MODULE 5

Transport layer

Responsibility of Transport layer

- End to end delivery of the data (Port to port delivery)
  - Example : (Process to process delivery)
- Reliability
  - In order delivery
  - No loss of data
- Error control
  - Checksum
- Congestion control (AIMD method)
- Flow control (Advertising the window from Rx)
- Multiplexing and demultiplexing
- Protocols :
  - TCP (Transmission control Protocol)
  - UDP (User datagram protocol)

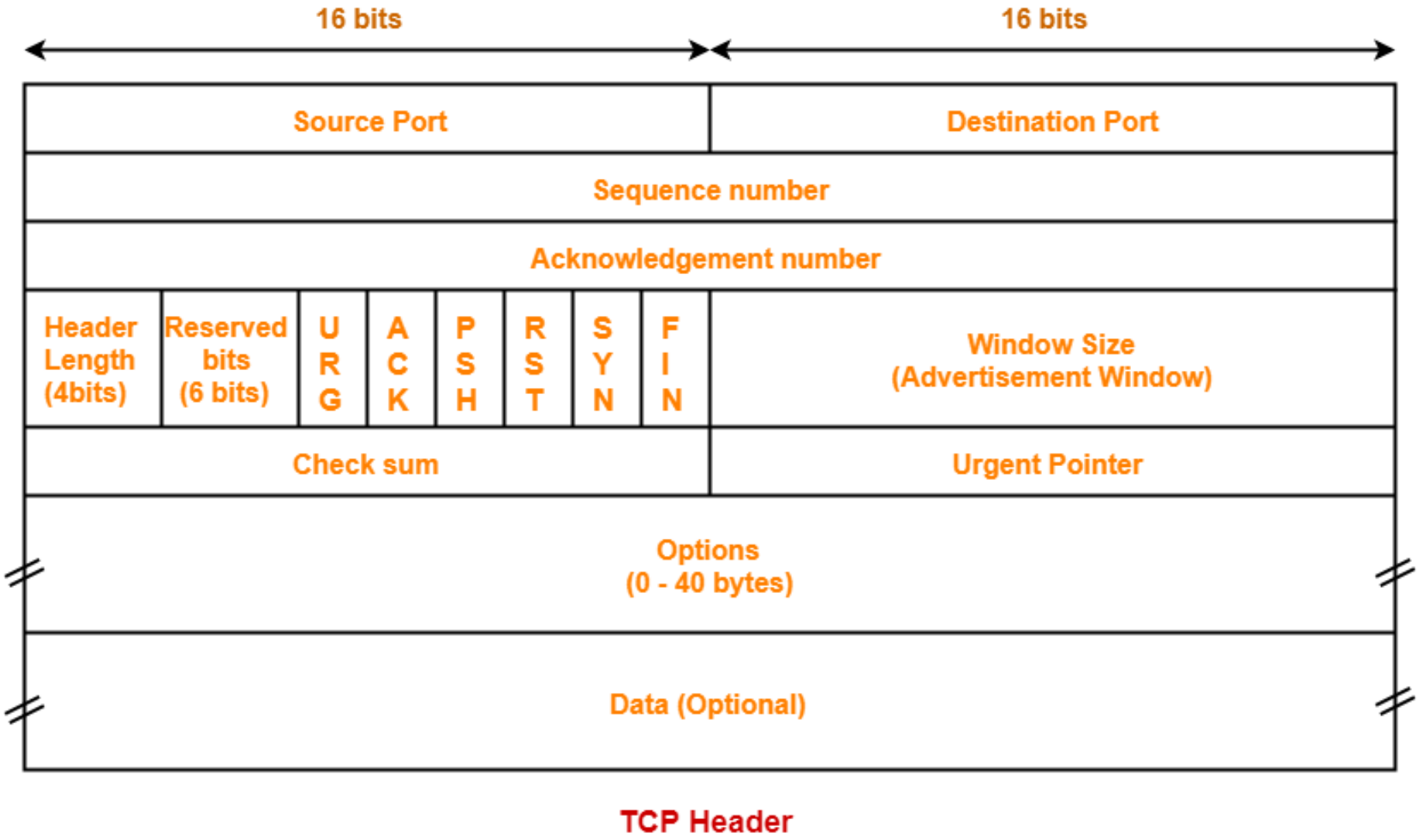
Concept of Socket address

- IP address [32 bits]
- Port address [16 bits]
- Socket address = IP address + Port address= 32 bits + 16 bits = 48 bits

