Lab 3 Report

Isaac Hartung and Sam Warnick

March 21, 2017

1 Purpose

The purpose of this lab was to add congestion control to our simulator. We used TCP Tahoe as the basis for our congestion control. We implemented TCP Tahoe with the following rules and features found in the lab specifications:

- **Slow start**: At the start of the connection, or after any kind of loss event, set cwnd (congestion window) to 1 MSS. Every time the sender receives an ACK for new data, increment cwnd by the number of new bytes of data acknowledged. Never increment cwnd by more than one MSS.
- **Threshold**: Stop slow start when cwnd exceeds or equals the threshold. Start with a threshold of 100,000 bytes.
- **Additive Increase**: Once cwnd is larger than the threshold, use additive increase. Every time the sender receives an ACK for new data, increment cwnd by MSS*b/cwnd, where MSS is the maximum segment size (1000 bytes) and b is the number of new bytes acknowledged.
- **Fast Retransmit**: A loss event is detected when there are three duplicate ACKs (meaning the fourth ACK in a row for the same sequence number), and TCP immediately retransmits instead of waiting for the retransmission timer.
- When a loss event is detected (a timeout or 3 duplicate ACKs), then set the threshold to max(cwnd/2,MSS) and set cwnd to 1 MSS.
- Ensure that cwnd is always a multiple of MSS. This means that for the Additive Increase rule above, rather than directly incrementing cwnd by MSS*b/cwnd, keep a separate variable, increment, that stores the increment each time. Once increment exceeds MSS, add one MSS to cwnd and decrease increment by one MSS. Reset the increment to zero when a loss event occurs.
- Likewise, when setting the threshold to half the previous cwnd, subtract off any bytes that are in excess of a multiple of MSS.

We ran several tests to ensure that our implementation was working as expected. The tests are described below.

2 Tests

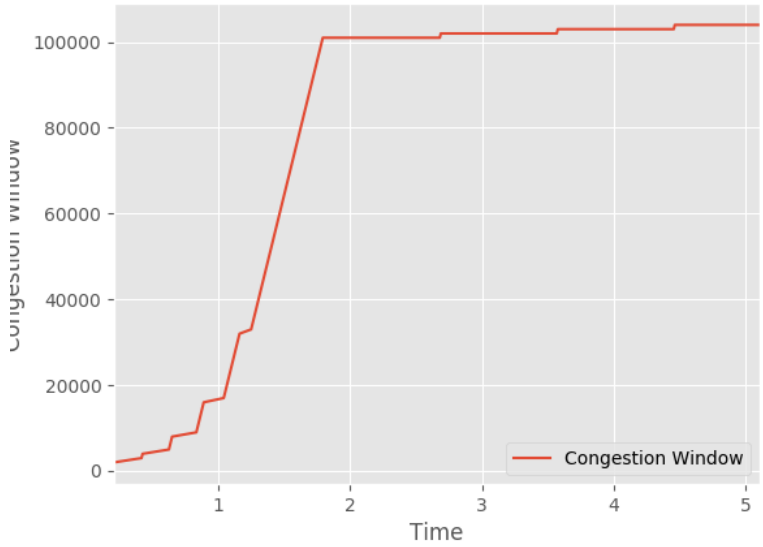
All tests use the same following network configuration:

```
#
# n1 — n2
#
n1 n2
n2 n1
```

Each link is 1 Mbps with 100 ms propagation delay.

2.1 Slow Start

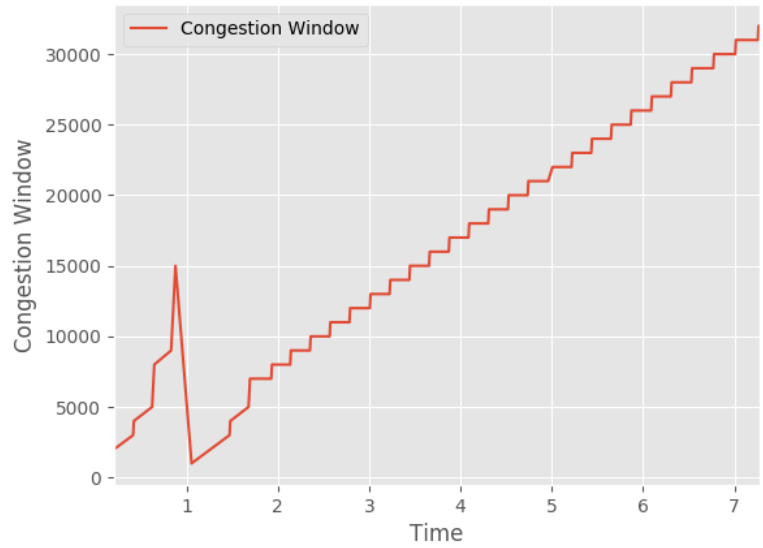
For this test we transferred the file with no packets dropped. Because the threshold is 100,000, we expected the window size to grow very rapidly.



