Problem 1

Suppose that a UDP receiver receives a packet. It then computes the Internet checksum for the received UDP segment, and finds that the result matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? Explain.

Write your solution to Problem 1 in this box

Problem 2

Consider the Go-Back-N protocol with a sender window size of 4 and a sequence number range of 0∼1023 (for simplicity, you do not need to consider the sequence number wrap-around). Suppose that at time t, the next in-order packet that the receiver is expecting has a sequence number of k. Assume that the medium does not reorder messages. Answer the following questions:

(a) List all possible sequence-number-sets (e.g. [k, k+1, k+2, k+3] could be a possible set) inside the sender's window at time t? Justify your answer.

(b) List all possible ACK numbers propagating back to the sender at time t? Justify your answer.

Write your solution to Problem 2 in this box

Problem 3

You are designing a reliable, sliding window, byte-stream protocol similar to TCP. It will be used for communication with a geosynchronous satellite network, for which the bandwidth is 1 Gbps ($1G = 10^9$) and the RTT is 200 ms. Assume the maximum segment lifetime is 10 seconds.

(a) What is the minimum number of bits you should use for ReceiveWindow and SequenceNum fields so that full utilization of the channel can be achieved? (Hints: The ReceiveWindow should be able to encode the maximum possible in-flight number of bytes. SequenceNum should be large enough so that the sequence number does not wrap around when the delayed segment is still in the channel.)

(b) If ReceiveWindow is 16 bits, what upper bound would that impose on the effective bandwidth?

Write your solution to Problem 3 in this box

Problem 4

In this question you are comparing among the following three schemes:

- Go-Back-N: cumulative ACK is used. The ACK number is the sequence number of corresponding segment (Figure 3.22 in the textbook).
- Selective Repeat: individual ACK is used. The ACK number is the sequence number of corresponding segment (Please use lecture slides' definition of Selective Repeat).
- TCP: cumulative ACK is used with no delayed ACK. The ACK number is the next expected sequence number.

Assume that timeout values for all three protocols are sufficiently long, such that 8 consecutive data segments and their corresponding ACKs can be received (if not lost in the channel) by the receiving host (Host B) and the sending host (Host A), respectively.

Suppose Host A sends 8 data segments to Host B, and the 5th segment (sent from A) is lost. No other segments are lost. In the end, all 8 data segments have been correctly received by Host B. The first segment starts from sequence number 1 and each segment contains 1 byte of data. The round trip time is greater than the transmission time for the 8 segments, so that the first ACK arrives after all 8 segments have been transmitted. For TCP, assume that the connection is already established and you don't need to count the connection establishment.

(a) How many segments has Host A sent in total and how many ACKs has Host B sent in total? What are their sequence numbers? Answer this question for all three protocols.

(b) If the timeout values for all three protocols are very long (e.g., longer than several RTTs), then which protocol successfully delivers all 8 data segments in shortest time interval?

Write your solution to Problem 4 in this box

Problem 5

Consider the evolution of a TCP connection with the following characteristics. Assume that all the following algorithms are implemented in TCP congestion control: slow start, congestions avoidance, fast retransmit and fast recovery, and retransmission upon timeout. If ssthresh equals to cwnd, use the slow start algorithm in your calculation.

- The TCP sender sends a number of data segments of 1 byte each.
- The TCP receiver follows cumulative ACK convention without delayed ACK, i.e. the receiver acknowledges every successfully received segment immediately.
- The RTT is 100 ms for all transmissions, consists of the network latency of 60 ms in sending a segment (header and payload) from the sender to the receiver and 40 ms in sending an acknowledgment (header only) from the receiver to the sender. Ignore packet-processing delays at the sender and the receiver.
- To simplify the setting, rather than estimating RTO from sampled RTT values, the RTO at the send is set to 500ms and does not change during the connection lifetime.
- Initially `ssthresh` at the sender is **set to 6**. Assume `cwnd` and `ssthresh` are measured in segments, and the transmission time for each segment is negligible.
- The connection (already established) starts to transmit data at time `t = 0`, and the **initial sequence number starts from 1**. TCP segment with **sequence number 6 is lost once** (i.e., it sees segment loss during its first transmission). No other segments are lost during transmissions.

What are the values for `cwnd` and `ssthresh` when the sender receives the TCP **ACK with number 15**? Show your intermediate steps or your diagram in your solution.

Note: In this question, we are taking the simplification that each TCP segment is exactly 1 byte, so that the integer parts of `cwnd` and `ssthresh` directly represent the number of segments. However, in practical TCP, `cwnd` and `ssthresh` are calculated in bytes, and the number of segments that can be sent is calculated by dividing `cwnd` by the MSS and rounding down to the nearest integer.

Problem 5 continued on next page...

Write your solution to Problem 5 in this box