Transmission Control Protocol (TCP)

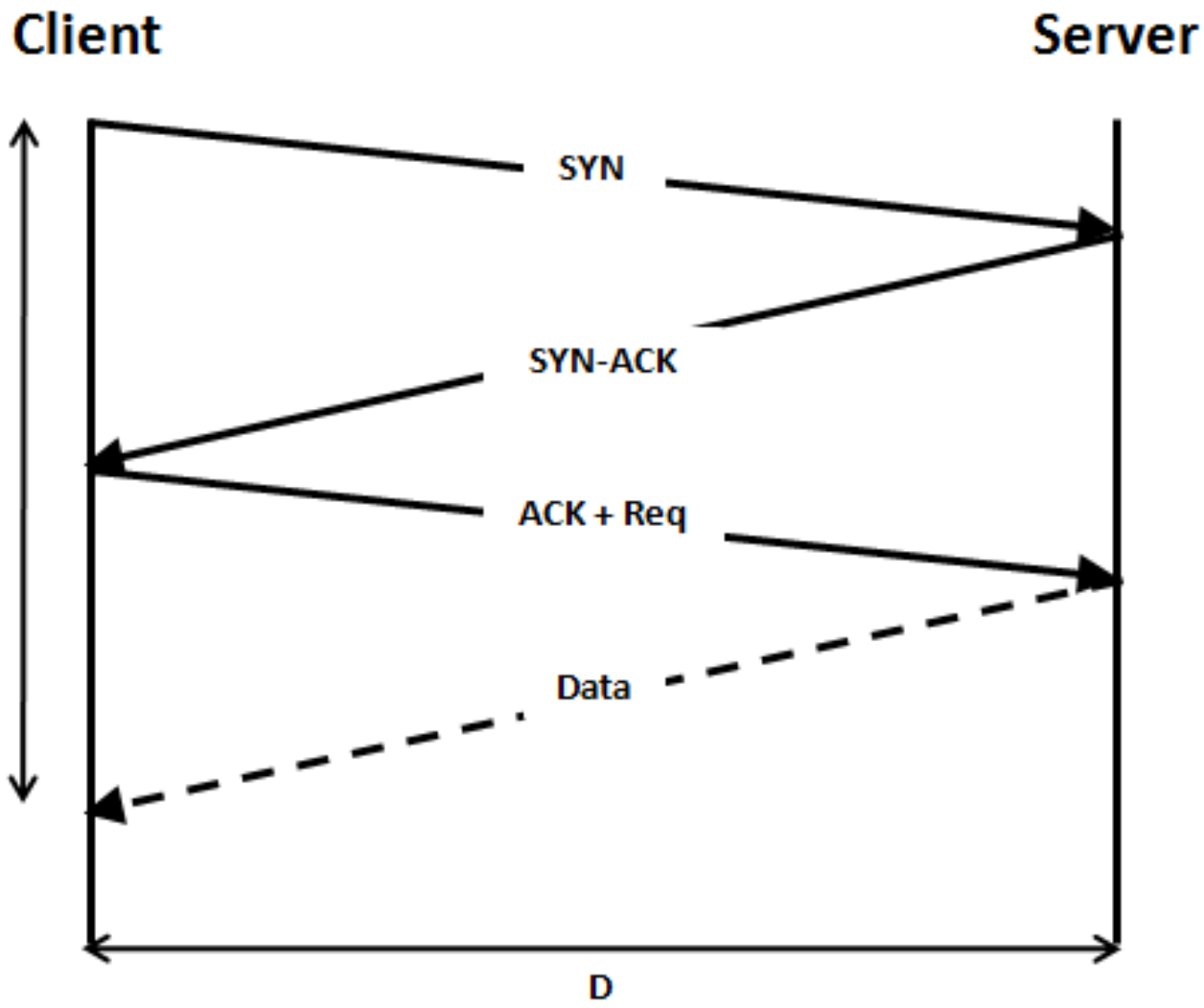
- Used in Trasport layer
- Trasport layer received the data from application layer
- Application layer defines the protocol to be used either TCP or UDP and add the header to the data
- Why to add header ?
  - **Byte streaming** (collection of bits and convert it into segments)
  - **Connection oriented** (For reliability of connection is achieved through it, TCP- 3 way handshaking)
  - **Full duplex** (Both can transmit the data at same time)
  - **Piggybacking** (whenever data is sent …acknowledgement is sent along with data)
    - Eg. Go back and Error, Selective Repeat
    - After receiving all the packets send only one acknowledgement
  - **Error control** (if the data is wrongly received inform sender)
  - **Flow control** (To manage with capacity of the receiver….send the data below that or equal to capacity)
  - **Congestion control** (Check for complete network congestion capacity)

