Lab 2 Report

Isaac Hartung and Sam Warnick

February 24, 2010

1 Basic Tests

Both tests use the same following network configuration:

```
#
# n1 — n2
#
n1 n2
n2 n1
```

Each link is 10 Mbps with 10 ms propagation delay. We ran a diff on the original file and the output file to ensure that they were transmitted correctly.

1.1 test.txt

For test.txt we used a window size of 3000 bytes. The file was then transferred with different four loss rates: 0%, 10%, 20%, and 50%.

output 0%

0.064 n1 (1) sending TCP segment to 2 for 10000
0.0732 n2 (2) received TCP segment from 1 for 9000
0.0732 application got 1000 bytes
0.0732 Sending this ack from handle data: 10000
0.0732 n2 (2) sending TCP ACK to 1 for 10000
0.0832 n1 (1) sending TCP segment to 2 for 10000
1.0832 n1 (1) retransmission timer fired

output 10%

0.074 n2 (2) received TCP segment from 1 for 9000
0.074 application got 1000 bytes
0.074 Sending this ack from handle data: 10000
0.074 n2 (2) sending TCP ACK to 1 for 10000
0.084 n1 (1) sending TCP segment to 2 for 10000
1.084 n1 (1) retransmission timer fired

output 20%

1.064 n1 (1) sending TCP segment to 2 for 8000
1.0748 n2 (2) received TCP segment from 1 for 8000

1.0748 application got 2000 bytes
1.0748 Sending this ack from handle data: 10000
1.0748 n2 (2) sending TCP ACK to 1 for 10000
1.0848 n1 (1) sending TCP segment to 2 for 10000
2.0848 n1 (1) retransmission timer fired

output 50%

4.0832 n1 (1) sending TCP segment to 2 for 7000
4.094 n2 (2) received TCP segment from 1 for 7000
4.094 application got 3000 bytes
4.094 Sending this ack from handle data: 10000
4.094 n2 (2) sending TCP ACK to 1 for 10000
4.104 n1 (1) sending TCP segment to 2 for 10000
5.104 n1 (1) retransmission timer fired

Loss (%)	time (s)
0	1.0832
10	1.084
20	2.0848
50	5.104

As can be seen from the data, as the data loss increases, the amount of time it takes to transmit the entire file increases. The difference between 0% and 10% data loss is pretty insignificant. But as the loss increases, the amount of time to send the file increases significantly (an entire second in the case of 20%). Obviously, the more retransmits required to transmit a file, the longer it will take.

1.2 internet_architecture.pdf

For internet_architecture.pdf we used a window size of 10000 bytes. The file was then transferred with two different loss rates: 0% and 50%.

output 0%

1.074416 Sending this ack from handle data: 514520
1.074416 n2 (2) sending TCP ACK to 1 for 514520
1.0816 n1 (1) sending TCP segment to 2 for 514520
1.0824 n1 (1) sending TCP segment to 2 for 514520
1.0832 n1 (1) sending TCP segment to 2 for 514520
1.084 n1 (1) sending TCP segment to 2 for 514520
1.084416 n1 (1) sending TCP segment to 2 for 514520
2.084416 n1 (1) retransmission timer fired

output 50%

30.602416 Sending this ack from handle data: 514520
30.602416 n2 (2) sending TCP ACK to 1 for 514520
30.608 n1 (1) sending TCP segment to 2 for 514520
30.6088 n1 (1) sending TCP segment to 2 for 514520
30.6096 n1 (1) sending TCP segment to 2 for 514520

30.6104 n1 (1) sending TCP segment to 2 for 514520
30.6112 n1 (1) sending TCP segment to 2 for 514520
30.612 n1 (1) sending TCP segment to 2 for 514520
30.612416 n1 (1) sending TCP segment to 2 for 514520
31.612416 n1 (1) retransmission timer fired

Loss (%)	time (s)
0	2.084416
50	31.612416

Just as the other tests show, significant data loss leads to increased time to transmit. This is due to the number of retransmits required to transmit the entire file.

2 Fast Retransmit

Again, this test uses the same following network configuration:

#
n1 — n2
#
n1 n2
n2 n1

Each link is 10 Mbps with 10 ms propagation delay.

We used a window size of 10000 bytes. The file was then transferred with two different loss rates: 0% and 20%, both with and without fast retransmit.

2.1 With fast retransmit

output 0 with FR%
1.0816 n1 (1) sending TCP segment to 2 for 514520
1.0824 n1 (1) sending TCP segment to 2 for 514520
1.0832 n1 (1) sending TCP segment to 2 for 514520
1.084 n1 (1) sending TCP segment to 2 for 514520
1.084416 n1 (1) sending TCP segment to 2 for 514520
2.084416 n1 (1) retransmission timer fired

output 20 with FR%
61.0472 n1 (1) sending TCP segment to 2 for 514520
61.048 n1 (1) sending TCP segment to 2 for 514520
62.048 n1 (1) retransmission timer fired
62.048 n1 (1) sending TCP segment to 2 for 514000
62.058416 n2 (2) received TCP segment from 1 for 514000
62.058416 n2 (2) sending TCP ACK to 1 for 514520
62.068416 n1 (1) sending TCP segment to 2 for 514520
63.068416 n1 (1) retransmission timer fired

