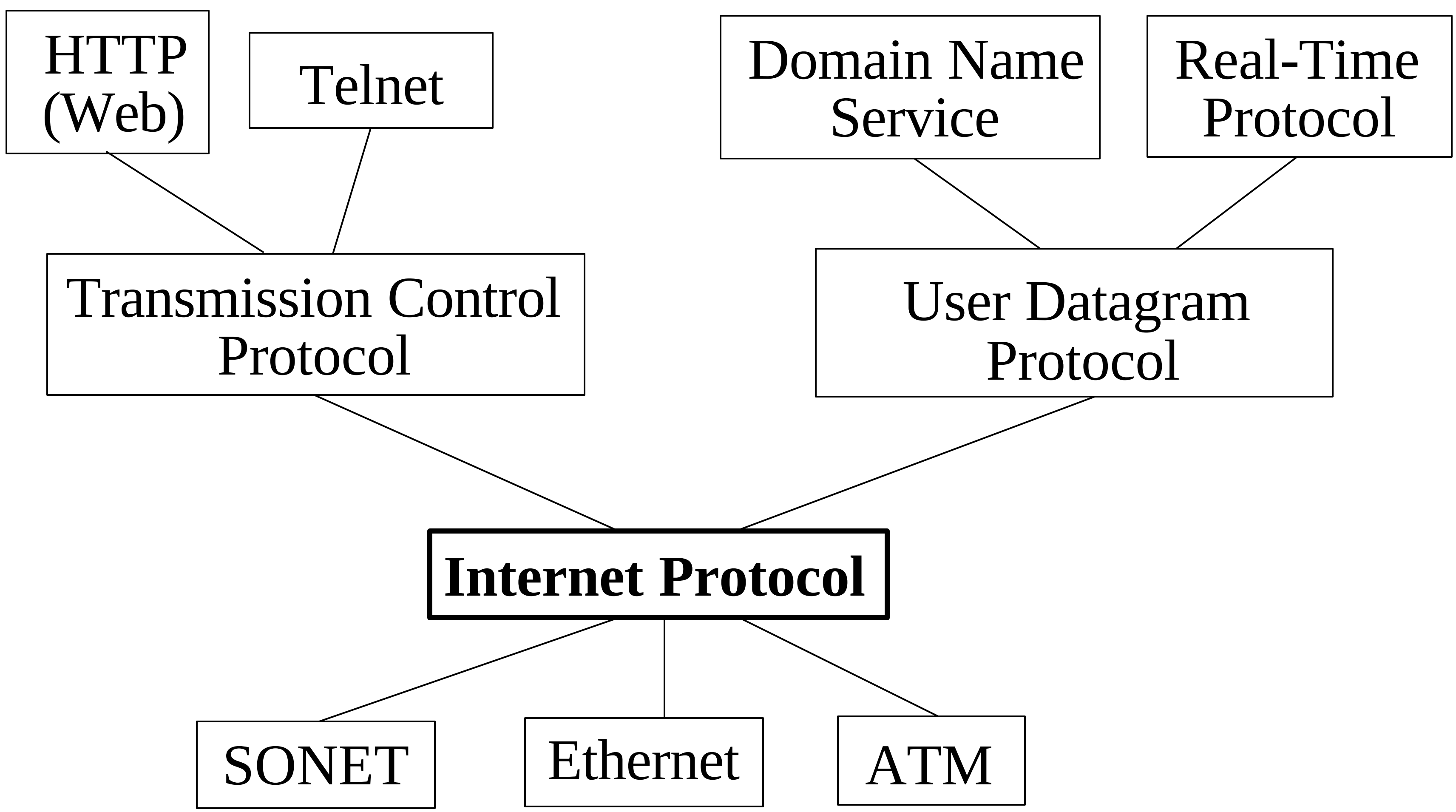
CS155a: E-Commerce

Lecture 4: Sept. 18, 2001

