Cpt S 411 Assignment Cover Sheet

(To be turned in along with each homework and program project submission)

Assignment # 1

For individual assignments:

Student name (Last, First): Cuevas, Jessica

For team projects:

List of all students (Last, First): Cuevas, Jessica

List of collaborative personnel (excluding team participants):
N/A

I[1] certify that I have listed above all the sources that I consulted regarding this assignment, and that I have not received or given any assistance that is contrary to the letter or the spirit of the collaboration guidelines for this assignment. I also certify that I have not referred to online solutions that may be available on the web or sought the help of other students outside the class, in preparing my solution. I attest that the solution is my own and if evidence is found to the contrary, I understand that I will be subject to the academic dishonesty policy as outlined in the course syllabus.

Please print your names.
Jessica Cuevas

Assignment Project Participant(s):
Jessica Cuevas

Today's Date:
09/17/20

[1] If you worked as a team, then the word "I" includes yourself and your team members.

Programming Project 1 Report

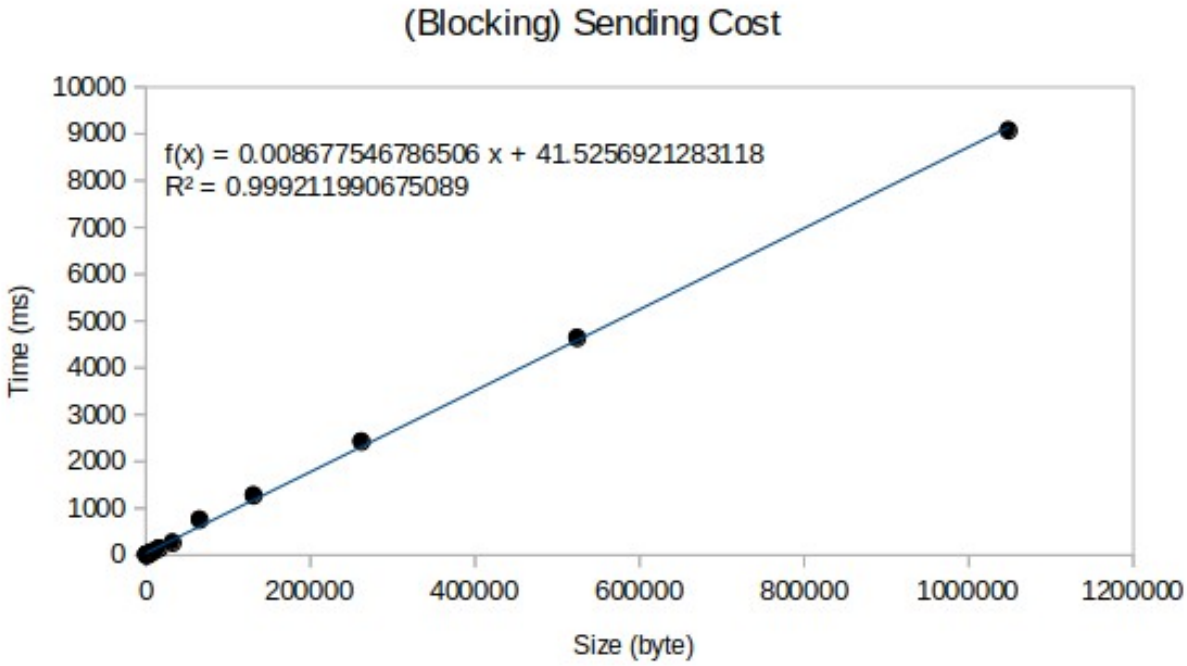
Blocking Test Tables and Chart Results:

