**Transport Layer**

* Provide logical communication between app processes running on different hosts
* Trans protocols run in end systems
  + Send side: breaks app messages into segments, passes to network layer
  + RCV side: assemble segments into messages, passes to app layer
* More than one transport protocol available to apple
  + Internet: TCP and UDP

Transport vs Network Layer

* Network Layer: logical communication between hosts
* Transport layer: logical communication between processes
  + Relies on, enhances, network layer services
* Internet Transport-layer protocols
  + Reliable, in-oder deliver (TCP
    - Congestion control
      * Flow control
      * Connection setup
  + Unreliable, underordered delivery: UDP
    - No-frills extension of “best effortt “IP
  + Services not available
    - Delay guarantees
    - Bandwidth guarantees
* Multiplexing/demultiplexing
  + Multiplexing at sender; handle data form multiple sockets, add transport header (later used for demultiplexing)
  + Demultiplexing at receiver: use header info to deliver received segments to correct socket.
* Demultiplexing
  + Host receives IP datagrams
  + Host uses IP address
* Connectionless demultiplexing
  + Recall: created socket has host-local port #.
    - DatagramSocket mySocket1 = new DatagramSocket(12535);
  + Recall: when creating datagram to send into UDP socket, must specify.
  + When host receives UDP segment
    - Checks destination port # in segment
    - Directs UDP
  + Something
  + Something