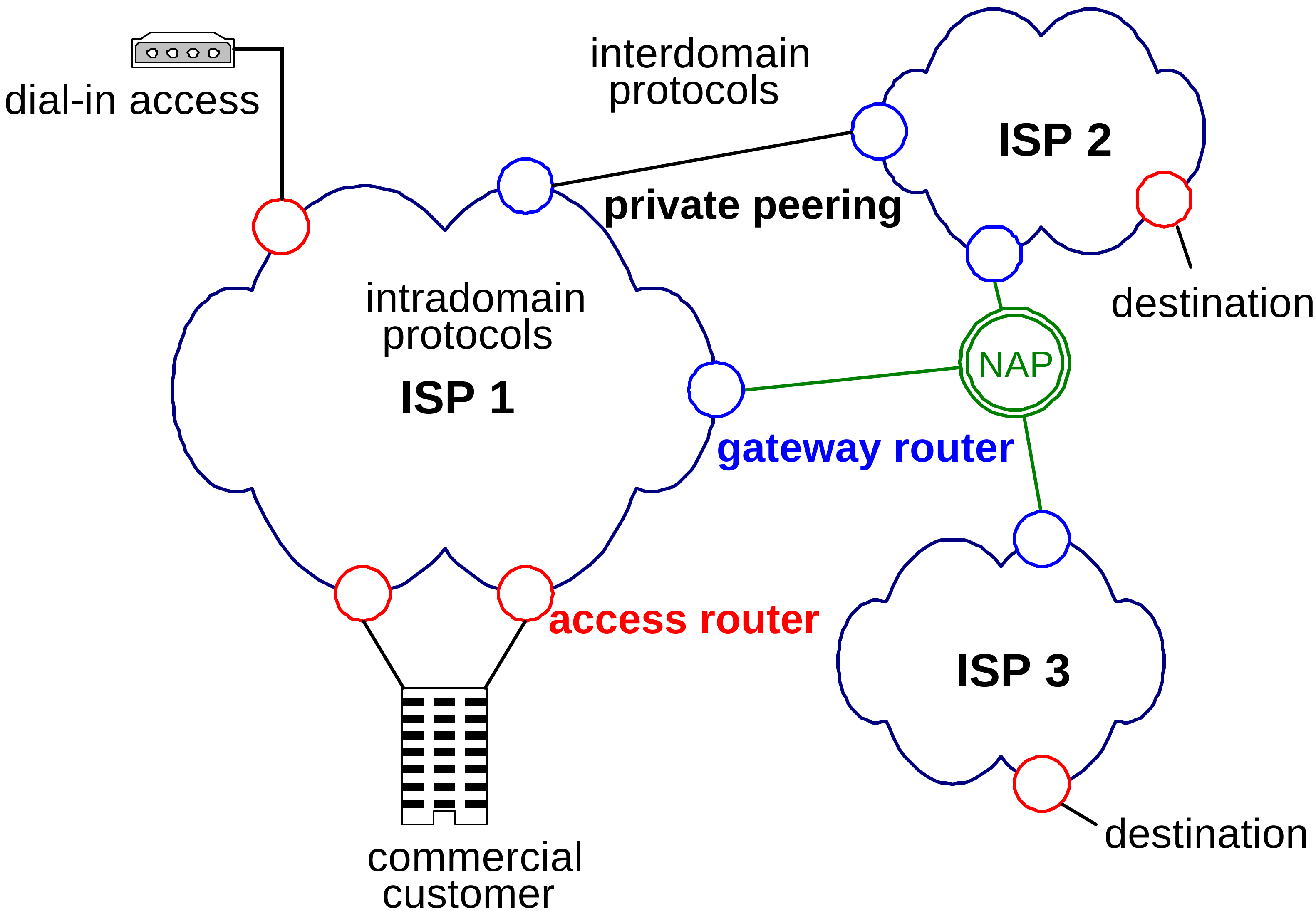
How Does the Internet Work?
(continued)

