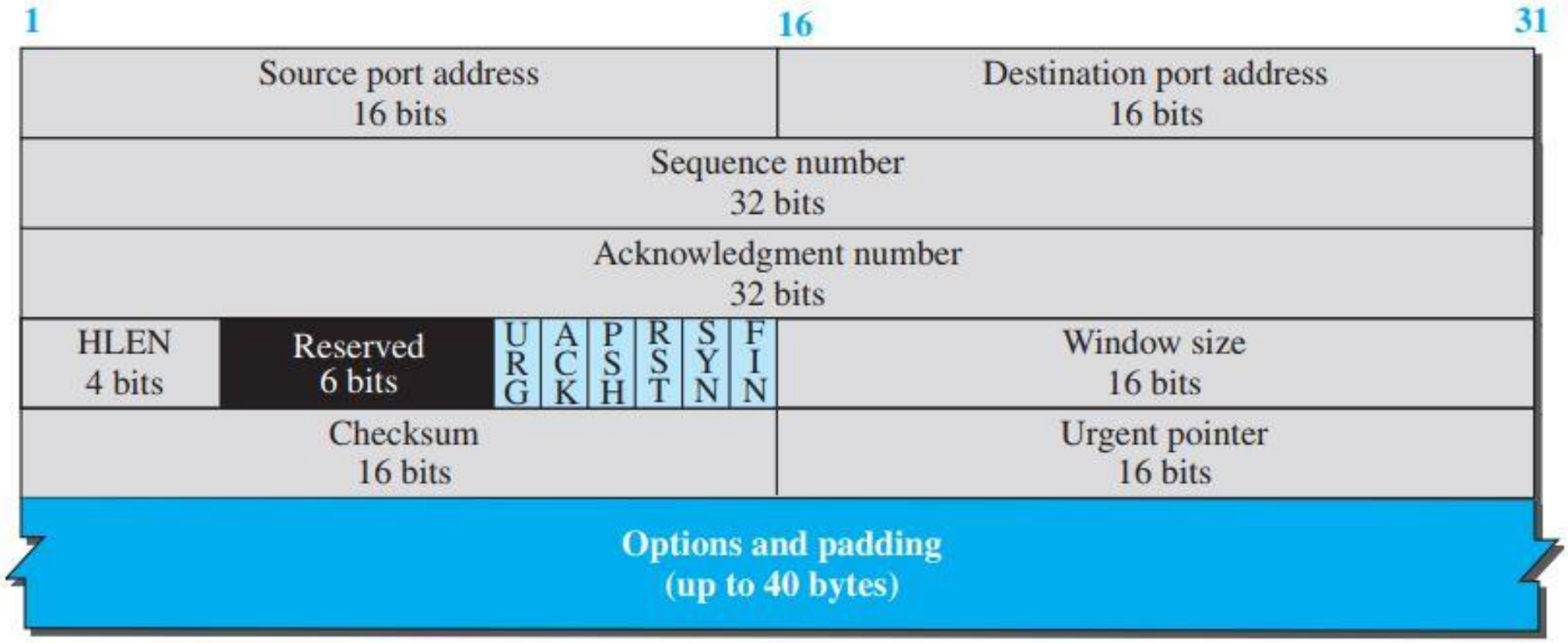
TCP Segment

a. Segment



b. Header

Fig 1. TCP segment format

TCP Segment Format

- **Source port address**
  - A 16-bit field that defines the port number of the application program running in the host that is sending the segment
- **Destination port address**
  - A 16-bit field that defines the port number of the application program running in the host that is receiving the segment
- **Sequence number**
  - A 32-bit field that defines the number assigned to the first byte of data in the segment
  - Sequence number tells the destination which byte in the sequence is the first byte in the segment
- **Acknowledgment number**
  - A 32-bit field that defines the byte number receiver is expecting
  - Acknowledgment and data can be piggybacked together

TCP Segment Format

- **Header length**
  - A 4-bit field indicates the number of 4-byte words in the TCP header

- **Control field**
  - One or more bits can be set at a time
  - The bits enable flow control, connection establishment and termination, connection abortion, and mode of data transfer in TCP

