CS155b: E-Commerce

Lecture 3: Jan. 21, 2003

How Does the Internet Work?
(continued)

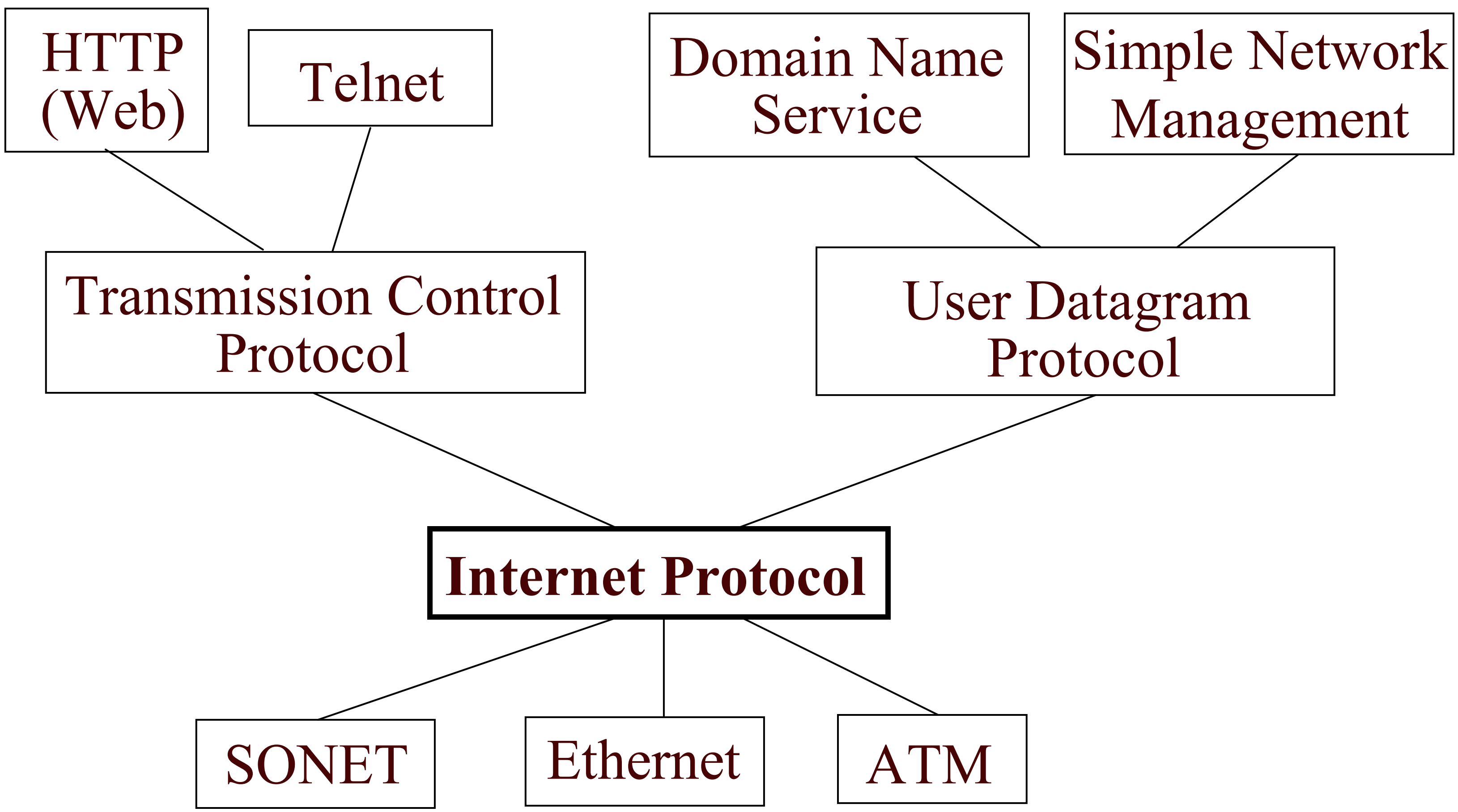
Acknowledgements: J. Rexford and V. Ramachandran

