CN [Day - 3]

UID: 24MCI10204

Name: Rahul Saxena

Branch: 24MCA – AI & ML

Question 1: A host H1 wants to send a large file (50 KB) to host H2 over a TCP connection. The connection is established using the standard three-way handshake. The maximum segment size (MSS) is 1 KB, the window size is 5 KB, and the RTT (Round Trip Time) is constant at 200 ms. The initial cwnd is 1 KB and the ssthresh is 16 KB. Assume that TCP Tahoe is used for congestion control.
Meanwhile, another application on H1 sends a 1 KB message to H2 over UDP without requiring reliability.
(a) Explain the steps of connection establishment using TCP three-way handshake.
(b) How many segments are sent in the first 1 second of the TCP connection assuming no losses?
(c) If a packet loss occurs at cwnd = 8 KB and is detected by timeout, what will be the new values of cwnd and ssthresh?
(d) Calculate the total transmission time for the 50 KB file assuming no losses and full utilization of window.
(e) Explain how UDP differs from TCP in the way it handles this transmission.
(f) What will happen if the receiver crashes during file transfer and comes back online after 1 minute? Explain with reference to TCP crash recovery.

Answer:

A) TCP Three-Way Handshake

TCP establishes a reliable connection using a three-step handshake:

1. **SYN (Synchronize):**
    - H1 sends a SYN packet to H2 to initiate the connection.
    - Includes Initial Sequence Number (ISN) from H1.
2. **SYN-ACK:**
    - H2 responds with a SYN + ACK.
    - Acknowledges H1's SYN and sends its own ISN.
3. **ACK:**
    - H1 sends an ACK confirming receipt of H2's SYN-ACK.

Connection established and data transfer begins.

B) How many segments are sent in the first 1 second? (Assuming no loss)

Given:

- **File size** = 50 KB
- **MSS** = 1 KB
- **RTT** = 200 ms
- **Initial cwnd** = 1 KB
- **ssthresh** = 16 KB

- **TCP Tahoe** (uses slow start until ssthresh)

Time	RTT Cycle	cwnd (KB)	Segments Sent
0 ms	0	1	1
200 ms	1	2	2
400 ms	2	4	4
600 ms	3	8	8
800 ms	4	16	16
1000 ms	5	32 (capped by file size)	N/A (beyond 1 sec)

Total Segments Sent in 1 second = 1 + 2 + 4 + 8 + 16 = **31 segments**

C) If a packet loss occurs at cwnd = 8 KB and is detected by timeout:

TCP Tahoe behavior:

- On **timeout**, it performs:
  - **ssthresh = cwnd / 2 = 8 / 2 = 4 KB**
  - **cwnd reset to 1 KB**

Answer:

- **New cwnd = 1 KB**
- **New ssthresh = 4 KB**

D) Total Transmission Time for 50 KB (Assuming No Loss)

Facts:

- Window size = 5 KB
- RTT = 200 ms
- MSS = 1 KB → So 5 segments can be sent per RTT.

Total Segments = 50 KB / 1 KB = 50 segments
→ In each RTT, 5 segments sent
→ 50 / 5 = 10 RTTs
Total time:

- 1 RTT = 200 ms
- 10 RTTs = 10 × 200 ms = 2000 ms = 2 seconds

Answer: Total transmission time = 2 seconds

E) How UDP differs from TCP for this transmission

Feature	TCP	UDP
Connection	Connection-oriented (3-way)	Connectionless
Reliability	Reliable, uses ACKs, retransmits	No guarantee of delivery
Congestion	Uses congestion control (Tahoe, Reno)	No congestion control
Overhead	Higher (due to headers, ACKs)	Lower overhead
Ordering	Maintains order	No ordering

F) If the receiver crashes and comes back after 1 minute

TCP behavior:

- **TCP uses timeouts** to detect failure.
- If **ACKs are not received**, the sender retransmits using exponential backoff.
- After **several timeouts**, the sender assumes **receiver is unreachable** and **resets connection** (typically after RTO * attempts).

After 1 minute:

- H2 (receiver) comes back, but:
  - TCP does **not recover old connection automatically**.
  - A new TCP connection must be **established from scratch** (new 3-way handshake).

Answer:

- TCP will **terminate the connection** after timeouts.
- On recovery, **new connection required**.
- TCP does **not maintain persistent connections** through crashes.

Question 2: A TCP sender is transmitting a large file over a network to a receiver. The Maximum Segment Size (MSS) is 1 KB, the initial value of cwnd (congestion window) is 1 KB, and the ssthresh (slow start threshold) is 16 KB. Assume that TCP Reno is used.
Below is the timeline of events (each line = 1 RTT):

RTT	Event
1	No loss; ACKs received
2	No loss; ACKs received
3	No loss; ACKs received
4	No loss; ACKs received
5	Segment lost; 3 duplicate ACKs received
6	Recovery begins; ACK received
7	No loss; ACKs received
8	No loss; ACKs received

Questions
(a) Plot the values of cwnd and ssthresh across these 8 RTTs.
(b) Describe how TCP Reno responds to the loss in RTT 5.
(c) At which RTT does the algorithm switch from Slow Start to Congestion Avoidance?
(d) If another loss had occurred in RTT 7 but was detected via timeout (not duplicate ACKs), how would Reno respond?
(e) What are the advantages of Fast Retransmit and Fast Recovery over timeout-based recovery?

Answer:

Plot of cwnd and ssthresh over 8 RTTs

TCP Reno behavior:

- Uses Slow Start until cwnd ≥ ssthresh
- Then enters Congestion Avoidance
- On 3 duplicate ACKs → uses Fast Retransmit + Fast Recovery
- On timeout → resets cwnd to 1 KB

RTT	cwnd	ssthresh	Phase
1	2	16	Slow Start
2	4	16	Slow Start
3	8	16	Slow Start
4	16	16	Slow Start (ends)
5	8	16	Fast Recovery starts
6	9	16	Congestion Avoidance
7	10	16	Congestion Avoidance
8	11	16	Congestion Avoidance

How TCP Reno responds to loss in RTT 5:

- When 3 duplicate ACKs are received:
  - TCP Reno performs Fast Retransmit (resend lost packet immediately)
  - Enters Fast Recovery
  - ssthresh = cwnd / 2 = 16 / 2 = 8 KB
  - cwnd = ssthresh = 8 KB
- After recovery is complete, it continues in Congestion Avoidance

No timeout happens → Reno avoids dropping cwnd to 1 KB.

When does it switch from Slow Start to Congestion Avoidance?

- Switch occurs after cwnd ≥ ssthresh
- RTT 4: cwnd becomes 16 KB = ssthresh
- Therefore, Congestion Avoidance starts from RTT 5, but due to loss, fast recovery happens instead.

Answer: RTT 5

If another loss occurs in RTT 7, detected by timeout (not dup ACKs):

If another loss occurs in RTT 7, detected by timeout (not dup ACKs):

- TCP Reno response to timeout:
  - ssthresh = cwnd / 2 = 10 / 2 = 5 KB
  - cwnd reset to 1 KB
  - Enters Slow Start again

Timeout is more aggressive than 3 dup ACKs.

Advantages of Fast Retransmit & Fast Recovery over timeout-based recovery:

Feature	Timeout-Based	Fast Retransmit & Recovery
Reaction Time	Slow (waits for timer)	Fast (immediate on 3 dup ACKs)
cwnd Reset	Full reset to 1 KB	Cuts cwnd to half
Throughput Impact	High	Lower
Efficiency	Lower	Higher (maintains better flow)
Preferred in Practice	No	Yes