Problem 1 (20 points).

1. The question refers to the Go-Back-N protocol. The sender’s code (from Lecture 6, slide 45) has been changed so that it is possible to send only one packet when a timeout happens. When there is a timeout, only the first packet (the packet that the send\_base pointer is pointing at) is resent. The code of the receiver has not been changed. Is the protocol correct after the change? If yes, justify and explain how it differs from the behavior of the original algorithm; if no, give an example of an incorrect behavior.

Answer:

The modified protocol is not correct after the change. The Go-Back-N protocol relies on the retransmission of all unacknowledged packets upon a timeout because the receiver only provides cumulative acknowledgments and does not buffer out-of-order packets.

By retransmitting only the first unacknowledged packet when a timeout occurs, the protocol breaks its synchronization mechanism. This creates scenarios where the receiver waits for packets in strict sequence, but the sender fails to retransmit subsequent lost packets, leading to deadlock.

For example, if the sender transmits packets 1, 2, 3, and 4, and packet 2 is lost, the receiver will send an acknowledgment for packet 1 and wait for packet 2. If the sender times out and retransmits only packet 2, the receiver will accept it and expect packet 3. However, if packet 3 is also lost or its acknowledgment is delayed, the sender will repeatedly retransmit only packet 2, leaving the receiver stuck waiting for packet 3, with no further progress.

2. Consider the sliding window protocol with a sender window size 15 and sequence number range from 1 to 532. Suppose that at time t, the last packet in the sender’s window has a sequence number 150. Assume that the medium does not reorder messages in both directions. What are the ACK messages with different sequence numbers that can be currently propagating back from the receiver to the sender at time t and what are their number assuming that:

1. The GO-BACK-N protocol is used.

2. The Selective Repeat protocol is used.

In the Go-Back-N protocol, at time t, only one unique ACK number can be propagating back from the receiver to the sender. This ACK corresponds to the highest in-order packet received by the receiver. For example, if the receiver has received all packets up to 145, it will repeatedly send an ACK for 145 until it receives packet 146. While the exact number of ACK messages in transit depends on the round-trip time and the rate of acknowledgment generation, all of these ACKs will carry the same sequence number, corresponding to the cumulative acknowledgment.

In the Selective Repeat protocol, multiple distinct ACK sequence numbers can propagate back to the sender simultaneously. Each correctly received packet generates an individual ACK, regardless of order. For instance, if packets 140, 141, 143, and 145 are successfully received by the receiver, ACKs for 140, 141, 143, and 145 will all be propagating at the same time. The number of distinct ACKs propagating depends on the number of packets successfully received within the sender’s window. In this case, up to 15 distinct ACKs can be propagating, as the sender window size is 15.

Problem 2 (20 points).

To get around the problem of sequence numbers wrapping around while old packets still exist, one could use 58-bit sequence numbers. However, theoretically, an optical fiber can run (i.e., transmit) at 7.5 Tbps (7.5 x 10^12 bits per second). Assume that each packet is of size 1500 bytes and has a lifetime after which it disappears if it does not arrive at its destination. Assume that each byte has its own sequence number, as TCP does. What is the biggest packet lifetime requested to ensure the networks do not have a wraparound problem? Notice that the optical fiber transmission rate mentioned above is in bits, not bytes. Try that the length of your answer will be no more than six lines.

The transmission rate of the optical fiber in bytes calculation:

Considering the packet size of 1500 bytes, the transmission rate in terms of packets per second is bytes per second.

Given there are unique sequence numbers as given, the total number of possible sequences relative to the packet size is

Therefore, the maximum packet lifetime is seconds.

Problem 3 (20 points)

Host A and host B are communicating over a TCP connection; the first segment sent from A to B has sequence number 1, and B has already received from A all (data) bytes up through byte 6000. Suppose A sends five new segments, S1, S2, S3, S4, and S5, to B back-to-back. S1 is sent first, S2 second, S3 third, S4 fourth, and S5 fifth. S1, S2, S3, S4, and S5 contain 100, 200, 300, 400, and 500 bytes of data, respectively. Assume that delayed-ack is not used (an ACK is sent as soon as a packet arrives).

1. If S2 arrive s at B before S4 and S5 and after S1 and S3, what is the acknowledgment number sent by B immediately after S2 arrives? Assume that the S3 arrives before S1 and S4 arrives before S5.

2. If S1 arrives at B after S2 and S3, and before S4 and S5, what is the acknowledgment number sent by B immediately after S1 arrives? Assume that the S3 arrives before S2 and S5 arrives before S4.

3. If S3 arrives at B before S4, and after S1, S2 and S5, what is the acknowledgment number sent by B immediately after S3 arrives? Assume that the S2 arrives before S1 and after S5.