Acknowledgements: J. Rexford and V. Ramachandran

Layering in the IP Protocols



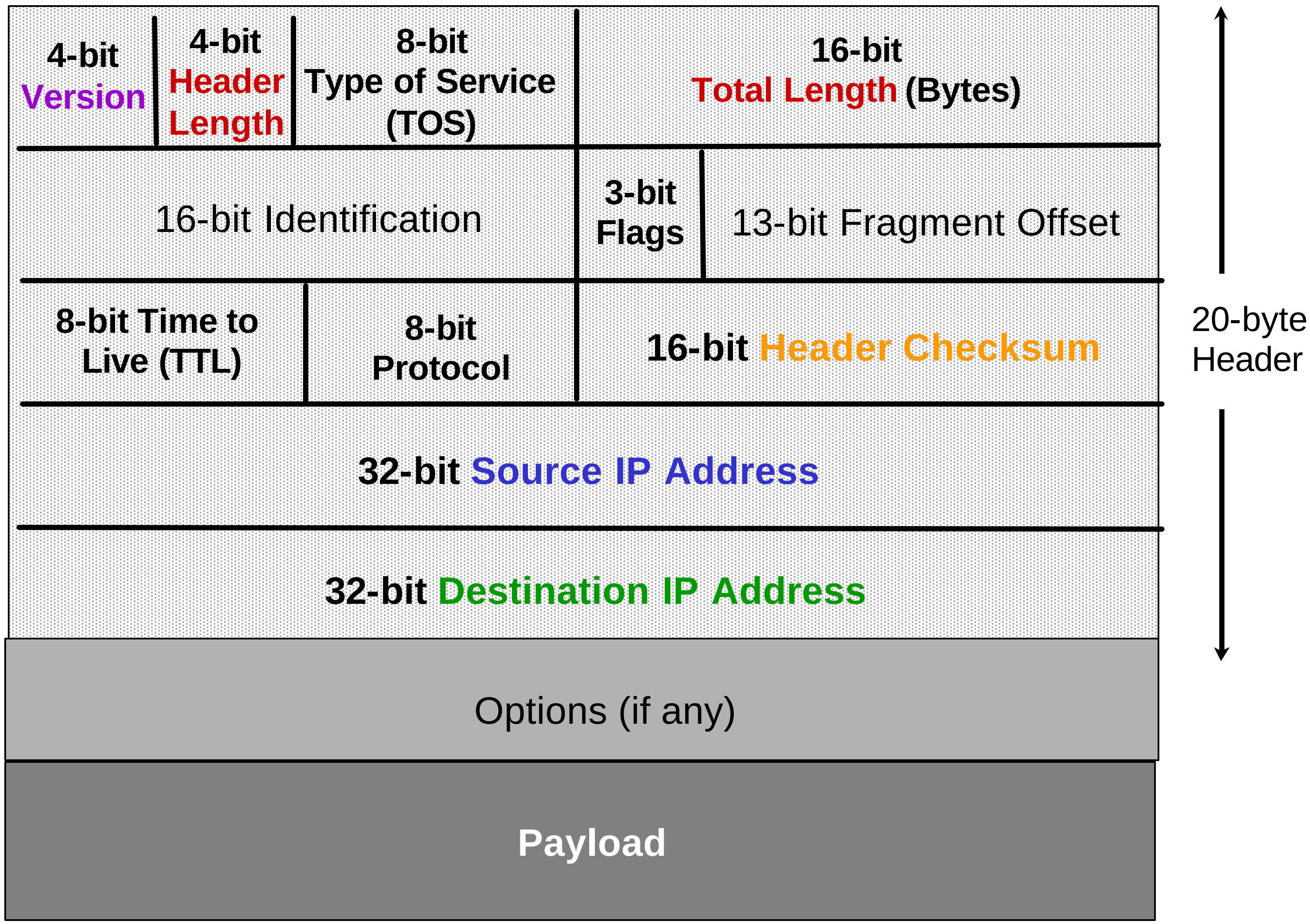
Internet Architecture



IP Connectionless Paradigm

- No error detection or correction for packet data
  - 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")
  - Send can slow down in response to loss or delay

IP Packet Structure



Main IP Header Fields

- **Version number** (e.g., version 4, version 6)
- **Header length** (number of 4-byte words)
- **Header checksum** (error check on header)
- **Source** and **destination** IP addresses
- Upper-level protocol (e.g., TCP, UDP)
- **Length** in bytes (up to 65,535 bytes)
- IP options (security, routing, timestamping, etc.)

Time-to-Live Field

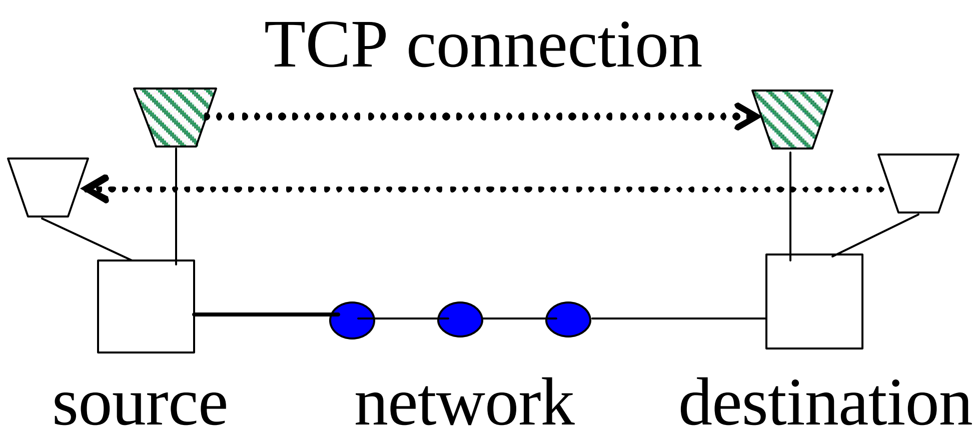
- Potential robustness problem
  - What happens if a packet gets stuck in a routing loop?
  - What happens if the packet arrives **much** later?
- Time-to-live field in packet header
  - TTL field decremented by each router on the path
  - Packet is discarded when TTL field reaches 0
  - Discard generates "timer expired" message to source
- Expiry message exploited in **traceroute** tool
  - Generate packets with TTL of i=1, 2, 3, 4, …
  - Extract router id from the "timer expired" message
  - Provides a way to gauge the path to destination

