UNIVERSITAS
GADJAH MADA

Transport Layer - TCP

Muhammad Alfian Amrizal

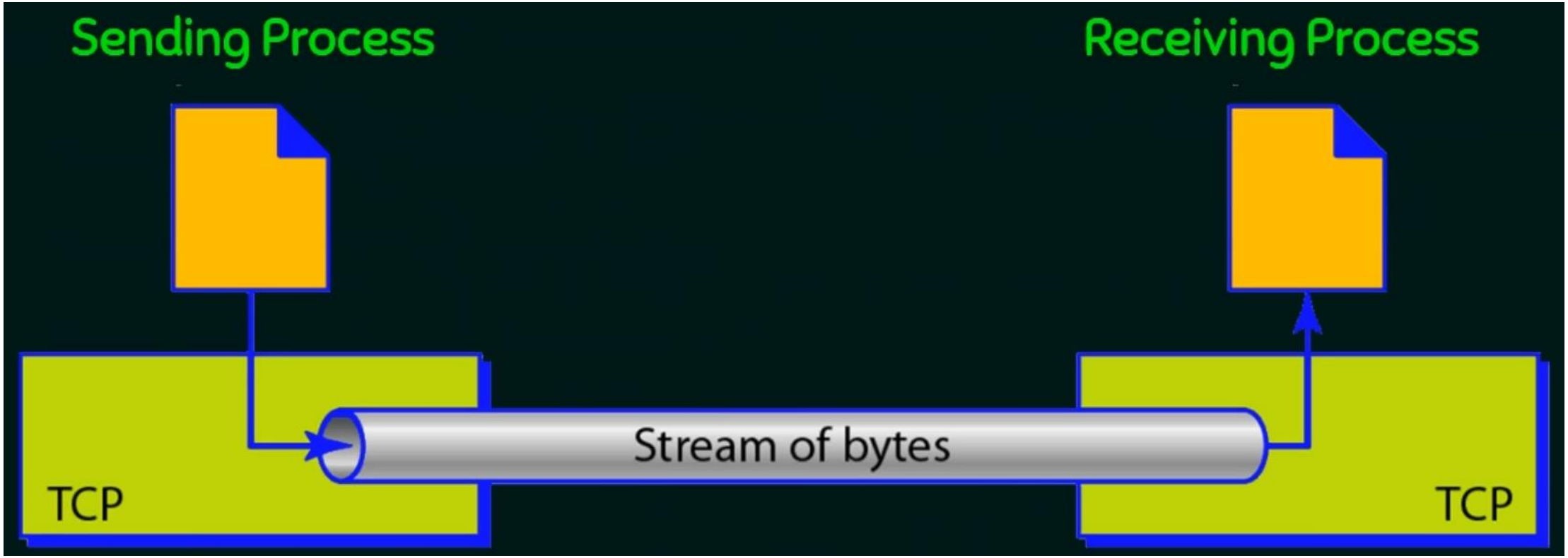
TCP

- TCP = Transmission Control Protocol
- Widely used transport layer protocol
- It creates a virtual connection between two TCPs to send data
- In addition, TCP uses flow and error control mechanisms at the transport level
- Used by applications that can tolerate delay but cannot tolerate loss
- Used by:
  - World Wide Web
  - HTTP
  - FTP
  - SMTP

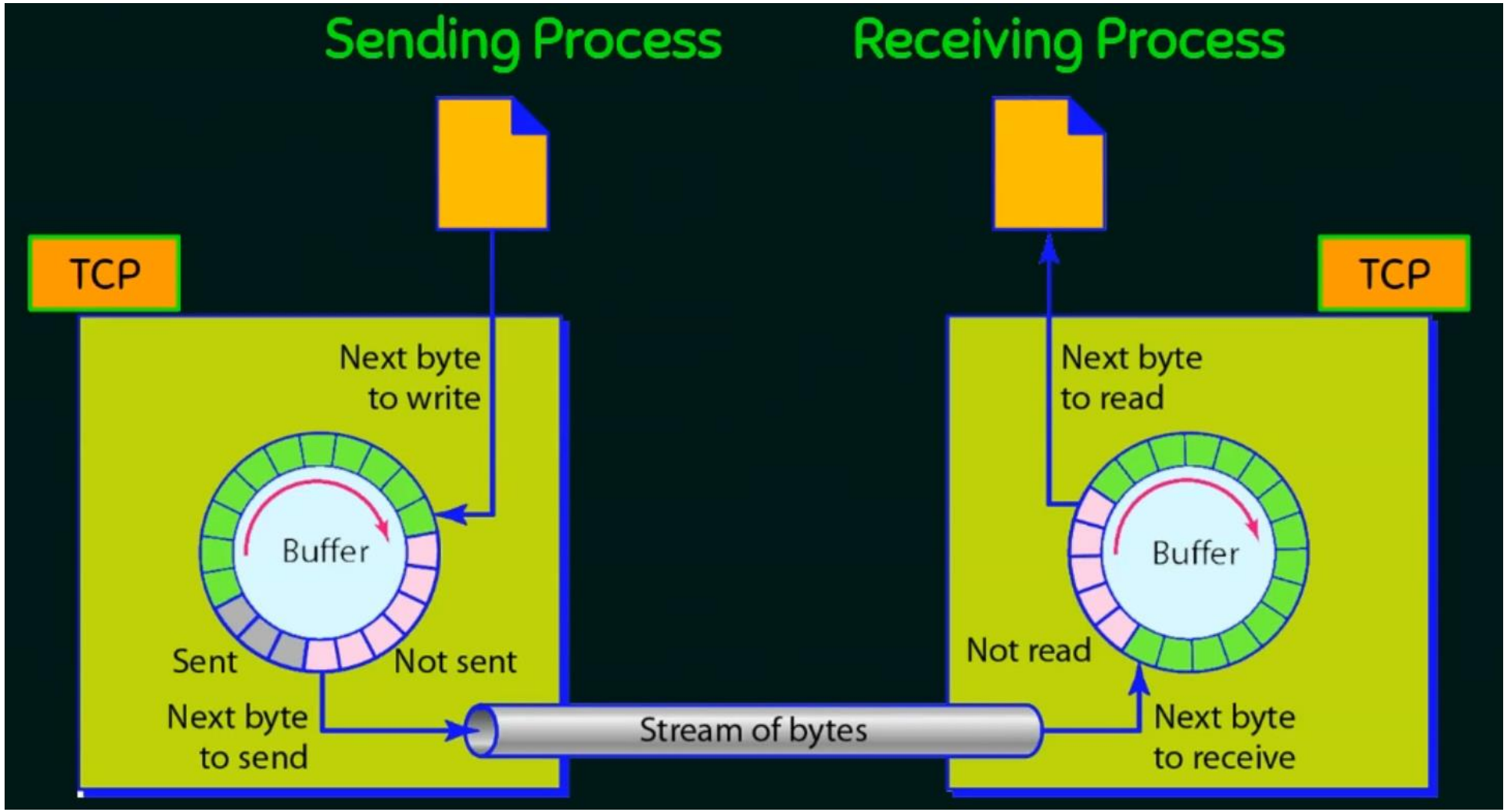
Features of TCP

- Connection oriented
- Reliable delivery
- Acknowledgement Oriented
- Retransmission
- Flow Control
- Error Control
- Congestion Control
- Segmentation and Reassembly
- Full Duplex Support