4. If S4 arrives at B after S1, S2, and S3 and before S5, what is the acknowledgment number sent by B immediately after S4 arrives? Assume that S3 arrives before S2 and after S1.

5. If S5 arrives at B after S1, S2, S3, and S4, what is the acknowledgment number sent by B immediately after S4 arrives? Assume that the S2 arrives before S3 and after S1 and that S4 arrives before S3 and after S2.

For each of the questions above, give an answer assuming (1) buffering is not used on the receiver side and (2) buffering is used on the receiver side.

**S1:** 100 bytes (6001–6100)

**S2:** 200 bytes (6101–6300)

**S3:** 300 bytes (6301–6600)

**S4:** 400 bytes (6601–7000)

**S5:** 500 bytes (7001–7500)

**Buffering *is* not used on the receiver side**

Host B only accepts in-order segments. Out-of-order segments are discarded, and the acknowledgment number remains at the last contiguous byte received plus one.

**Buffering *is* used on the receiver side**

Host B can accept and buffer out-of-order segments. It acknowledges the highest contiguous byte received.

**1.**

Buffering is *not* used on the receiver side:

1. **S3** arrives first (Received Bytes: 6301–6600)

Out-of-order, discard S3, ACK Sent: 6001

1. **S1** arrives next (Received Bytes: 6001–6100)

In-order, Next Expected Byte: 6101, ACK Sent: 6101

1. **S2** arrives next (Received Bytes: 6001–6300)

In-order, Next Expected Byte: 6301, **ACK Sent: 6301**

Buffering is used on the receiver side:

1. **S3** arrives first (Received Bytes: 6301–6600)

Missing Bytes: 6001–6300

ACK Sent: 6001 (since bytes 6001–6300 are missing)

1. **S1** arrives next (Received Bytes: 6001–6100 and 6301–6600)

Missing Bytes: 6101–6300

ACK Sent: 6101 (since bytes 6101–6300 are still missing)

1. **S2** arrives after S1 (Received Bytes: 6001–6600)

Missing Bytes: None up to 6600

Next Expected Byte: 6601, **ACK Sent: 6601**

**2.**

Buffering is not used on the receiver side:

1. **S3** arrives first and then **S2**, both out of order, discard S3 and S2, ACK Sent: 6001
2. **S1** arrives next (Received Bytes: 6001–6100)

Missing Bytes: None up to 6300

Next Expected Byte: 6101, **ACK Sent:** **6101**

Buffering is used on the receiver side:

1. **S3** arrives first (Received Bytes: 6301–6600)

Missing Bytes: 6001–6300

ACK Sent: 6001 (since bytes 6001–6300 are missing)

1. **S2** arrives next (Received Bytes: 6101–6600)

Missing Bytes: 6001–6100

ACK Sent: 6001 (since bytes 6001–6100 are still missing)

1. **S1** arrives after S2 (Received Bytes: 6001–6600)

Missing Bytes: None up to 6300

Next Expected Byte: 6601, **ACK Sent:** **6601**

**3.**

Buffering is not used on the receiver side:

1. **S5** arrives first and then **S2**, both out of order, discard S5 and S2, ACK Sent: 6001
2. **S1** arrives next (Received Bytes: 6001–6100)

ACK Sent: 6101

1. **S3** arrives next, out of order, discard S3, **ACK Sent:** **6101**

Buffering is used on the receiver side:

1. **S5** arrives first (Received Bytes: 7001–7500)

Missing Bytes: 6001–7000

ACK Sent: 6001 (since bytes 6001–7000 are missing)

1. **S2** arrives next (Received Bytes: 6101–6300 and 7001–7500)

Missing Bytes: 6001–6100 and 6301–7000

ACK Sent: 6001 (since bytes 6001–6100 and 6301–7000 are missing)

1. **S1** arrives next (Received Bytes: 6001–6300 and 7001–7500)

Missing Bytes: 6301–7000

ACK Sent: 6301 (since bytes 6300–7000 are missing)

1. **S3** arrives next (Received Bytes: 6001–6600 and 7001–7500)

Missing Bytes: 6601–7000

Next Expected Byte: 6601, **ACK Sent:** **6601**

**4.**

Buffering is not used on the receiver side:

1. **S1** arrives first (Received Bytes: 6001–6100)

Missing Bytes: none up to 6100,

ACK Sent: 6101

1. **S3** arrives next, out of order, discard S3, ACK Sent: 6101
2. **S2** arrives next, in order, (Received Bytes: 6001–6300)

ACK Sent: 6301

1. **S4** arrives next ,out of order, discard S4, **ACK Sent:** **6301**

Buffering is used on the receiver side:

1. **S1** arrives first (Received Bytes: 6001–6100)

