Transport Layer; class 5

Responsible for end to end communication over a network. It is a logical communication between application running on different hosts

Only run in start and end systems. Create a virtual line between the 2 apps and don’t know anything else

It possess SEGMENTS, with an AppID source and an AppID destination.

The sending one breaks the app messages into segments and passes it to network layer.

The receiving one reassembles the message and passes it to app layer

There is 2 protocol for transport layer :

## UDP: User Datagram Protocol

Unreliable, unordered delivery protocol. No control mechanism, doesn’t care about data loss.

When doesn’t require reliability.

It has ports and segments

8 bit segment

Useful for where loss is acceptable and for short messages

## Transmission Control Protocol

Connection oriented so responsible for reliability.

Acknowledgement & retransmissions

Flow control : adapt to the receiver bandwidth and processing abilities, so sender adapt depending on the receiver abilities

Congestion control (of a network) : routers have buffer, if full drops packets

Source & destination port = 2 bytes

TD :

* Ethernet Header
  + Mac destination (6 bytes) : 48 45 20 06 06 0d
  + Mac source (6 bytes) : a0 1b 29 7e 4c a6
  + Protocol (2 bytes) : 08 00 => IPv4
* IP header
  + Version(4 bits) : 4
  + Header Length (4 bits): 5
  + Type of service (1 byte) : 00
  + Total Length (2 bytes) : 00 34
    - 52 bytes
    - 20 : header
    - 32 : payload
  + ID (2 bytes) : e9 1a
  + Flags ( 4 bits ) : 4
  + Fragment Offset (12 bits ) : 0 00
  + TTL (1 byte) : 39
  + Protocol (1 byte) : 06 => TCP
  + Checksum (2 bytes) : 5e 18
  + IP Source (4 bytes) : 5d b8 dc 1d = 93.184.220.29
  + IP Destination (4 bytes) : c0 a8 00 13 = 192.168.0.19
* TCP Header
  + Source port (2 bytes) : 00 50 => port 80
  + Destination port (2 bytes) : c4 cc => port 50 380
  + Sequence number (4 bytes) : 43 8f 8a 09
  + Acknowledgement number (4 bytes) : c0 cf 81 54
  + TCP header length (4 bits ) : 8
    - 8 \* 4 = 32 Bytes
  + Null ( 6 bits ) : 0000 00
  + URG / ACK / PSH / RST / SYN / FIN (6 bits) : 0 / 1 / 0 / 0 / 0 / 0
  + Window size (2 bytes) : 01 20
  + Checksum (2 bytes) : 24 eb
  + Urgent pointer (2 bytes) : 00 00
  + Options (12 bytes left) (optional) : 01 01 05 0a c0 cf 81 53 c0 cf 81 54

### TCP Connection Management

3 way handshake

### TCP Segments & flow control

Acknowledge the next expected byte

### TCP Timer management

RTT is calculated by measuring how long ACKs took

### TCP Congestion Control

Slow start : exponentially increase the number of messages send over the same period of time (multiplying by 2)

Additive increase (+1)