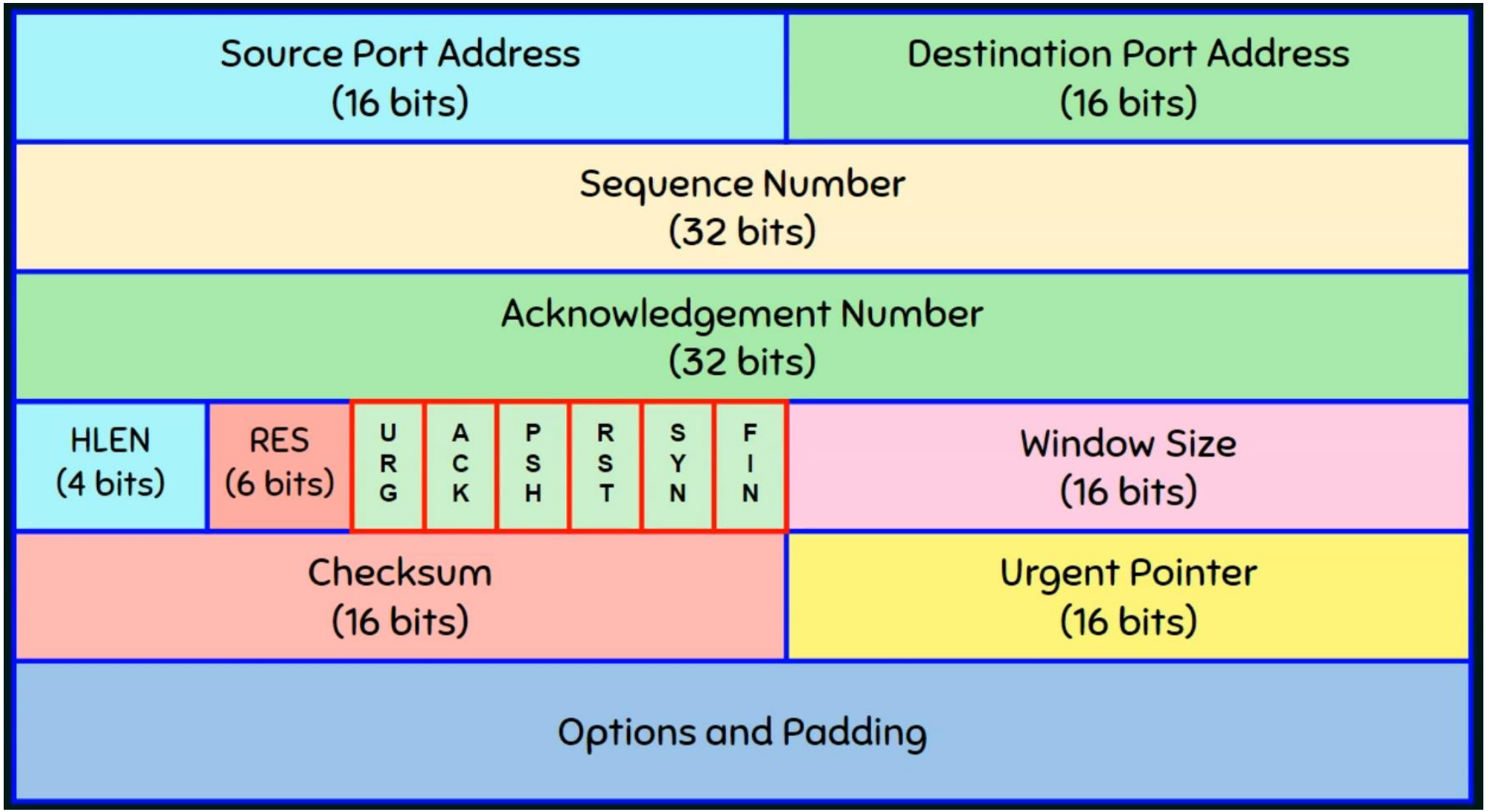
TCP Stream Delivery



TCP Sending and Receiving Buffers



TCP Header Format



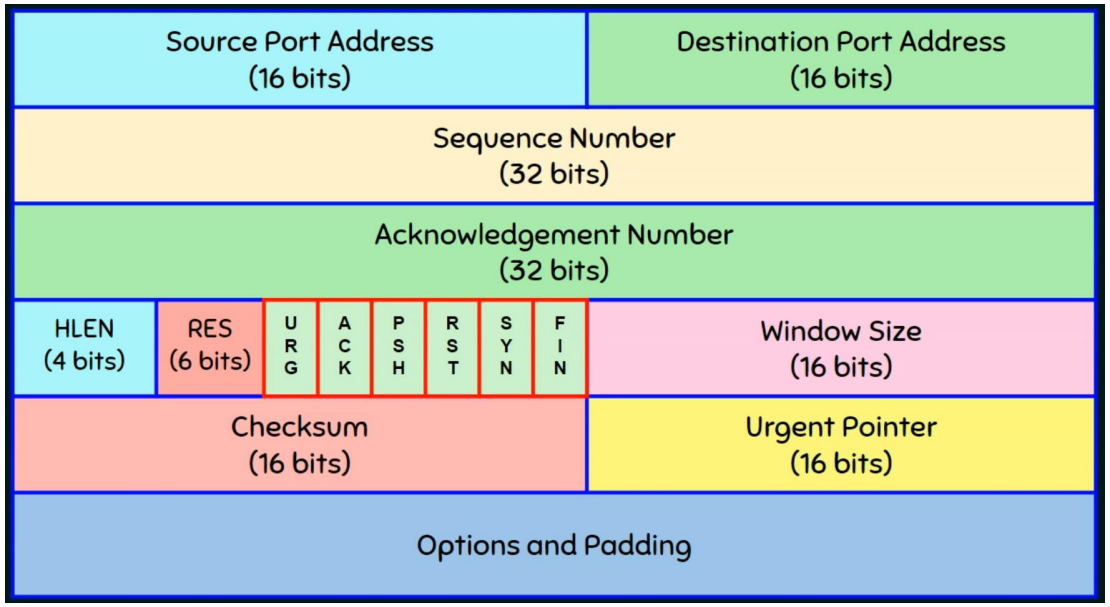
TCP Header Format

- The segment consists of a 20 to 60 bytes header, followed by data from the application layer

- The header is 20 bytes if there are no options and up to 60 bytes if it contains options

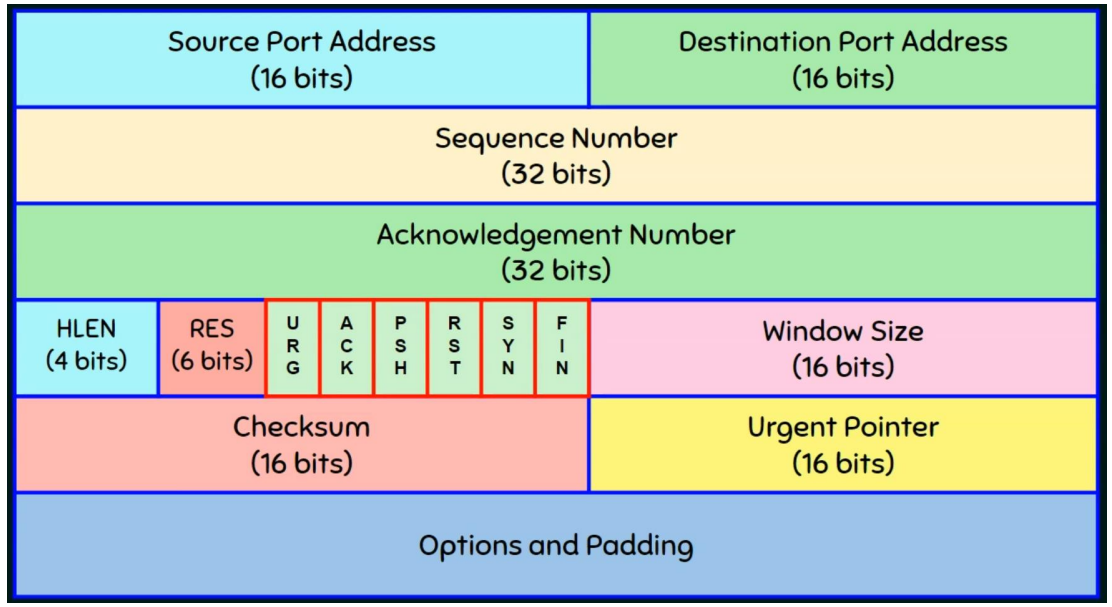
TCP Header Format

- Source Port Address
  - It defines the port number of the application program in the host that is sending the segment
  - This serves the same purpose as the source port address in the UDP header
- Destination Port Address
  - Also similar to that of UDP
- Sequence Number
  - Defines the number assigned to the first byte of data contained in this segment
  - If SYN flag is set to 1, this shows the initial segment