Announcement

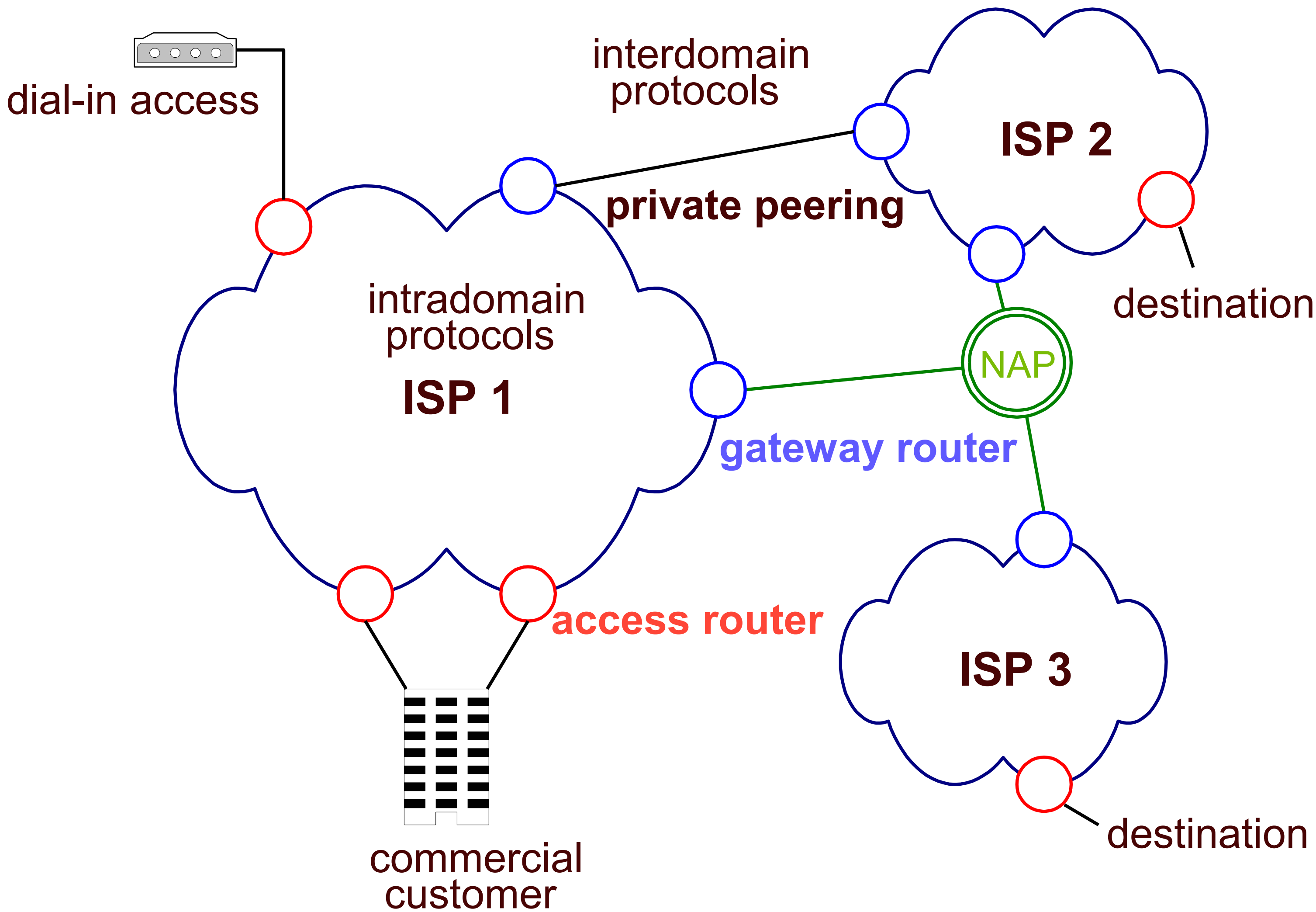
Professor Feigenbaum's office hours are canceled on Thursday, 1/23.

The TA will hold usual office hours on Wednesday, 1/22, from 3-4pm.

