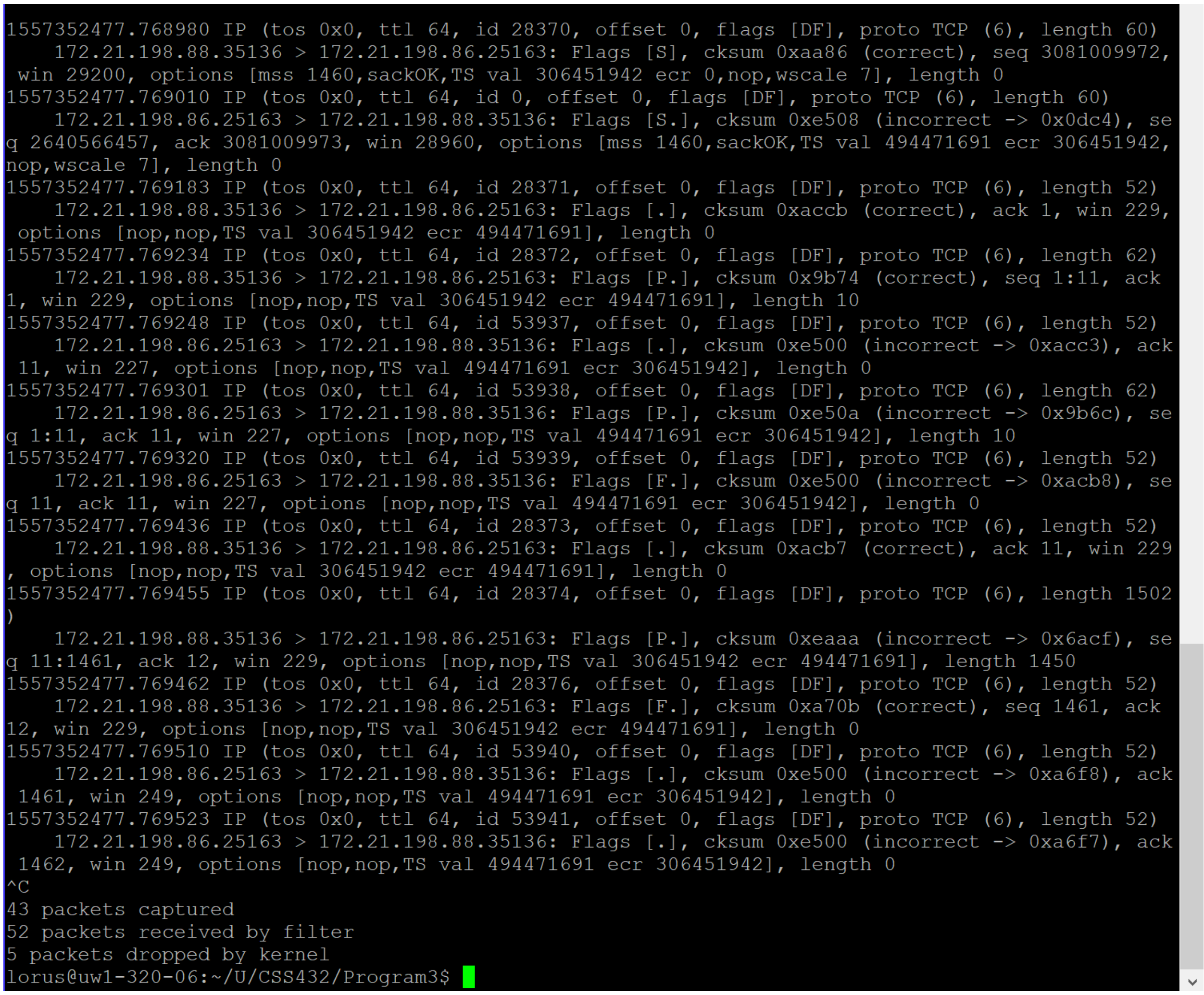
Jarod Guerrero
CSS 432
5/9/19

NET P3

Test 1: Running tcpdump and hw3

Output:

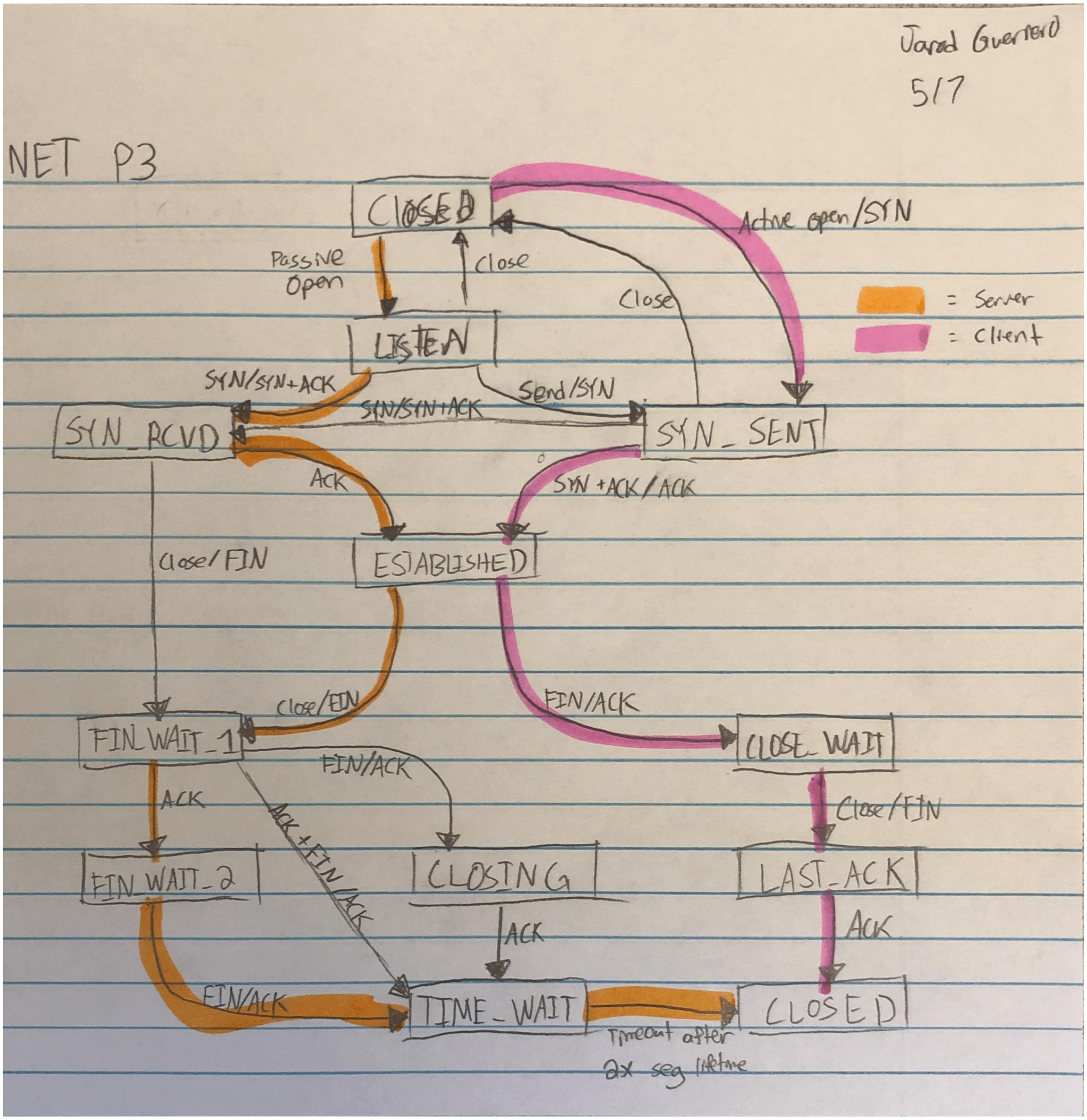
Ignore the packets captured/received by filter. I had to run hw3 multiple times to get the proper output Proper amount should be 12 captured, 12 received, and 0 dropped by kernel.