TCP Header Format

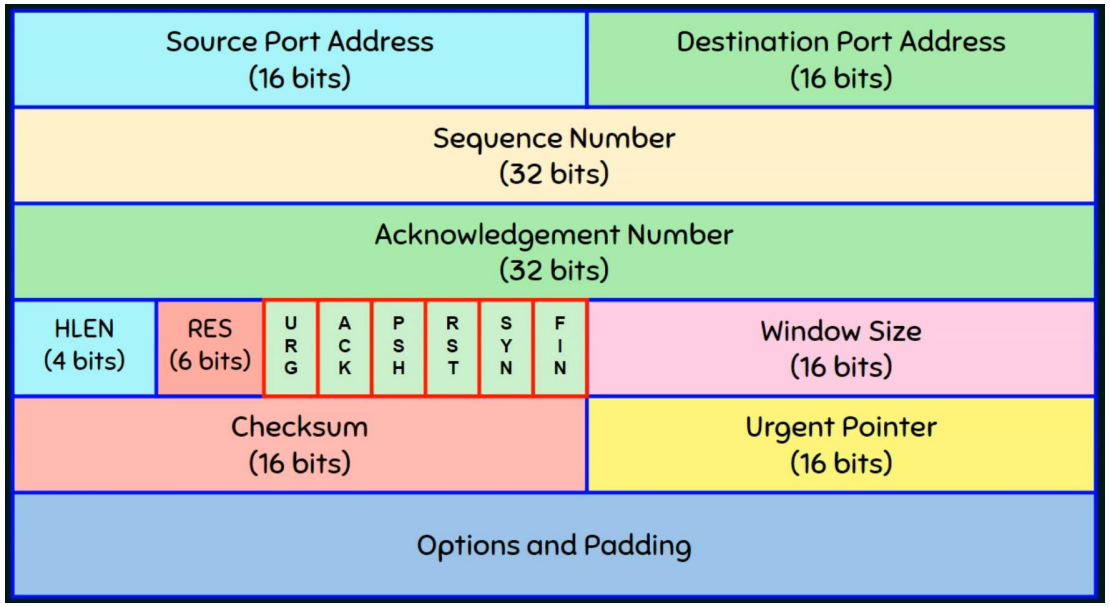
- Acknowledgement Number:
  - It defines the byte number that the receiver of the segment is expecting to receive from the other party
  - If the receiver of the segment has successfully received byte number x from the other party, it defines x+1 as the acknowledgement number
  - Acknowledgement and data can be piggybacked together



- Header Length:
  - It indicates the number of 4-byte words in the TCP header
  - The length of the header can be between 20 and 60 bytes
  - Therefore, the value of this field can be between 5 (5x4=20) and 15 (15x4=60)
- Reserved: as it name implies, it is reserved for future use

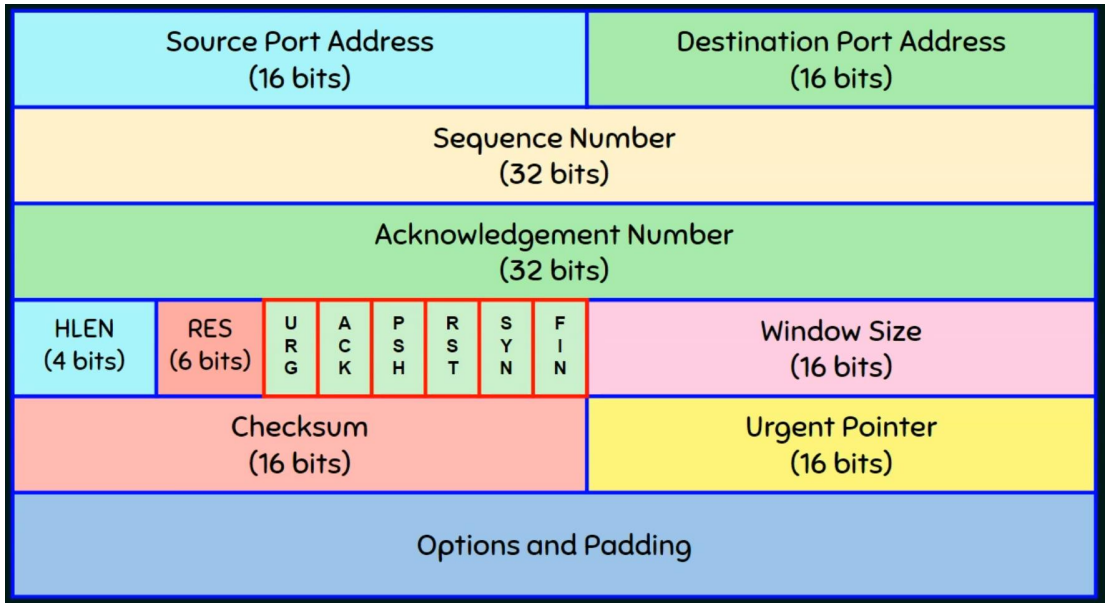
TCP Header Format

- Flags
  - This field defines 6 different control bits or flags
  - One or more of these bits can be set at a time
- Types of Flags
  - URG (1 bit): Indicates that the Urgent Pointer field is significant
  - ACK (1 bit): Indicates that the Acknowledgement field is significant
  - PSH (1 bit): Push function. Asks to push the buffered data to the receiving application
  - RST (1 bit): Reset the connection
  - SYN (1 bit): Synchronize sequence numbers
  - FIN (1 bit): Last packet from sender



TCP Header Format

- Window Size
  - This field defines the size of the window, in bytes, that the other party must maintain
- Checksum
  - To check if there is any error in the segment
- Urgent Pointer
  - Valid only if the urgent flag is set, is used when the segment contains urgent data
- Options and Padding
  - Options have up to three field
  - Padding is composed of zeros to make sure that the TCP header ends on a 32-bit boundary



TCP Connection & Flow Control

Muhammad Alfian Amrizal

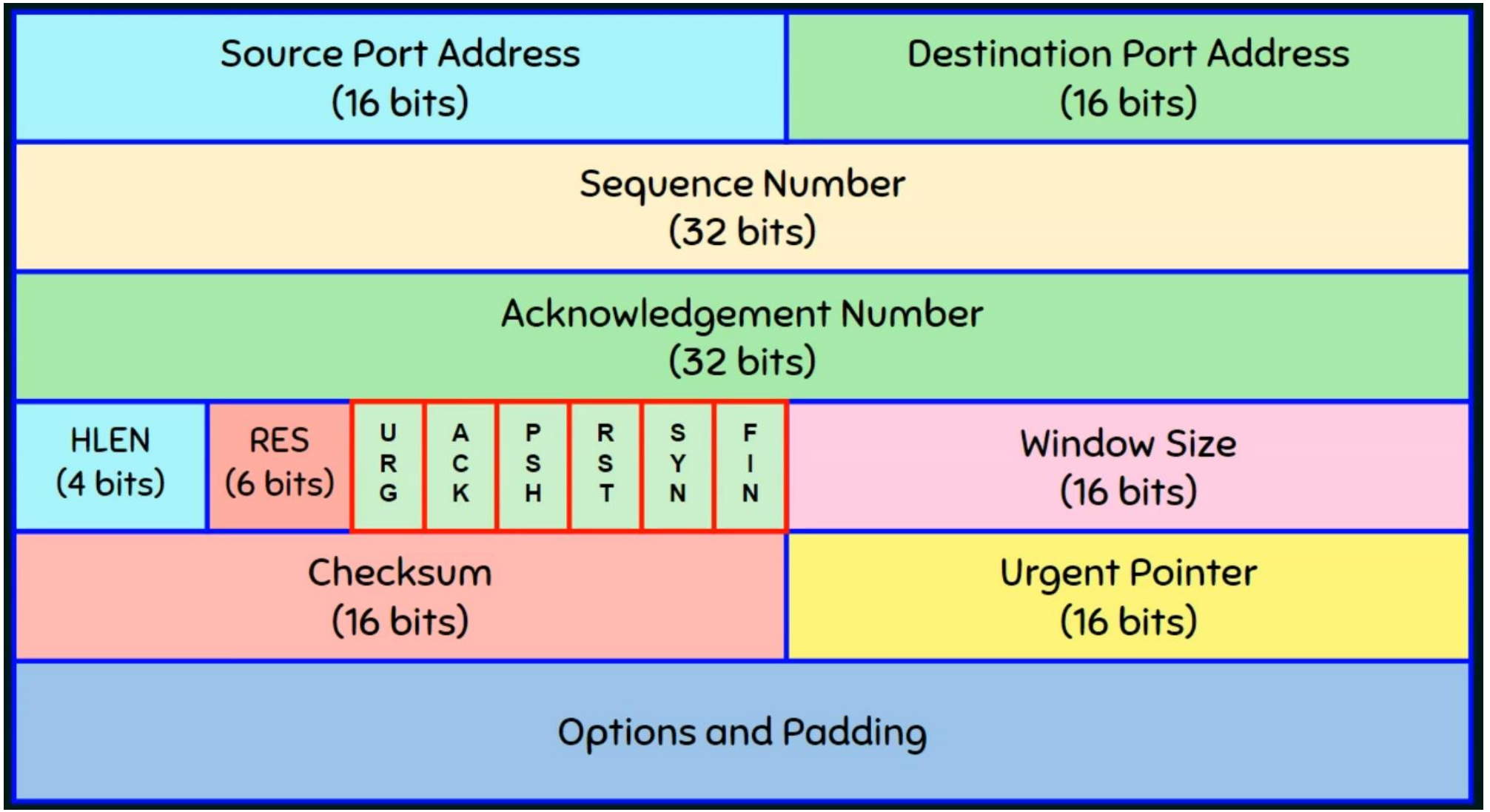
TCP Connection

- Connection-oriented
- Virtual path
- Acknowledgement process
- Retransmission of lost or damaged segments
- Full-duplex mode
- Approval from other party

