SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 167
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

Course Name – 19IT401 Computer Networks

II Year / IV Semester

Unit 4 – Transport Layer

Topic 2 – TCP

TCP

- Transmission Control Protocol (TCP) is a connection-oriented, reliable protocol.
- TCP explicitly defines connection establishment, data transfer, and connection teardown phases to provide a connection-oriented service.
- TCP uses a combination of GBN and SR protocols to provide reliability.
- To achieve this goal, TCP uses checksum (for error detection), retransmission of lost or corrupted packets, cumulative and selective acknowledgments, and timers.

TCP - Services

TCP SERVICES

Process-to-Process Communication

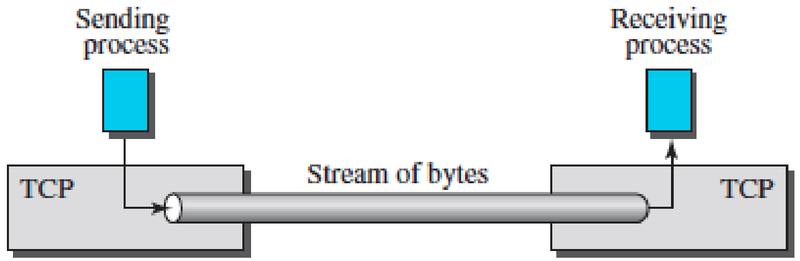
- TCP provides process-to-process communication using port numbers.

Stream Delivery Service

- TCP, unlike UDP, is a stream-oriented protocol.

- TCP allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.

- The sending process produces (writes to) the stream and the receiving process consumes (reads from) it. It uses sending buffers and receiving buffers for this purpose.



TCP - Services

TCP SERVICES

Full-Duplex Communication

TCP offers full-duplex service, where data can flow in both directions at the same time.

Multiplexing and Demultiplexing

TCP performs multiplexing at the sender and demultiplexing at the receiver.

Connection-Oriented Service

1. The two TCP's establish a logical connection between them.
2. Data are exchanged in both directions.
3. The connection is terminated.

Reliable Service

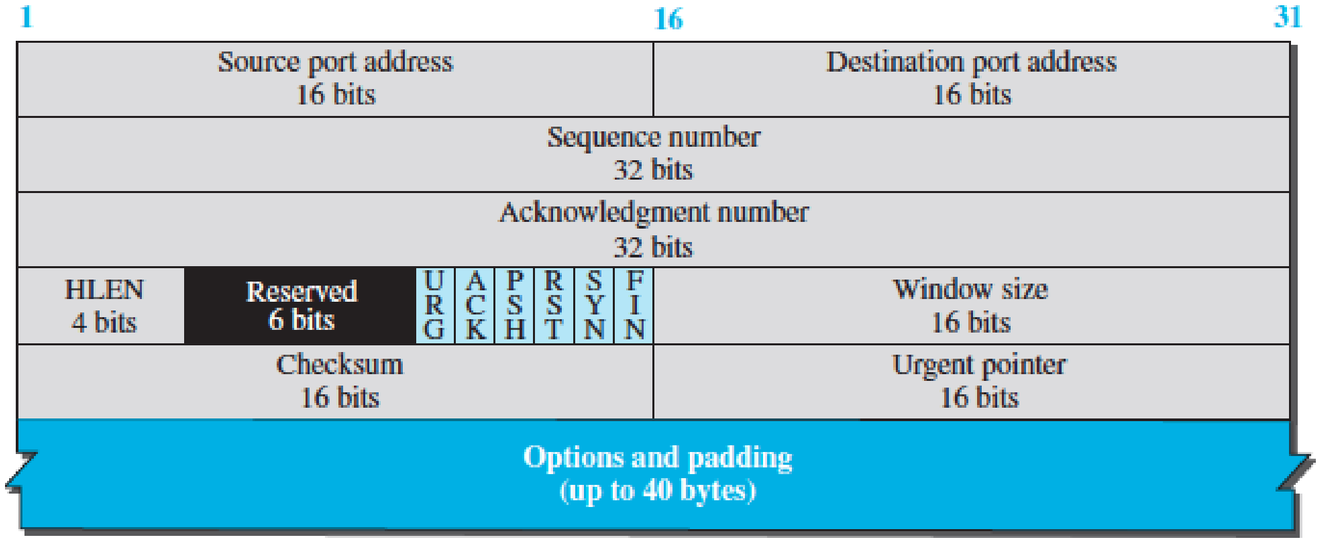
TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data