Type-of-Service Bits

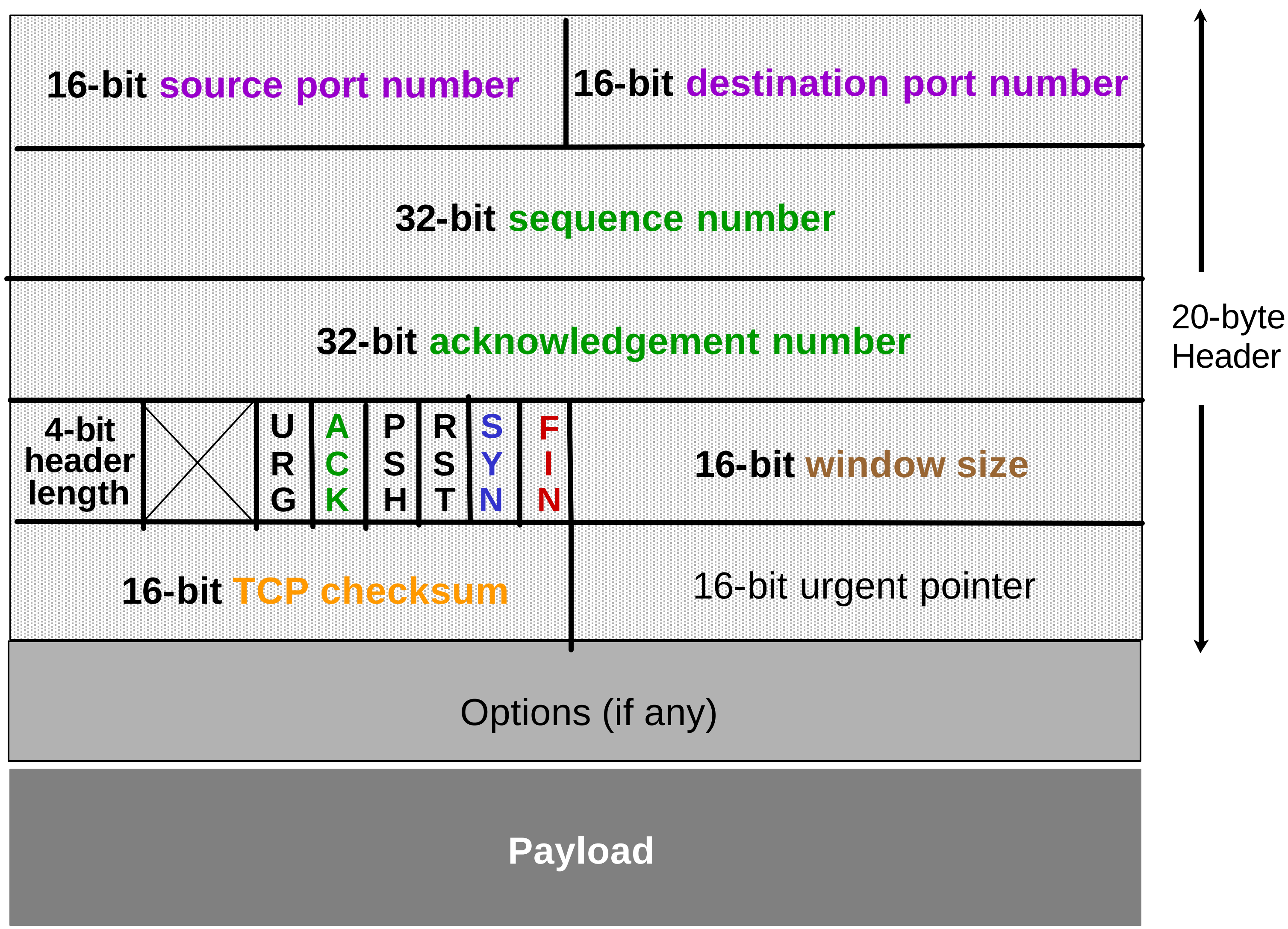
- Initially, envisioned for type-of-service routing
  - Low-delay, high-throughput, high-reliability, etc.
  - However, current IP routing protocols are static
  - And, most routers have first-in-first-out queuing
  - So, the ToS bits are ignored in most routers today
- Now, heated debate for differentiated services
  - ToS bits used to define a small number of classes
  - Affect router packet scheduling and buffering polices
  - Arguments about consistent meaning across networks

