Transport Layer

The transport layer is used for:

- **Demultiplexing**: decide which application should receive the data.

    This is accomplished using port numbers, which are in the transport header and designate the application (Web server is port 80 for example).

    UDP (User Datagram Protocol) does <u>only</u> this.

    A unit in the transport layer is called a segment.

- If segments get lost, retransmission has to be done

- If the routes change, we should reorder the segments to get the original order the segments were sent in (multiple segments may constitute a single file)

    TCP does the above two.
    ↳ Transmission Control Protocol

→ It does not send segment (n+1) until segment (n) has been received

The issue is that routers do not cooperate to ensure reliable data transfer. We should be able to infer packet loss!

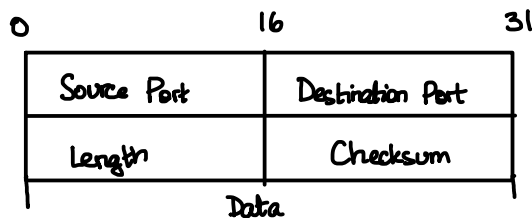If we want something between UDP and TCP, we should use UDP and build our own protocol on top of it.

The mechanism via which we drop packets at a router if the <u>queue becomes full</u> is called a drop-tail mechanism.   congestion

TCP is able to infer that congestion has occurred and does congestion control: the segment sending rate is reduced Enough of the input links back off to allow the queue to clear up.
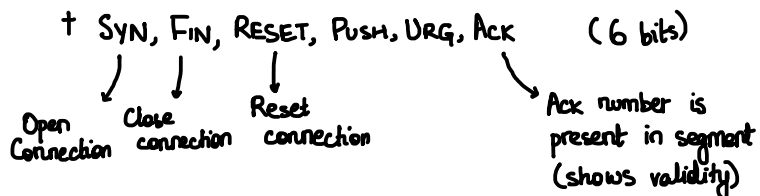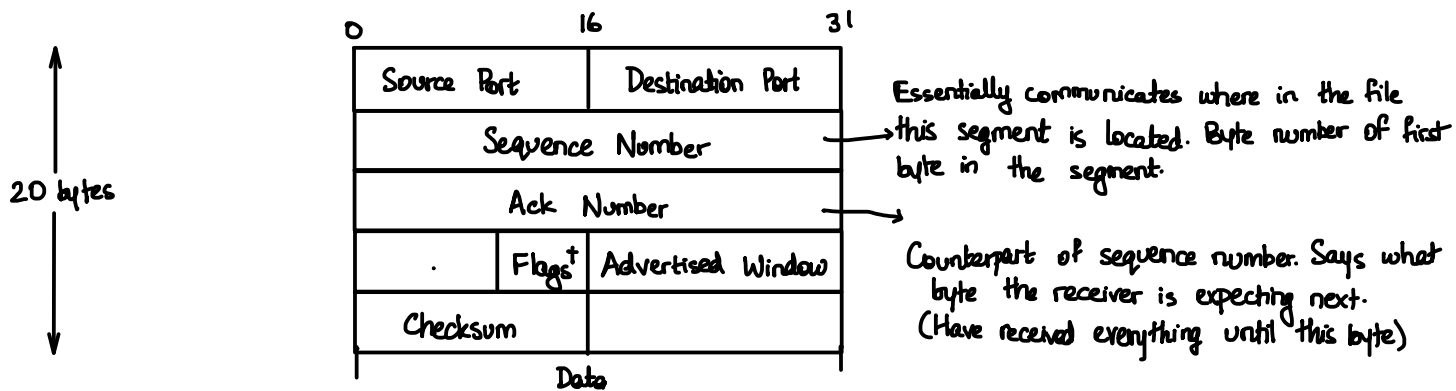
TCP also does flow control, which is just congestion control at the destination. This is different because the source and destination can communicate. The destination may take time to process information, and may not have read all the information from the transport layer yet (there is a buffer).

Van Jacobson and Sally Floyd did a lot of work on TCP in the 80s.

The UDP header is just

0	16	31
Source Port	Destination Port	
Length	Checksum	
Data		

The TCP header on the other hand, looks like

↕ 20 bytes

0	16	31
Source Port	Destination Port	
Sequence Number		
Ack Number		
·	Flags†	Advertised Window
Checksum		
Data		

Sequence Number → Essentially communicates where in the file this segment is located. Byte number of first byte in the segment.

Ack Number → Counterpart of sequence number. Says what byte the receiver is expecting next. (Have received everything until this byte)

† SYN, FIN, RESET, PUSH, URG, ACK   (6 bits)

SYN → Open Connection
FIN → Close connection
RESET → Reset connection
ACK → Ack number is present in segment (shows validity)

The protocol field in the IP layer says whether the transport layer is using TCP or UDP.
                                                    TCP → 6   UDP → 17