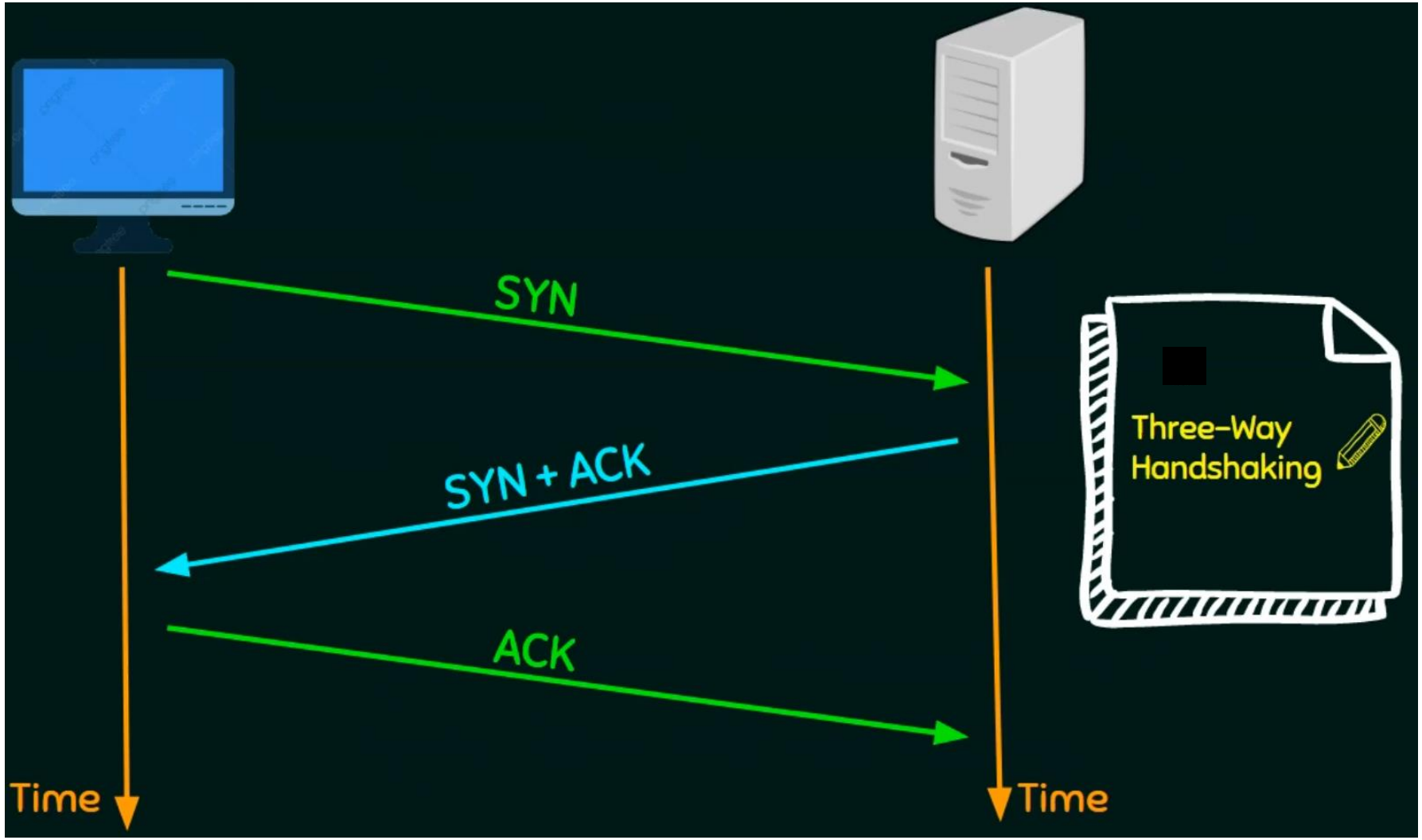
Three Phases of TCP Connection

- Connection Establishments
- Data Transfer
- Connection Termination

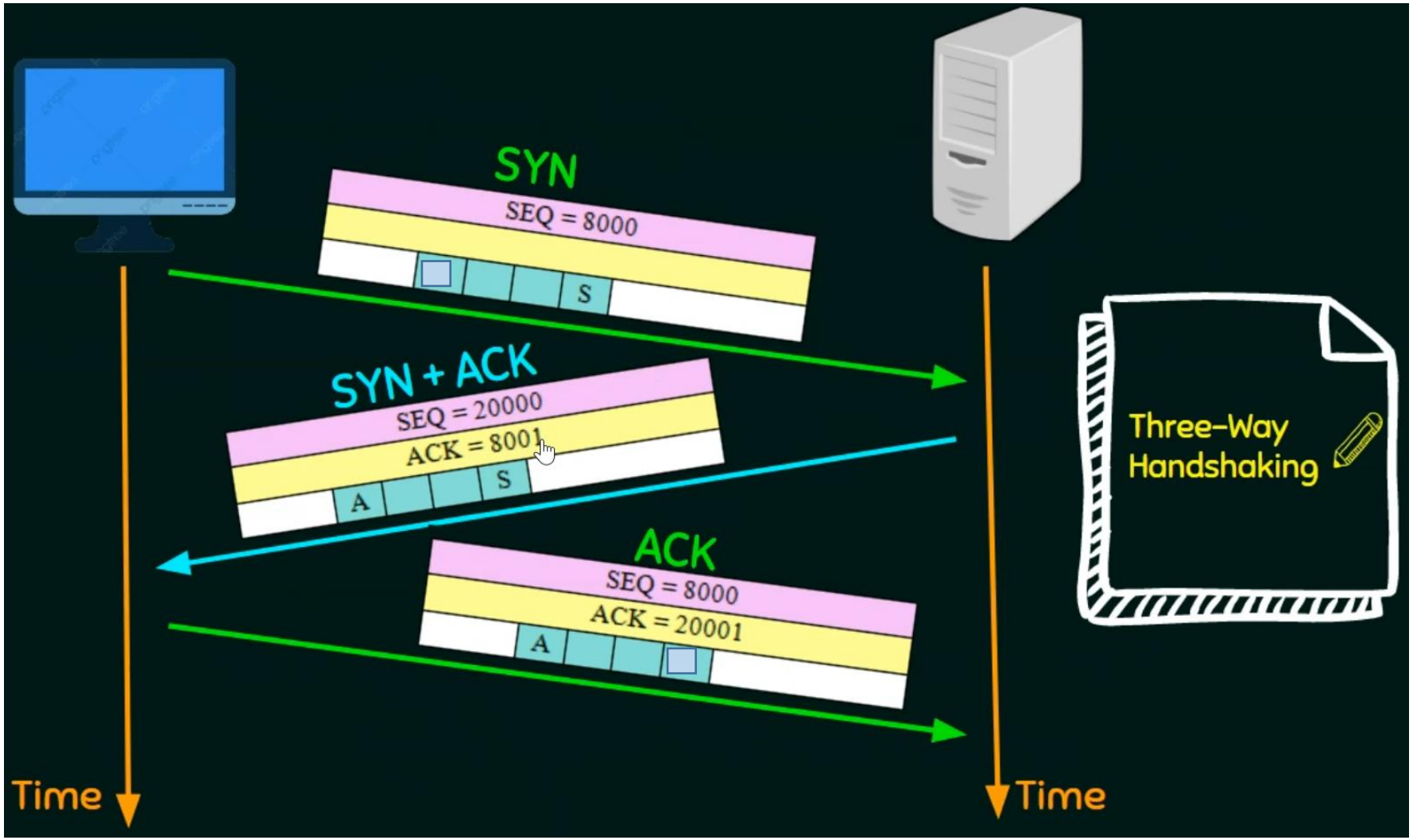
Review: TCP Header Format



TCP Connection Establishment



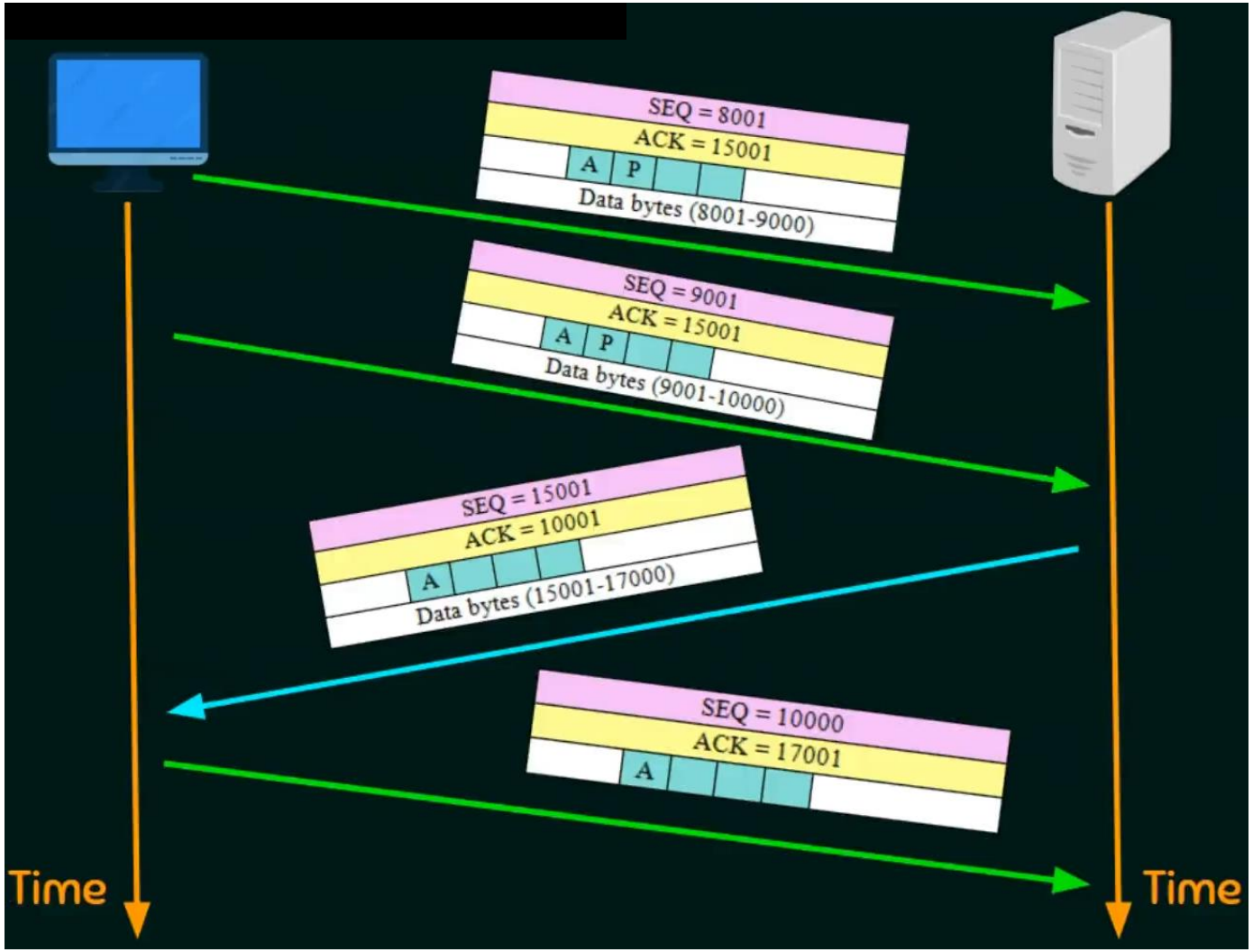
TCP Connection Establishment



TCP Data Transfer

- Bidirectional data transfer
- Piggybacking
- The acknowledgement is piggybacked with data
- Push and Urg Flaps