Transmission Control Protocol (TCP)

- Byte-stream socket abstraction for applications
- **Retransmission** of lost or corrupted packets
- **Flow-control** to respond to network congestion
- Simultaneous transmission in both directions
- **Multiplexing** of multiple logical connections

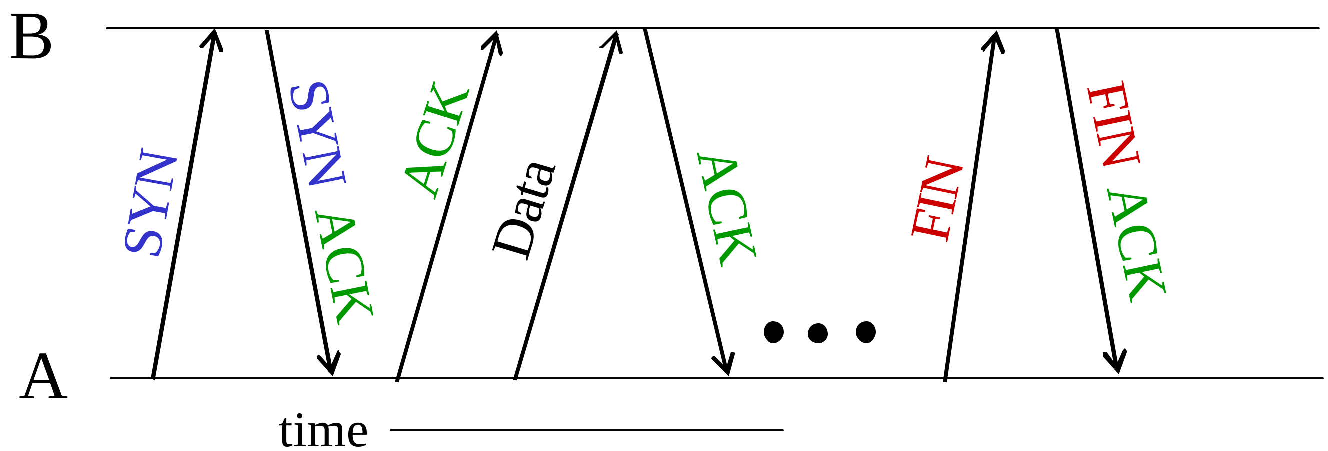


TCP Header



16-bit source port number
16-bit destination port number
32-bit sequence number
32-bit acknowledgement number
4-bit header length
U R G
A C K
P S H
R S T
S Y N
F I N
16-bit window size
16-bit TCP checksum
16-bit urgent pointer
Options (if any)
Payload
20-byte Header

Establishing a TCP Connection



- Three-way handshake to establish connection
  - Host A sends a SYN (open) to the host B
  - Host B returns a SYN acknowledgement (ACK)
  - Host A sends an ACK to acknowledge the SYN ACK
- Closing the connection
  - Finish (FIN) to close and receive remaining bytes (and other host sends a FIN ACK to acknowledge)
  - Reset (RST) to close and not receive remaining bytes

Lost and Corrupted Packets

- Detecting corrupted and lost packets
  - Error detection via **checksum** on header and data
  - Sender sends packet, sets timeout, and waits for *ACK*
  - Receiver sends ACKs for received packets
- Retransmission from sender
  - Sender retransmits lost/corrupted packets
  - Receiver reassembles and reorders packets
  - Receiver discards corrupted and duplicated packets

Packet loss rates are high (e.g., 10%), causing significant delay (especially for short Web transfers)!

TCP Flow Control

- Packet loss used to indicate network congestion
  - Router drop packets when buffers are (nearly) full
  - Affected TCP connection reacts by backing-off
- **Window-based** flow control
  - Sender limits number of outstanding bytes
  - Sender reduces **window size** when packets are lost
  - Initial slow-start phase to learn a good window size
- TCP flow-control header fields
  - **Window size** (maximum # of outstanding bytes)
  - **Sequence number** (byte offset from starting #)
  - **Acknowledgement number** (cumulative bytes)