TCP - Segment

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

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

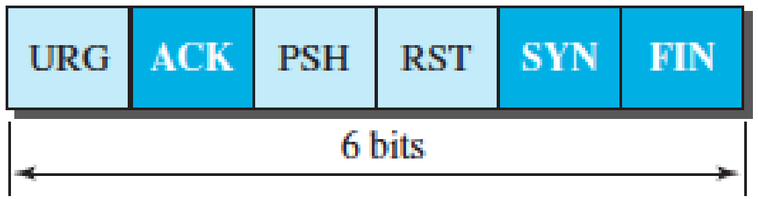
a. Segment



b. Header

TCP - Segment

Header length - 4-byte words in the TCP header. Value of this field is always between 5 (5x4 =20) and 15 (15x4 = 60).



URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH : Request for push
RST : Reset the connection
SYN: Synchronize sequence numbers
FIN : Terminate the connection

Window size - This field defines the window size of the sending TCP in bytes. the maximum size of the window is 65,535 bytes.

Urgent pointer - This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data.

TCP - Connection

In TCP, connection-oriented transmission requires three phases:

1. Connection establishment,
2. Data transfer, and
3. Connection termination.

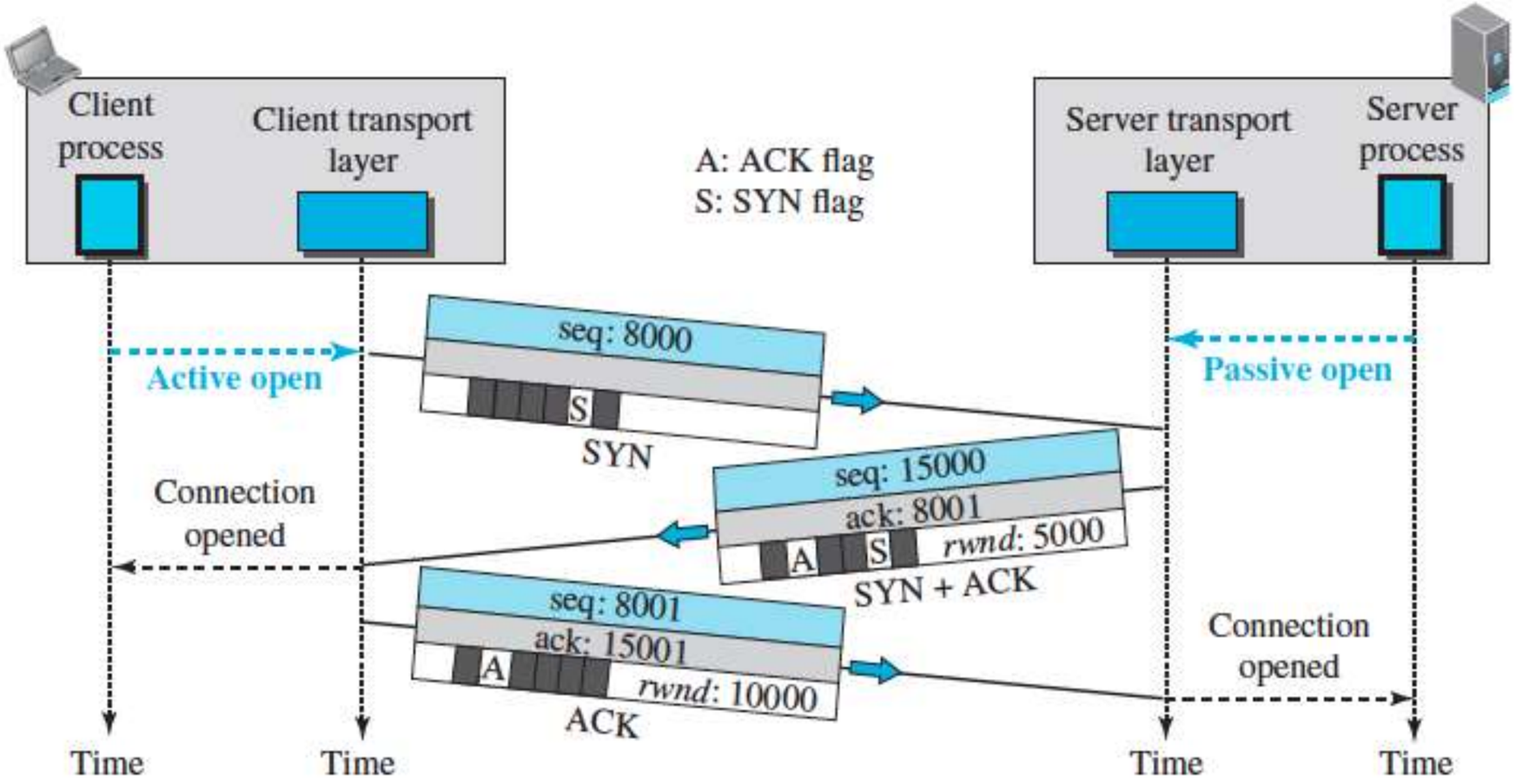
Connection Establishment

