Computer Networks

BCST -502 BCSP- 502

B.Tech (CSE) 5th Semester

Course Instructor: Dr Bishwajeet Pandey

New 2020 Syllabus

Unit –I

Computer Network: Definitions, goals, components, Architecture, Classifications & Types.Layered Architecture: Protocol hierarchy, Design Issues, Interfaces and Services, ConnectionOriented & Connectionless Services, Service primitives, Design issues & its functionality. ISOOSI Reference Model: Principle, Model, Descriptions of various layers and its comparison with TCP/IP. Principals of physical layer: Media, Bandwidth, Data rate and Modulations

Unit-II

Data Link Layer: Need, Services Provided, Framing, Flow Control, Error control. Data Link Layer Protocol: Elementary &Sliding Window protocol: 1-bit, Go-Back-N, Selective Repeat, Hybrid ARQ. Protocol verification: Finite State Machine Models & Petri net models. ARP/RARP/GARP

Unit-III

MAC Sub layer: MAC Addressing, Binary Exponential Back-off (BEB) Algorithm, Distributed Random Access Schemes/Contention Schemes: for Data Services (ALOHA and Slotted- ALOHA), for Local-Area Networks (CSMA, CSMA/CD, CSMA/CA), CollisionFree Protocols: Basic Bit Map, BRAP, Binary Count Down, MLMA Limited Contention Protocols: Adaptive Tree Walk, Performance Measuring Metrics. IEEE Standards 802 series & their variant.

New 2020 Syllabus

Unit-IV

Network Layer: Need, Services Provided, Design issues, Routing algorithms: Least CostRouting algorithm, Dijkstra's algorithm, Bellman-ford algorithm, Hierarchical Routing,Broadcast Routing, Multicast Routing. IP Addresses, Header format, Packet forwarding,Fragmentation and reassembly, ICMP, Comparative study of IPv4 & IPv6

Unit-V

Transport Layer: Design Issues, UDP: Header Format, Per-Segment Checksum, CarryingUnicast/Multicast Real-Time Traffic, TCP: Connection Management, Reliability of DataTransfers, TCP Flow Control, TCP Congestion Control, TCP Header Format, TCP TimerManagement. Application Layer: WWW and HTTP, FTP, SSH, Email (SMTP, MIME, IMAP), DNS, Network Management (SNMP).

ABOUT COURSE INSTRUCTOR

- PhD from Gran Sasso Science Institute, Italy
- PhD Supervisor Prof Paolo Prinetto from Politecnico Di Torino, World Rank 13 in Electrical Engineering
- MTech from Indian Institute of Information Technology, Gwalior
- Visited 32 Countries Across The Globe
- Written 200+ Research paper with 193 Researcher from 63 Universities
- Scopus Profile: https://www.scopus.com/authid/detail.uri?authorId=57203239026
- Google Scholar: https://scholar.google.com/citations?user=UZ_8yAMAAAAJ&hl=hi
- Contact: gyancity@gyancity.com, +91-7428640820 (For any help @ BIAS and Guidance for future MS from Europe and USA after BIAS)

Course Objectives

- The course aims to develop an understanding of the fundamentals of Computer Network among the students

- The course explores different components of computer network, types of protocols, modern network technologies and their applications.

Course Outcomes

After completing this course the student will be well equipped with the following concepts:

1. The student will be able to recognise the technological trends of Computer Networking.
2. The student will be able to discuss the key technological components of the Network and evaluate the challenges in building the network and find solutions for the same.
3. The student could understand the basic computer network technology as an isolating concept.

Course Outcomes

4. The student will be thorough in concepts of Data Communication system and its components

5. The student will be able to identify and distinguish between different types of network topologies and protocols.

6. The student will have in depth knowledge of the the layers of the OSI model and TCP/IP and will able to explain the function(s) of each layer.

Course Outcomes

7. The student will be able to identify the different types of network devices and their functions within a network

8. The student would have the skill to understand the building skills of subnetting and routing mechanisms.

9. Upon familiarity with the above concepts the student will be able to assist in network design and its implementation in real time.

