Week 6

* Transport services and protocols
  + Service: provides logical communication between application processes running on different hosts
  + Transport protocols actions in end systems
    - Sender: breaks application messages into segments, passes to network layer
    - Receiver: reassembles segments into messages, passes to application layer
  + Mentioned protocols
    - TCP
      * More complex
      * Connection oriented transport
      * Reliable
      * Congestion control
      * Flow control
      * Connection setup
    - UDP
      * User Datagram Protocol
      * Connectionless transport
      * Unreliable unordered delivery
      * No frills extension of best effort IP
    - Neither delay guarantee rates
    - Bandwidth guarantees
* Transport Layer Actions
  + Sender:
    - Is passed an application layer message
    - Determines segment header fields values
    - Creates segment
    - Passes the segment on to the IP
  + Receiver
    - Receives segment from IP
    - Checks header values
    - Extracts application- layer message
    - Demultiplexes message up to application via socket
* Multiplexing vs. Demultiplexing
  + Process of gathering data chunks at the source host from different sockets, encapsulating each data chunk with header information to create segments, and passing the segments to the network layer
  + Demultiplexing
    - Each datagram has a source IP address as well as a destination IP
    - Each datagram carries one transport layer segment
    - Each segment has source, destination port number’’
* Connectionless Demultiplexing
  + When receiving host receives UDP segment
    - Checks destination port # in segment
    - Directs UDP segment to socket with that port #
* User Datagram Protocol
  + More of a bare bones internet transport protocol
  + “Best effort” service, UDP
    - Segments might be lost or out of order
    - Not much insurance in terms of quality control
  + Connectionless
    - No handshaking between UDP sender, receiver
    - Each UDP segment handled independently of others
  + Why UDP?
    - Much simpler than TCP
    - Smaller header size
* UDP uses
  + Streaming multimedia apps (loss tolerant, rate sensitive)
    - If a pixel or segment gets mishandled, the human won’t notice, so the protocol doesn’t need to hold up the entire operation
  + DNS
  + SNMP (simple network management protocol)
  + HTTP/3
* Internet Checksum
  + Goal: detect errors in transmitted segment
    - Sender:
      * Treat contents of UDP segment (including header) as sequence of 16 bit integers
    - Receiver:

Thursday

* Principles of reliable data transfer – Overview
  + Reliable data transder
    - Getting data there despite unreliable channel
    - Important for application, transport and link layers
  + Our approach
    - Start simple, and build increasingly sophisticated protocol to handle problems
    - Characteristics of unreliable channel will determine complexity of reliable data transfer protocol