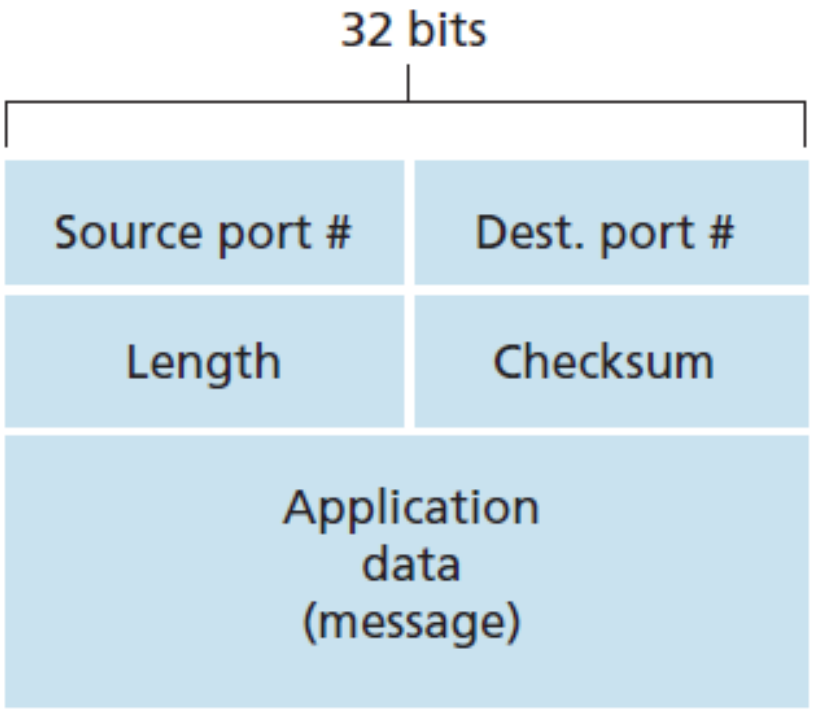
CS321: Computer Networks

UDP and TCP

Dr. Manas Khatua
Assistant Professor
Dept. of CSE
IIT Jodhpur
E-mail: manaskhatua@iitj.ac.in

Introduction of UDP

- UDP (User Datagram Protocol)
  - Transport-layer protocol
  - Connectionless
  - Simple
  - Efficient
  - Unreliable



UDP Services

- Process-to-process communication
  - Need socket address (IP + Port)
- Connectionless service
  - No sequence number
  - No relation between UDP datagrams
  - No connection establishment
  - Datagram can travel through different path
  - No segmentation (message size < (65535-8))
- No flow control
  - No window mechanism
- No error control
  - Error detection through checksum but no control

Cont…

- No congestion control
  - Assumption is that congestion will not be created as UDP datagrams are small in size
- Encapsulation and decapsulation
  - Needs to send message from one process to another
- Multiplexing and demultiplexing
  - One UDP, but several process in application layer wants to use its services
- Queuing
  - Queues are associated with port

Cont…

- Checksum
  - Consider three parts: Pseudoheader, UDP header, message from the upper layer; it is not mandatory;
  - Datalink layer has error detection mechanism. Why do we need checksum in transport layer?
    - neither link-by-link reliability nor in-memory error detection is guaranteed w.r.t. end-to-end service
  - Why do we need pseudoheader?
    - Socket-address need to be correct.
    - To ensure intended receiver (avoids in-memory error)

UDP Checksum

- What value is sent for the checksum in each one of the following hypothetical situations?
  - The sender decides not to include the checksum.
  - The sender decides to include the checksum, but the value of the sum is all 1s.
  - The sender decides to include the checksum, but the value of the sum is all 0s.
- Solution:
  - All 0s
  - When the sender complements the sum, the result is all 0s; the sender complements the result again before sending. The value sent for the checksum is all 1s. The second complement operation is needed to avoid confusion with the previous case. Note that this does not create confusion because the value of the checksum is never all Is in a normal situation
  - This situation never happens because it implies that the value of every term included in the calculation of the sum is all 0s, which is impossible.

UDP Applications

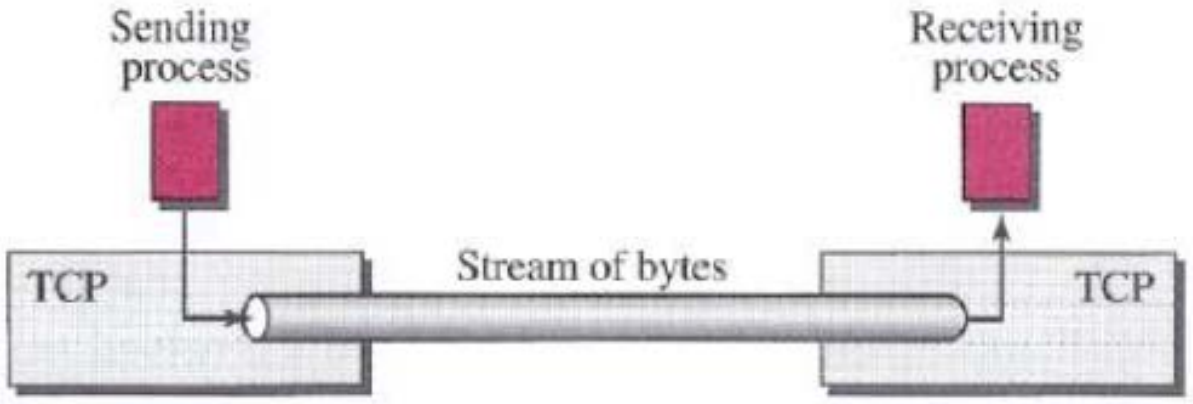
- If the request and response can each fit in a single user datagram, a connectionless service may be preferable.
  - E.g. DNS request and response
  - But, not suitable in SMTP as e-mail size could be large
- Lack of error control is advantageous sometimes
  - E.g. real time communication through Skype
  - But, not suitable for file download
- Lack of congestion control
  - Advantageous in error-prone network