TCP transmits data in full-duplex mode.

Three-Way Handshaking

- The connection establishment in TCP is called three-way handshaking
- The server program tells its TCP that it is ready to accept a connection. This request is called a **passive open.**
- The client program issues a request for an **active open** . A client that wishes to connect to an open server tells its TCP to connect to a particular server.
- TCP can now start the three-way handshaking process

TCP - Connection



TCP - Connection

1. The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers.

2. The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK. This segment has a dual purpose. First, it is a SYN segment for communication in the other direction. The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client.

The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client.

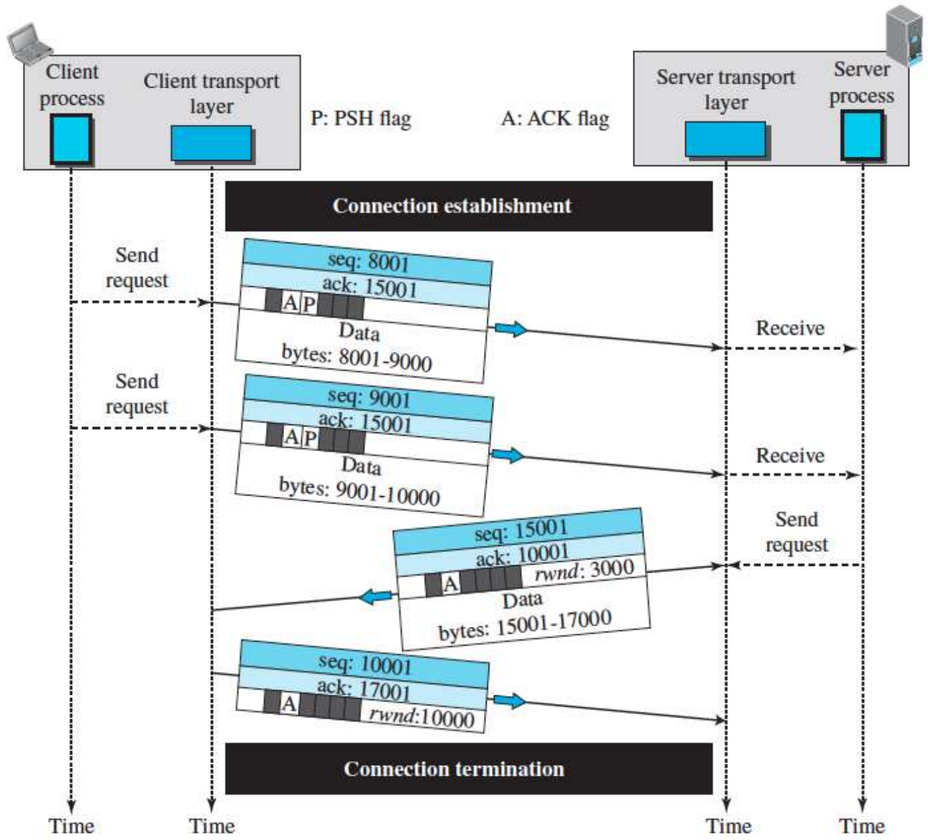
It also needs to define the receive window size, rwnd.