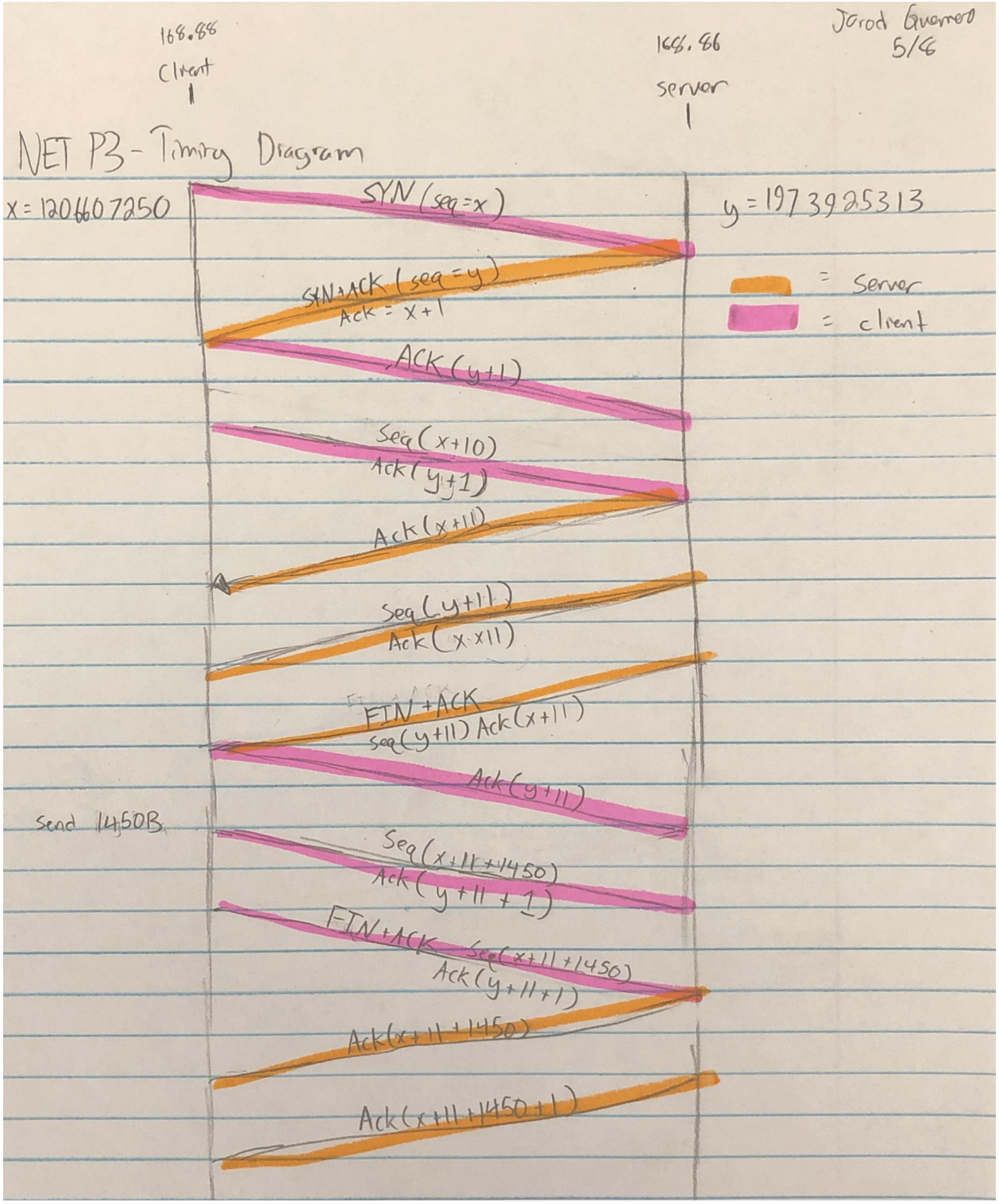
```
1557352477.768980 IP (tos 0x0, ttl 64, id 28370, offset 0, flags [DF], proto TCP (6), length 60)
    172.21.198.88.35136 > 172.21.198.86.25163: Flags [S], cksum 0xaa86 (correct), seq 3081009972,
 win 29200, options [mss 1460,sackOK,TS val 306451942 ecr 0,nop,wscale 7], length 0
1557352477.769010 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    172.21.198.86.25163 > 172.21.198.88.35136: Flags [S.], cksum 0xe508 (incorrect -> 0x0dc4), se
q 2640566457, ack 3081009973, win 28960, options [mss 1460,sackOK,TS val 494471691 ecr 306451942,
nop,wscale 7], length 0
1557352477.769183 IP (tos 0x0, ttl 64, id 28371, offset 0, flags [DF], proto TCP (6), length 52)
    172.21.198.88.35136 > 172.21.198.86.25163: Flags [.], cksum 0xaccb (correct), ack 1, win 229,
 options [nop,nop,TS val 306451942 ecr 494471691], length 0
1557352477.769234 IP (tos 0x0, ttl 64, id 28372, offset 0, flags [DF], proto TCP (6), length 62)
    172.21.198.88.35136 > 172.21.198.86.25163: Flags [P.], cksum 0x9b74 (correct), seq 1:11, ack
1, win 229, options [nop,nop,TS val 306451942 ecr 494471691], length 10
1557352477.769248 IP (tos 0x0, ttl 64, id 53937, offset 0, flags [DF], proto TCP (6), length 52)
    172.21.198.86.25163 > 172.21.198.88.35136: Flags [.], cksum 0xe500 (incorrect -> 0xacc3), ack
 11, win 227, options [nop,nop,TS val 494471691 ecr 306451942], length 0
1557352477.769301 IP (tos 0x0, ttl 64, id 53938, offset 0, flags [DF], proto TCP (6), length 62)
    172.21.198.86.25163 > 172.21.198.88.35136: Flags [P.], cksum 0xe50a (incorrect -> 0x9b6c), se
q 1:11, ack 11, win 227, options [nop,nop,TS val 494471691 ecr 306451942], length 10
