

## Transport Layer

### Points:

- process to process communication
- addressing port numbers
- MUX: data from different ports sent as one (M:1) allows multiple apps to use network at the same time
- DEMUX: data received as one and to-be-sent to different ports (1:M)
- flow control: manages data delivery in case of congestion on host
- error control: detect and discard corrupt packets, keep track of lost and discarded packets and resend them, discard duplicate packets, buffer out-of-order packets until missing packet arrives
- reliable delivery: error control, sequence control, loss control, duplication control
- moves data between applications on devices in network
- segmentation: easier to perform error checking on segments, eases transfer of data by lower layers, allows multiplexing, header is attached to every segment
- encapsulation: data+header sent as one message
- decapsulation: data extracted from message and header is used to get info
- 

### Protocols:

UDP (user datagram): connectionless, no ACKs, low overhead because of no ACKs, no reordering of data packets, no retransmission, Mux, Demux

UDP header: 8-byte, src (2), dest (2), length of data (2), checksum (optional) (2)

Uses: multicasting, broadcasting, routing, real-time transmission

TCP (transmission control): connection-oriented, ACKs, stream-oriented, no broadcasting, supports error and flow control, data is carried through imaginary tube between two end points, maintains buffer, seq# tells how to reconstruct out-of-order segments, full-duplex

TCP header: 20-60 byte, src (2), dest (2), seq# (4), ack# (4), header-length (4bits), Reserved (6bits), Flags (6bits), window size (2), checksum (2), urgent pointer (2), options and padding (40)

Flags: URG – urgent pointer points to important data which is treated as normal data but is told to App layer that it's important, ACK – acknowledgment, PSH – data is being pushed, RST – connection reset when seq# limit reached or something else, SYN – to establish connection, FIN – to terminate connection

SCTP (stream control transfer):

Simple Protocol: no error control, no flow control, no ACKs, no handshake

Stop and Wait Protocol: provides flow and error control (via checksum), S sends packet, S goes in Block state until ACK is received. After ACK is received it goes in Ready state and sends next packet.

Selective Protocol: many packets are sent by S with not considering ACKs. If R gets an out-of-order segment, it puts it in buffer and wait for right segment. After S is done, R sends seq# of corrupted or lost segments. S re-sends lost segments.