3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field

TCP - Connection

Data Transfer

After connection is established, bidirectional data transfer can take place. The client and server can send data and acknowledgments in both directions.



TCP - Connection

Data Transfer

Pushing data

The application program at the sender can request a push operation. This means that the sending TCP must not wait for the window to be filled. It must create a segment and send it immediately.

The sending TCP must also set the push bit (PSH) to let the receiving TCP know that the segment includes data that must be delivered to the receiving application program as soon as possible and not to wait for more data to come.

Urgent Data

There are occasions in which an application program needs to send urgent bytes, some bytes that need to be treated in a special way by the application at the other end. The solution is to send a segment with the URG bit set.

TCP - Connection

Connection Termination

Either of the two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client.

Most implementations today allow two options for connection termination: three-way handshaking and four-way handshaking with a half-close option.

Three-Way Handshaking

1. In this situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.

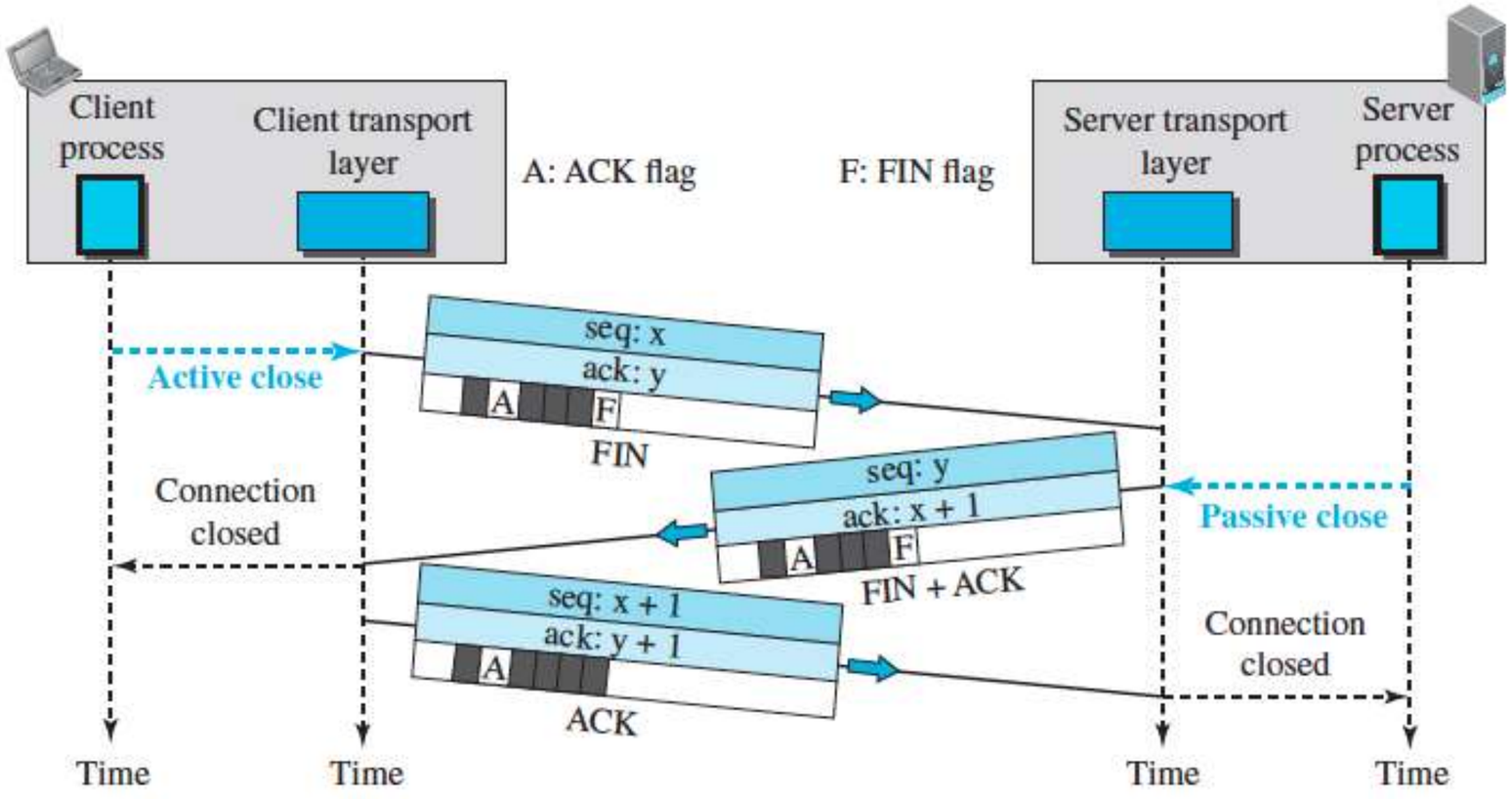
2. The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN + ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction.