Introduction to TCP

- TCP (Transmission Control Protocol)
  - Connection-oriented (but not virtual-circuit)
  - Reliable
  - Create connection, do data transfer, tear down connection
  - Uses ARQ protocols (GBN and SR)
  - Checksum for error detection
  - Retransmission of lost / corrupted packets
  - Cumulative / selective ACK
  - Commonly used in Internet

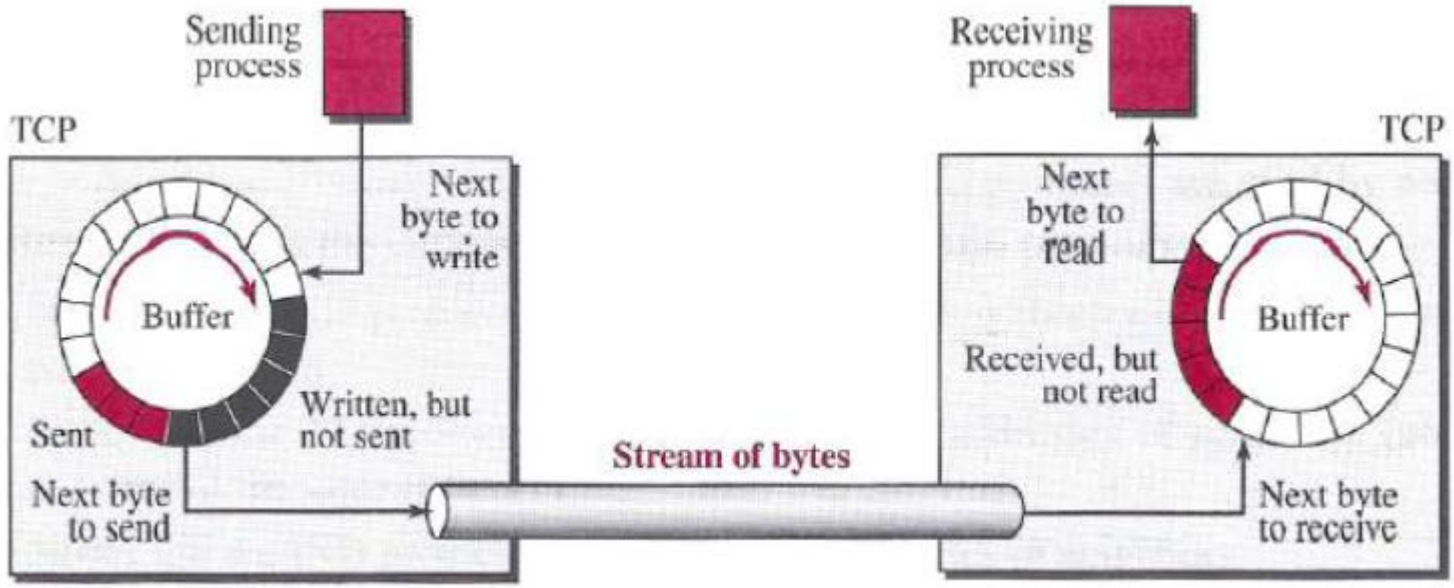
TCP Services

- Process-to-process communication
  - Needs socket address
- Stream Delivery Service



