP Packets?

- IP proposed in the early 1970s
 - Defense Advanced Research Project Agency (DARPA)
- Goal: connect existing networks
 - To develop an effective technique for multiplexed utilization of existing interconnected 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
 - Previous DARPA project
 - Demonstrated 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

4-bit Version	4-bit Header Length	8-bit Type of Service (TOS)	16-bit Total Length (Bytes)	
16-bit Identification			3-bit Flags	13-bit Fragment Offset
8-bit Time to Live (TTL)		8-bit Protocol	16-bit Header Checksum	
32-bit Source IP Address				
32-bit Destination IP Address				
Options (if any)				
Payload				

IP Header: Version, Length, ToS

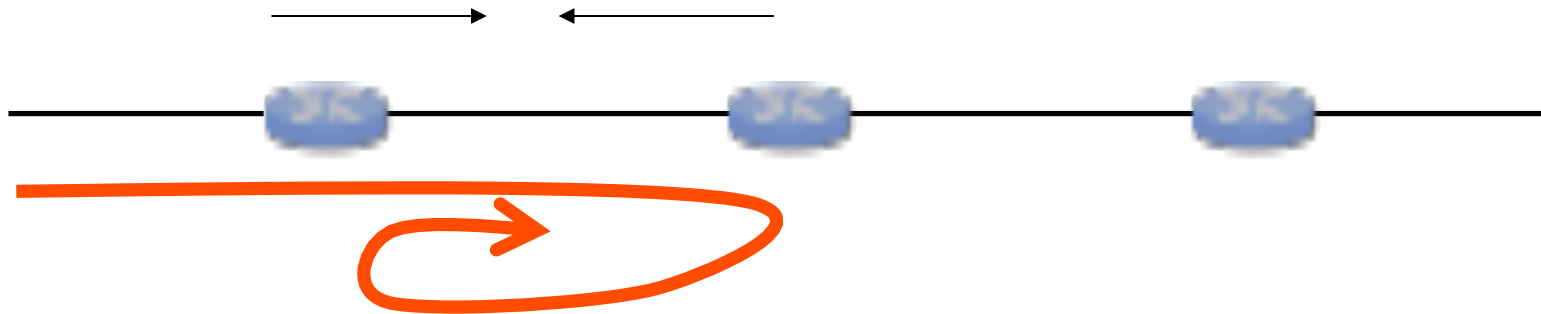
- **Version number (4 bits)**
 - Indicates the version of the IP protocol
 - Necessary to know what other fields to expect
 - Typically “4” (for IPv4), and sometimes “6” (for IPv6)
- **Header length (4 bits)**
 - Number of 32-bit words in the header
 - Typically “5” (for a 20-byte IPv4 header)
 - Can be more when “IP options” are used
- **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

IP Header: Length, Fragments, TTL

- **Total length (16 bits)**
 - Number of bytes in the packet
 - Maximum size is 63,535 bytes ($2^{16} - 1$)
 - ... though underlying links may impose harder limits
- **Fragmentation information (32 bits)**
 - Packet identifier, flags, and fragment offset
 - Supports dividing a large IP packet into fragments
 - ... in case a link cannot handle a large IP packet
 - ... so why do we typically send max MTU packets?
- **Time-To-Live (8 bits)**
 - Used to identify packets stuck in forwarding loops
 - ... and eventually discard them from the network

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

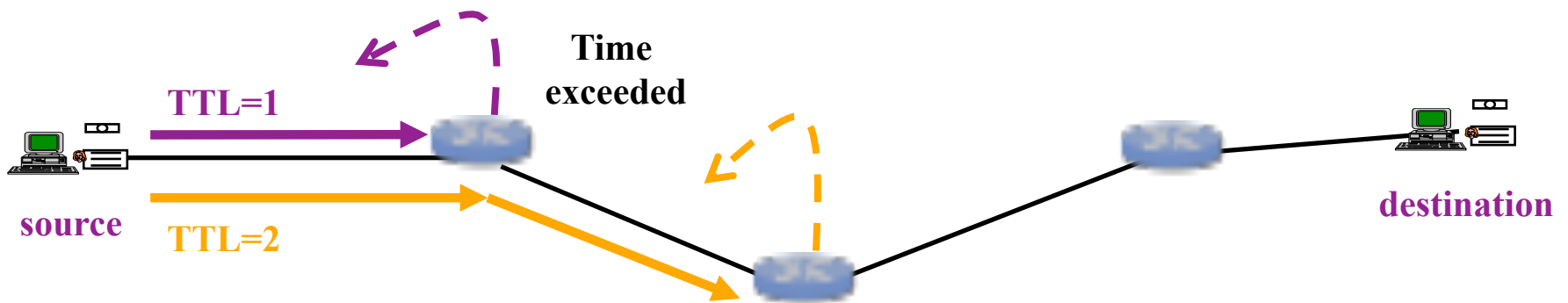
- **Potential robustness problem**
 - Forwarding loops can cause packets to cycle forever
 - Confusing if the packet arrives much later