3. The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server.

TCP - Connection

Half-Close

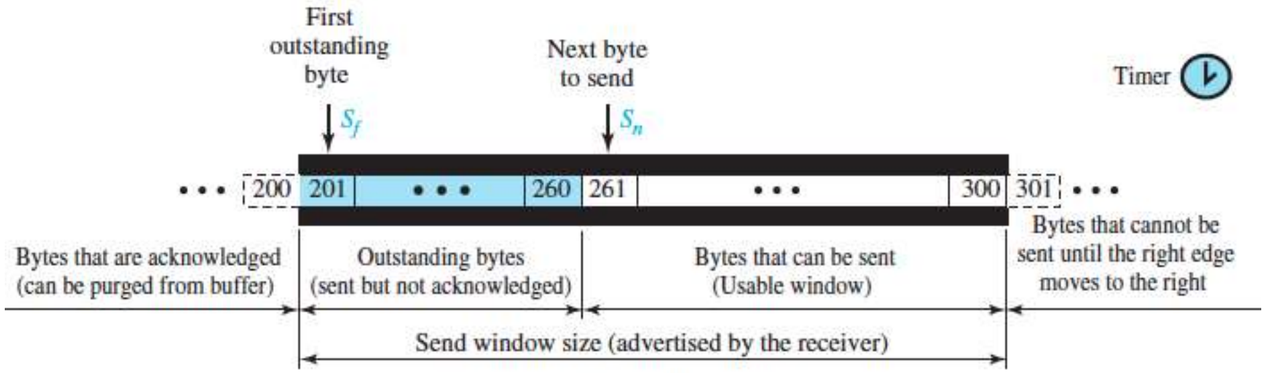
In TCP, one end can stop sending data while still receiving data. This is called a halfclose. Either the server or the client can issue a half-close request



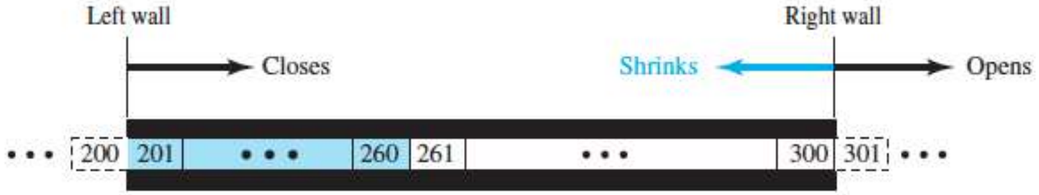
Windows in TCP

TCP uses two windows (send window and receive window) for each direction of data transfer, which means four windows for a bidirectional communication.