Layering in the IP Protocols

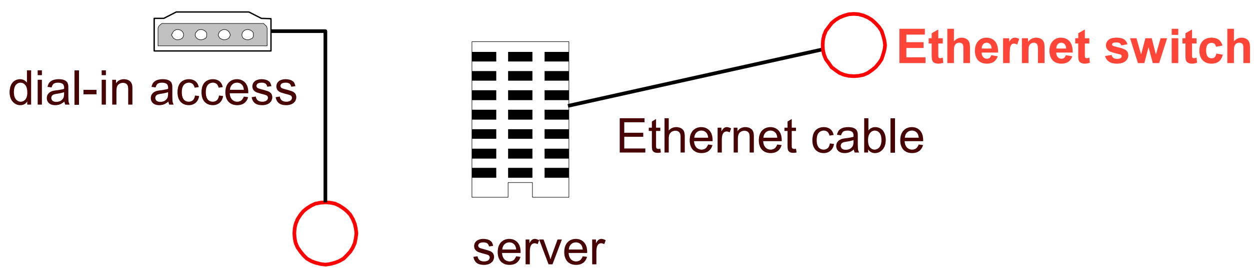


Internet Architecture



The Physical Layer

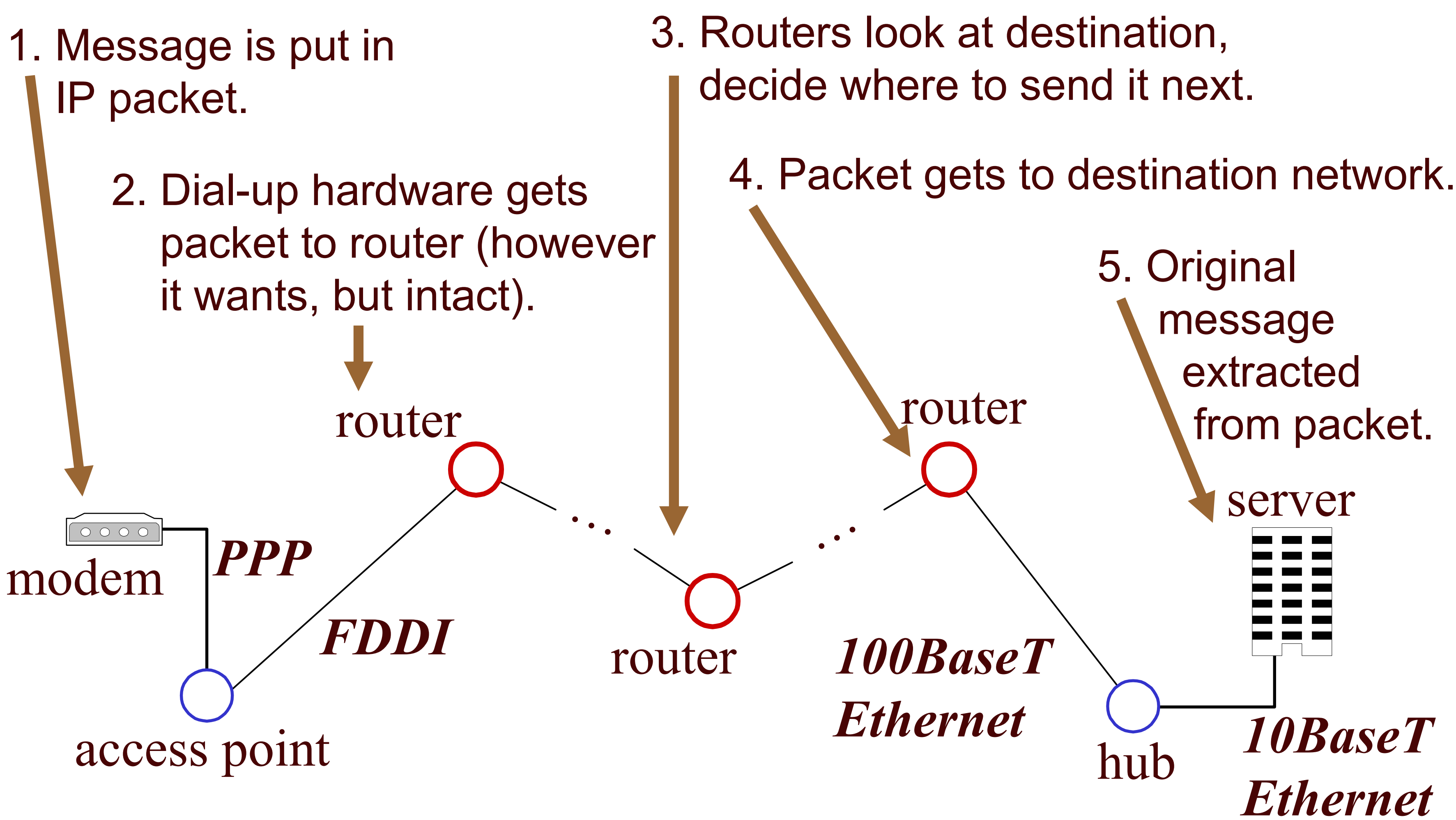
- A network spans different hardware.



- Physical components can work however they want, as long as the **interface between them is consistent**.
- Then, different hardware can be connected.

The Role of the IP Layer

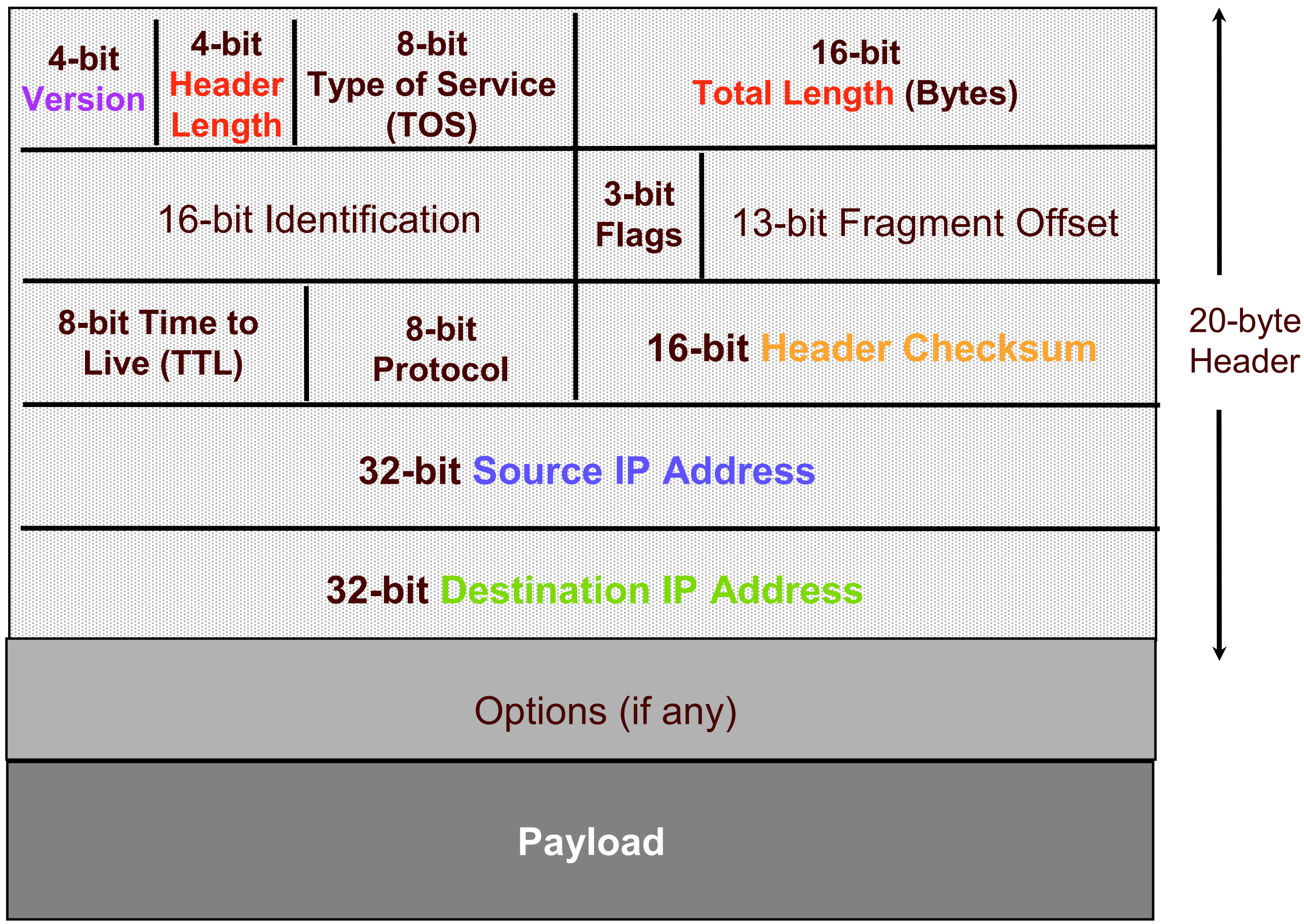
- **Internet Protocol (IP):** gives a standard way to "package" messages across different hardware types.