1557352477.769320 IP (tos 0x0, ttl 64, id 53939, offset 0, flags [DF], proto TCP (6), length 52)
    172.21.198.86.25163 > 172.21.198.88.35136: Flags [F.], cksum 0xe500 (incorrect -> 0xacb8), se
q 11, ack 11, win 227, options [nop,nop,TS val 494471691 ecr 306451942], length 0
1557352477.769436 IP (tos 0x0, ttl 64, id 28373, offset 0, flags [DF], proto TCP (6), length 52)
    172.21.198.88.35136 > 172.21.198.86.25163: Flags [.], cksum 0xacb7 (correct), ack 11, win 229
, options [nop,nop,TS val 306451942 ecr 494471691], length 0
1557352477.769455 IP (tos 0x0, ttl 64, id 28374, offset 0, flags [DF], proto TCP (6), length 1502
)
    172.21.198.88.35136 > 172.21.198.86.25163: Flags [P.], cksum 0xeaaa (incorrect -> 0x6acf), se
q 11:1461, ack 12, win 229, options [nop,nop,TS val 306451942 ecr 494471691], length 1450
1557352477.769462 IP (tos 0x0, ttl 64, id 28376, offset 0, flags [DF], proto TCP (6), length 52)
    172.21.198.88.35136 > 172.21.198.86.25163: Flags [F.], cksum 0xa70b (correct), seq 1461, ack
12, win 229, options [nop,nop,TS val 306451942 ecr 494471691], length 0
1557352477.769510 IP (tos 0x0, ttl 64, id 53940, offset 0, flags [DF], proto TCP (6), length 52)
    172.21.198.86.25163 > 172.21.198.88.35136: Flags [.], cksum 0xe500 (incorrect -> 0xa6f8), ack
 1461, win 249, options [nop,nop,TS val 494471691 ecr 306451942], length 0
1557352477.769523 IP (tos 0x0, ttl 64, id 53941, offset 0, flags [DF], proto TCP (6), length 52)
    172.21.198.86.25163 > 172.21.198.88.35136: Flags [.], cksum 0xe500 (incorrect -> 0xa6f7), ack
 1462, win 249, options [nop,nop,TS val 494471691 ecr 306451942], length 0
^C
43 packets captured
52 packets received by filter
5 packets dropped by kernel
lorus@uw1-320-06:~/U/CSS432/Program3$
```