About Course Outline

- UNIT 1:
    - Theory Lecture No 1-4, Lecture 29
    - Lab on Vivado: Lecture 9-11
- UNIT 2: Theory Lecture No 5-8
- UNIT 3: Theory Lecture No 14-18
- UNIT 4:
    - Theory Lecture No 12-13, 19-21, 36
    - Lab on Packet Tracer and C: Lecture 24-28
- UNIT 5: Theory Lecture No 30-35
- Student Assignment Presentation: 22-23
- Lecture No 37-42: Discuss Previous Year Question of UTU

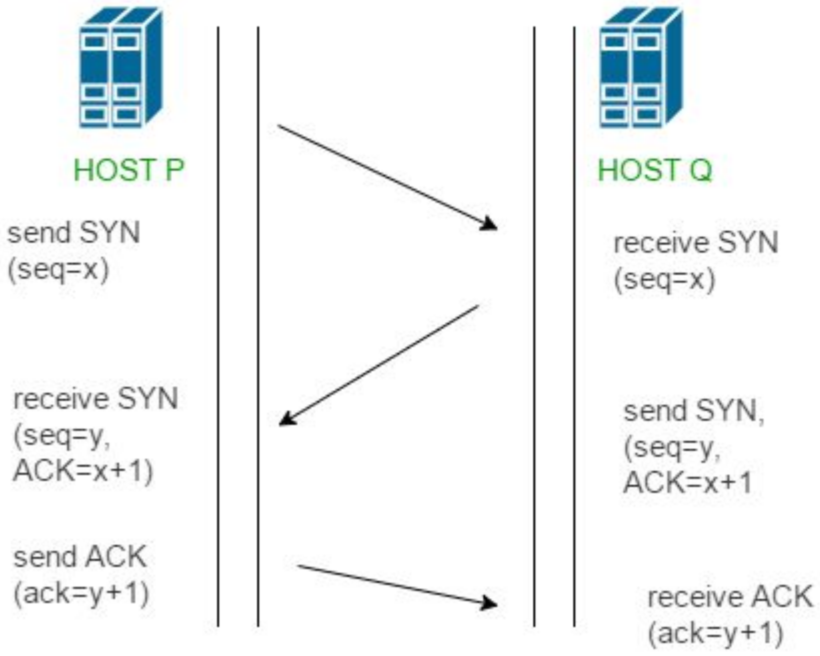
OUTLINE OF LECTURE 33

- TCP:
  - Connection Management,
  - Reliability of Data Transfers,
  - TCP Flow Control,
  - TCP Congestion Control,

TCP: Connection Management

- TCP 3-Way Handshake Process/TCP Connection Establishment
- TCP Connection Termination

TCP 3-Way Handshake Process



- The steps 1, 2 establish the connection parameter (sequence number) for one direction and it is acknowledged.
- The steps 2, 3 establish the connection parameter (sequence number) for the other direction and it is acknowledged.
- With these, a full-duplex communication is established.

TCP 3-Way Handshake Process

- **Step 1 (SYN) :** In the first step, client wants to establish a connection with server, so it sends a segment with SYN(Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segments with
- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with
- **Step 3 (ACK) :** In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer

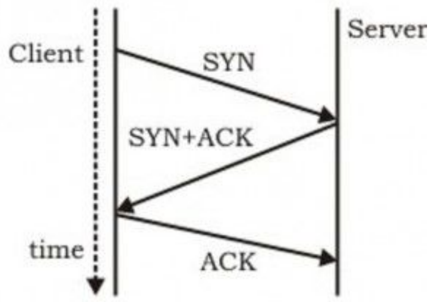
TCP Connection Establishment

- Since the connection establishment phase of TCP makes use of 3 packets, it is also known as 3-way Handshaking **(SYN, SYN + ACK, ACK).**

- TCP is a connection oriented protocol and every connection oriented protocol needs to establish connection in order to reserve resources at both the communicating ends.

Gate IT 2008 | Question 67

The three way handshake for TCP connection establishment is shown:



Which of the following statements are TRUE?

(S1) Loss of SYN + ACK from the server will not establish a connection

(S2) Loss of ACK from the client cannot establish the connection

(S3) The server moves LISTEN → SYN_RCVD → SYN_SENT → ESTABLISHED in the state machine on no packet loss

(S4) The server moves LISTEN → SYN_RCVD → ESTABLISHED in the state machine on no packet loss.

(A) S2 and S3 only

(B) S1 and S4

(C) S1 and S3

(D) S2 and S4

Gate IT 2008 | Question 67

Answer: (B)

Explanation: Before Three Way Hand Shake both client and server are in closed state for start sending or receiving both client and server comes in Listen state.