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

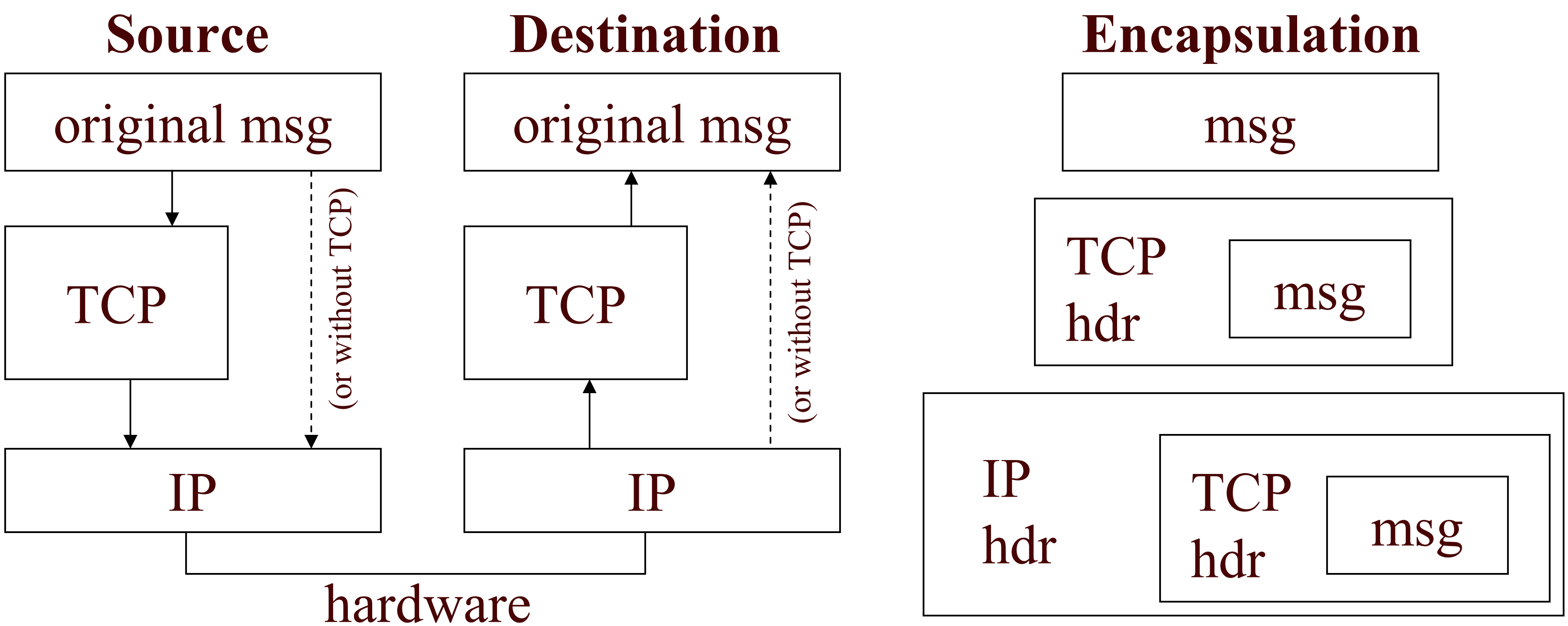


Main IP Header Fields

- 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.)
- TTL (prevents messages from looping around forever; packets "die" if they "get lost")

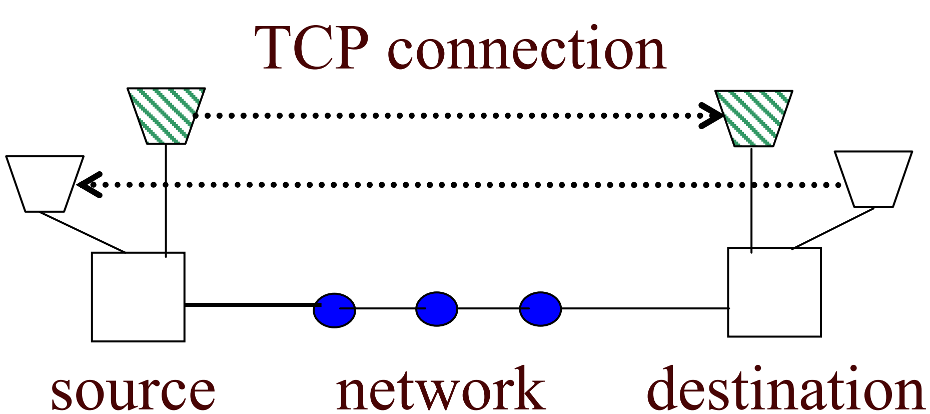
Adding Some Functionality

- More guarantees, *e.g.*, that packets go in order, require more work at both ends.
- Solution: add another layer (*e.g.*, TCP)