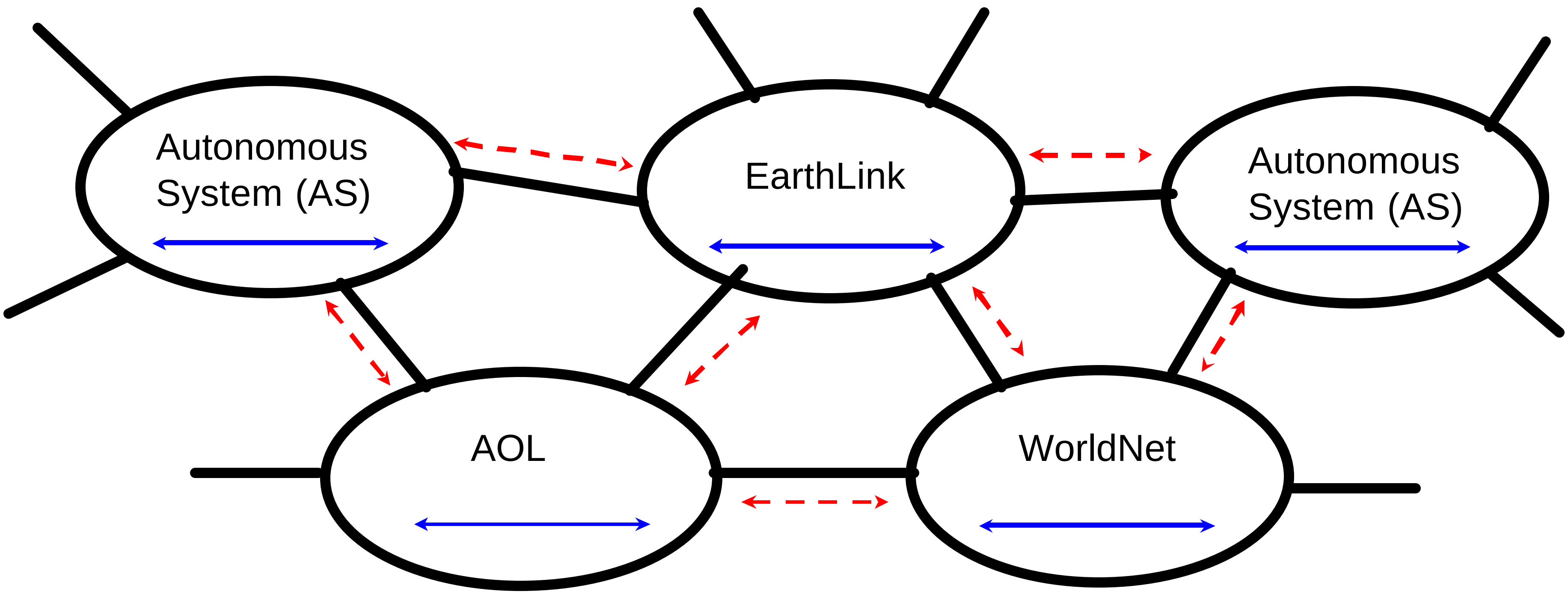
User Datagram Protocol (UDP)

- Some applications do not want or need TCP
  - Don’t need recovery from lost or corrupted packets
  - Don’t want flow control to respond to loss/congestion
- Amount of UDP packets is rapidly increasing
  - Commonly used for multimedia applications
  - UDP traffic interferes with TCP performance
  - But, many firewalls do not accept UDP packets
- Dealing with the growth in UDP traffic
  - Pressure for applications to apply flow control
  - Future routers may enforce “TCP-like” behavior
  - Need better mathematical models of TCP behavior

Getting an IP Packet From A to B

- Host must know at least three IP addresses
  - Host IP address (to use as its own source address)
  - Domain Name Service (to map names to addresses)
  - Default router to reach other hosts (e.g., gateway)
- Simple customer/company
  - Connected to a single service provider
  - Has just one router connecting to the provider
  - Has a set of IP addresses allocated in advance
  - Does not run an Internet routing protocol

Connecting Networks



Autonomous System: A collection of IP subnets and routers under the same administrative authority.