Steps:

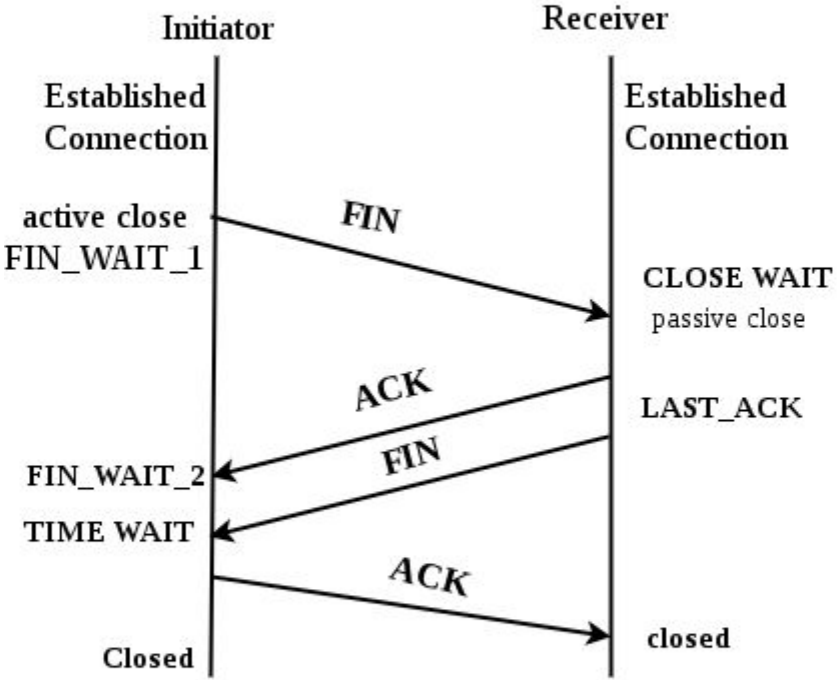
1) Client sent SYN packet which will be received by the server.

2) Server will SYN + ACK packet so as to establish the connection of client. Now Client is ready to send the data.

3) Then Client will send ACK packet to server when this packet is received by server the server will also be in established state.

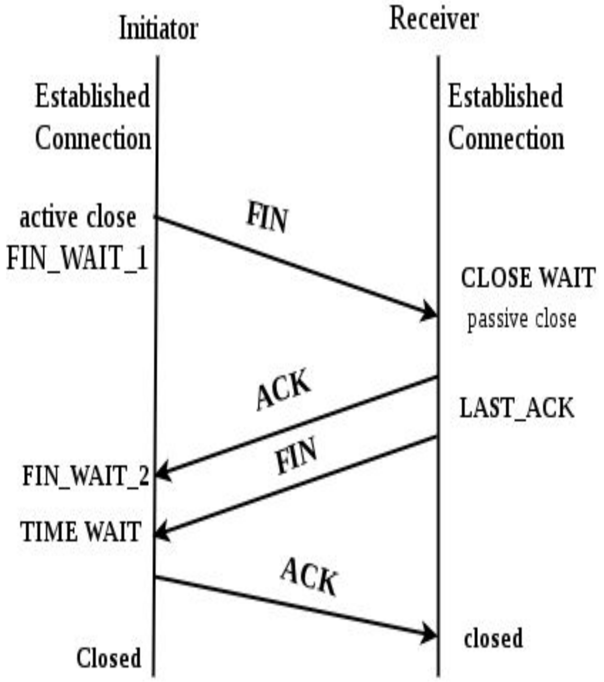
TCP Connection Termination

- Here we will also need to send bit segments to server which **FIN** bit is set to 1.



TCP Connection Termination

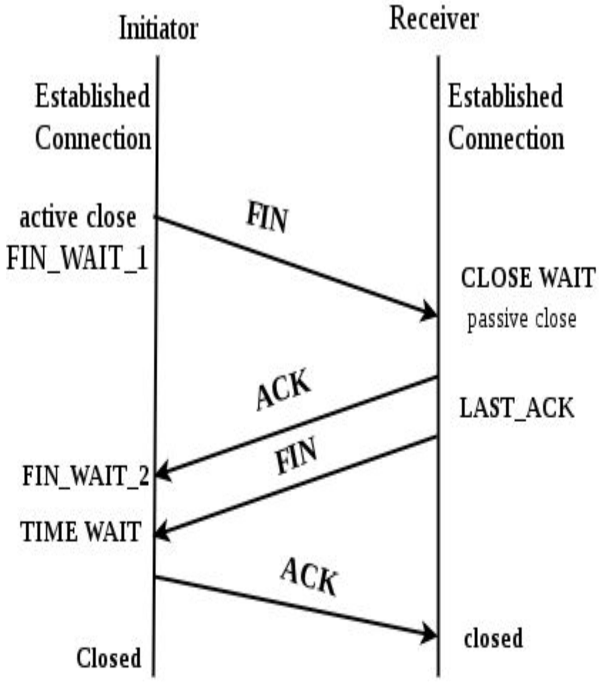
- **Step 1 (FIN From Client) –** Suppose that the client application decides it wants to close the connection. (Note that the server could also choose to close the connection).
  - This causes the client send a TCP segment with the **FIN** bit set to **1** to server and to enter the **FIN_WAIT_1** state. While in the **FIN_WAIT_1** state, the client waits for a TCP segment from the server with an acknowledgment (ACK).