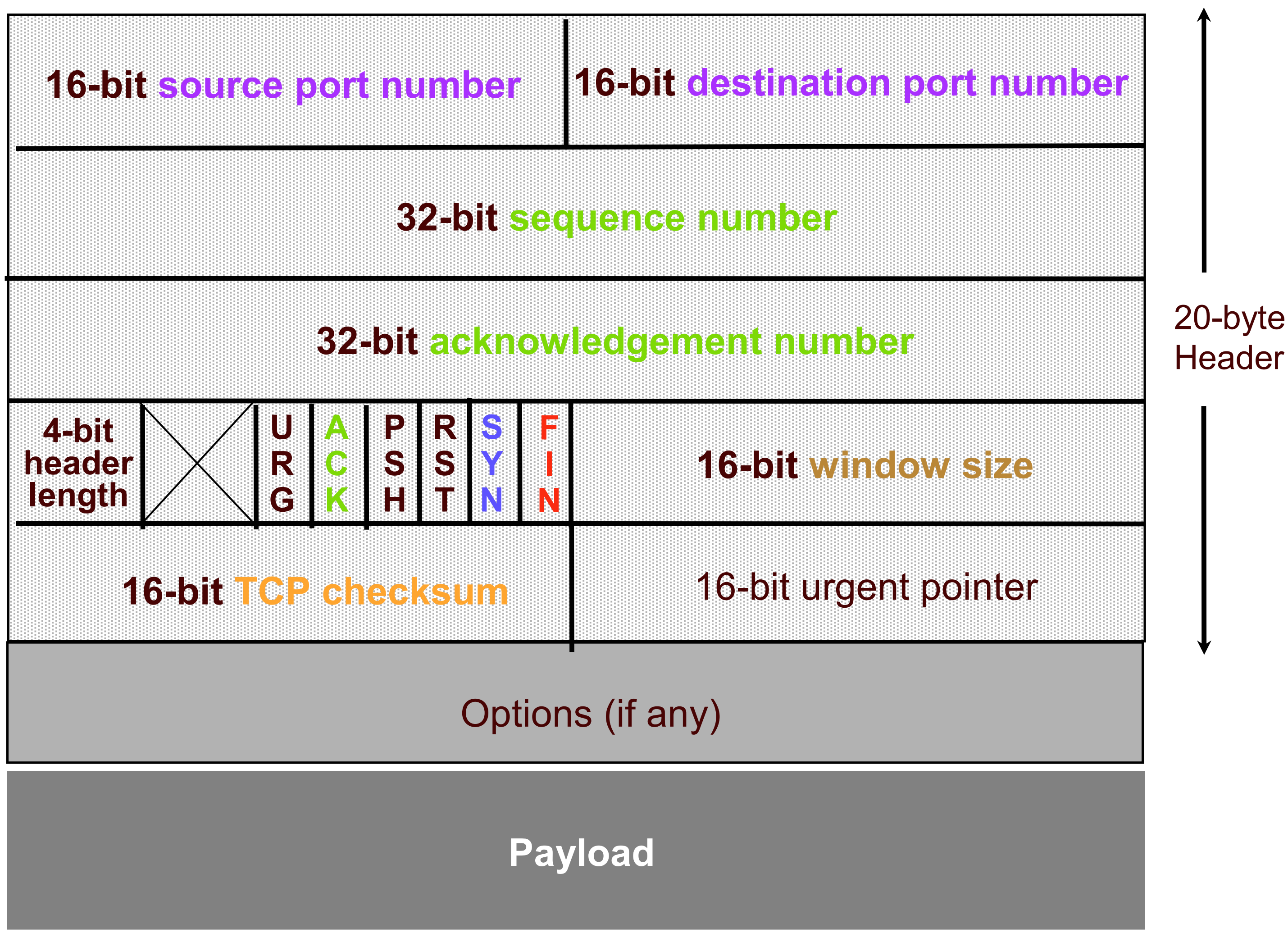


Transmission Control Protocol (TCP)

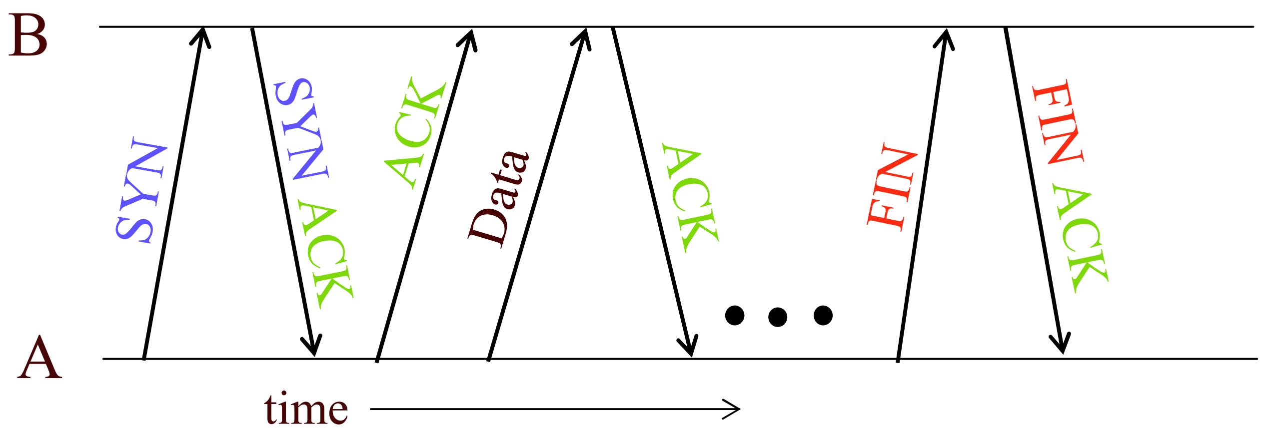
- Byte-stream socket abstraction for applications
- **Retransmission** of lost or corrupted packets
- **Flow-control** to respond to network congestion
- Simultaneous transmission in both directions
- **Multiplexing** of multiple logical connections