Loss (%)	time (s)
0	2.0844616
20	63.068416

2.2 Without fast retransmit

output 0%
1.074416 Sending this ack from handle data: 514520
1.074416 n2 (2) sending TCP ACK to 1 for 514520
1.0816 n1 (1) sending TCP segment to 2 for 514520
1.0824 n1 (1) sending TCP segment to 2 for 514520
1.0832 n1 (1) sending TCP segment to 2 for 514520
1.084 n1 (1) sending TCP segment to 2 for 514520
1.084416 n1 (1) sending TCP segment to 2 for 514520
2.084416 n1 (1) retransmission timer fired

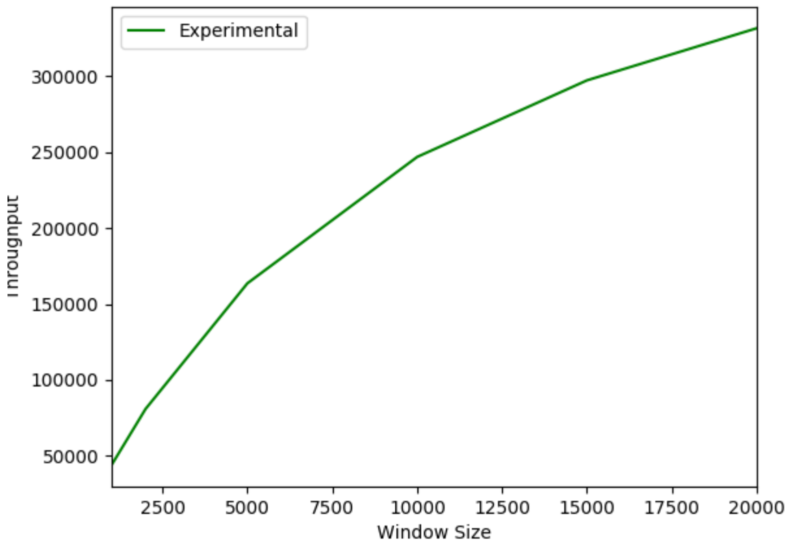
output 20%
75.099216 n2 (2) received TCP segment from 1 for 514000
75.099216 n2 (2) sending TCP ACK to 1 for 514520
75.1056 n1 (1) sending TCP segment to 2 for 514520
75.1088 n1 (1) sending TCP segment to 2 for 514520
75.109216 n1 (1) sending TCP segment to 2 for 514520
76.109216 n1 (1) retransmission timer fired

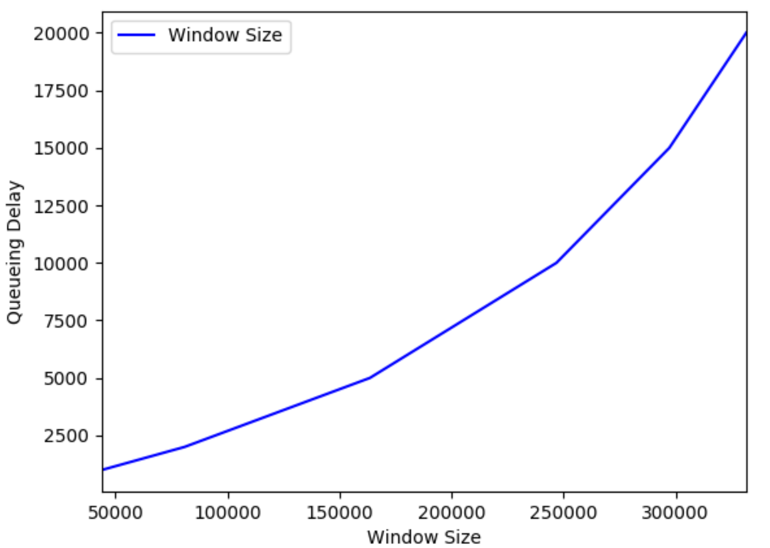
Loss (%)	time (s)
0	2.084416
20	76.10916

With 0%, there is no difference between using fast retransmit and not using fast retransmit. this is expected, since there is no need to retransmit. When loss is increased to 20%, we found that fast retransmit does in fact allow the file to be transmitted more quickly. It was transmitted 13.04 seconds more quickly. This is a significant difference.

3 Experiments

For the experiments dealing with window size, we used the same network as the earlier tests, with a queue size of 100 and a loss rate of 0%. We transmitted internet_architecture.pdf using window sizes of 1000, 2000, 5000, 10000, 15000, and 20000 bytes. We then divided the file size (514,520 bytes) by the time it took to transmit to determine the throughput. So, in this case, fast transmit was a benefit.





As the first graph shows, as the window size increases, the throughput does as well. But it can be seen that the rate at which the throughput increases slows. This is related to the queueing delay caused by increasing the window size. With the near infinite arrival rate, it is hard to calculate average queue delay using the traditional formulae. But we are able to calculate queueing delay relative to the fastest overall transfer time. We accomplish this by subtracting the fastest transition time from all of the others. The difference that results is attributable to queueing delay because the other delays remain constant, and there are no network related delays because loss is set to 0. We then divide the resulting time by the number of segments sent and this gives us a measure of average queueing delay relative to our fastest time.