TCP Connection Termination

Step 2 (ACK From Server) – When Server received FIN bit segment from Sender (Client), Server Immediately send acknowledgement (ACK) segment to the Sender (Client).

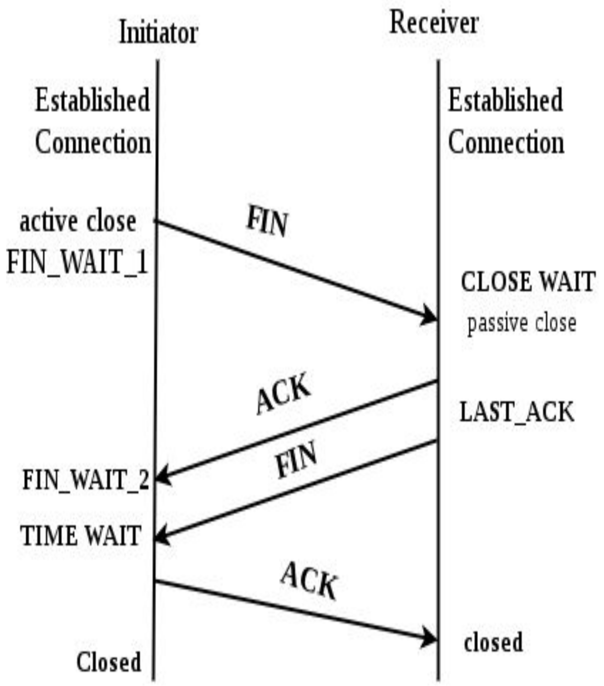
Step 3 (Client waiting) – While in the **FIN_WAIT_1** state, the client waits for a TCP segment from the server with an acknowledgment. When it receives this segment, the client enters the **FIN_WAIT_2** state. While in the **FIN_WAIT_2** state, the client waits for another segment from the server with the FIN bit set to 1.



TCP Connection Termination

Step 4 (FIN from Server) –

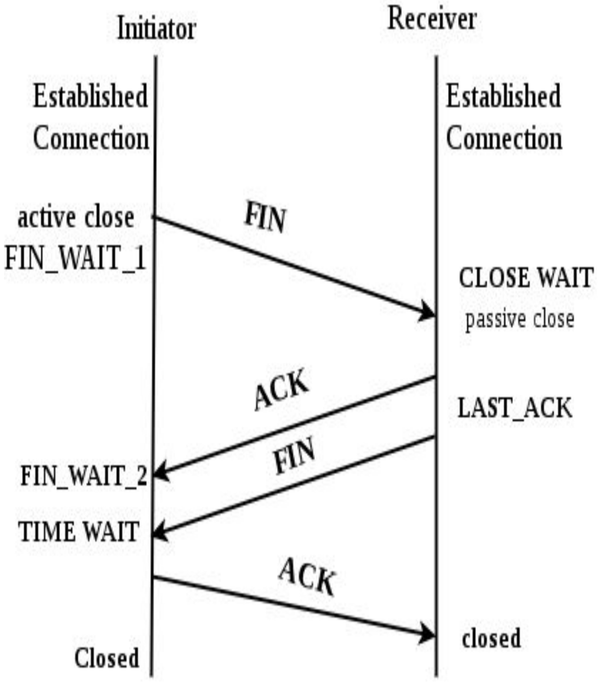
Server sends FIN bit segment to the Sender(Client) after some time when Server send the ACK segment (because of some closing process in the Server).