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 drops 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)

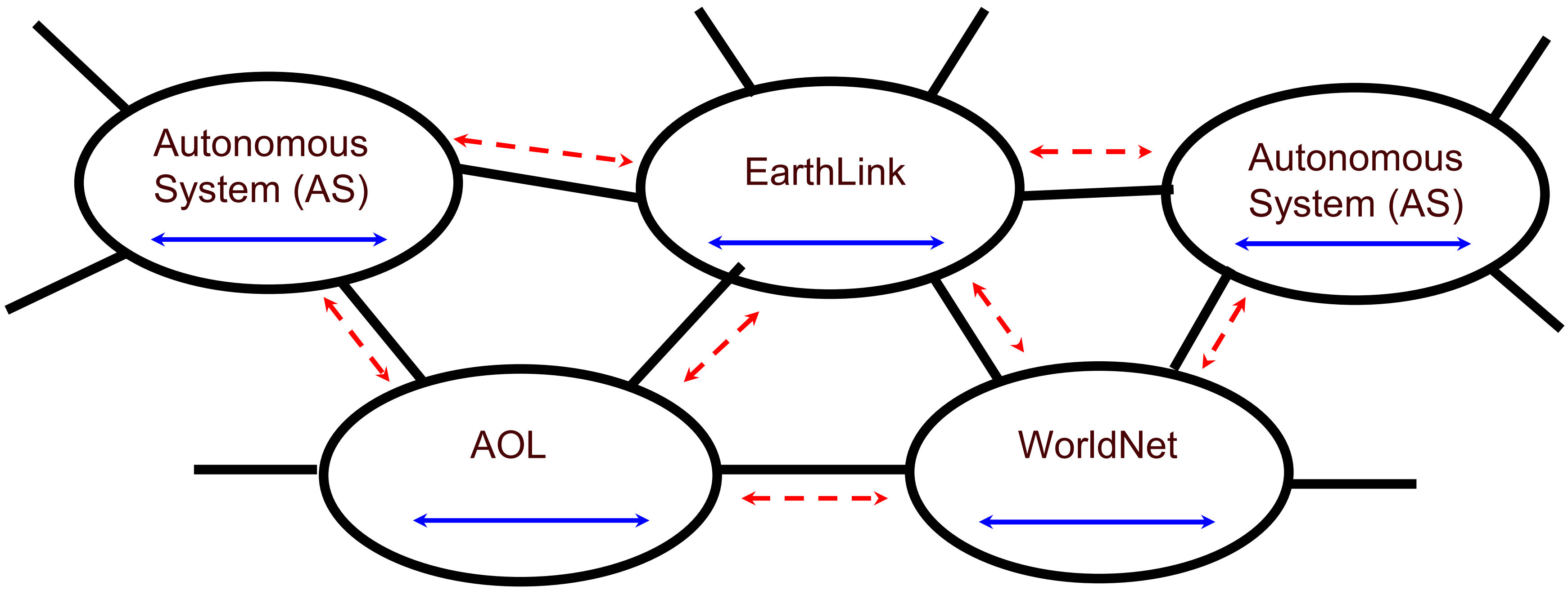
User Datagram Protocol (UDP)

- 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
- Fraction 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 from A to B: Summary

- Need IP addresses for:
  - Self (to use as source address)
  - DNS Server (to map names to addresses)
  - Default router to reach other hosts (e.g., gateway)
- Use DNS to get destination address
- Pass message through TCP/IP handler
- Send it off! **Routers** will do the work:
  - Physically connecting different networks
  - Deciding where to next send packets **(HOW??)**

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)