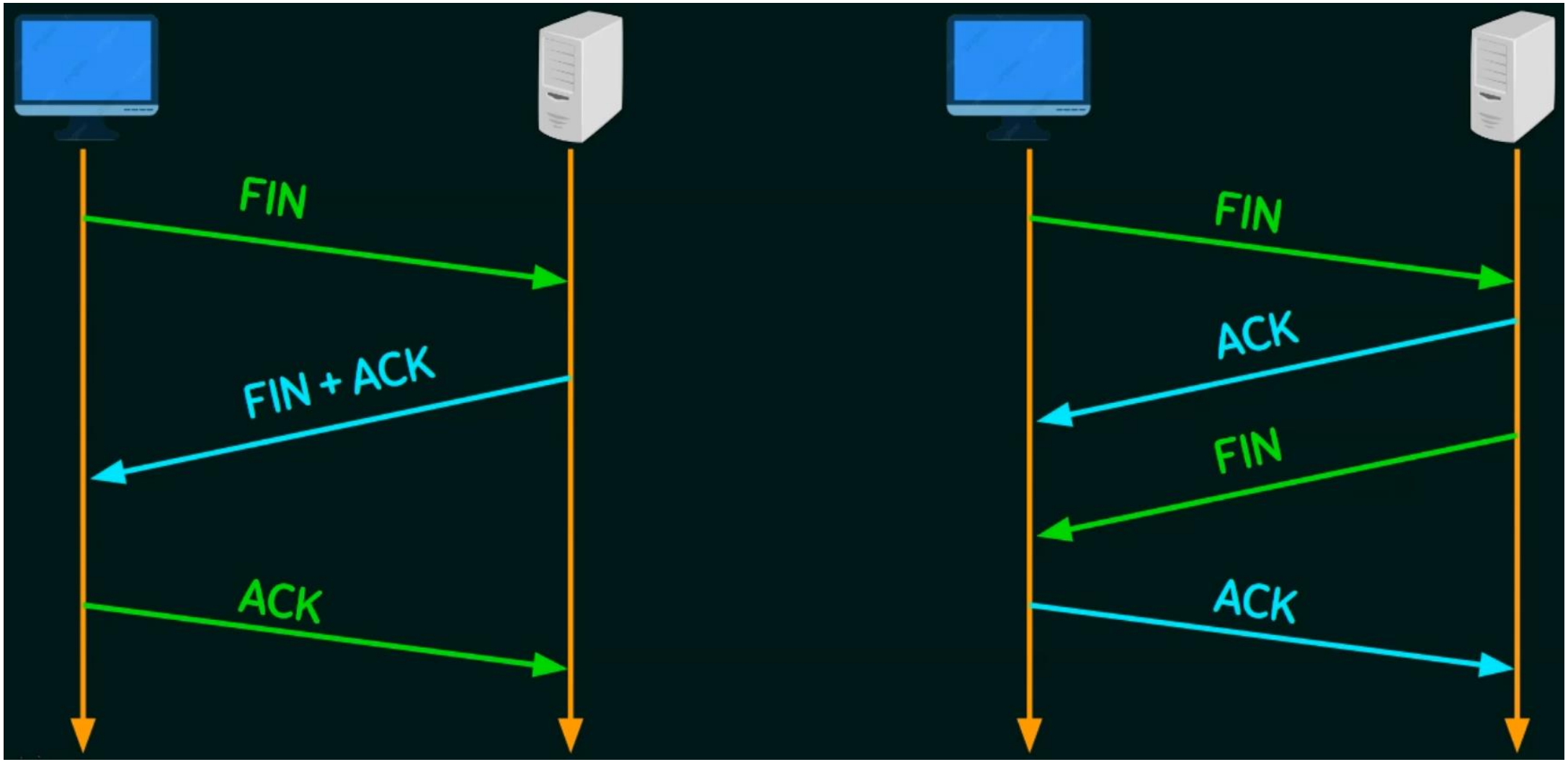
TCP Data Transfer



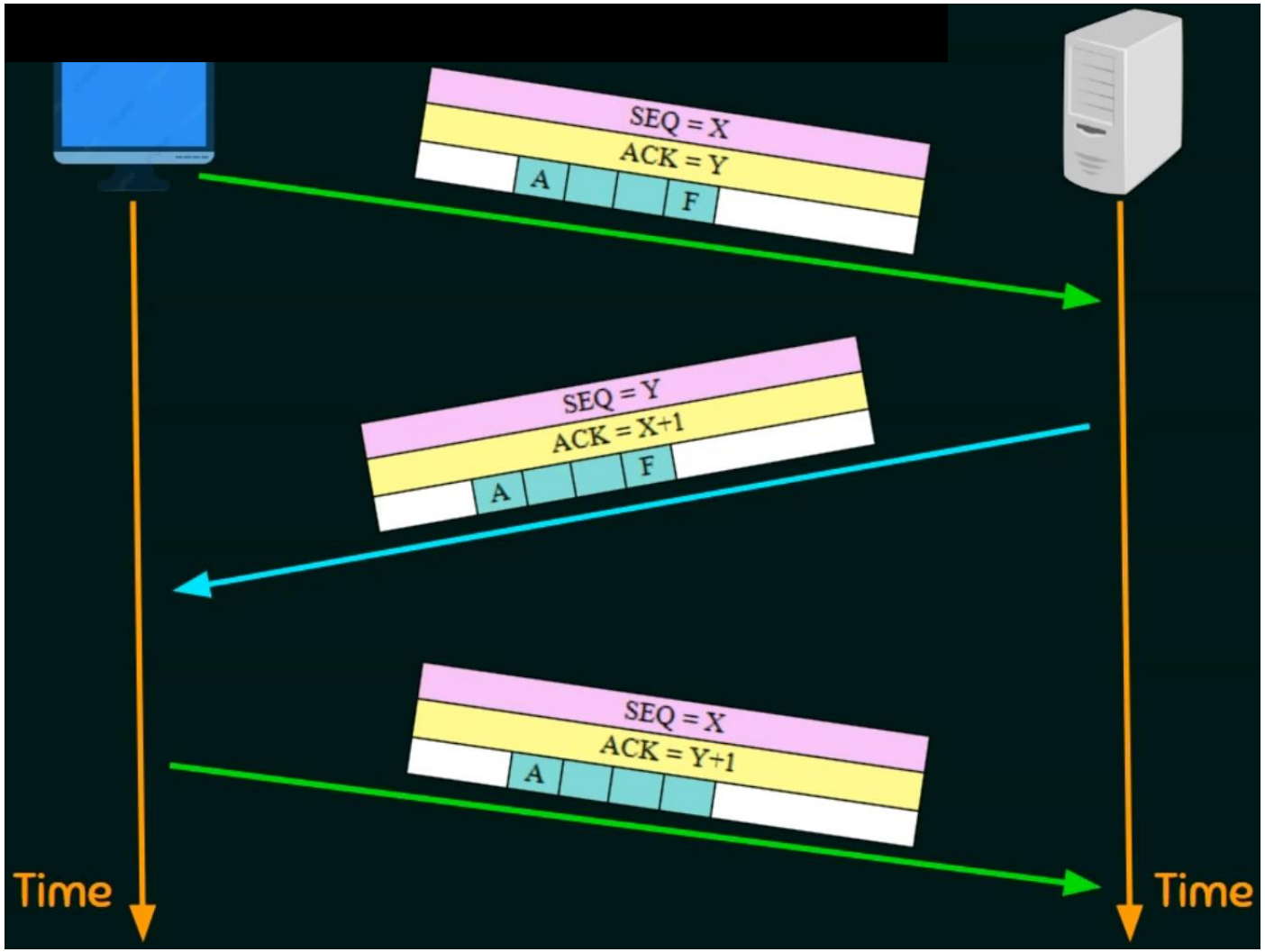
TCP Connection Termination

- Usually initiated by the client
- Two options for connection termination
  - Three-way handshaking