Send Window



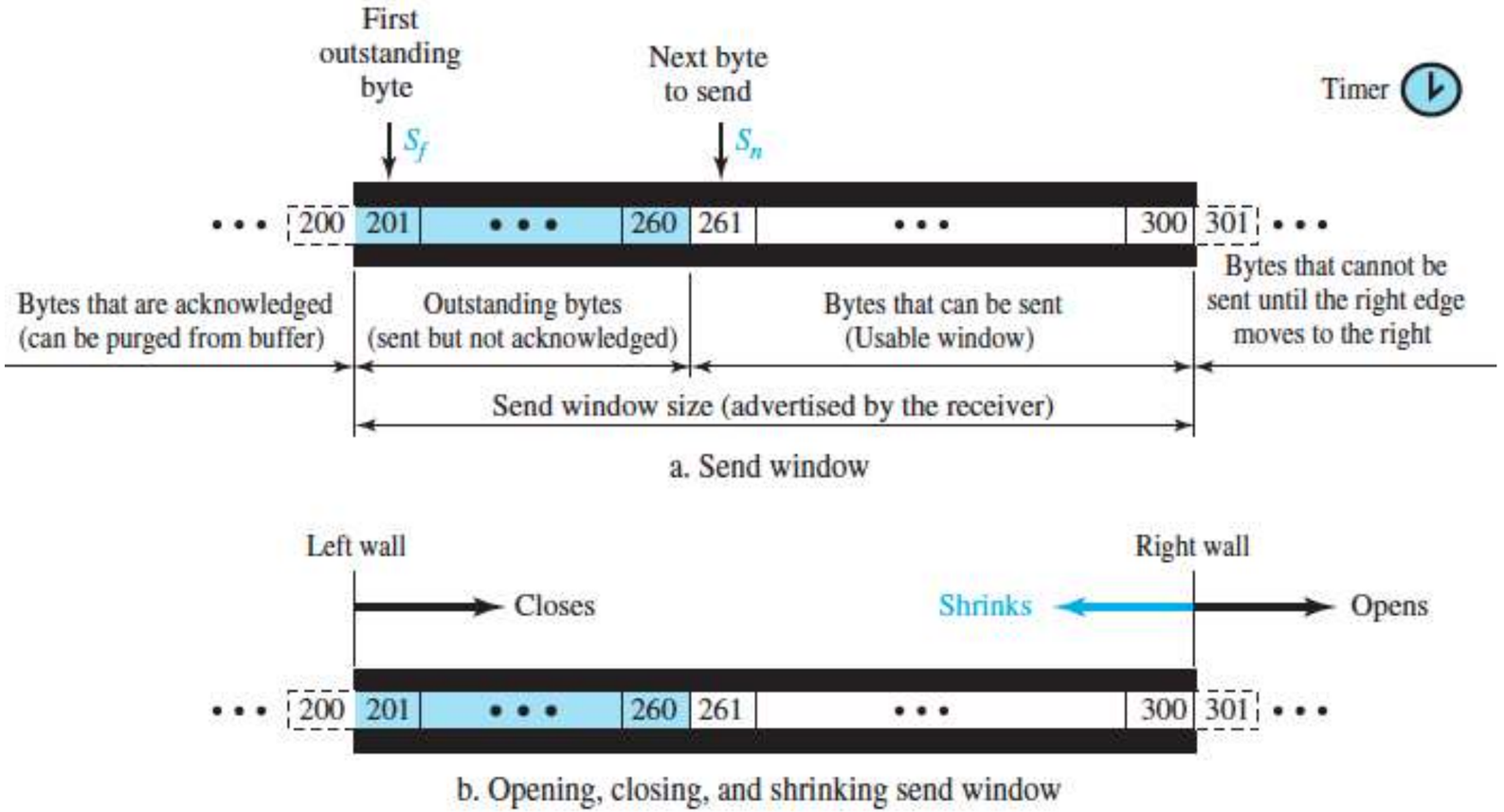
a. Send window



b. Opening, closing, and shrinking send window

Windows in TCP

Receiver Window



THANK YOU