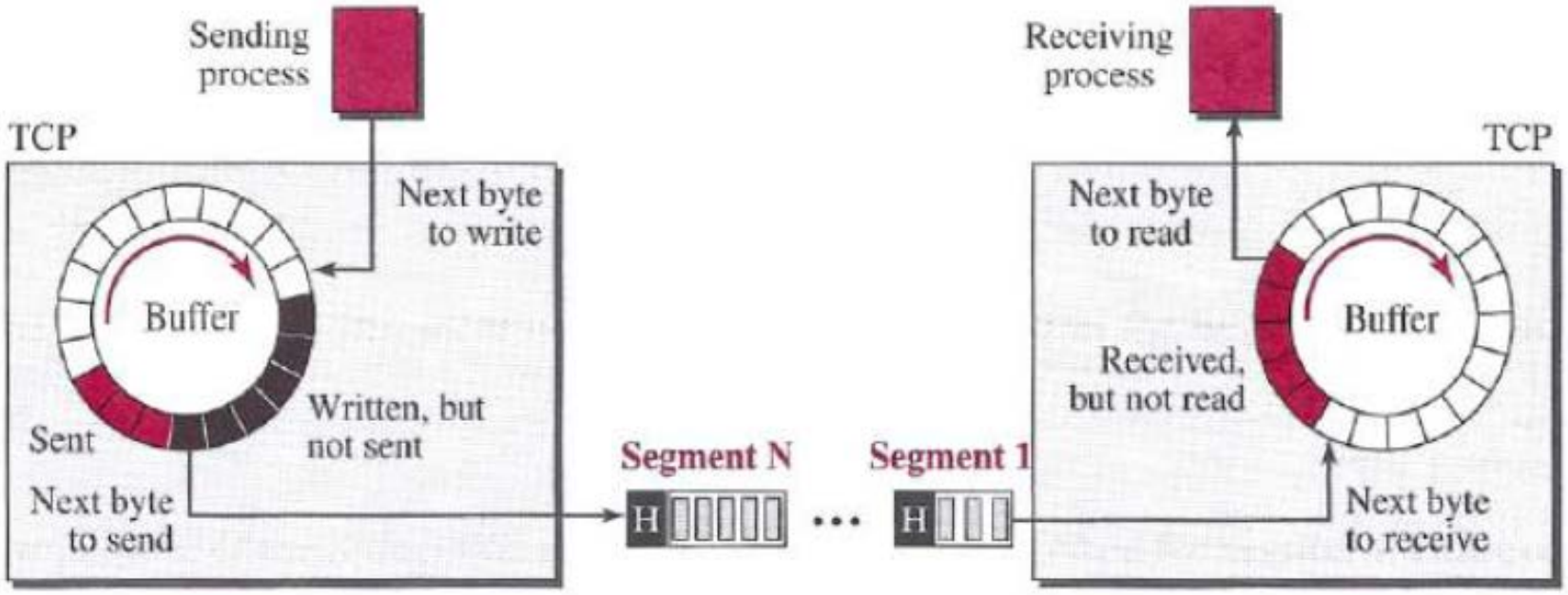
Cont…

- Sending and receiving buffer
  - may not necessarily write or read data at the same rate
  - Buffer allows to have flow control
- Full-Duplex Communication
  - Each TCP endpoint has its own sending and receiving buffer



Cont…

- Segmentation



- Reliable service
  - Using ACK

Cont…

- Multiplexing and Demultiplexing
  - But need connection between processes

- Connection-oriented service
  - two TCP's establish a logical connection
  - This is not a virtual-circuit switching
  - Connections are at two end systems only
  - Data are exchanged in both directions
  - The connection is terminated at the end of transmission

Numbering in TCP

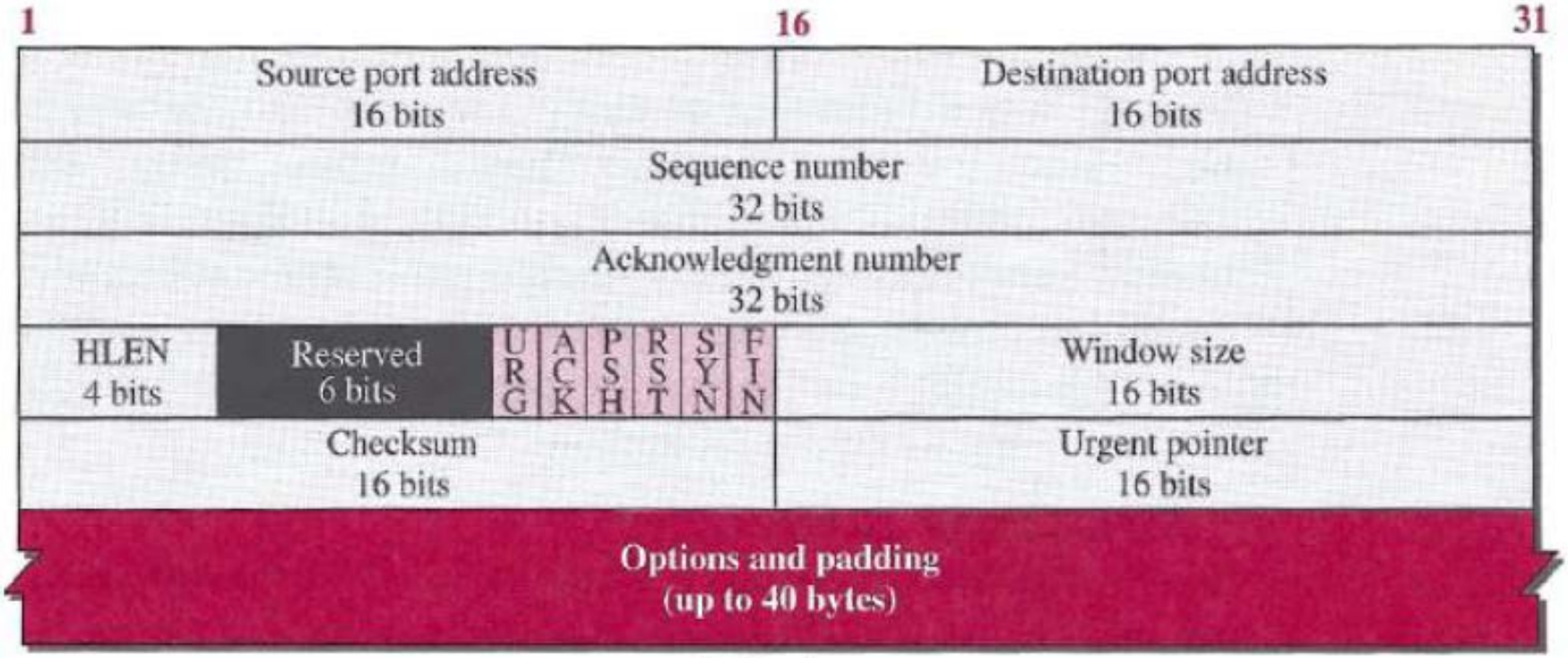
- No segment number
- Instead, we have sequence number and acknowledgement number
  - These are byte numbers, but not segment number
- Number is independent in each direction
- TCP numbers all data bytes transmitted in a connection
- TCP chooses an arbitrary number between 0 and 2^{32}- 1 for numbering the first byte