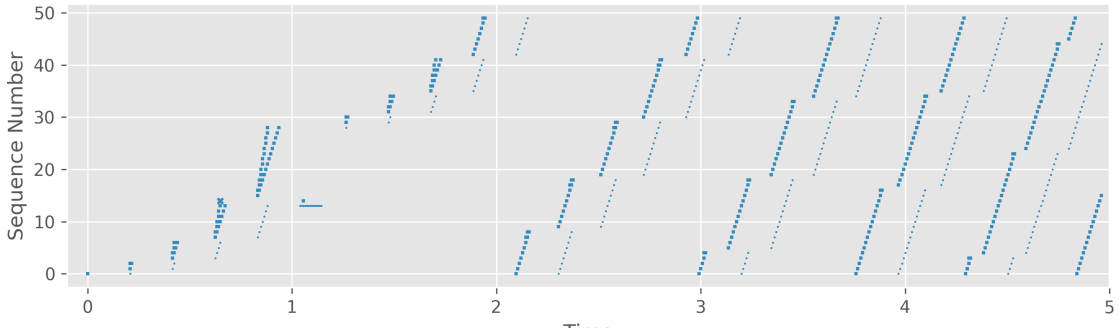
From this graph, we can see that slow start is working as we would expect. cwnd grows very quickly at the beginning. As soon as cwnd passes the threshold of 100000, additive increase begins, as expected.

2.2 One packet loss

For this test we transferred the file and dropped sequence number 14000.



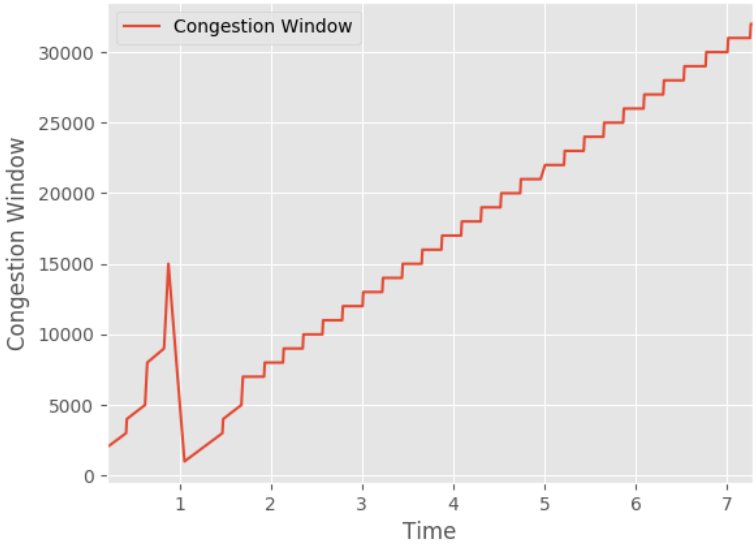
cwnd behaves as expected. The first drop happens before the initial threshold is reached, so it is slow start until the first drop. After the drop, cwnd is set to 1000 and slow start occurs again. The threshold is hit at 7000 and additive increase begins.