Jarod Guerrero
CSS 432
5/9/19

Analysis 1: TCP State and Timing Diagrams



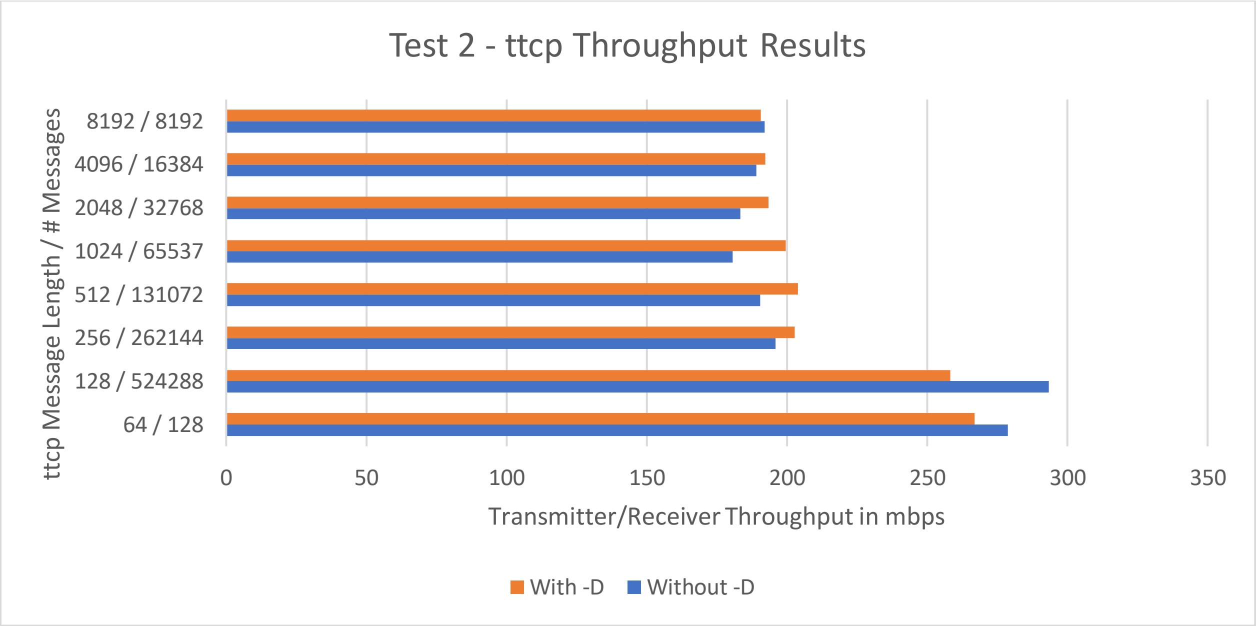
Jarod Guerrero
CSS 432
5/9/19



Output from running my coded version of hw3 is in hw3dupOutput.txt. Produces the same output at above.

Jarod Guerrero
CSS 432
5/9/19

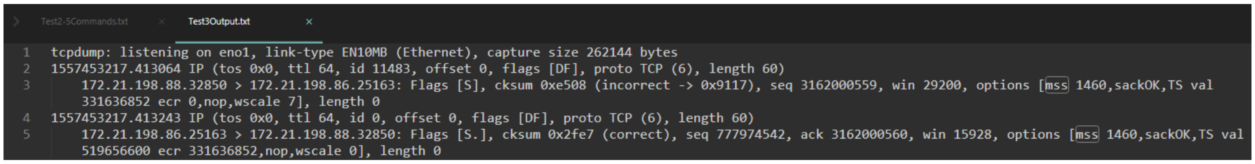
Analysis 2: ttcp Throughput by Message Length and Size



Increasing the buffer length past 128 significantly lowers the average throughput.