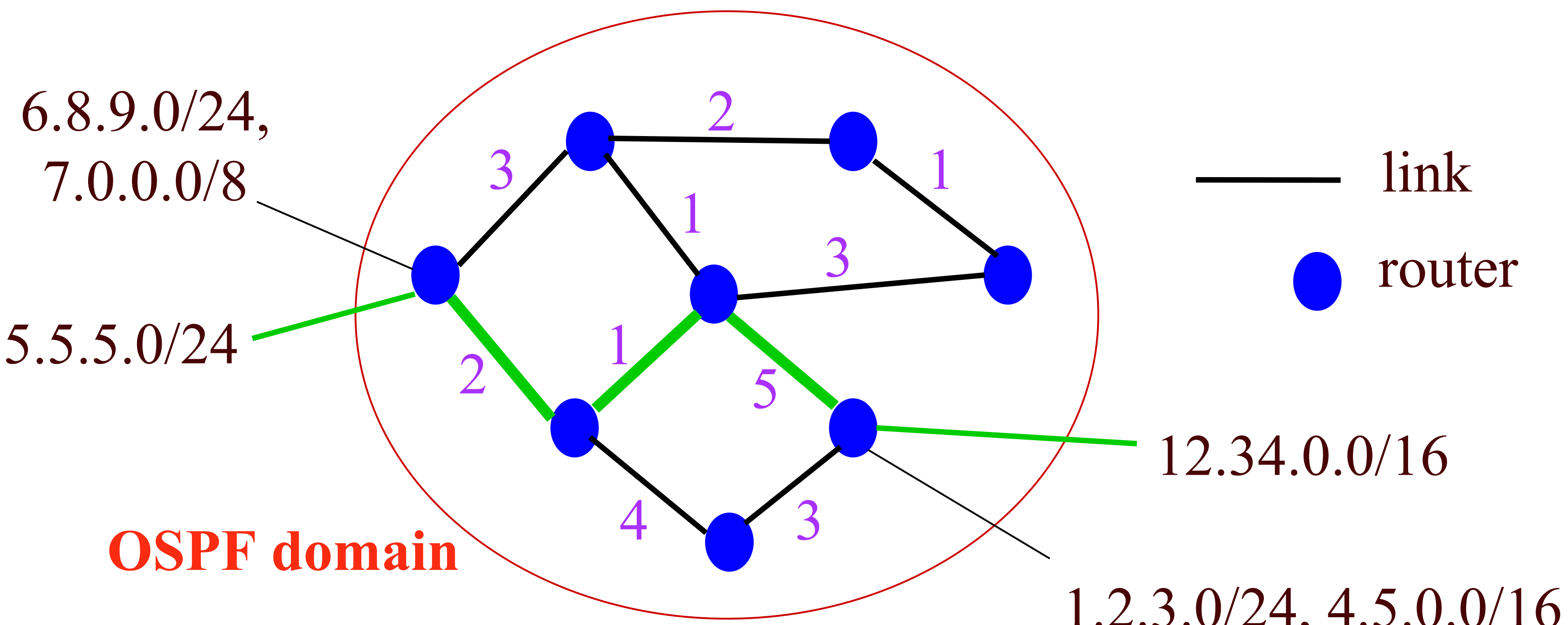
Where to Go Next

- Routers contain a **forwarding table** that pairs destination with next hop (on what physical wire to send msg.).
- The table gets populated with information learned **internally** (*e.g.*, OSPF) and **externally** (*e.g.*, BGP).
- OSPF and BGP are protocols that communicate ***knowledge about destinations*** between routers.