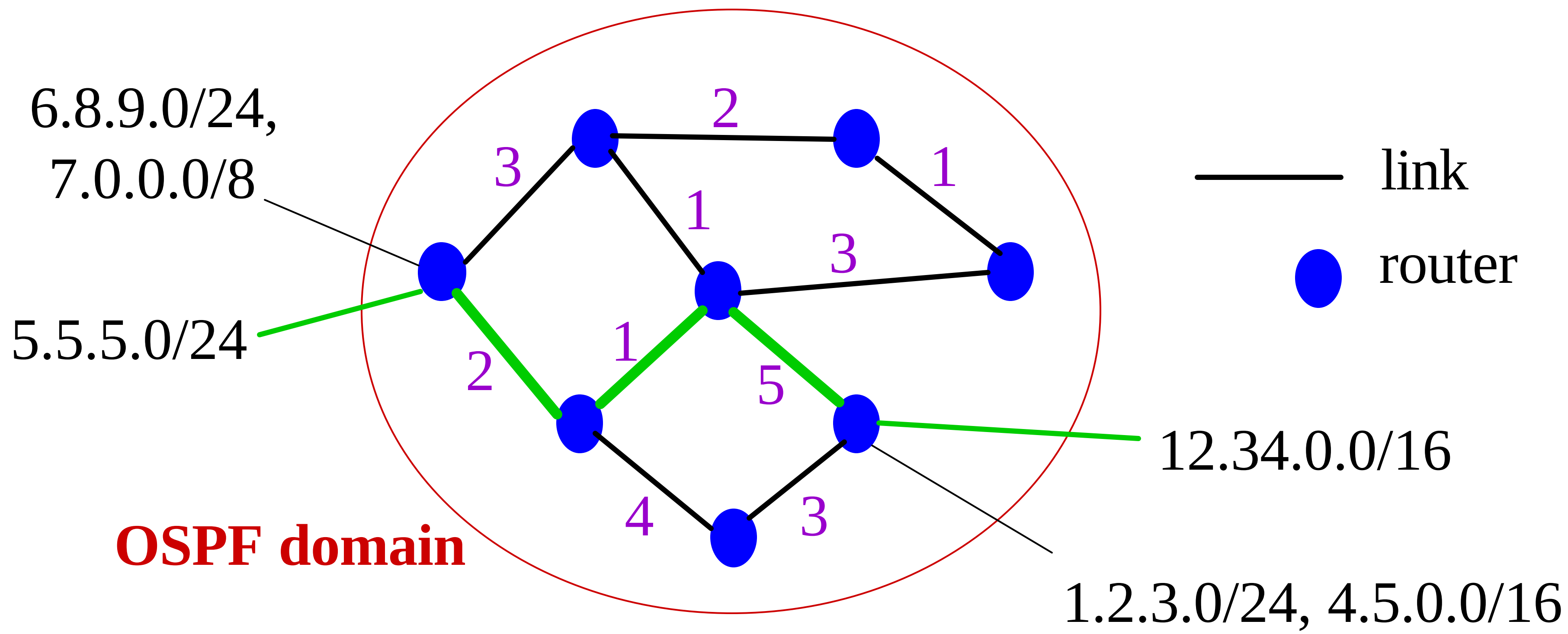
Interior Routing Protocol (e.g., Open Shortest Path First)

Exterior Routing Protocol (e.g., Border Gateway Protocol)

Open Shortest-Path First (OSPF) Routing

- Network is a graph with **routers** and **links**
  - Each unidirectional link has a **weight** (1-63,535)
  - **Shortest-path** routes from sum of link weights
- **Weights** are assigned statically (configuration file)
  - Weights based on capacity, distance, and traffic
  - Flooding of info about weights and IP addresses
- Large networks can be divided into multiple **domains**