Missing Bytes: none up to 6100, ACK Sent: 6101

1. **S3** arrives next (Received Bytes: 6001–6100 and 6301–6600)

Missing Bytes: 6101–6300

ACK Sent: 6101 (since bytes 6101–6300 are missing)

1. **S2** arrives next (Received Bytes: 6001–6600)

Missing Bytes: none up to 6600

ACK Sent: 6601 (since bytes 6301–7000 are missing)

1. **S4** arrives next (Received Bytes: 6001–7000)

Missing Bytes: none up to 7000

Next Expected Byte: 7001, **ACK Sent: 7001**

**5.**

Buffering is not used on the receiver side:

1. **S1** arrives first (Received Bytes: 6001–6100)

Missing Bytes: none up to 6100

ACK Sent: 6101

1. **S2** arrives next (Received Bytes: 6001–6300)

In order, Missing Bytes: none up to 6300, ACK Sent: 6301

1. **S4** arrives next, out of order, discard S4, **ACK Sent: 6301**

Buffering is used on the receiver side:

1. **S1** arrives first (Received Bytes: 6001–6100)

Missing Bytes: none up to 6100, ACK Sent: 6101

1. **S2** arrives next (Received Bytes: 6001–6300)

Missing Bytes: none up to 6300, ACK Sent: 6301

1. **S4** arrives next (Received Bytes: 6001–6300 and 6601–7000)

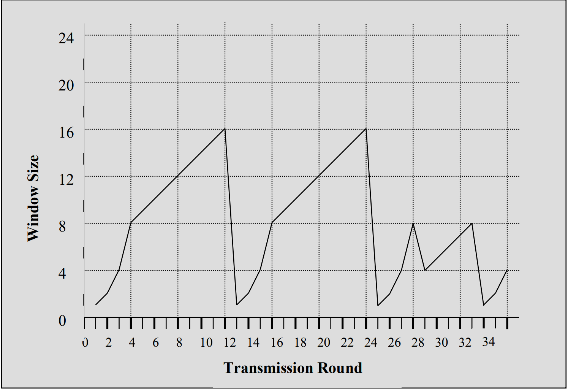
Missing Bytes: 6301–6600

**ACK Sent: 6301** (since bytes 6301–6600 are missing)

|  |  |  |
| --- | --- | --- |
| **#** | **Without buffering** | **With buffering** |
| 1 | 6301 | 6601 |
| 2 | 6101 | 6601 |
| 3 | 6101 | 6601 |
| 4 | 6301 | 7001 |
| 5 | 6301 | 6301 |

Problem 4

Problem 5 (20 points). Consider the following plot of TCP window size as a function of time that is, as a function of the number of rounds. (In each round, the sender transmits all the segments in its congestion window and either receives acknowledgments for them or there is a loss event.)



If TCP Reno is the protocol experiencing the above behavior, answer the following questions.

a.Identify all the intervals of time when TCP slow start is operating, and identify the intervals of time when TCP congestion avoidance is operating.

**Answer:**

Slow Start: [1,4], [13,16], [25, 28], [34,36]

Congestion Avoidance: [4,12], [16, 24], [29, 33]

Slow Start: Begins with a congestion window of 1 and grows exponentially.

Congestion Avoidance: Starts at a threshold and increases the window linearly until a timeout.

b. What are all the transmission rounds in which segment loss is detected by a triple duplicate ACK?

**Answer: 28-29**

In our lecture, we learned that TCP Reno exits the slow start phase after receiving three duplicate ACKs. Additionally, during the 28th transmission round, it encounters a timeout. Starting from the 29th round, TCP Reno resumes increasing the congestion window linearly instead of exponentially.

c. What are all the transmission rounds in which segment loss is detected by a timeout?

**Answer: 12, 24, 33**.

We observe that timeouts occur at the peaks of the graph, causing the congestion window (CongWin) dropping to 1.

d. What are all the values of the Threshold during the first 34 transmission rounds (starting from round 1)?

**Answer:**

the threshold is calculated by : congestion window / 2.

the threshold is 8 until the 12th round, then it remains 8 until the 24th round, and remains 8 until the 28th round. now he becomes 4 () and it will remain 4 in the 34th round.

e. During what transmission round is the 75th segment sent?

**Answer:** in the 10th Transmission Round

|  |  |  |
| --- | --- | --- |
| Round | Sent | total |
| 1 | 1 | 1 |
| 2 | 2 | 3 |
| 3 | 4 | 7 |
| 4 | 8 | 15 |
| 5 | 9 | 24 |
| 6 | 10 | 34 |
| 7 | 11 | 45 |
| 8 | 12 | 57 |
| 9 | 13 | 70 |
| 10 | 14 | 84 |