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



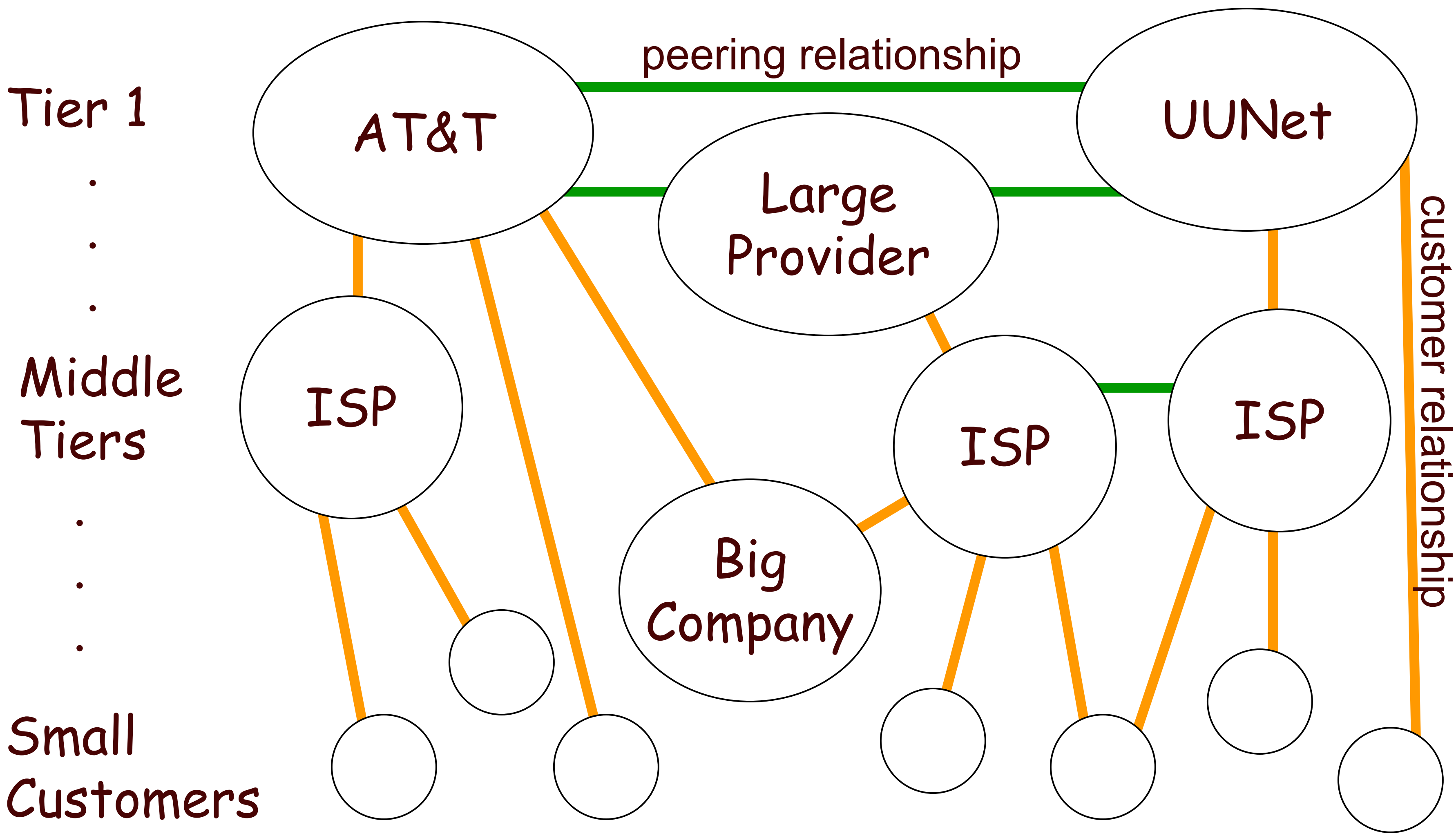
IP Routing in OSPF

- 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

OSPF Won't Work Between Companies

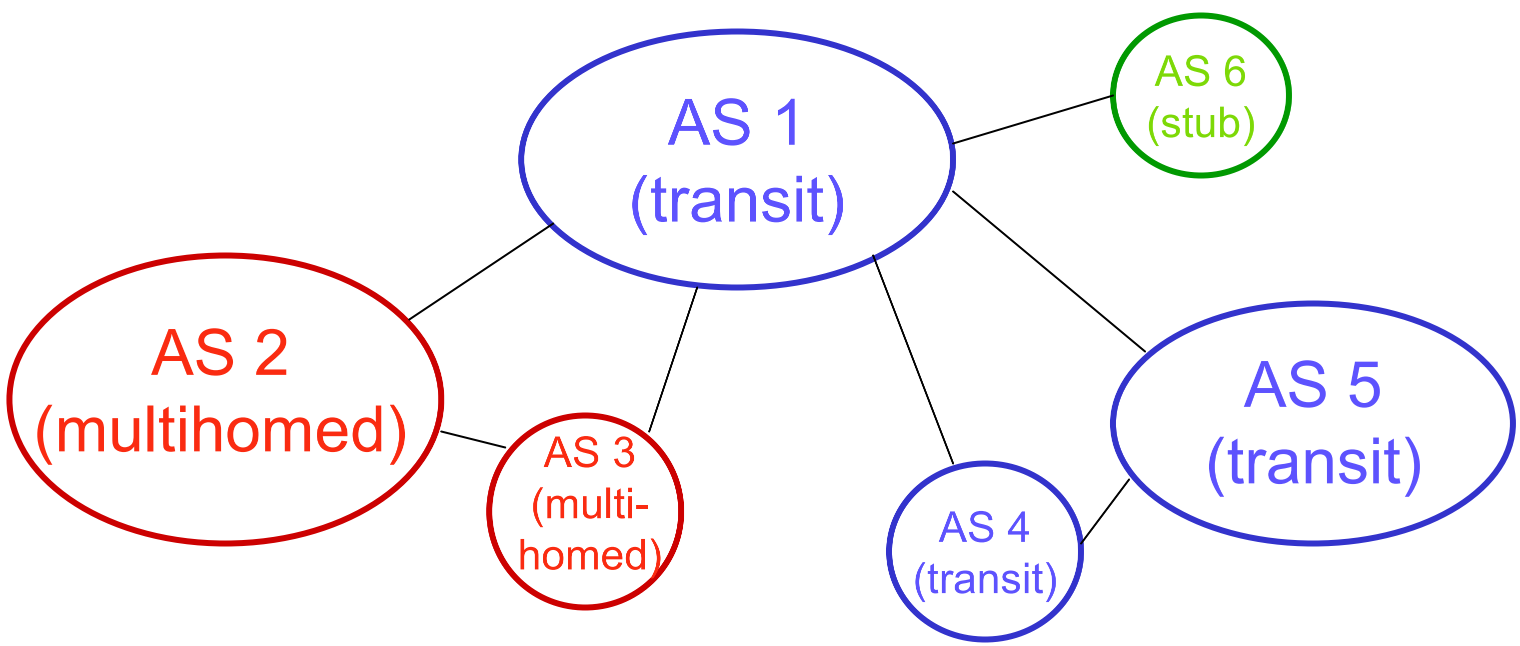
- OSPF nodes are managed by the same authority. They have a common goal (find shortest path).
- Domain is small enough that nodes can flood each other with information.
- Across companies, ***business relationships*** determine routing policy. More complicated!

Business Relationships Connect the Internet



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) Concepts

- **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.

BGP Preferences

- Nodes have to choose a path from all those advertised by their neighbors.
- BGP table contains all the collected routes and their **local preference**.
- Choose route with highest rank.
- How to set rank?
  - Based on routing policy: prefer customers first, then peers, then upstream providers.
  - Other factors? Geography, special agreements with neighbors (see **HW problem** for example).

References

- For more information, see:

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

  or:

  RFCs that define the protocols (see "Useful Links" page on course home page).

Homework Assignment For January 23

- Chapter 2 of Text.
- Chapter 4 of Blown to Bits, Evans and Wurster, HBS Press: 1999. (Available in print form only)
- First written HW, due 1/28, is now available online. (http://zoo.cs.yale.edu/classes/cs155/spr03/hw1.pdf)