TCP Connection Termination

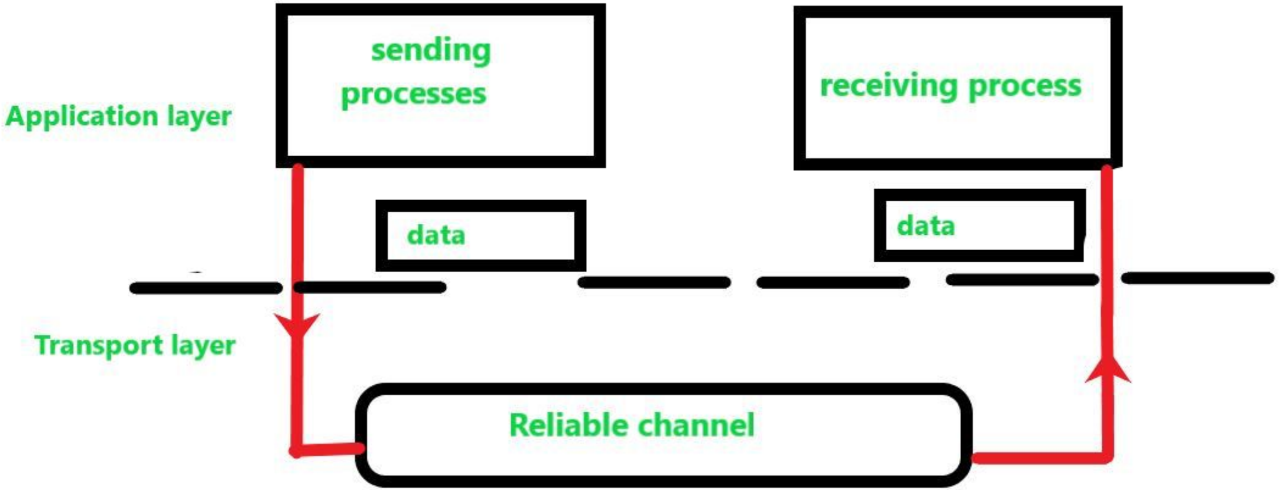
Step 5 (ACK from Client) –

- When Client receive FIN bit segment from the Server, the client acknowledges the server's segment and enters the **TIME_WAIT** state.
- The **TIME_WAIT**state lets the client resend the final acknowledgment in case the **ACK** is lost.
- The time spent by client in the **TIME_WAIT** state is depend on their implementation, but their typical values are 30 seconds, 1 minute, and 2 minutes.
- After the wait, the connection formally closes and all resources on the client side (including port numbers and buffer data) are released.



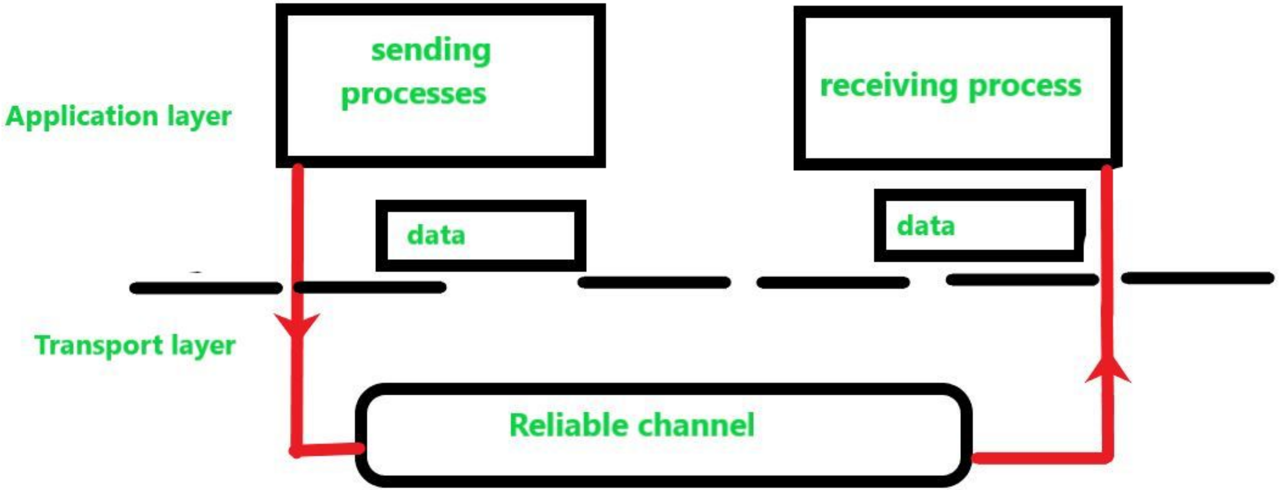
Reliability of Data Transfers in TCP

- TCP (Transmission Control Protocol) is a reliable data transfer protocol that is implemented on top of an unreliable layer, i.e., Internet Protocol (IP) is an end to end network layer protocol.