Transmission Control Protocol (TCP) header

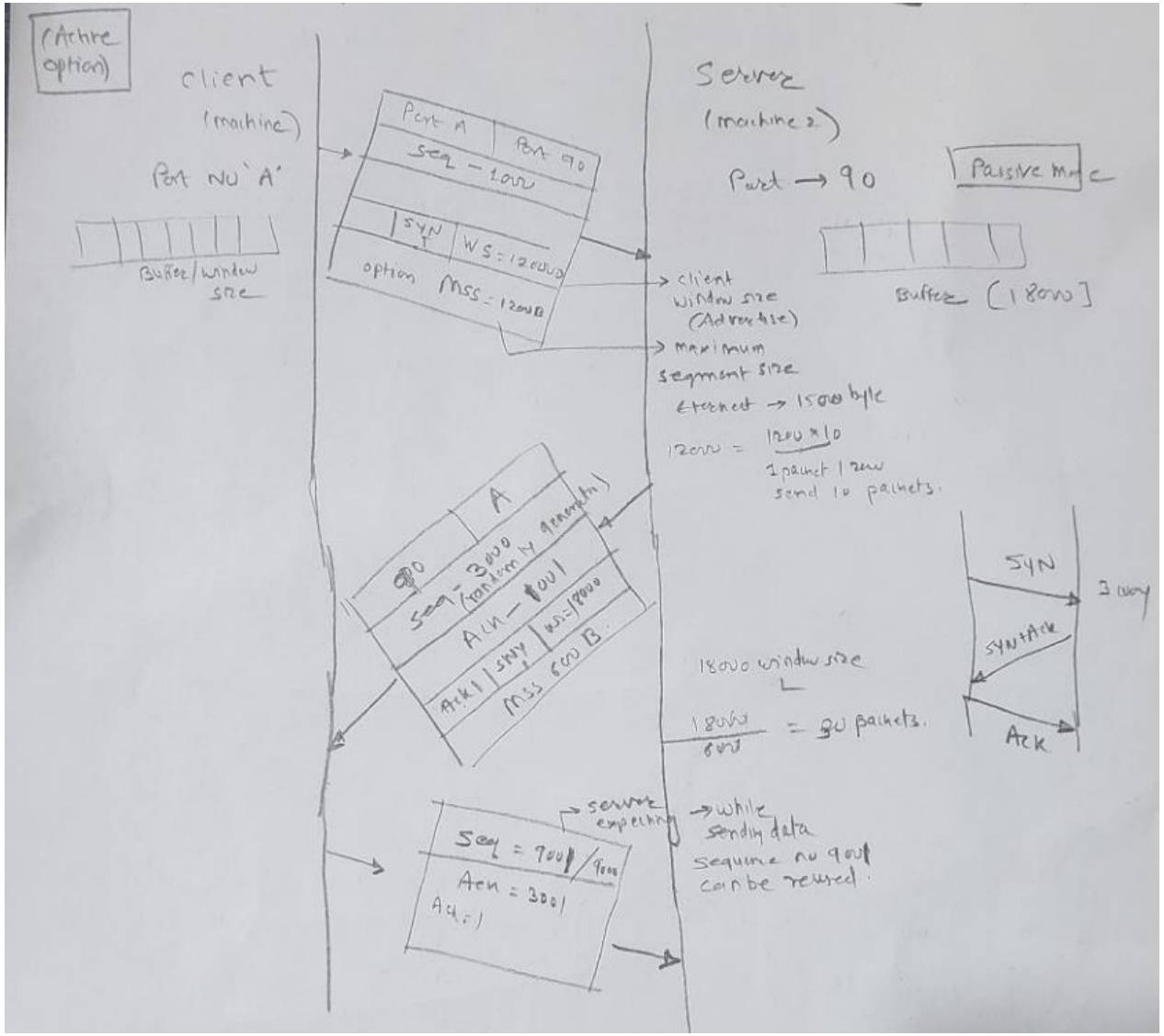


TCP Header