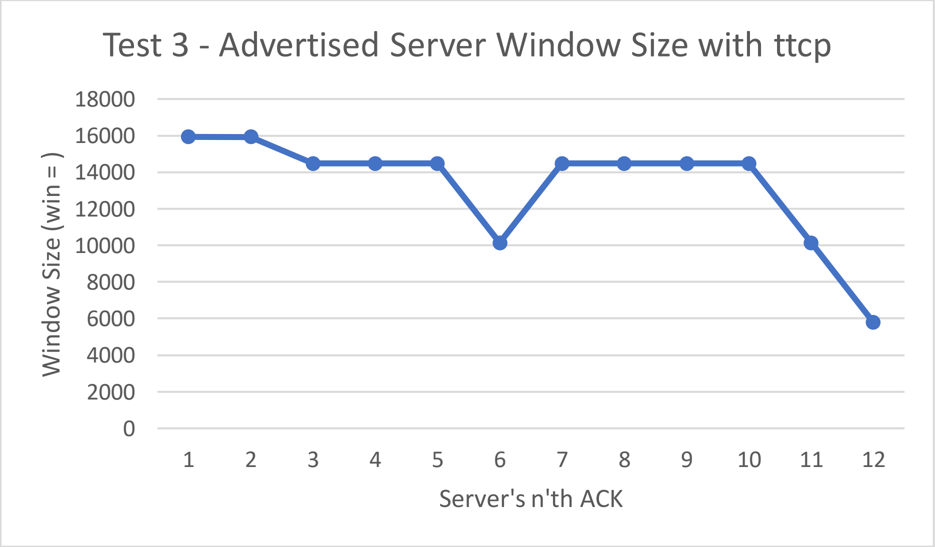
It's hard to say whether the -D TCP_NODELAY option affected the results. The throughput at lengths 64 and 128 are slightly lower with TCP_NODELAY, but slightly higher for all other message lengths.

Analysis 3: Running tcpdump with ttcp



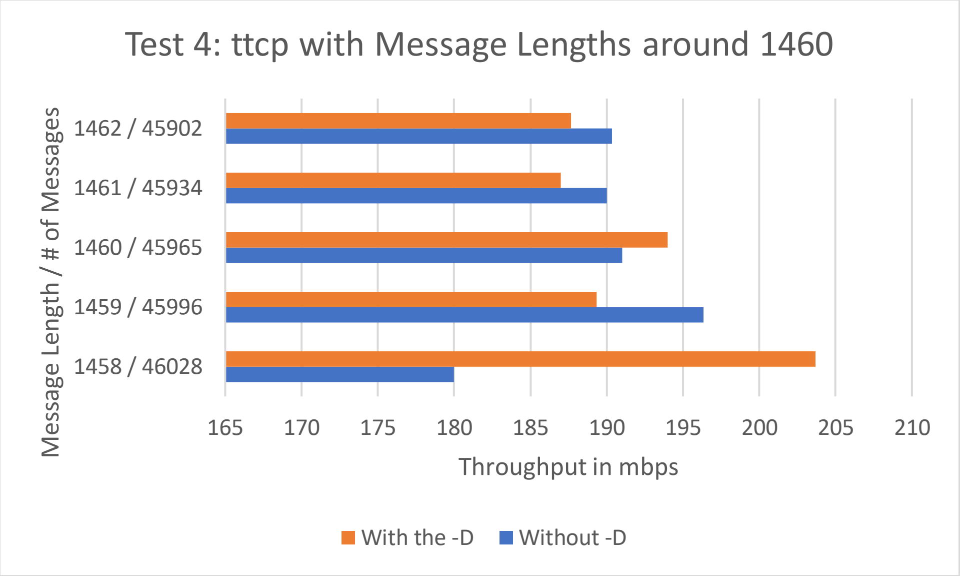
```
tcpdump: listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
1557453217.413064 IP (tos 0x0, ttl 64, id 11483, offset 0, flags [DF], proto TCP (6), length 60)
    172.21.198.88.32850 > 172.21.198.86.25163: Flags [S], cksum 0xe508 (incorrect -> 0x9117), seq 3162000559, win 29200, options [mss 1460,sackOK,TS val
    331636852 ecr 0,nop,wscale 7], length 0
1557453217.413243 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    172.21.198.86.25163 > 172.21.198.88.32850: Flags [S.], cksum 0x2fe7 (correct), seq 777974542, ack 3162000560, win 15928, options [mss 1460,sackOK,TS val
    519656600 ecr 331636852,nop,wscale 0], length 0
```

