Program 3 Performance Results

Aika usui

# Test 1

## The hw3 tcpdump results

**A screenshot of a newspaper

Description automatically generated**

## My program’s tcpdump result

**A close up of a newspaper

Description automatically generated**

# Analysis 1

## State diagram

**A close up of a map

Description automatically generated**

## Timing Chart

**A close up of a piece of paper

Description automatically generated**

# Analysis 2

Table: ttc Mbps for each message condition and with D flag and without

|  |  |  |
| --- | --- | --- |
| **-l (message length)/-n (# messages)** | **Without D (Mbps)** | **With D (Mbps)** |
| 64/1048576 | 451.048 | 409.159 |
| 128/524288 | 662.713 | 648.475 |
| 256/262144 | 512.436 | 603.386 |
| 512/131072 | 537.99 | 448.914 |
| 1024/65536 | 930.127 | 597.348 |
| 2048/32768 | 540.237 | 466.247 |
| 4096/16384 | 763.967 | 581.416 |
| 8192/8192 | 1037.78 | 942.247 |

1. The effect of buffer length **without using -D option**

Reference to the graph above, without using the -D flag option for ttcp, the throughput increases as a buffer length increases. There are some outlier when the buffer length are 256 and 2048.

1. The effect of **-D option**

Compare the throughput values on with and without -D option, most of the case, without -D option is greater than with -D option; however, the difference is not significant. I observed the biggest throughput difference between with and without -D option when message length of 1024, number of messages of 65536. It is interesting to see that both of the trends are similar.

# Analysis 3

First 10 window size from Server ACK

|  |  |
| --- | --- |
| ACK | Window Size |
| 1 | 26880 |
| 2 | 8948 |
| 3 | 8948 |
| 4 | 8948 |
| 5 | 13422 |
| 6 | 13422 |
| 7 | 13422 |
| 8 | 13422 |
| 9 | 13422 |
| 10 | 6711 |

1. Does the growth of the advertised window follow additive increment, slow start, or perhaps a different algorithm?

It is hard to tell which congestion control algorithms this program follows. However, the change in window size from ACK 9 to ACK 10 is exactly half. With that said, the growth of the advertised window might follow slow start.

1. How large is MSS in TCP? Is it 1448 or 1460 bytes?

MSS in TCP as found is 8960. It turns out to be neither 1448 or 1460.

# Analysis 4

ttcp throughput in slight difference in message lengths with and without -D option

|  |  |  |
| --- | --- | --- |
| **-l (message length) / -n (# messages)** | **Without -D (Mbps)** | **With -D (Mbps)** |
| 1458/46028 | 849.769 | 558.857 |
| 1459/45996 | 770.727 | 473.468 |
| 1460**/**45965 | 439.961 | 587.057 |
| 1461/45934 | 765.966 | 494.151 |
| 1462/45902 | 515.878 | 568.653 |

1. The effect of buffer length **without using -D option**

There doesn’t seem to be a trend in throughput in increasing or decreasing. However, there are some outlier at 1460/45965 and 1462/45902.

1. The effect of **-D option**

Compare to ttcp without -D option, the throughput for ttcp with -D option is more stable. There are some outlier but without -D option seems to be better because it has bigger throughput.

# Analysis 5

## Client: netstat -st | grep segments

**Without -D**

Before:

A picture containing table, knife

Description automatically generated

After:

A picture containing table

Description automatically generated

Difference:

Received: -6720

Send out: -10029

Retransmitted: -1

Bad segments received: 0

**With -D**

Before:

A screenshot of a cell phone

Description automatically generated

After:

A screenshot of a cell phone

Description automatically generated

Difference:

Received: -8458

Send out: -9026

Retransmitted: -2

Bad segments received: 0

## Server: netstat -st | grep segments

**Without -D**

Before:

A picture containing table

Description automatically generated

After:

A picture containing table

Description automatically generated

Difference:

Received: -9831

Send out: -6774

Retransmitted: 0

Bad segments received: 0

**With -D**

Before:

A picture containing table

Description automatically generated

After:

A picture containing table

Description automatically generated

Difference:

Received: -8526

Send out: -8539

Retransmitted: -1

Bad segments received: 0

## OS to execute each write system call

Time differences of without -D and with -D over the first 20 write()

|  |  |  |
| --- | --- | --- |
| Write() | Without -D  Time difference | With -D  Time difference |
| 1 | **0.000024** | 0.000023 |
| 2 | 0.000016 | 0.000022 |
| 3 | 0.000028 | 0.000028 |
| 4 | 0.000015 | 0.000020 |
| 5 | 0.000027 | 0.000027 |
| 6 | 0.000015 | 0.000021 |
| 7 | 0.000028 | 0.000021 |
| 8 | 0.000015 | 0.000022 |
| 9 | 0.000016 | 0.000021 |
| 10 | 0.000024 | 0.000022 |
| 11 | 0.000015 | 0.000022 |
| 12 | 0.000018 | 0.000022 |
| 13 | 0.000016 | 0.000022 |
| 14 | 0.000029 | 0.000022 |
| 15 | 0.000015 | 0.000022 |
| 16 | 0.000017 | 0.000021 |
| 17 | 0.000016 | 0.000028 |
| 18 | 0.000041 | 0.000022 |
| 19 | 0.000016 | 0.000022 |
| 20 | 0.000016 | 0.000022 |

1. the number of TCP packets sent, received, and retransmitted

Between the client and server, there is no big differences in sends, receives, or retransmits between using the D flag and not using it. However, I found out that by using -D option, it can reduce the retransmits. Also, for server side, using -D option can reduce the number of segments that is sent out.

1. the time required for OS to execute each write system call (focus on the first 20 writes.)

According to the table above, most of the time, with -D option has greater time differences. This is clear because with -D option, it requires some overhead because it doesn’t buffer when it sends. Thus, the additional overheads make each segments to processed slower.