EXERCISE

Q. let the size of congestion window of a TCP Connection in two cases when
Case1: timeout occur
case2: 3 ACK Recived
is 32KB. the RTT of a connection is 100m sec and MSS = 2KB. the time taken (m sec.) by TCP connection to get back to 32KB congestion Window is _____ and _____ respectively.

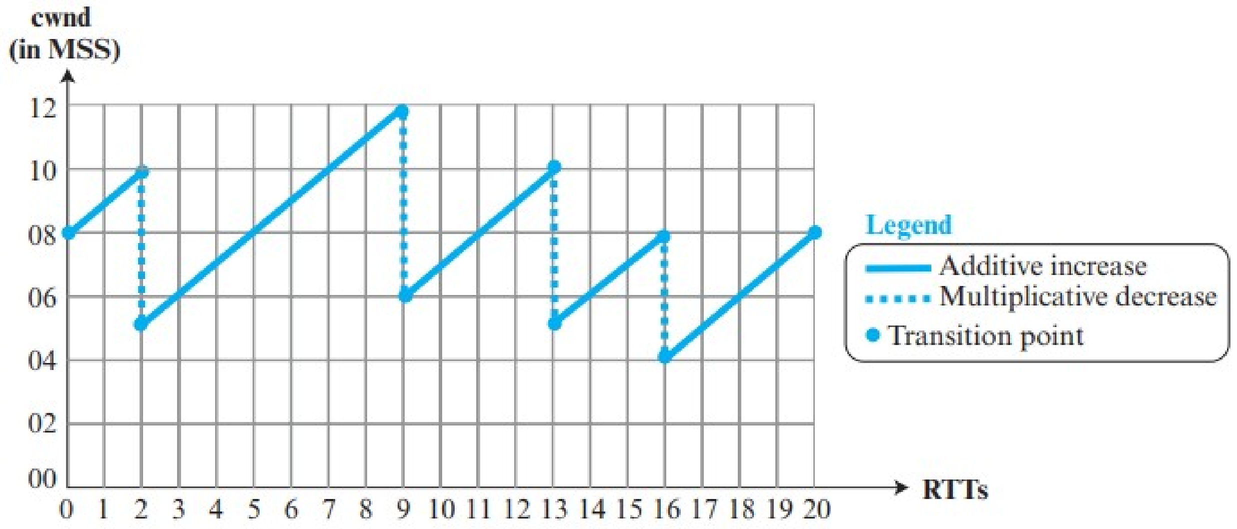
- Given:
- • Initial congestion window after event = ?
- • Target cwnd = 32 KB
- • RTT = 100 ms
- • MSS = 2 KB
- Case 1: Timeout occurs
- When timeout occurs, TCP resets cwnd to 1 MSS (2 KB) and enters Slow Start.
- Slow Start doubles cwnd every RTT until it reaches ssthresh or target:
- Target cwnd = 32 KB
- MSS = 2 KB → So 32 KB / 2 KB = 16 segments
- Growth: 1 → 2 → 4 → 8 → 16 (doubles each RTT)

- Number of RTTs = log2(16) = 4 RTTs
- Time = 4 × 100 ms = 400 ms

TCP THROUGHPUT

- The throughput for TCP, which is based on the congestion window behavior, can be easily found if the cwnd is a constant (flat line) function of RTT.
- The throughput with this unrealistic assumption is **throughput = cwnd / RTT**. In this assumption, TCP sends a cwnd bytes of data and receives acknowledgement for them in RTT time.
- But we know that, behaviour of TCP is like a saw teeth, therefore the **Throughput = 0.75 Wmax / RTT** where Wmax is the average of window sizes when the congestion occurs.

TCP Throughput Example



If MSS = 10KB and RTT = 100 ms then calculate the throughput for TCP ?

$W_{max} = (10 + 12 + 10 + 8 + 8) / 5 = 9.6$ MSS

Throughput $= (0.75\ W_{max} / \text{RTT}) = 0.75 \times 960$ kbps / 100 ms = 7.2 Mbps

TCP TIMERS

- To perform their operations smoothly, most TCP implementations use at least four timers:
  - Retransmission
  - Persistence
  - Keep-Alive
  - TIME-WAIT.

Retransmission Timer

- To retransmit lost segments, TCP employs one retransmission timer (for the whole connection period) that handles the retransmission time-out (RTO), the waiting time for an acknowledgment of a segment.
- We can define the following rules for the retransmission timer:
  - When TCP sends the segment in front of the sending queue, it starts the timer.
  - When the timer expires, TCP resends the first segment in front of the queue, and restarts the timer.
  - When a segment or segments are cumulatively acknowledged, the segment or segments are purged from the queue.
  - If the queue is empty, TCP stops the timer; otherwise, TCP restarts the timer.

PERSISTENT TIMER

- **Deadlock in TCP:**
  - If the receiving TCP announces a window size of zero, the sending TCP stops transmitting segments until the receiving TCP sends an ACK segment announcing a nonzero window size. This ACK segment can be lost.
  - Remember that ACK segments are not acknowledged nor retransmitted in TCP. If this acknowledgment is lost, the receiving TCP thinks that it has done its job and waits for the sending TCP to send more segments. There is no retransmission timer for a segment containing only an acknowledgment. The sending TCP has not received an acknowledgment and waits for the other TCP to send an acknowledgment advertising the size of the window. Both TCP's might continue to wait for each other forever (a deadlock).
- To correct this deadlock, TCP uses a persistence timer for each connection. When the sending TCP receives an acknowledgment with a window size of zero, it starts a persistence timer.
- When the persistence timer goes off, the sending TCP sends a special segment called a probe. This segment contains only 1 byte of new data. It has a sequence number, but its sequence number is never acknowledged; it is even ignored in calculating the sequence number for the rest of the data. The probe causes the receiving TCP to resend the acknowledgment.

KEEP-ALIVE TIMER

- A keepalive timer is used in some implementations to prevent a long idle connection between two TCP's.
- Suppose that a client opens a TCP connection to a server, transfers some data, and becomes silent. Perhaps the client has crashed. In this case, the connection remains open forever.
- To remedy this situation, most implementations equip a server with a keepalive timer. Each time the server hears from a client, it resets this timer. The time-out is usually 2 hours. If the server does not hear from the client after 2 hours, it sends a probe segment. If there is no response after 10 probes, each of which is 75 seconds apart, it assumes that the client is down and terminates the connection.

Time-Wait Timer

- The TIME-WAIT (2MSL) timer is used during connection termination. The maximum segment life time (MSL) is the amount of time any segment can exist in a network before being discarded.
- The implementation needs to choose a value for MSL. Common values are 30 seconds, 1 minute, or even 2 minutes.
- The 2MSL timer is used when TCP performs an active close and sends the final ACK. The connection must stay for 2 MSL amount of time to allow TCP resend the final ACK in case the ACK is lost. This requires that the RTO timer at the other end times out and new FIN and ACK segments are resent.