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

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**

Example Traceroute: Berkeley to CNN

Hop number, IP address, DNS name

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 response from router

No name resolution

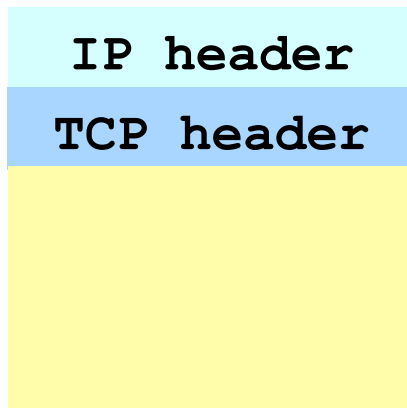
Try Running Traceroute Yourself

- On UNIX machine
 - Traceroute
 - E.g., “traceroute cnn.com” or “traceroute 12.1.1.1”
- On Windows machine
 - Tracert
 - E.g., “tracert cnn.com” or “tracert 12.1.1.1”
- Common uses of traceroute
 - Discover the topology of the Internet
 - Debug performance and reachability problems

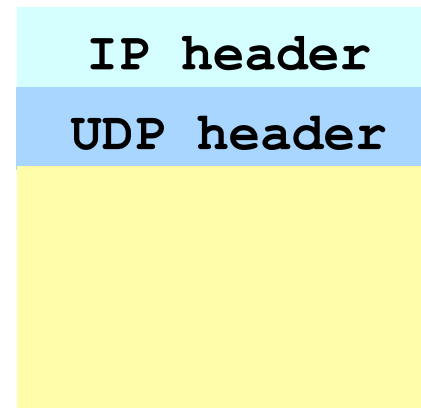
IP Header Fields: Transport Protocol

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

`protocol=6`



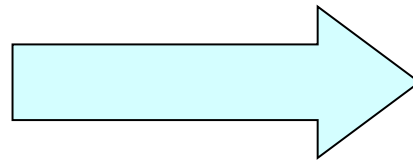
`protocol=17`



IP Header: Checksum on the Header

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

$$\begin{array}{r} 134 \\ + 212 \\ \hline = 346 \end{array}$$



$$\begin{array}{r} 134 \\ + 216 \\ \hline = 350 \end{array}$$

Mismatch!

IP Header: To and From Addresses

- **Two IP addresses**
 - Source IP address (32 bits)
 - Destination IP address (32 bits)
- **Destination address**
 - Unique identifier for the receiving host
 - Allows each node to make forwarding decisions
- **Source address**
 - Unique identifier for the sending host
 - Recipient can decide whether to accept packet
 - 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, anyway?
 - You could send packets with any source you want
- Why would someone want to do this?
 - Launch a denial-of-service attack
 - Send excessive packets to the destination
 - ... to overload the node, or the links leading to the node
 - Evade detection by “spoofing”
 - But, the victim could identify you by the source address
 - So, you can put someone else's source address in the packets
 - Also, an attack against the spoofed host
 - Spoofed host is wrongly blamed
 - Spoofed host may receive return traffic from the receiver

Summary: Packet Switching Review

- **Efficient**
 - Can send from any input that is ready
- **General**
 - Multiple types of applications
- **Accommodates bursty traffic**
 - Addition of queues
- **Store and forward**
 - Packets are self contained units
 - Can use alternate paths – reordering
- **Contention (i.e., no isolation)**
 - Congestion
 - Delay

Next Lecture

- IP routers
 - Packet forwarding
 - Components of a router
- Reading for this week
 - Chapter 3: Sections 3.1 and 3.4
 - Chapter 4: Sections 4.1.1 -- 4.1.4
- Please subscribe to the course mailing list
 - <https://lists.cs.princeton.edu/mailman/listinfo/cos461>