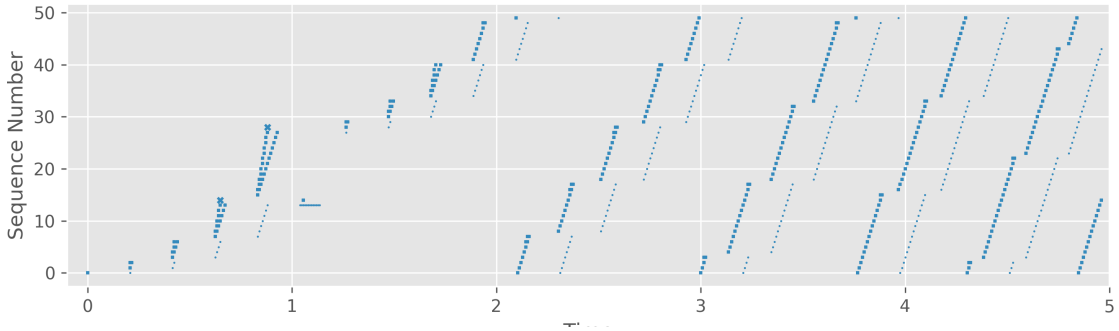
This graph looks as expected, compared with the "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP" paper. We see that the packet is dropped, there are multiple ACKs received for the dropped packet, and cwnd is clearly reset to 1000 and increases with a slow start until the threshold.

2.3 Two packet loss

For this test, we again transferred the file and dropped sequence number 14000 in addition to sequence number 28000.



Again, cwnd behaves as expected. The first drop happens before the initial threshold is reached, so it is slow start until the first drop. After the drop, cwnd is set to 1000 and slow start occurs again. The threshold is hit at 7000 and additive increase begins.

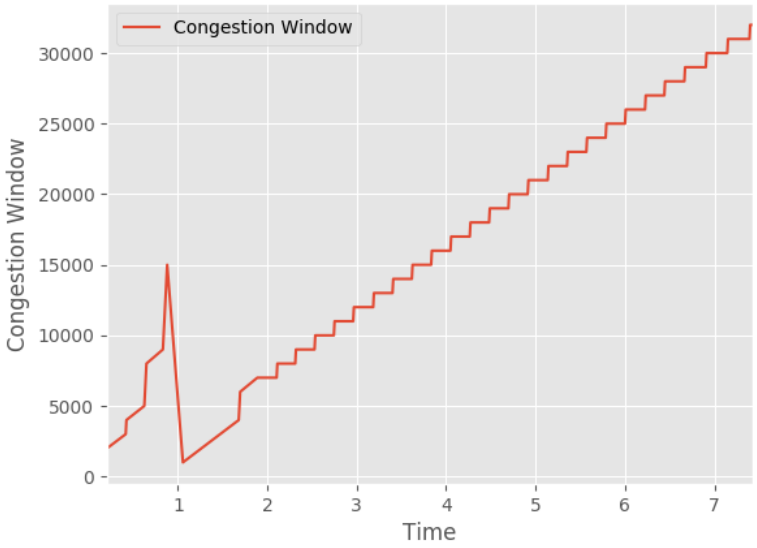


This graph looks as expected, compared with the "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP" paper. The second packet is dropped before a triple ACK is received for the first dropped

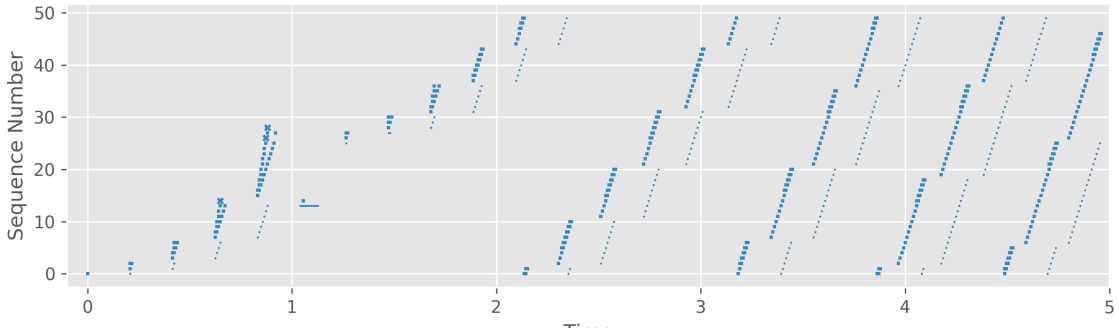
packet and slow start begins again.

2.4 Three packet loss

For this test, we transferred the file and dropped sequence numbers 14000, 26000, and 28000.



Again, cwnd behaves as expected.



This graph looks as expected, compared with the "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP" paper. After the first dropped packet is retransmitted, the second dropped packet is retransmitted with the beginning of slow start.