SEQ and ACK numbers

- After the bytes have been numbered, TCP assigns a SEQ number to each segment that is being sent
  - For 1st segment: SEQ number is random
  - For any other segment: SEQ number of previous segment + number of byte in the previous segment
- Each party also uses an ACK number to confirm the bytes it has received.
- The ACK number defines the number of the next byte that the party expects to receive.
- The ACK number is cumulative

TCP Segment Format

a. Segment



b. Header

Fields in a Segment

- Source & Destination port address.
- Sequence number & Acknowledgement number.
- Header length.
- Control. (6 control flags)
- Window size (window size of the sending TCP in bytes)
- Checksum. (pseudo header + TCP header + data)
- Urgent pointer (valid only if urgent flag is set)
- Options (optional information in the TCP header)

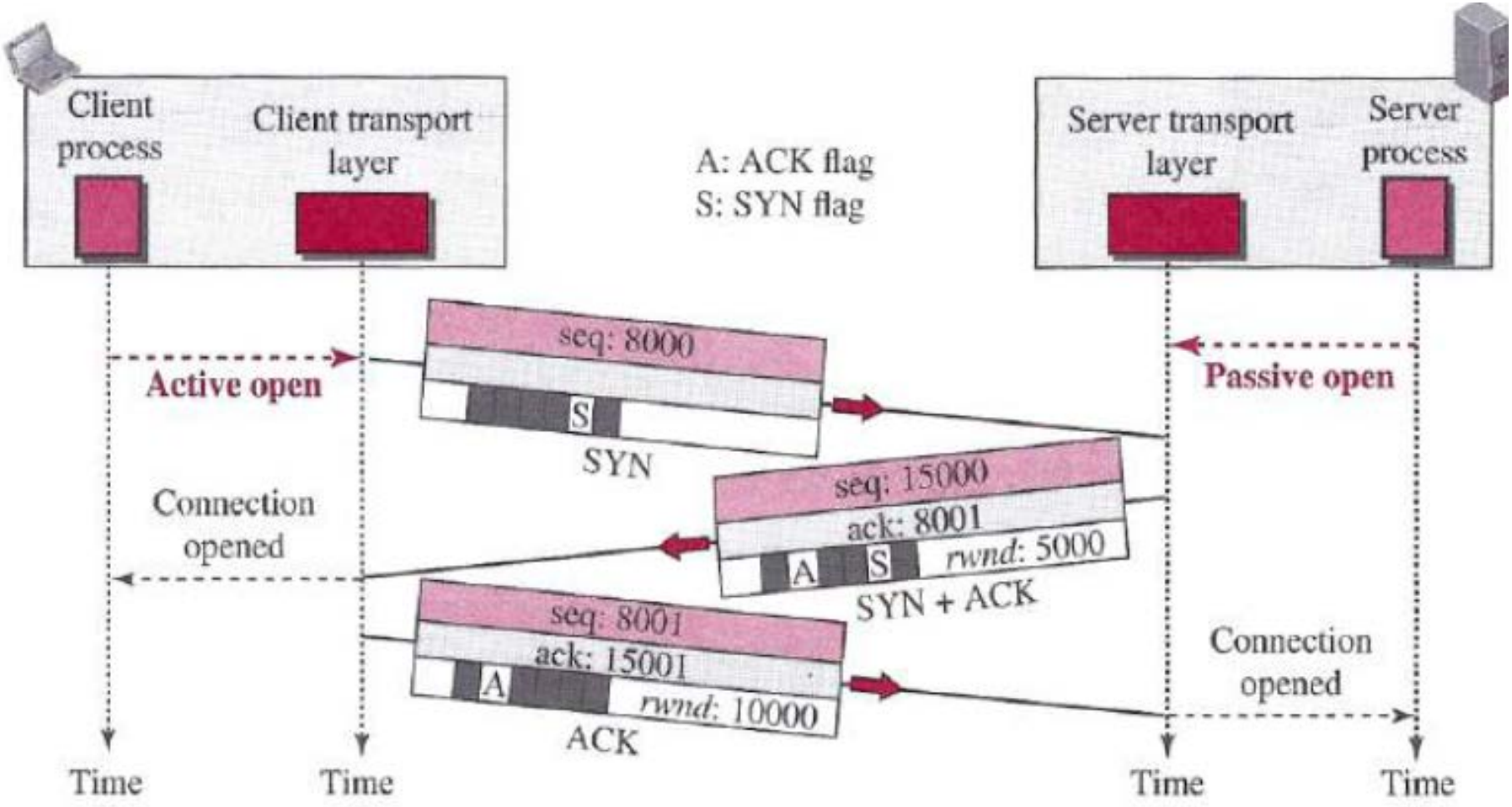
TCP Connection

- TCP connection is logical, not physical.
- TCP is connection-oriented although IP is connectionless
- TCP operates in full-duplex mode
- TCP uses three-way-handshaking

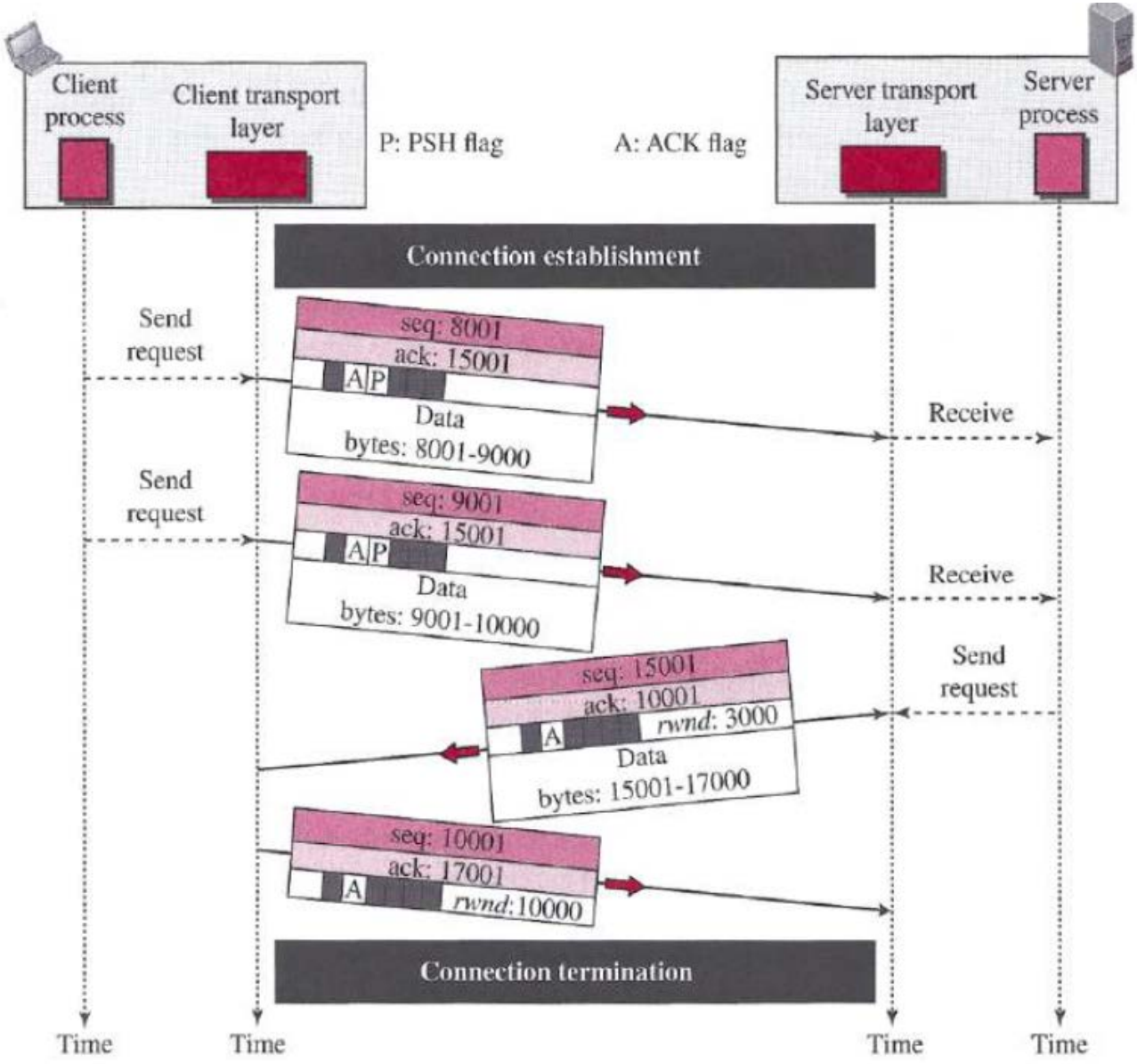
- Let, an application program, called the client, wants to make a connection with another application program, called the server, using TCP
- The process starts with the server.
  - Passive open (server process informs transport layer of server that it is ready)
  - Active open (client issues request to client transport layer)
  - Now Client transport layer starts three-way-handshaking