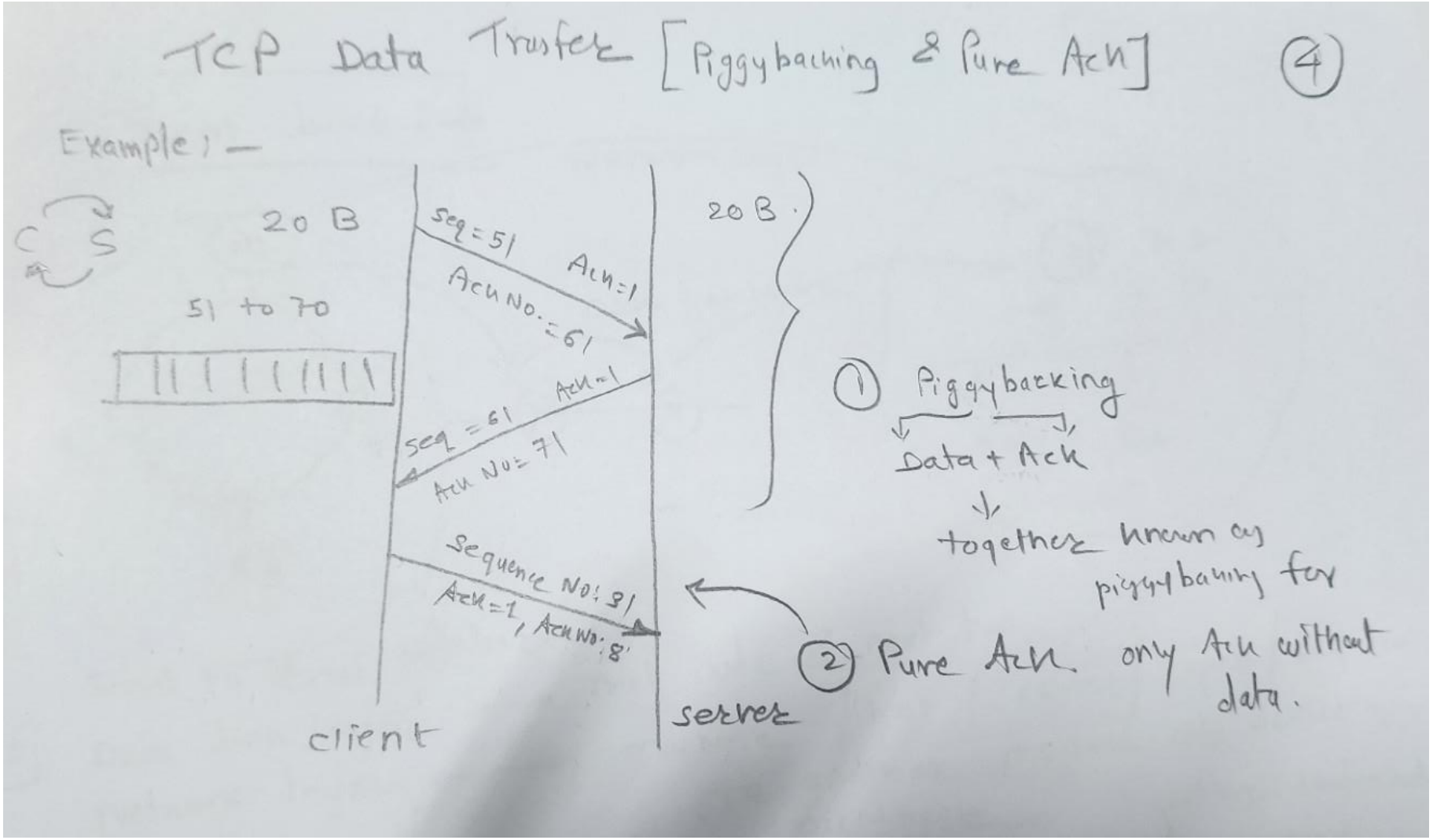
TCP Connection Establishment and Connection Termination