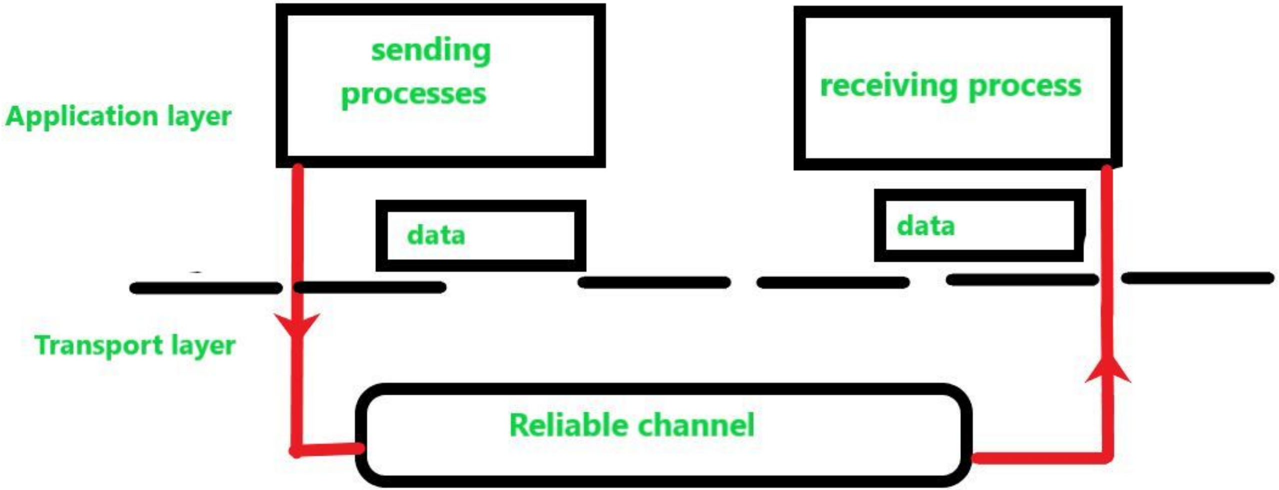
Reliability of Data Transfers in TCP

- In the reliable transfer of data the layer receives the data from the above layer breaks the message in the form of segment and put the header on each segment and transfer. Below layer receives the segments and remove the header from each segment and make it a packet by adding to header.

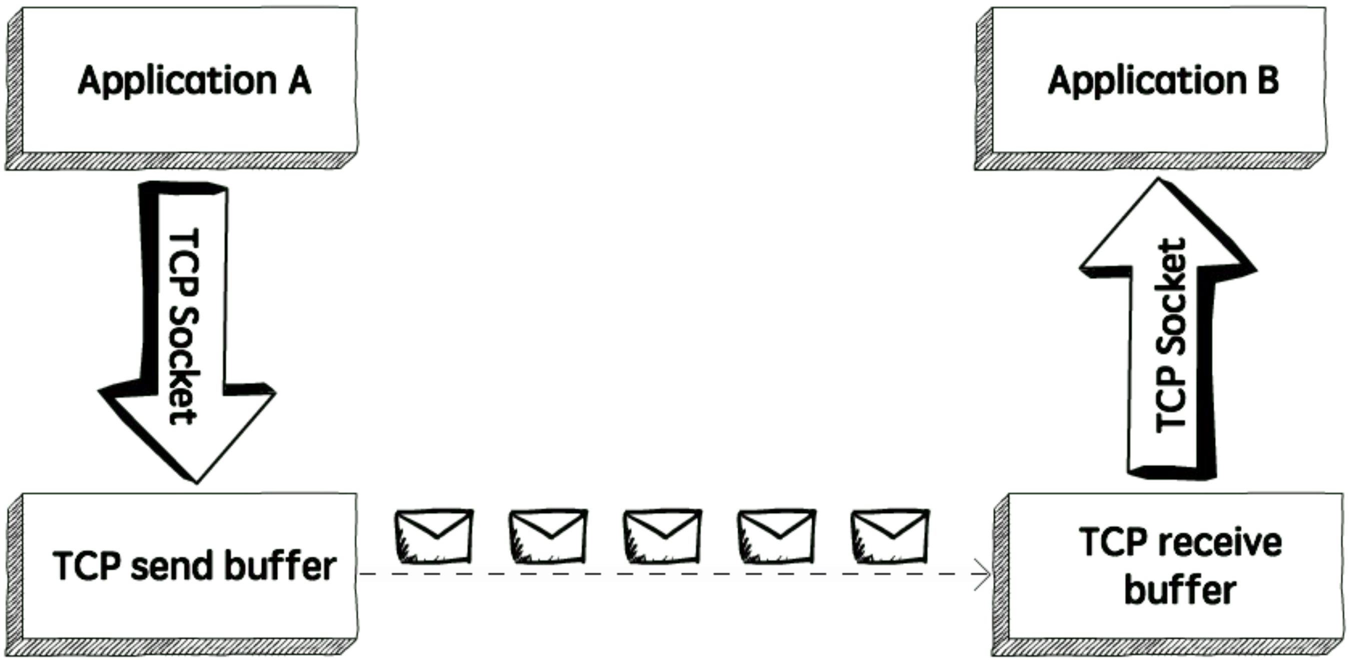


Reliability of Data Transfers in TCP

- The data which is transferred from the above has no transferred data bits corrupted or lost, and all are delivered in the same sequence in which they were sent to the below layer this is reliable data transfer protocol. This service model is offered by TCP to the Internet applications that invoke this transfer