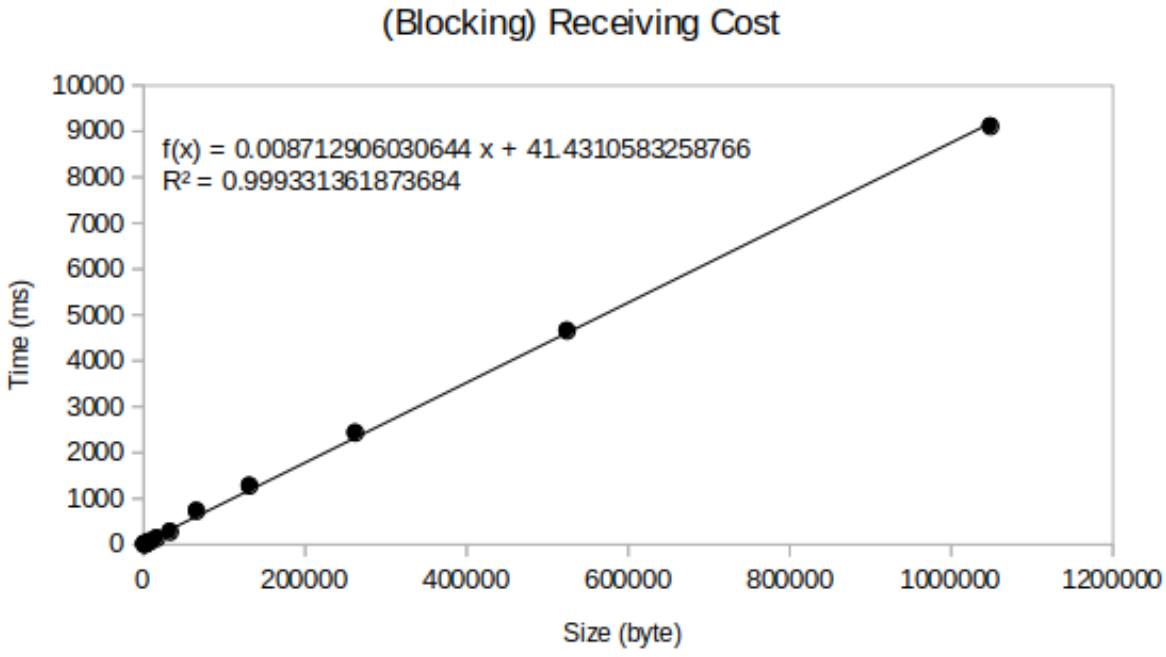
Table 1.

Send	Size (byte)	Time (ms)	Receive	Size (byte)	Time (ms)
1	1	2	0	1	3
1	2	2	0	2	2
1	4	2	0	4	2
1	8	2	0	8	3
1	16	2	0	16	2
1	32	2	0	32	3
1	64	2	0	64	2
1	128	2	0	128	3
1	256	2	0	256	3
1	512	2	0	512	5
1	1024	6	0	1024	9
1	2048	12	0	2048	18
1	4096	31	0	4096	34
1	8192	66	0	8192	68
1	16384	143	0	16384	140
1	32768	265	0	32768	277
1	65536	757	0	65536	733
1	131072	1275	0	131072	1280
1	262144	2425	0	262144	2434
1	524288	4638	0	524288	4657
1	1048576	9072	0	1048576	9110

Table 2.

Send	Size (byte)	Time (ms)	Receive	Size (byte)	Time (ms)
1	1	20064	0	1	20218
1	1	20059	0	1	20212
1	1	19993	0	1	20225
1	1	19955	0	1	20211
1	1	20012	0	1	20212


(Blocking) Sending Cost
f(x) = 0.008677546786506 x + 41.5256921283118
R² = 0.999211990675089
Time (ms)
Size (byte)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
200000
400000
600000
800000
1000000
1200000


(Blocking) Receiving Cost
f(x) = 0.008712906030644 x + 41.4310583258766
R² = 0.999331361873684
Time (ms)
Size (byte)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
200000
400000
600000
800000
1000000
1200000

Blocking Test Derivations

Processor /Rank	Average Latency (secs)	Bandwidth (bytes/sec)	Network Buffer Size Approximation (bytes)
Send	0.020015	115239943.34	8192
Receive	0.020213	114772269.61	

Comments of Deriving Parameters:

The latency was calculated for both the send and receiver by first running the send/receive message blocking algorithm with a size of one byte before increasing the size in a for loop. This code ran up to five times to obtain multiple readings and calculate the average of the latency for both the send and receiver.

```
if(rank==1)
{
    //destination tag
    int dest = 0;

    char test_x = 0;

    //-----------------------LATENCY TEST-----------------------------------//
    //start timer
    gettimeofday(&t1,NULL);

    MPI_Send(&test_x,1,MPI_CHAR,dest,0,MPI_COMM_WORLD);

    //End of timer.
    gettimeofday(&t2,NULL);

    int tSend = (t2.tv_sec-t1.tv_sec)*1000000 + (t2.tv_usec-t1.tv_usec);
    printf("%d,%d,%d\n", rank,1,tSend);
    //-----------------------LATENCY TEST-----------------------------------//
```

The bandwidth for both send and receiver were derived starting from 512 to 1048576 (1MB) message bytes. A fitted linear line was set in the graph using linear regression. The once the slop of each linear fit was calculated the slope was inverted and then converted into seconds instead of microseconds.

The network buffer size was mostly approximated by looking closely at the sending time. I obtained the difference of time between the sender and the receiver, and then found the moment in which the senders time started to be really close to the receivers time, and made that size into the network buffer size.

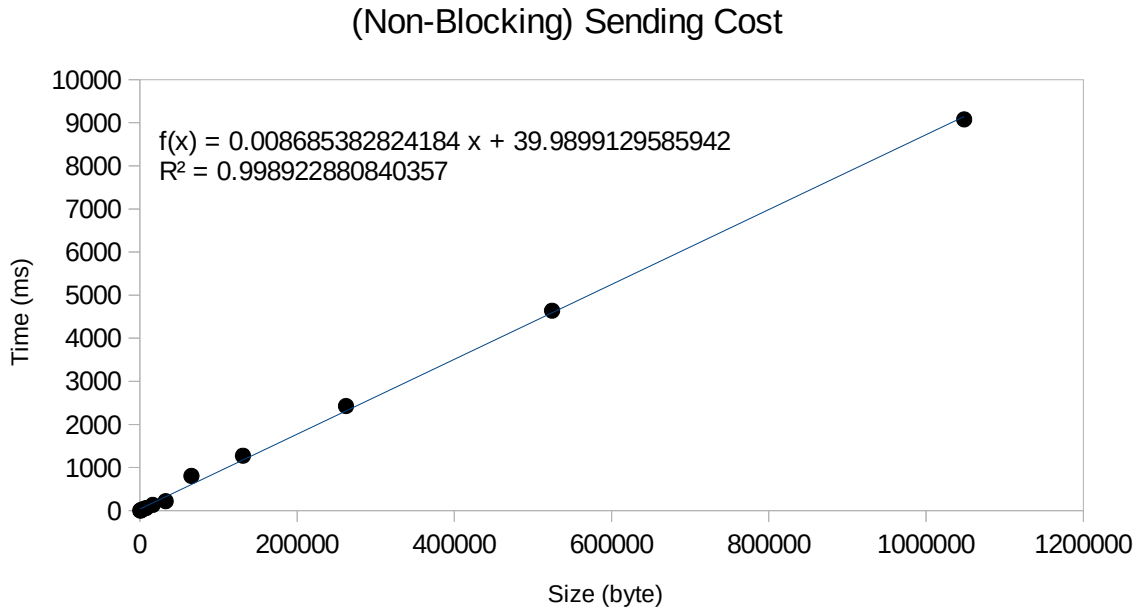
Non-Blocking Test Tables and Chart Results:

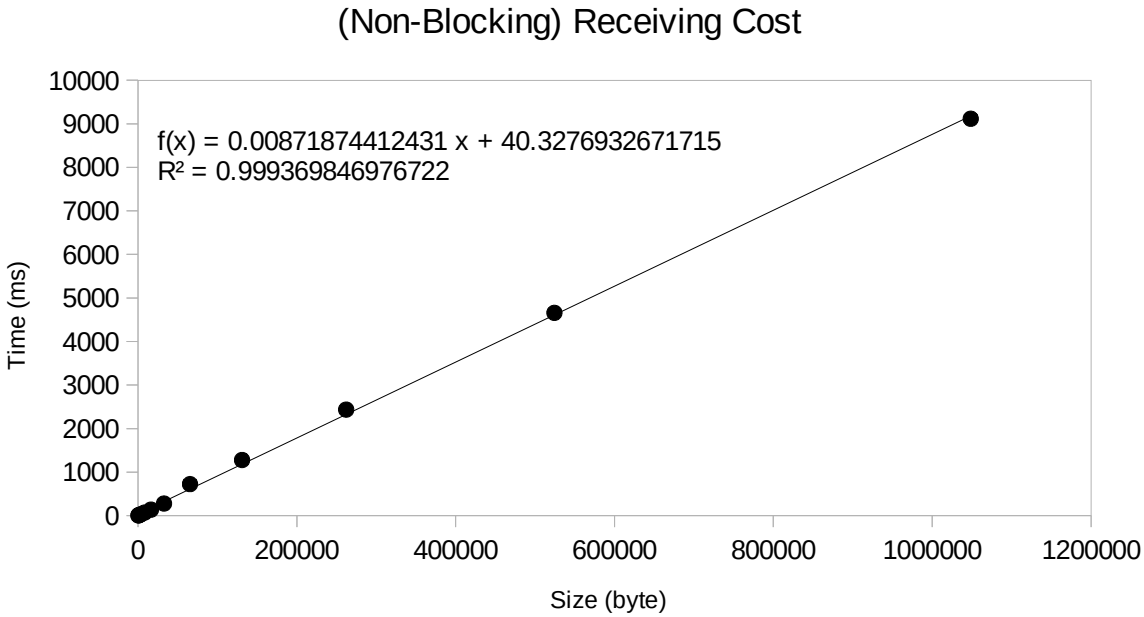
Table 3.

Send	Size (byte)	Time (ms)	Receive	Size (byte)	Time (ms)
1	1	2	0	1	3
1	2	2	0	2	2
1	4	2	0	4	2
1	8	2	0	8	1
1	16	2	0	16	3
1	32	2	0	32	2
1	64	2	0	64	2
1	128	3	0	128	3
1	256	3	0	256	6
1	512	2	0	512	5
1	1024	4	0	1024	10
1	2048	16	0	2048	17
1	4096	34	0	4096	35
1	8192	61	0	8192	70
1	16384	133	0	16384	138
1	32768	219	0	32768	278
1	65536	805	0	65536	724
1	131072	1273	0	131072	1278
1	262144	2426	0	262144	2435
1	524288	4639	0	524288	4658
1	1048576	9078	0	1048576	9116

Table 4.

Send	Size (byte)	Time (ms)	Receive	Size (byte)	Time (ms)
1	1	19989	0	1	20221
1	1	20013	0	1	20221
1	1	19996	0	1	20220
1	1	19992	0	1	20219
1	1	19944	0	1	20213


(Non-Blocking) Sending Cost
f(x) = 0.008685382824184 x + 39.9899129585942
R² = 0.998922880840357
Time (ms)
Size (byte)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
200000
400000
600000
800000
1000000
1200000


(Non-Blocking) Receiving Cost
f(x) = 0.00871874412431 x + 40.3276932671715
R² = 0.999369846976722
Time (ms)
Size (byte)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
200000
400000
600000
800000
1000000
1200000

Non-Blocking Test Derivations

Processor /Rank	Average Latency (secs)	Bandwidth (bytes/sec)	Network Buffer Size Approximation (bytes)
Send	0.019985	115135972.73	1024
Receive	0.020216	114695417.8	

Comments of Deriving Parameters:

Just as the blocking test, for the non-blocking test the latency was calculated for both the send and receiver by first running the send/receive message blocking algorithm with a size of one byte before increasing the size in a for loop. This code ran up to five times to obtain multiple readings and calculate the average of the latency for both the send and receiver.

The bandwidth for both send and receiver were derived starting from 512 to 1048576 (1MB) message bytes. A fitted linear line was set in the graph using linear regression. The once the slop of each linear fit was calculated the slope was inverted and then converted into seconds instead of microseconds.

The network buffer size was mostly approximated by looking closely at the sending time. I obtained the difference of time between the sender and the receiver, and then found the moment in which the senders time started to be really close to the receivers time, and made that size into the network buffer size.

Final Thoughts:

Overall for the blocking and non-blocking test we can observe a very similar values for both latency and bandwidth, while the approximation of the network buffer was slightly lower for non-blocking compared to blocking.