Connection Creation : step 1

- Using three-way-handshaking

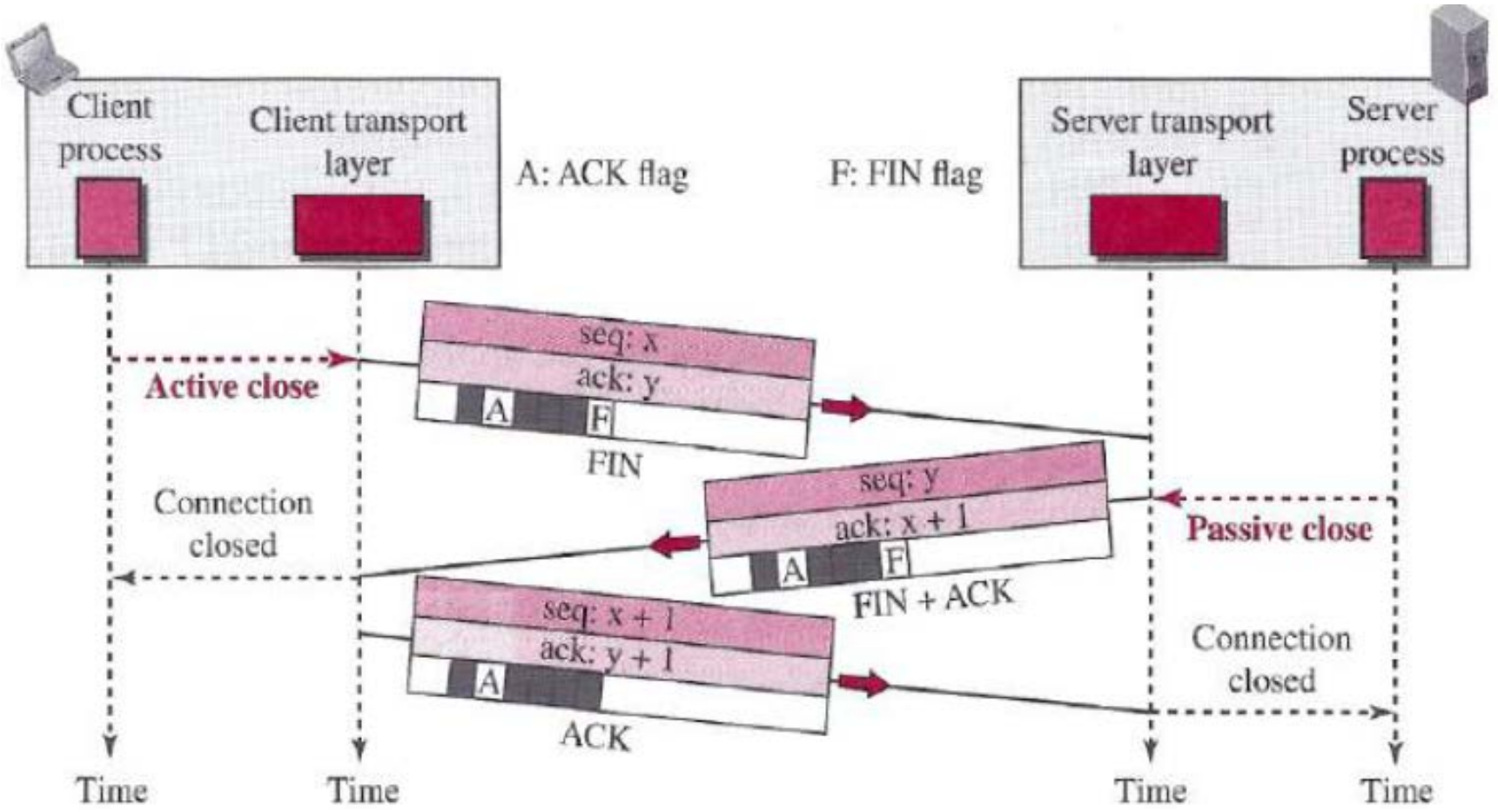


Data Transfer : step 2



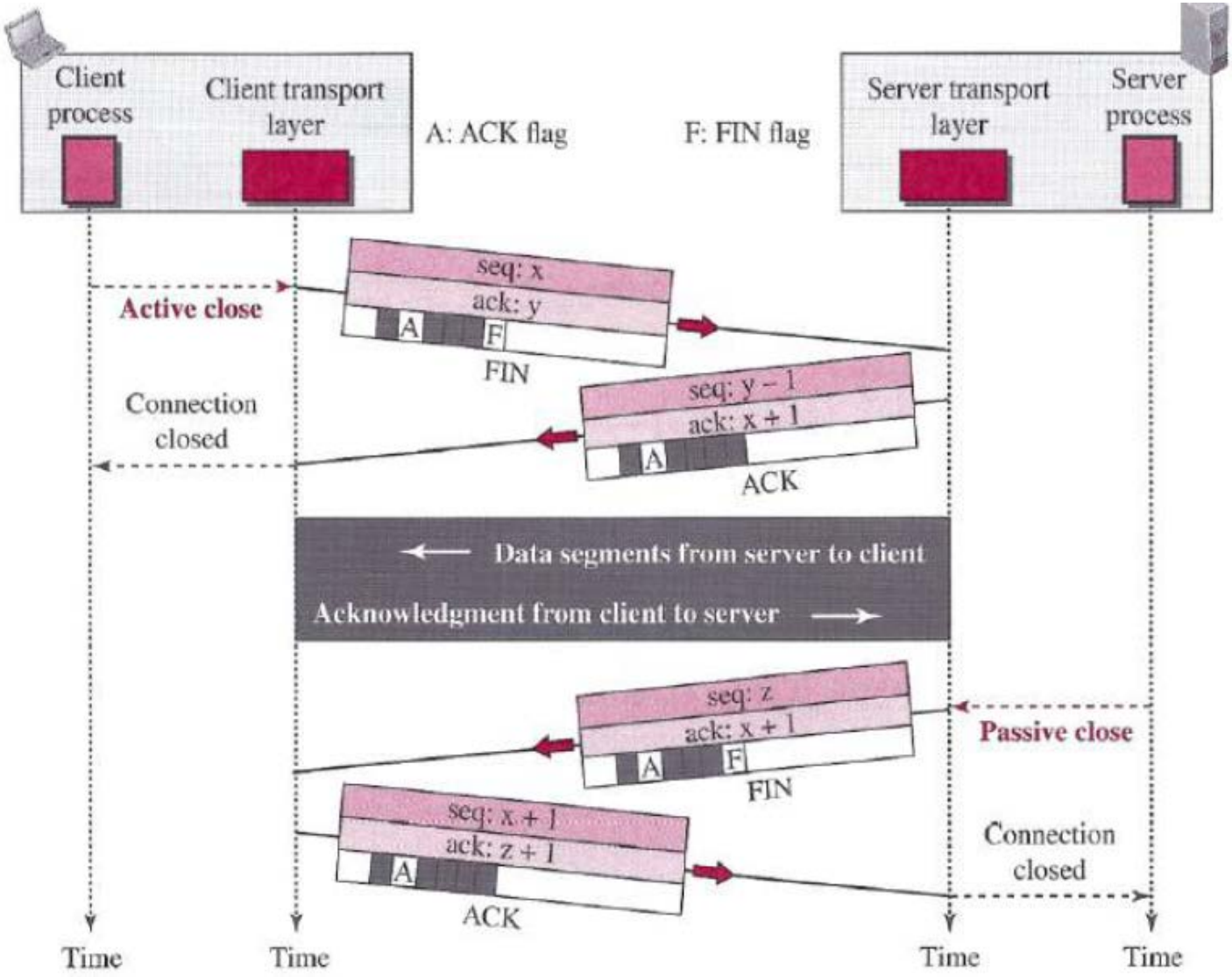
Connection Termination : step 3

- Using three-way-handshaking