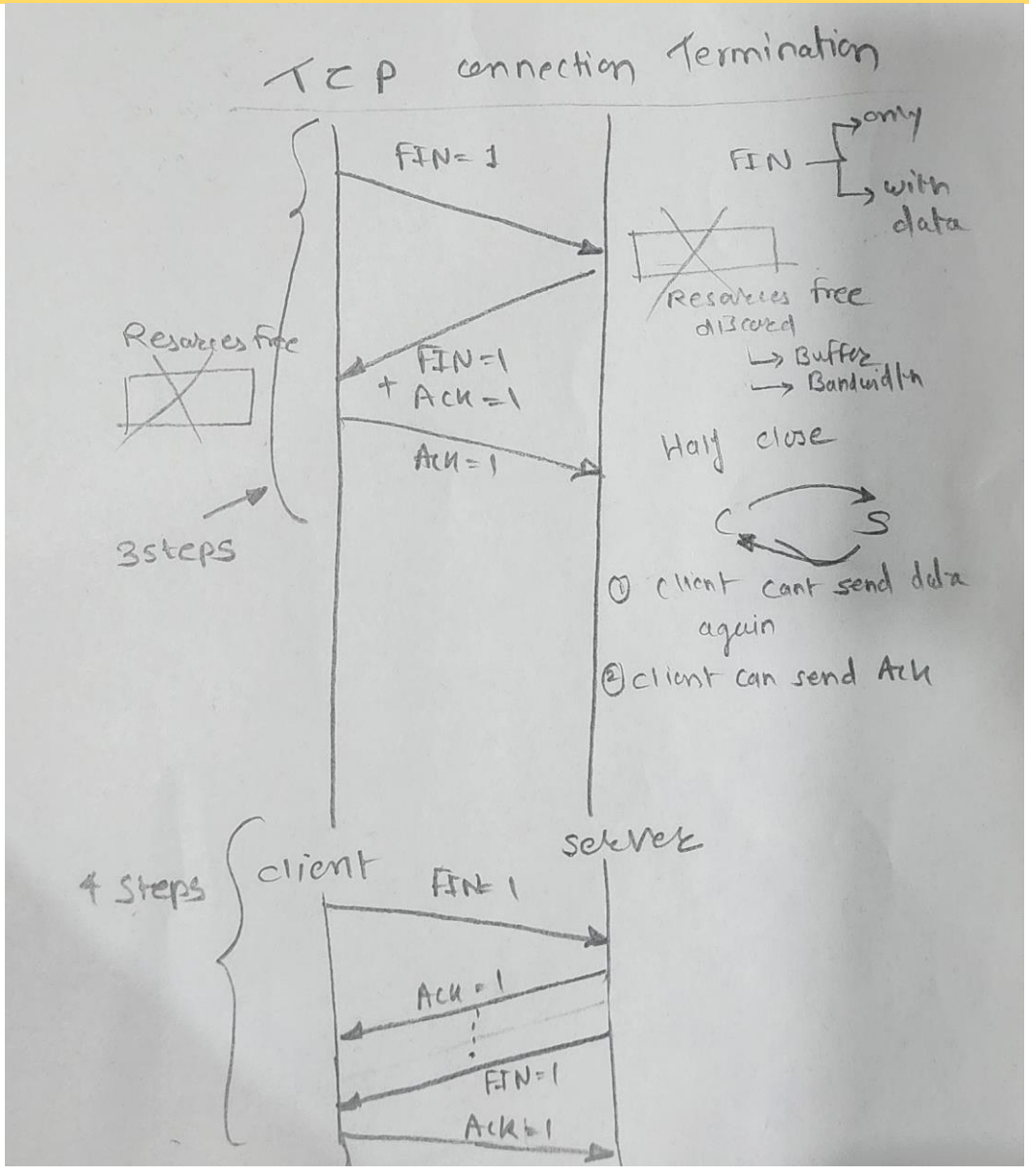
TCP Connection Establishment and Connection Termination