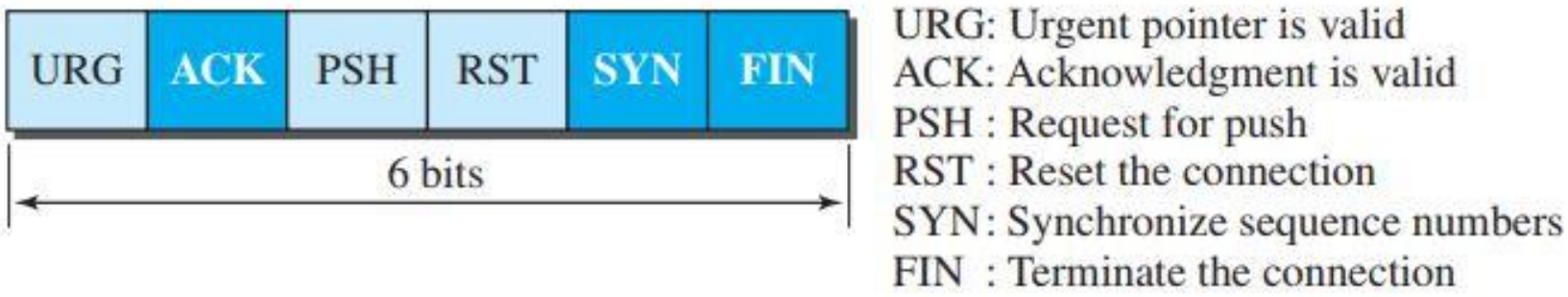


Fig 2. TCP control field

TCP Segment Format

- **Window size**
  - This field defines the window size of the sending TCP data in bytes
  - Normally known as receiving window (rwnd) and is determined by the receiver
  - Sender must obey the dictation of the receiver in this case

- **Checksum**

- **Urgent pointer**
  - A 16-bit field valid only when urgent flag is set
  - It is used only when the segment contains the urgent data

- **Options**

TCP Connection

- TCP is a connection-oriented protocol which establishes a logical path between the source and destination
- All segments belonging to a message are sent over the logical path
- TCP connection is logical, not physical. It operates at a higher level
- TCP uses the services of IP layer to deliver individual segments and controls the connection
- If a segment is lost or corrupted, it is retransmitted and IP is unaware of this retransmission
- TCP connection requires three phases: *connection establishment, data transfer, and connection termination*

Connection Establishment

- Connection establishment in TCP is called *three-way handshaking*
- Client makes connection to the server using TCP
- Server program tell its TCP that it is ready to accept the connection. This request is called a *passive open*
- A client tell its TCP to connect to a particular server and this request is called *active open*