  - Four-way handshaking with a half-close option

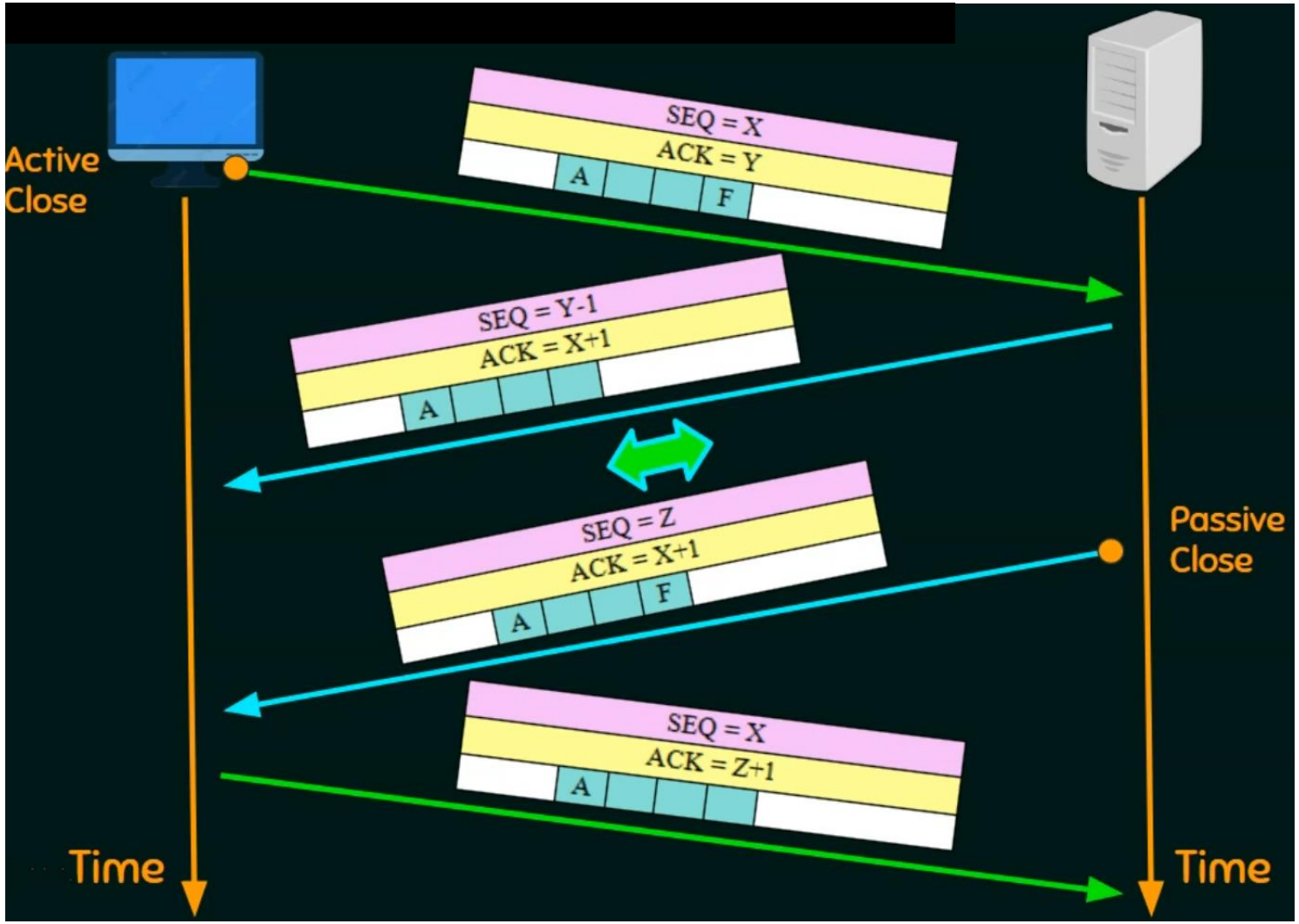
TCP Connection Termination



Three-way Handshaking



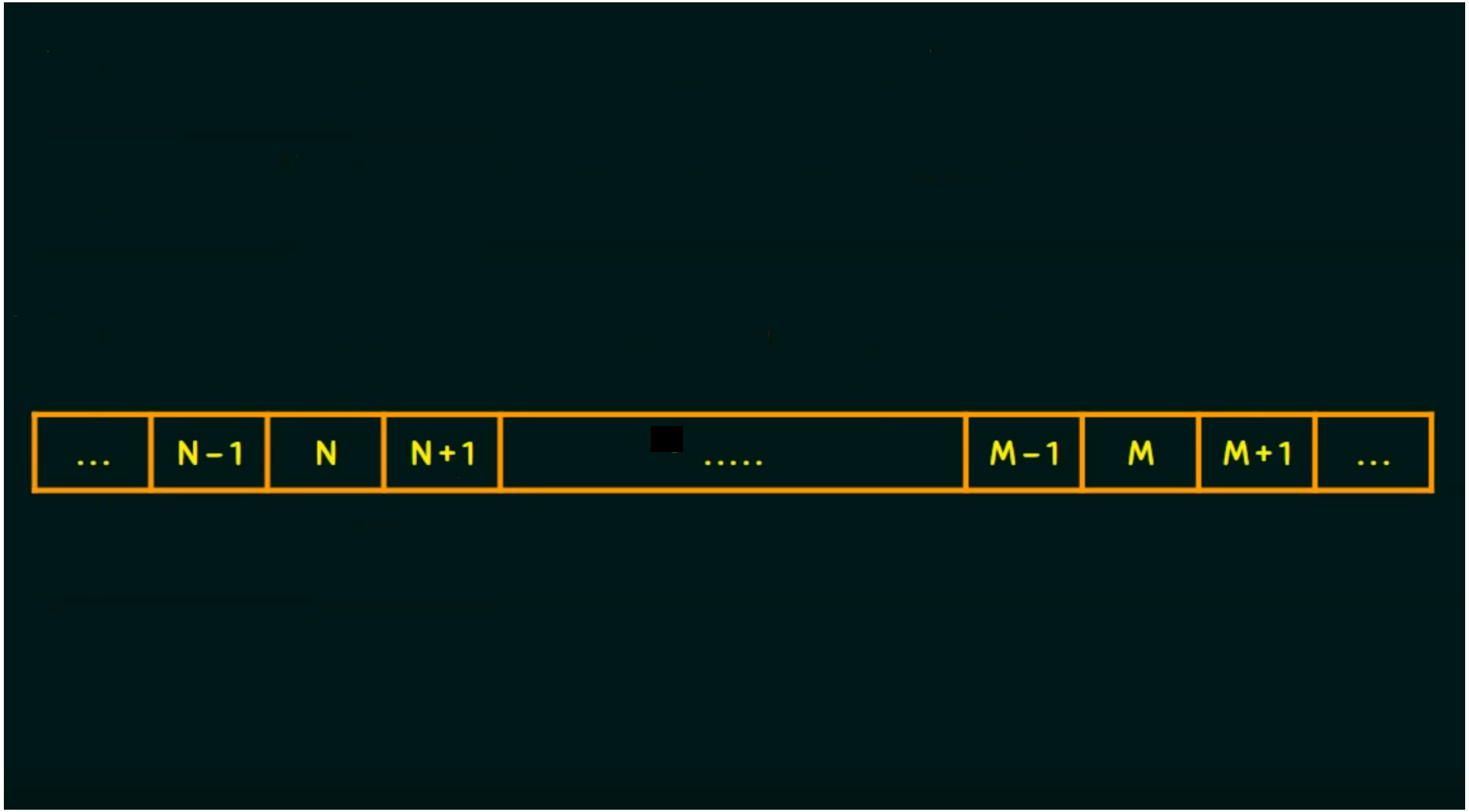
Four-way Handshaking w/ Half Close Option



