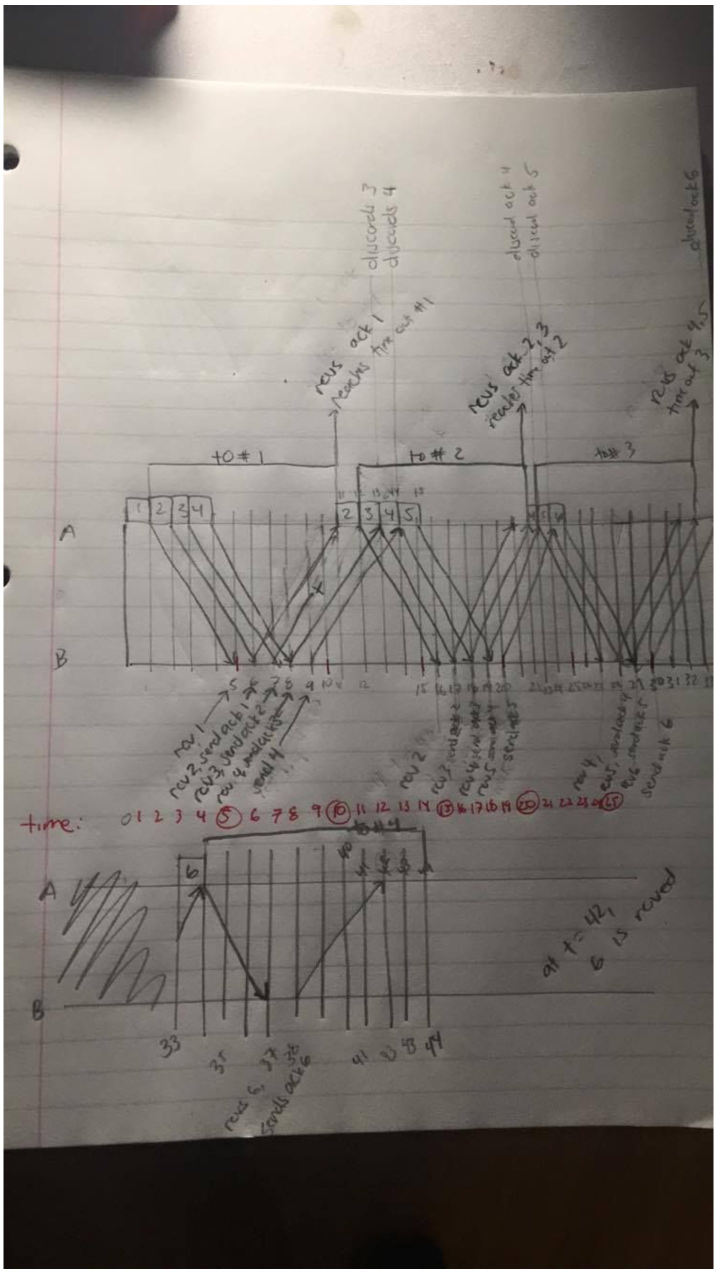
Corey Kipp (ID: 57723335)

Kevin Teer (ID: 27649116)

CS 132 Homework 3

1. TCP Reliability



Total time until all packets are sent and acknowledged: 42 ms

2. **Chapter 3, Problem 40**

    a. [1,6] and [23,26]

    b. [6,16] and [17,22]

    c. Triple Duplicate ACK

    d. Timeout

    e. 32

    f. 21

    g. 13

    h. 7

    i. Cwnd = 7, ssthresh = 4

    j. Cwnd = 4, ssthresh = 21

    k. 52 packets

Chapter 3, Problem 45

a. $\frac{W}{2} + (\frac{W}{2} + 1) + ... + W = \sum_{n=0}^{W/2}(\frac{W}{2} + n)$

$$= (\frac{W}{2} + 1)\ \frac{W}{2} + \sum_{n=0}^{W/2} n$$

$$= (\frac{W}{2} + 1)\ \frac{W}{2} + \frac{\frac{W}{2}(\frac{W}{2}+1)}{2}$$

$$= \frac{W^2}{4} + \frac{W}{2} + \frac{W^2}{8} + \frac{W}{4}$$

$$= \frac{3}{8}W^2 + \frac{3}{4}W$$

Therefore Loss Rate is $\frac{1}{\frac{3}{8}W^2+\frac{3}{4}W}$

b. $L \approx \frac{3}{8}W^2$ meaning that $W = \sqrt{\frac{8}{3L}}$

Average rate = $\frac{0.75\,(W)\,(MSS)}{RTT}$

$$= \frac{3}{4}\sqrt{\frac{8}{3L}} \cdot \frac{MSS}{RTT}$$

$$= \frac{3}{4} \cdot \sqrt{\frac{8}{3}} \cdot \frac{MSS}{RTT \cdot \sqrt{L}}$$

$$= \frac{1.22 \cdot MSS}{RTT \cdot \sqrt{L}}$$

3. Wireshark Lab #2

1) Client Computer (Source):