The minimum segment size (mss) in TCP and for this test is 1460.



It doesn't appear to be slow start nor AIMD. The window starts high so it can't be slow start. When it decreases, it doesn't halve so it does not follow the saw tooth AIMD pattern. It's probably another algorithm listed on https://en.wikipedia.org/wiki/TCP_congestion_control

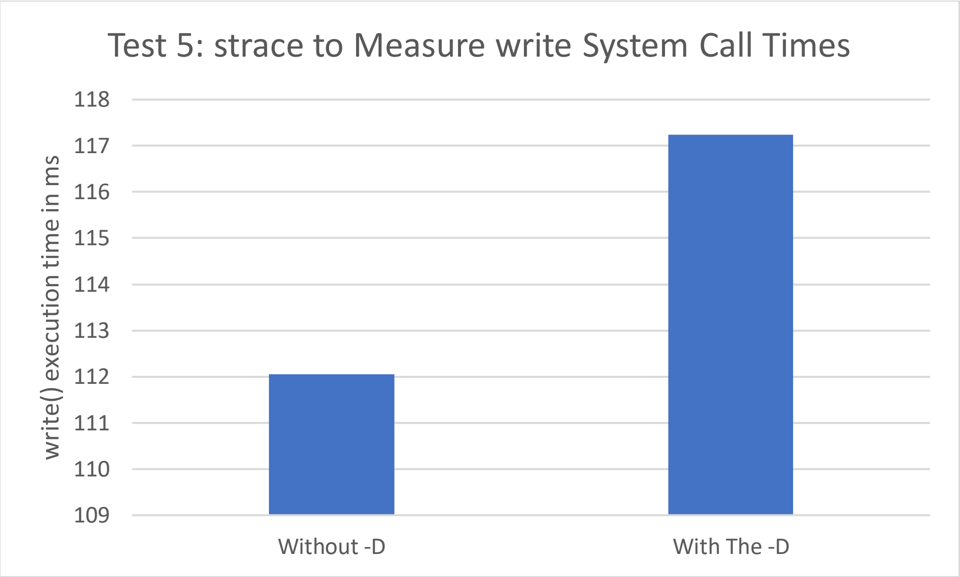
Analysis 4: ttcp Throughput around 1460 mss



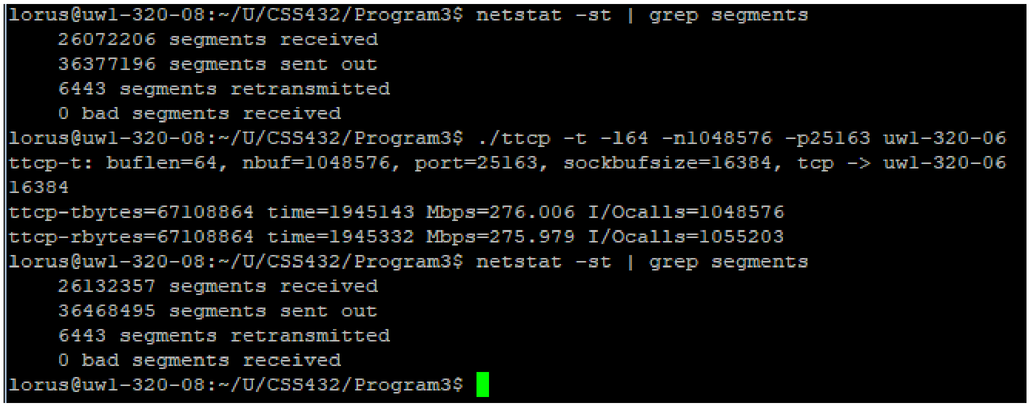
Buffer length and the TCP_NODELAY option did not significantly affect the results for buffer sizes 1459-1462. Without the -D option, a buffer size of 1458 had a throughput of only 180mbps, which was improved when turning on TCP_NODELAY to around 204mbps.