Half-close Connection

Example:
Sorting
at server



Scenario

Create Connection

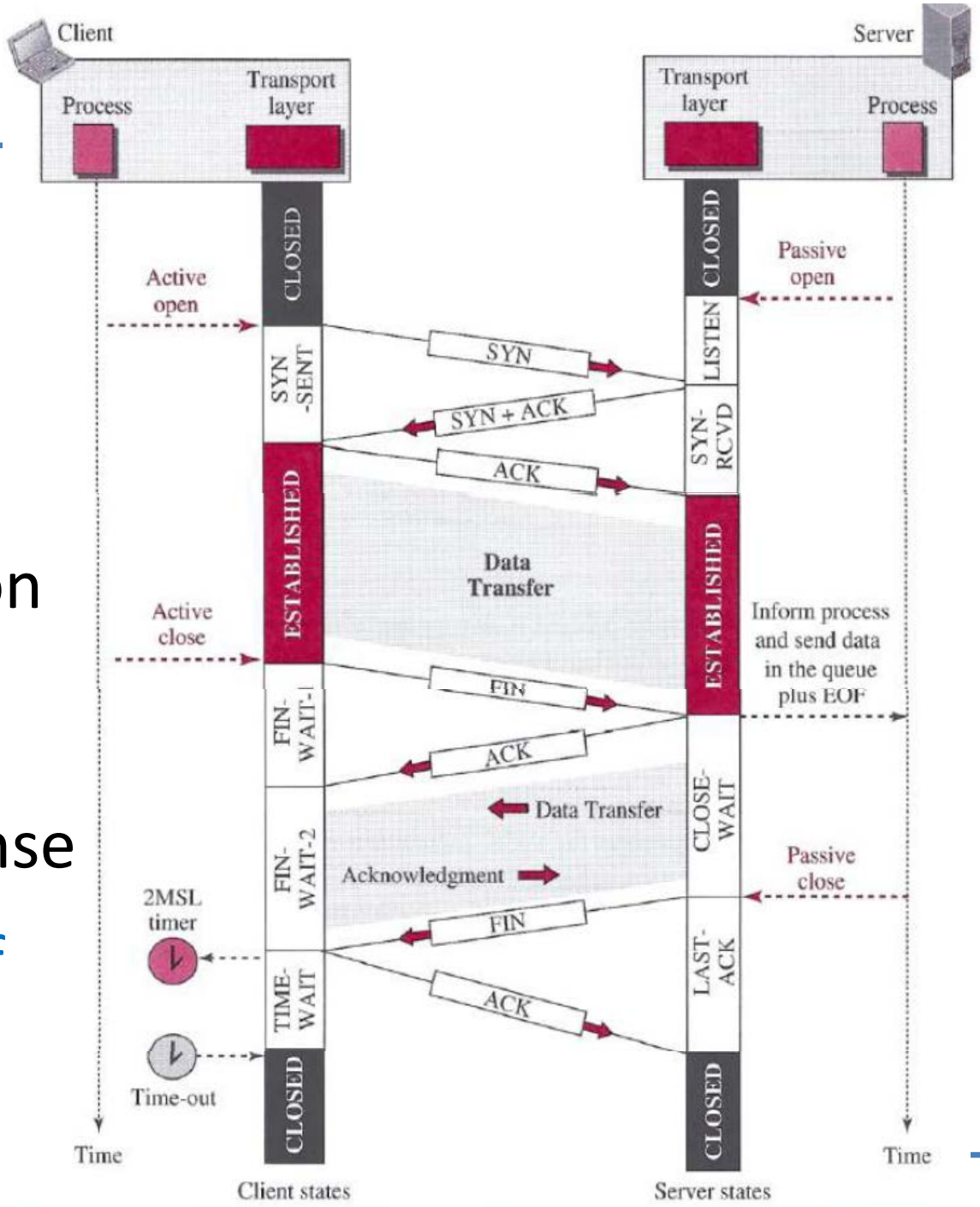
Data Transfer in both direction

Half-close

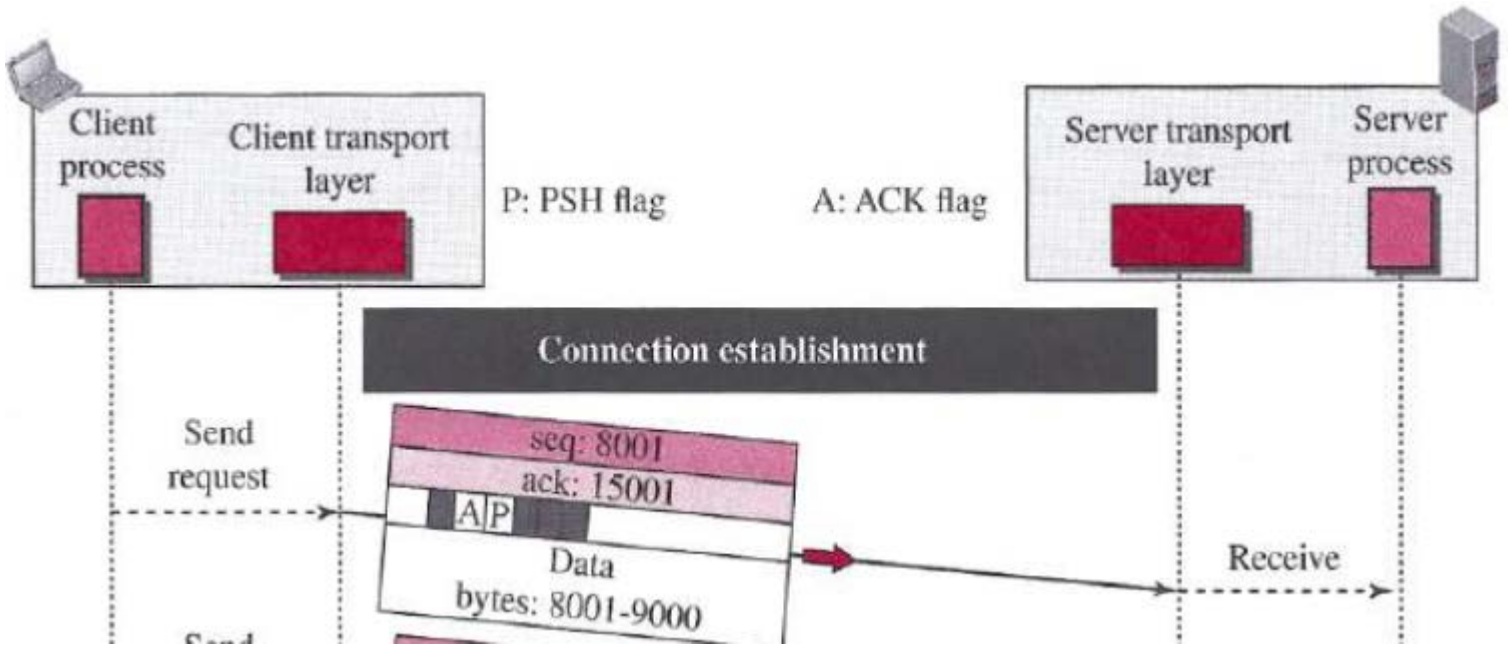
Receive Response

Close other half

Closed status



PUSH Flag



- PUSH means sending TCP must not wait for the window to be filled, and then send the segment
- PUSH flag informs the receiving TCP to deliver the received segment immediately to application program

Thanks!