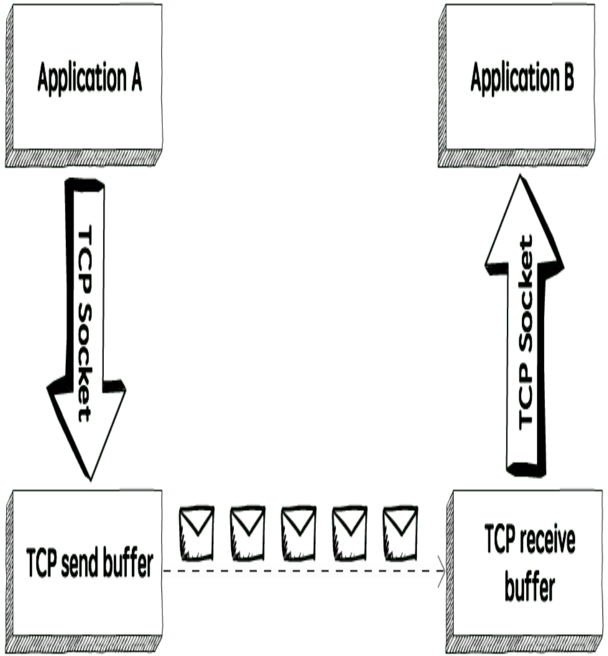


TCP Flow Control



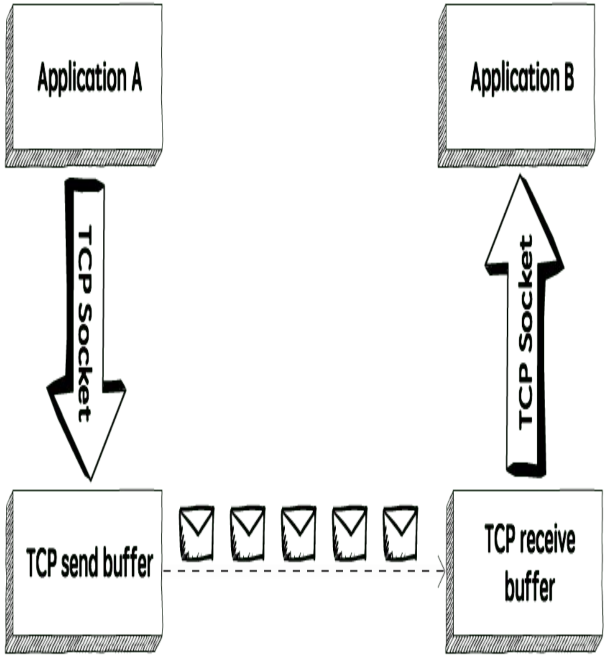
TCP Flow Control

- The sender application writes data to a socket, the transport layer (in our case, `TCP`) will wrap this data in a segment and hand it to the network layer (e.g. `IP`), that will somehow route this packet to the receiving node.



TCP Flow Control

- On the other side of this communication, the network layer will deliver this piece of data to `TCP`, that will make it available to the receiver application as an exact copy of the data sent, meaning if will not deliver packets out of order, and will wait for a retransmission in case it notices a gap in the byte stream.



TCP Congestion Control

- TCP uses a congestion window and a congestion policy that avoid congestion.
- Previously, we assumed that only receiver can dictate the sender’s window size. We ignored another entity here, the network.
- If the network cannot deliver the data as fast as it is created by the sender, it must tell the sender to slow down.
- In other words, in addition to the receiver, the network is a second entity that determines the size of the sender’s window.

TCP Congestion Control

Congestion policy in TCP –

1. Slow Start Phase: starts slowly increment is exponential to threshold
2. Congestion Avoidance Phase: After reaching the threshold increment is by 1
3. Congestion Detection Phase: Sender goes back to Slow start phase or Congestion avoidance phase.

Slow Start Phase

- **Slow Start Phase : exponential increment –** In this phase after every RTT the congestion window size increments exponentially.
  - `Initially cwnd = 1`
  - `After 1 RTT, cwnd = 2^(1) = 2`
  - `2 RTT, cwnd = 2^(2) = 4`
  - `3 RTT, cwnd = 2^(3) = 8`

Congestion Avoidance Phase

- **Congestion Avoidance Phase : additive increment –** This phase starts after the threshold value also denoted as *ssthresh*.
- The size of *cwnd*(congestion window) increases additive.
- After each RTT cwnd = cwnd + 1.
- `Initially cwnd = i`
- `After 1 RTT, cwnd = i+1`
- `2 RTT, cwnd = i+2`
- `3 RTT, cwnd = i+3`

Congestion Detection Phase

- **Congestion Detection Phase : multiplicative decrement –** If congestion occurs, the congestion window size is decreased.
- The only way a sender can guess that congestion has occurred is the need to retransmit a segment.
- Retransmission is needed to recover a missing packet which is assumed to have been dropped by a router due to congestion.