TCP DATA Transfer



Connection termination in TCP



TCP Timers

1. Retransmission Timers
2. Keep alive Timers
3. Persistent Timer
4. Time wait timer

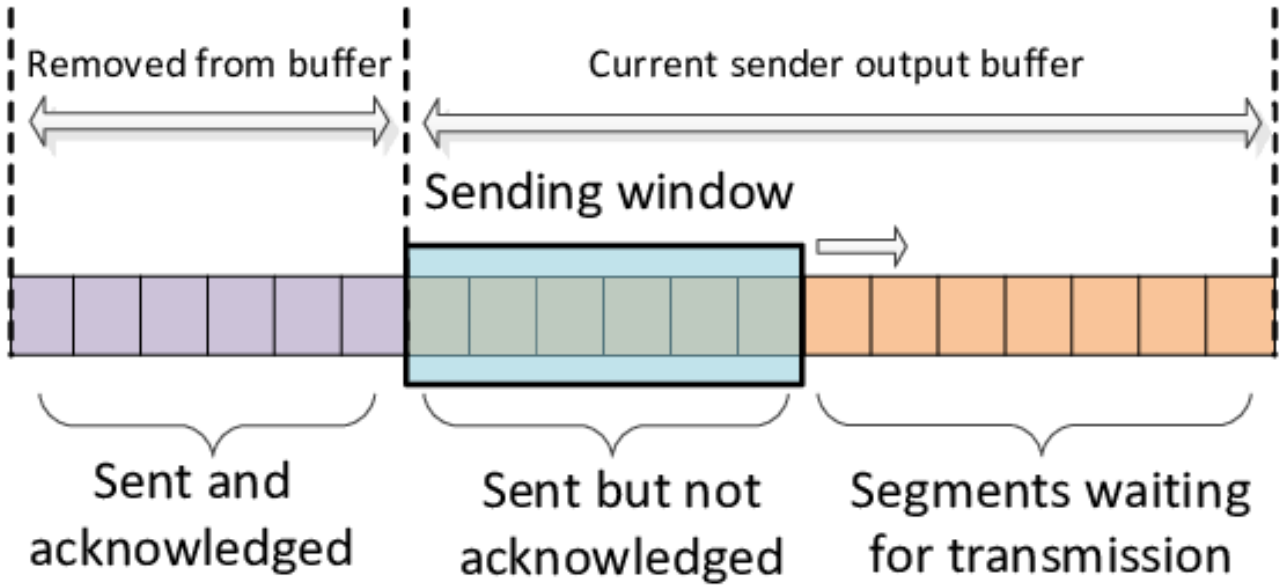
Error control

1. Normal operation
2. Lost segment
3. Lost Acknowledgment
4. Corrupted Segment
5. Out of order
6. Duplicate segment

Flow control

It regulates the amount of data a sender can send receiver can receive at a time.

- Sliding Window protocol



Silly window syndrome

- **Syndrome created by sender**
  - Assume sender has to send 5 bytes of data but at time only sends 1 byte of date.
  - For sending 1 byte of data 40 bytes of header is attached (20 byte TCP+ 20 bytes IP), thus 41 Bytes of data would be transmitted over the network.
  - So such 5 segment each carrying 1 byte of data world be transmitted causing 205 bytes of network data will be utilized.
  - If 5 bytes of data was sent in single segments with 40 bytes of header, only 45 bytes of network data would be utilized.

Silly window syndrome

- **Nagle's solution**
  - Step1 : Send the 1st segment as it is even if it is of 1 byte
  - Step 2: Start the buffer and the timer, collect all the segment in the buffer
  - Step 3 : Empty the buffer either when timer expires of buffer is full.

Silly window syndrome

- **Syndrome created by receiver:'**
  - In this assume receivers buffer size is 1000 bytes and the buffer is full, so it announce window = 0B to sender and hence sender stop sending the stop sending the segment. Receiver consumes 1 byte and announces window size = 10 byte to sender & hence sender send 1B of data with 40 Byte of header (20 BTCP + 20 BIP), causing 41 bytes of network data utilization.
  - Receiver accepts 1 byte of data, the buffer becomes full and it repeats.
- **Solution:**
  - Clark Solution:

    Don't advertise the window size till it is half empty.

Congestion control

- Congestion control focuses on reducing the overall traffic congestion in the network.
- For congestion control TCP uses 3 phases:
  - Slow start (Exponential increase)
  - Congestion avoidance (Additive increase)
  - Congestion detection (Multiplicative decrease)