    a) IP ADDRESS: 10.39.7.12

    b) PORT: 55720

2) Destination (gaia.cs.umass.edu):

    a) IP ADDRESS: 128.119.245.12

    b) PORT: 80

3) 216.58.217.13 PORT: 443

4) Sequence number 0 is sequence number of the TCP SYN segment used to initiate the TCP connection between the client computer and gaia.cs.umass.edu. The SYN flag is set to 1 and it indicates the segment to be a SYN segment.

5) a) The sequence number of the SYNACK segment sent by gaia.cs.umass.edu to client computer in reply to the SYN is 1 (relative ack number).

b) The value of the ACKnowledgement is 1.

c) gaia.cs.umass.edu determined that value by adding 1 to the sequence number of the previous segment.

d) This segment is identified by a synack segment ack and syn bits are both set.

6) The sequence number of the TCP segment containing the HTTP Post command is 1.

7)

segment	Relative Segment number	Segment number	Time sent	Acknowledgement Received	RTT	Estimated RTT
1	1	0dd601f	.026	.054	.028	.028
2	566	0dd6042	.042	.077	.035	.035
3	2026	0dd609d	.054	.124	.070	.70
4	3486	0dd60f9	.055	.169	.114	.114
5	4946	0dd60f9	.077	.217	.140	.140
6	6406	0dd61af	.078	.268	.190	.190

Estimated RTT of packet 1: .875 * .028 + .125 * .28 = .028

Estimated RTT of packet 2: .875 * .042 + .125 * .035 = .035

Estimated RTT of packet 3: .875 * .054 + .125 * .070 = .070

Estimated RTT of packet 4: .875 * .055 + .125 * .114 = .114

Estimated RTT of packet 5: .875 * .077 + .125 * .140 = .140

Estimated RTT of packet 6: .875 + .078 + .125 * .190 = .190

8)

Segment 1: 566 - 1 = 565 bytes

Segment 2: 2026 - 566 = 1460 bytes

Segment 3: 3486 - 2026 = 1460 bytes

Segment 4: 4946 - 3486 = 1460 bytes

Segment 5: 6406 - 4946 = 1460 bytes

Segment 6: 1460 bytes

9) The minimum amount of buffer space at gaia.cs.umass.edu for the entire trace is 5840 bytes, which is highlighted in the first acknowledgement from the server trace. The sender never throttled as a result of the lack of receiver buffer space.

10) There are no retransmitted segments. We looked for repeating segment numbers; but did not find any.

11) The receiver typically acks 1460 bytes; however there are cases where the receiver is acking every other received segment (data is doubled)... 2920 = 1460*2

12) The throughput is calculated is the total size divided by the total time; here the total size is the difference between the sequence number of the first TCP segment and last ACK, therefore: 164091 - 1 = 164090 bytes; and the whole transmission time s the difference of the first instance of the first TCP segment and the last ACK, total time is: 5.455830 - .026477 = 5.494 seconds. The throughput for the TCP connection is 164090/5.494 = 30.22 KB/s

13) TCP slow start begins at the start of the connection, when the HTTP POST segment is sent; it begins at time zero and ends about .15 seconds according to the graph. We cannot obtain the value of the congestion window size from the Time-Sequence-Graph (Stevens). It depends on the value of the congestion window of this TCP sender. The measured data is using a fraction of the window size instead of the idealized .33 to .50. Idealized behavior of TCP assumes senders are aggressive, traffic can congest the network; therefore TCP senders should follow AIMD alg so their sending window size should decrease when they detect network congestion. TCP behaviors largely depend on the application; like in this example, the TCP sender can send out data, there isn't data available for transmission. In a web app, some web objs are very small in size. Before the end of slow start phase, the transmission has ended... hence the transmission of these small objects unnecessarily suffer longer delays because of the slow start of TCP.

14) TCP slow start begins at the start of the connection, when the HTTP POST segment is sent; it begins at time zero and ends about .15 seconds according to the graph. Congestion avoidance takes over at about .19 seconds because it is cutting down the amount being sent.