Congestion Detection Phase

- Retransmission can occur in one of two cases: when the RTO timer times out or when three duplicate ACKs are received.
- **Case 1 : Retransmission due to Timeout –** In this case congestion possibility is high.
  (a) **ssthresh is reduced to half of the current window size.**
  (b) set cwnd = 1
  (c) start with slow start phase again.

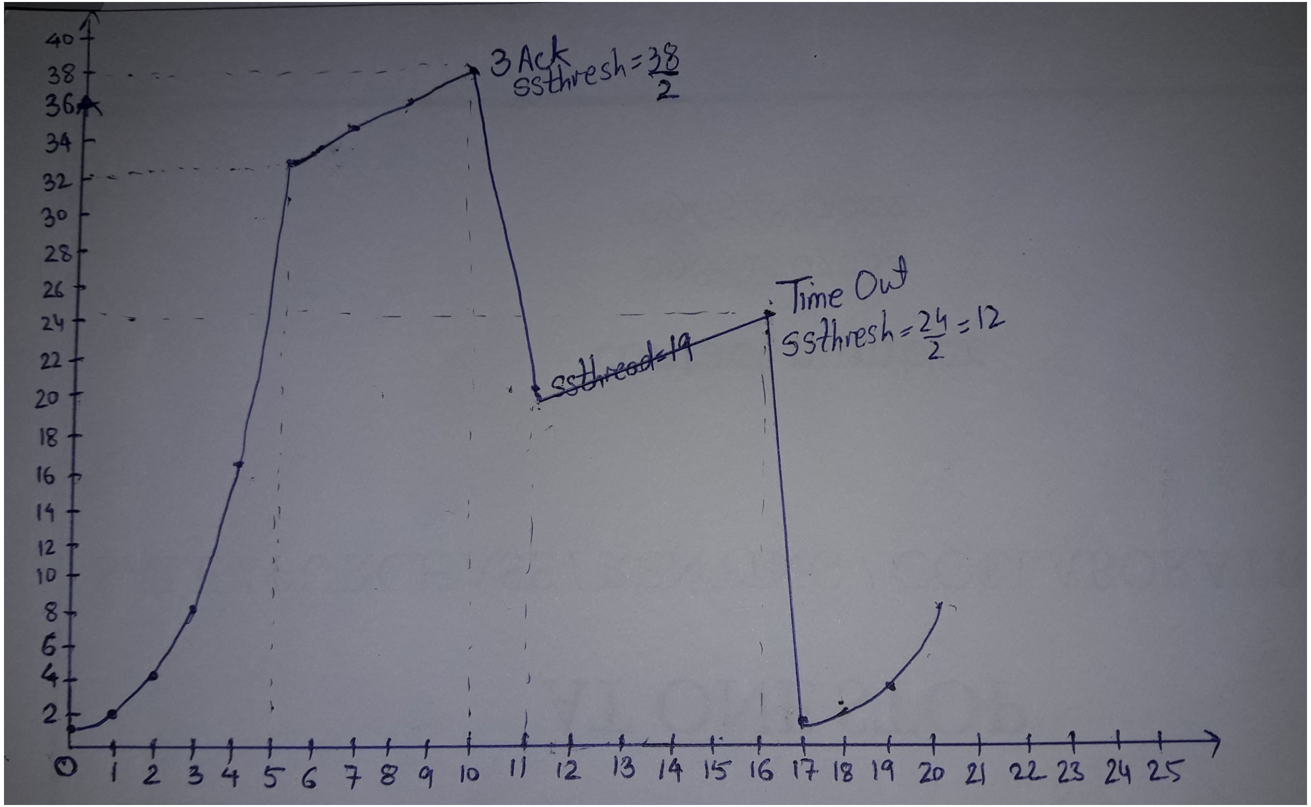
Congestion Detection Phase

- Retransmission can occur in one of two cases: when the RTO timer times out or when three duplicate ACKs are received.
- **Case 2 : Retransmission due to 3 Acknowledgement Duplicates –** In this case congestion possibility is less.
  (a) **ssthresh value reduces to half of the current window size.**
  (b) set cwnd= ssthresh
  (c) start with congestion avoidance phase

Example

Assume a TCP protocol experiencing the behavior of slow start. At 5th transmission round with a threshold (ssthresh) value of 32 goes into congestion avoidance phase and continues till 10th transmission. At 10th transmission round, 3 duplicate ACKs are received by the receiver and enter into additive increase mode. Timeout occurs at 16th transmission round. Plot the transmission round (time) vs congestion window size of TCP segments.

Example



3 Ack
ssthresh = 38/2
~~ssthresh=19~~
Time Out
ssthresh = 24/2 = 12
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25