TCP_NODELAY disables Nagle's algorithm so that data is transmitted as soon as possible instead of waiting on the algorithm.

Analysis 5: netstat and strace with/without TCP_NODELAY

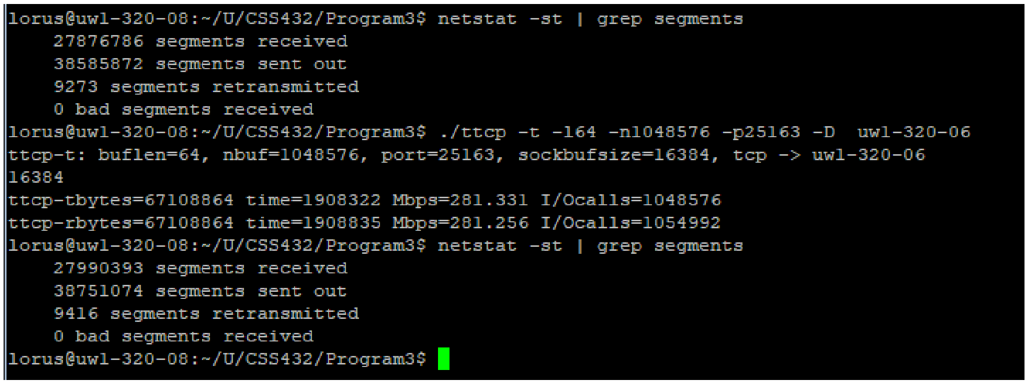


Adding the TCP_NODELAY flag increases the execution time by around 10%. It's hard to say whether this is significant or just variance in the network.



```
lorus@uwl-320-08:~/U/CSS432/Program3$ netstat -st | grep segments
    26072206 segments received
    36377196 segments sent out
    6443 segments retransmitted
    0 bad segments received
lorus@uwl-320-08:~/U/CSS432/Program3$ ./ttcp -t -l64 -n1048576 -p25163 uwl-320-06
ttcp-t: buflen=64, nbuf=1048576, port=25163, sockbufsize=16384, tcp -> uwl-320-06
16384
ttcp-tbytes=67108864 time=1945143 Mbps=276.006 I/Ocalls=1048576
ttcp-rbytes=67108864 time=1945332 Mbps=275.979 I/Ocalls=1055203
lorus@uwl-320-08:~/U/CSS432/Program3$ netstat -st | grep segments
    26132357 segments received
    36468495 segments sent out
    6443 segments retransmitted
    0 bad segments received
lorus@uwl-320-08:~/U/CSS432/Program3$
```

Turning on TCP_NODELAY also slightly increases the number of packets received, sent out, and retransmitted.



```
lorus@uwl-320-08:~/U/CSS432/Program3$ netstat -st | grep segments
    27876786 segments received
    38585872 segments sent out
    9273 segments retransmitted
    0 bad segments received
lorus@uwl-320-08:~/U/CSS432/Program3$ ./ttcp -t -l64 -n1048576 -p25163 -D  uwl-320-06
ttcp-t: buflen=64, nbuf=1048576, port=25163, sockbufsize=16384, tcp -> uwl-320-06
16384
ttcp-tbytes=67108864 time=1908322 Mbps=281.331 I/Ocalls=1048576
ttcp-rbytes=67108864 time=1908835 Mbps=281.256 I/Ocalls=1054992
lorus@uwl-320-08:~/U/CSS432/Program3$ netstat -st | grep segments
    27990393 segments received
    38751074 segments sent out
    9416 segments retransmitted
    0 bad segments received
lorus@uwl-320-08:~/U/CSS432/Program3$
```