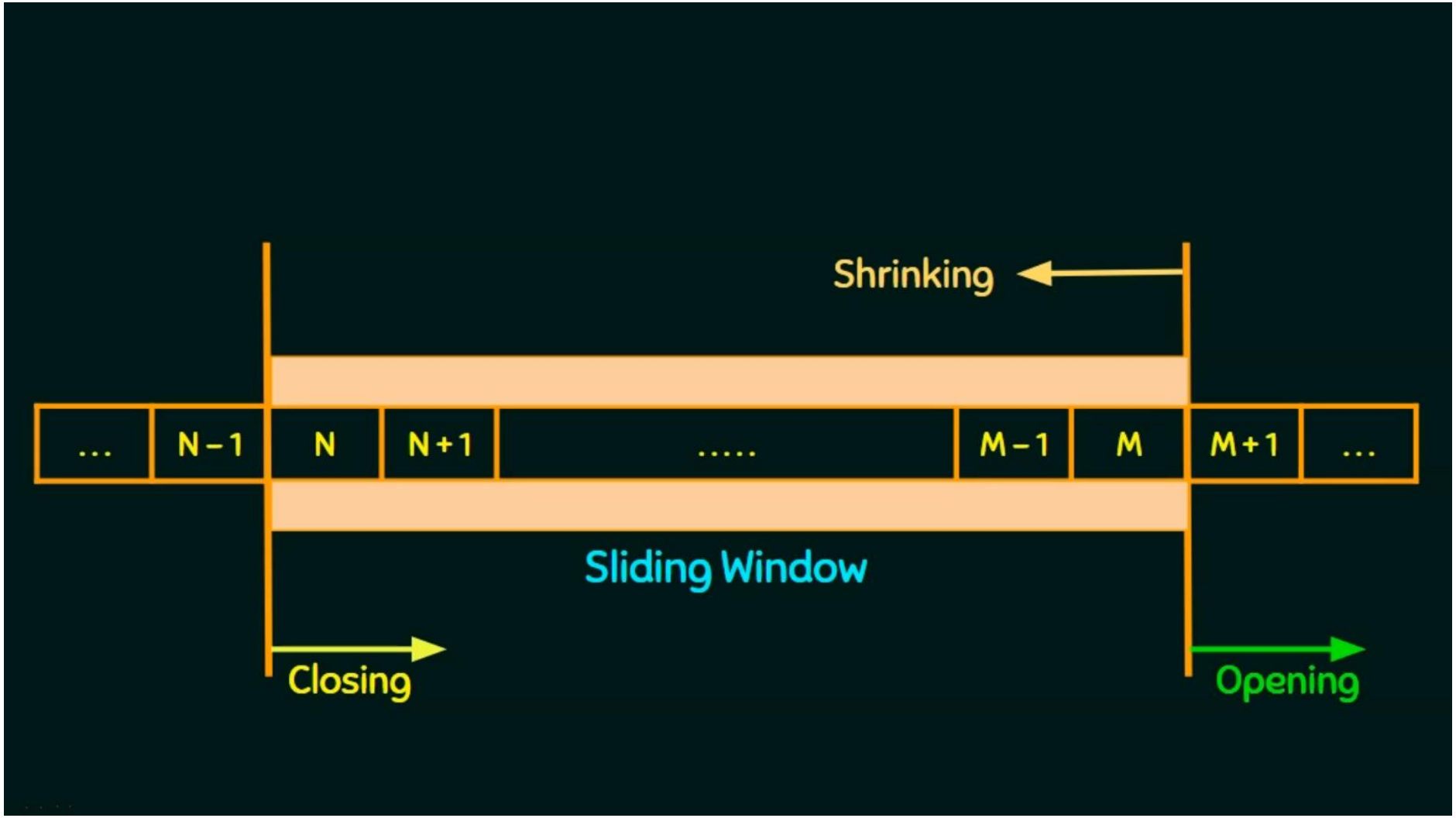
Flow Control

- TCP uses a sliding window to handle flow control
- TCP Sliding window is byte-oriented
- TCP's sliding window is of variable size
- Imaginary Window