Example Network and Shortest Path



IP Routing in OSPF

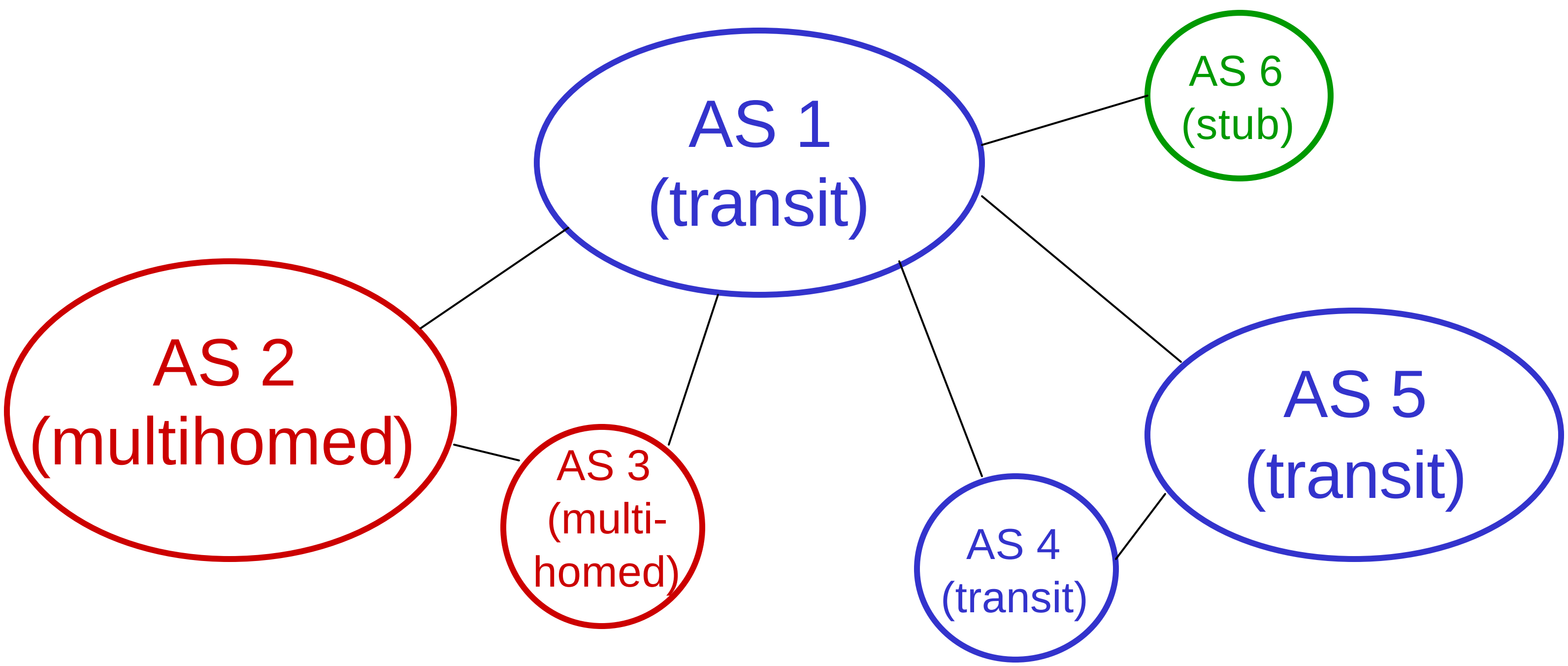
- Each router has a **complete view of the topology**
  - Each router transmits information about its links
  - Reliable flooding to all routers in the domain
  - Updates periodically or on link failure/installation
- Each router computes **shortest path(s)**
  - Maintenance of a complete link-state database
  - Execution of Dijkstra's shortest-path algorithm
- Each router constructs a **forwarding table**
  - Forwarding table with next hop for each destination
  - Hop-by-hop routing independently by each router

Routing Software

- Routing protocol software
  - Checking connection with neighboring routers (“hello”)
  - Exchanging link-state information with other routers
  - Computing shortest paths and IP forwarding table
  - Handling of packets with IP options selected
  - Exchanging routing information between providers
- Router management and configuration
  - Configuration files to configure addresses, routing, etc.
  - Command-line interface to inspect/change configuration
  - Logging of statistics in management information base
  - More complex traffic measurement (e.g., NetFlow)

Border Gateway Protocol (BGP)

- BGP routes traffic through a network where the AS's can be connected in any way.
- Three types of AS's: **stub** (local traffic only); **multihomed** (multiple connections but local traffic only); **transit** ("thru" and local traffic).



Border Gateway Protocol (BGP)

- **Reachability:** from one AS, what other AS's can be reached from it?
- Every AS has a **BGP Speaker** node that advertises its reachability info by sending **complete paths** to reachable networks.
- Given advertised updates, we calculate **loop-free** routes to networks.
- Problem of scale: too many networks; don't know how an AS works, so it's hard to determine cost to send through each.

Connecting With Our Neighbors

- Public peering
  - Network Access Points (e.g., MAE East, MAE West)
  - Public location for connecting routers
  - Routers exchange data and routing information
- Private peering
  - Private connections between two peers (e.g., MCI)
  - Private peers exchange direct traffic (no transit)
  - Private peers must exchange similar traffic volumes
- Transit networks
  - Provider pays another for transit service (e.g., BBN)
  - Improve performance and reach more addresses

Reference

- For more information, see:

  Peterson and Davie, Computer Networks: A Systems Approach. Morgan Kaufmann Publishers, 1999.

Reading Assignment For September 20

- "The Heavenly Jukebox," Charles C. Mann, The Atlantic Monthly, September 2000. (http://www.theatlantic.com/issues/2000/09/mann.htm)
- Addendum to Chapter 4 (Sections 106, 107, 109 of U.S. Copyright Law) and Chapter 5 of The Digital Dilemma. (http://books.nap.edu/html/digital_dilemma/)
- (Optional) Chapter 1 and Appendix E of The Digital Dilemma.