Connection Establishment

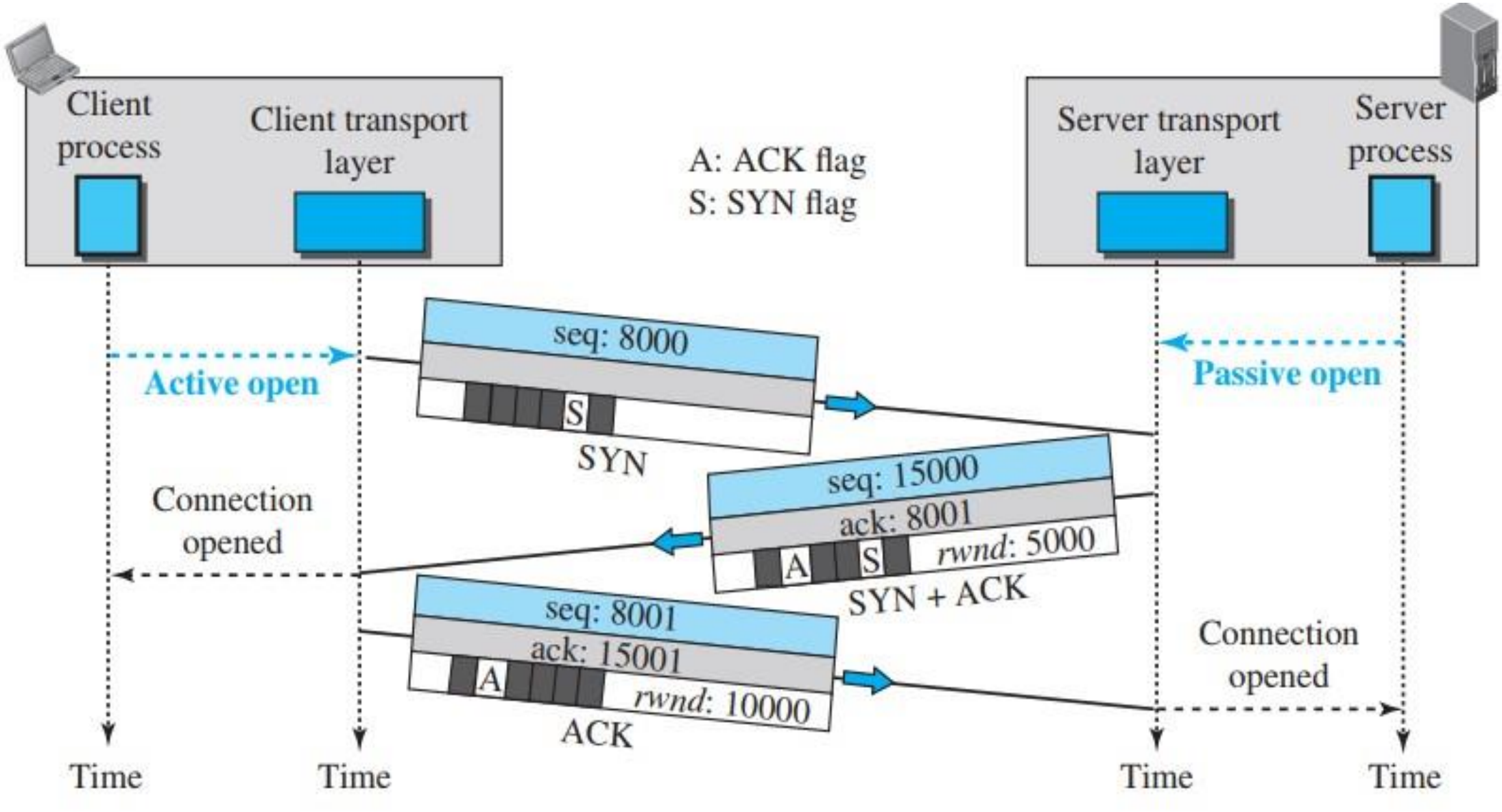


Fig 3. Three-way handshake

Connection Establishment

- The client sends the first segment, called a ***SYN*** segment, which is used to synchronize sequence numbers
- The client chooses a random number as the first sequence number (Initial Sequence Number) and sends this number to the server
- The ***SYN*** segment doesn't contain an acknowledgment number or define a window size
- It is a control segment and carries no data
- It consumes one sequence number because it needs to be acknowledged

Connection Establishment

- The server sends the second segment, a ***SYN + ACK*** segment.
- The server use this segment to initialize the sequence number
- The server acknowledges the reception of ***SYN*** segment from the client by setting the ***ACK*** flag and displaying the next sequence number it expects from the client
- The segment contains acknowledgment and, therefore, needs to define the receive window size (***rwnd***) that needs to be followed by the client
- This segment also needs to be acknowledged by the client and consumes one sequence number

Connection Establishment

- The client sends the third segment which is just an *ACK* segment
- Client acknowledges the reception of second segment
- The *ACK* segment (third segment) doesn't consume sequence number if it doesn't carry data
- The *ACK* segment (third segment) consumes sequence number if it carries data

SYN Flooding Attack

- One or more malicious attackers send many *SYN* segments to a server
- The attackers pretend to be clients by faking the source IP address in the datagram
- The server assumes the clients are issuing an active open and allocates the necessary resources
- The TCP server sends *SYN + ACK* segments, which are lost
- The server waits for the third step of the handshaking process, and if the number of *SYN* segments is large, the server will run out of resources
- This *SYN* flooding belongs to a group of security attacks called *"Denial of Service Attack"*

SYN Flooding Attack

- **How to alleviate denial of service attack**
  - Limit of connection for specific period
  - Filter out datagrams coming from unwanted source address
  - Using Cookie