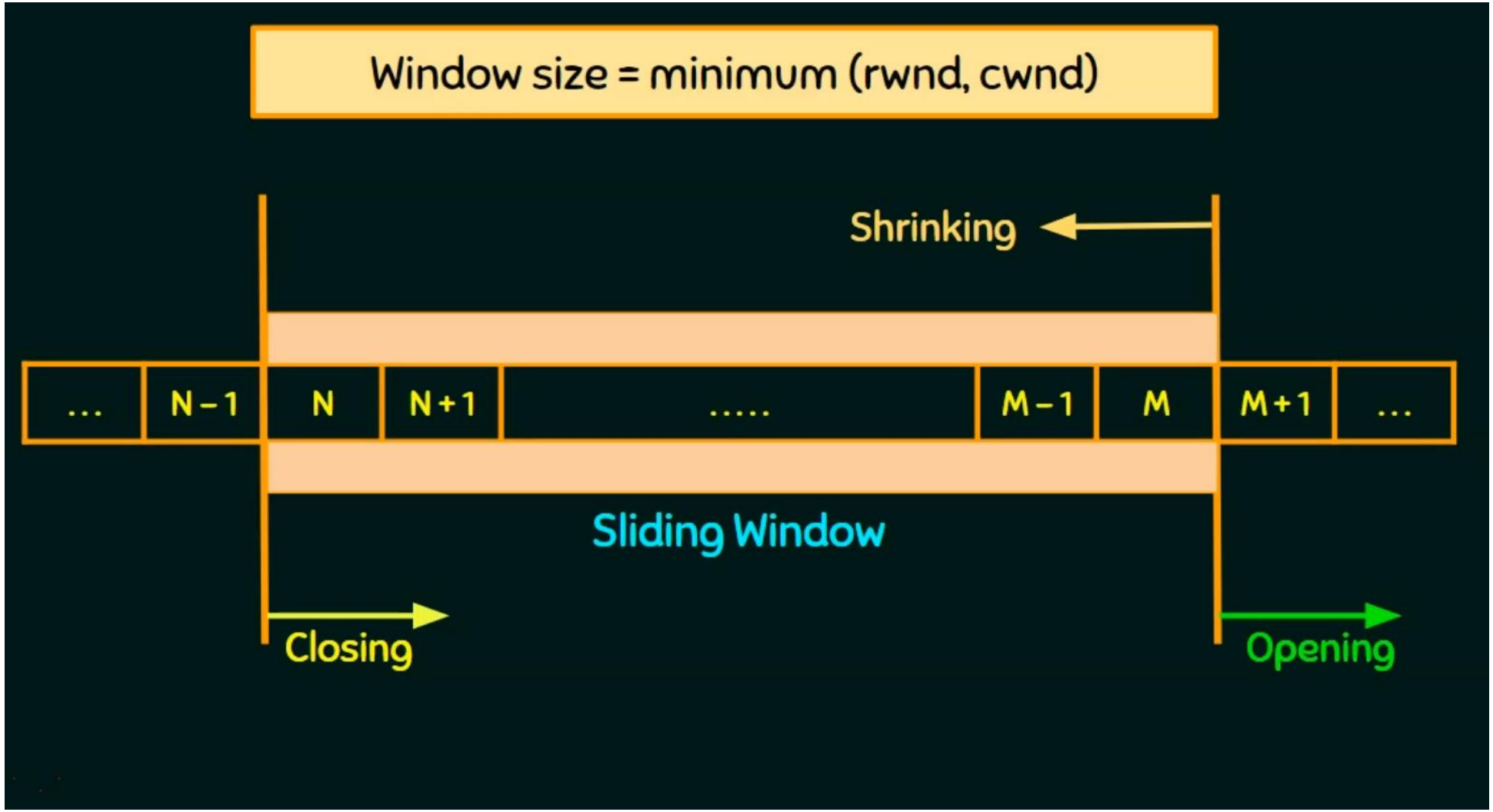
TCP Sliding Window



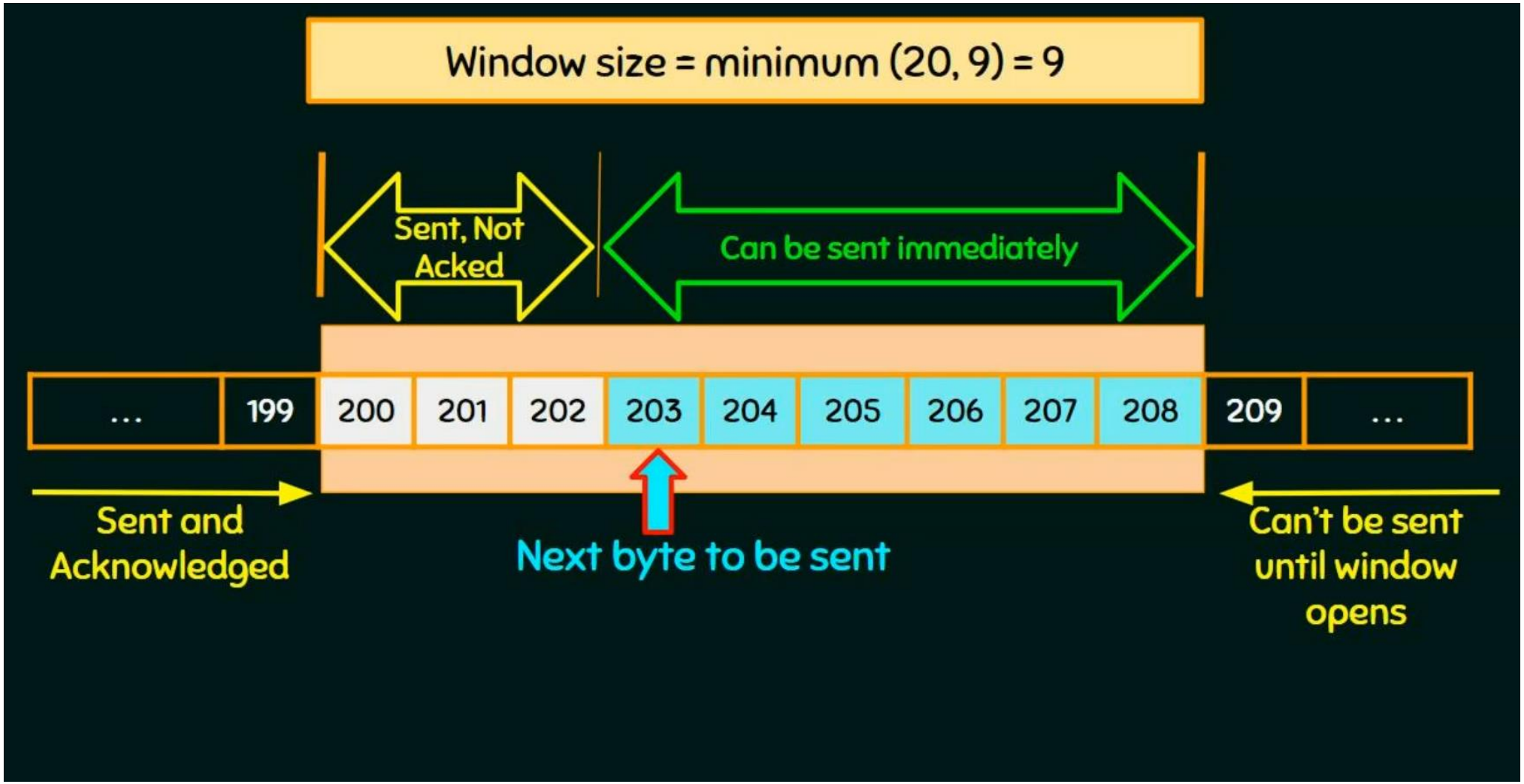
TCP Sliding Window



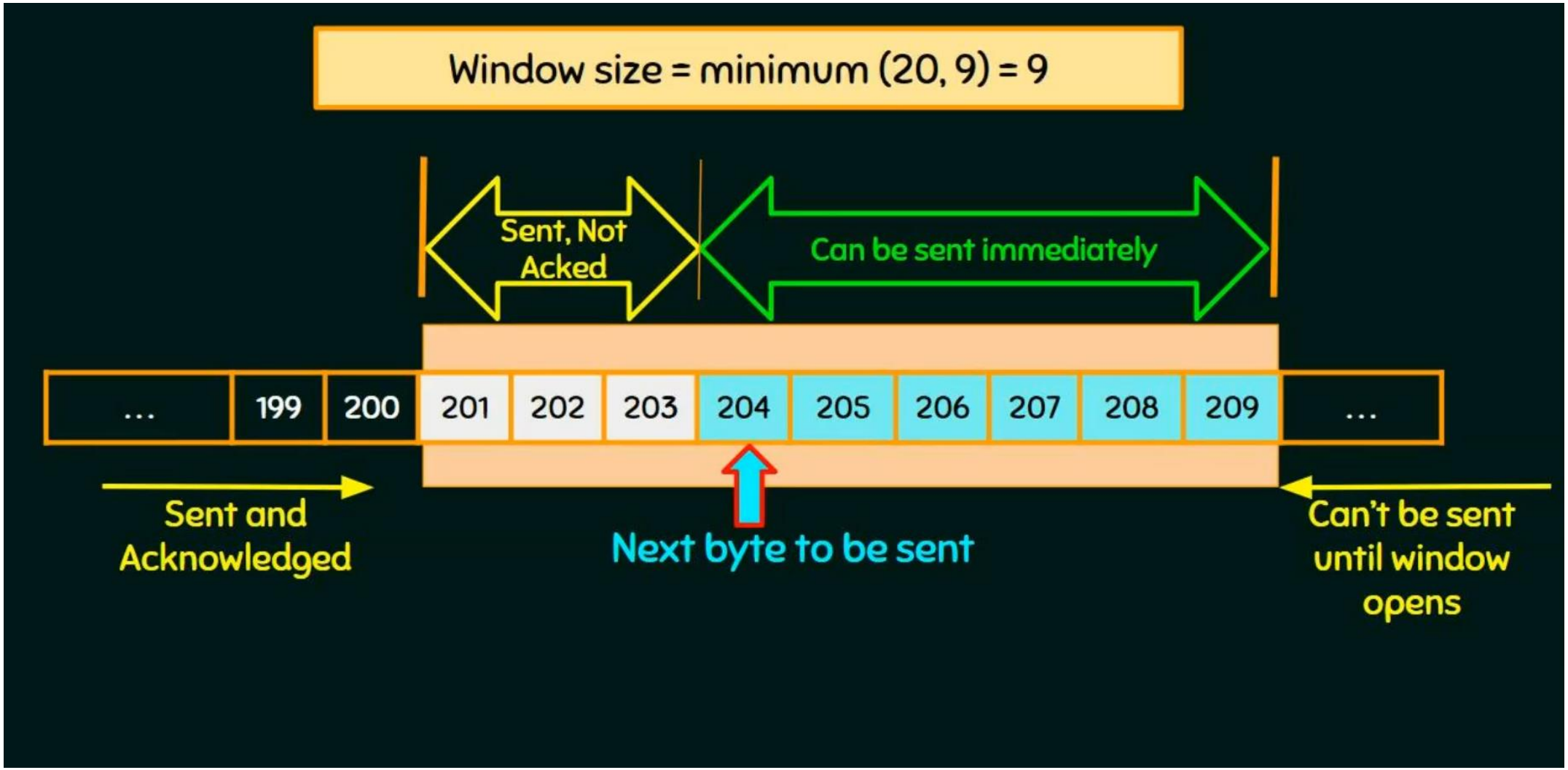
TCP Sliding Window



TCP Sliding Window



TCP Sliding Window



Important Points About TCP Sliding Window

- Window size = minimum(rwnd, cwnd)
- The source does not have to send a full window's worth of data
- The window can be opened or closed by the receiver, but should not be shrunk
- The receiver can send an ACK at any time