Assignment # 1

For team projects:

List of all students (Last, First):

Memon, Samra

Araiza, Jovan

I and Jovan Araiza 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.

Memon, Samra

Araiza, Jovan

Assignment Project Participant(s):

Memon, Samra

Araiza, Jovan

Today’s Date: 09/17/2019

"Professional" Report Project 1

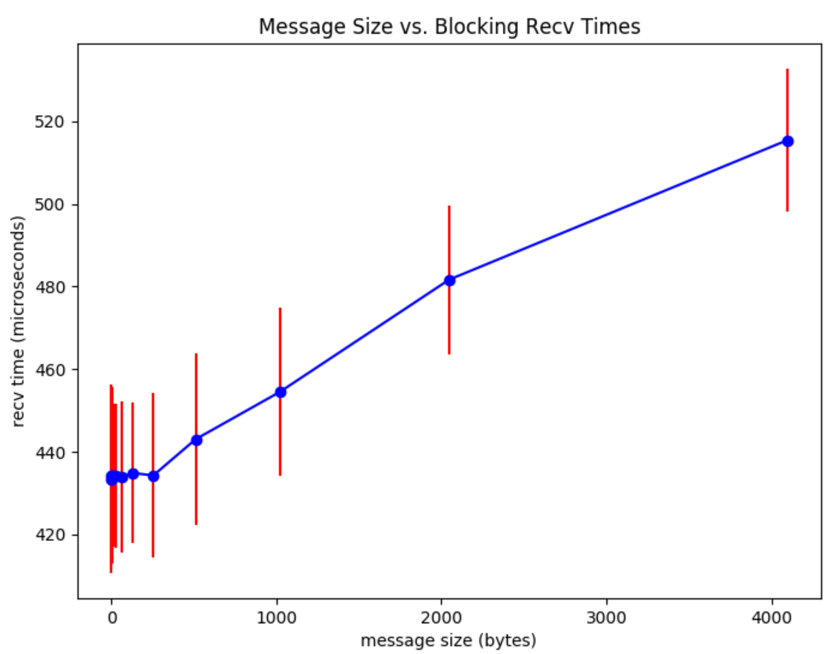
Experiment

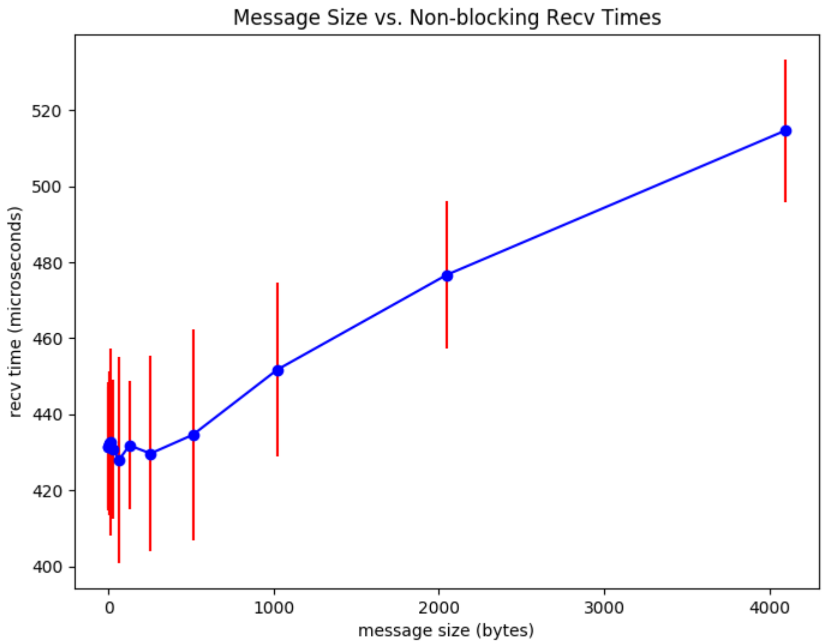
For our experiment setup we created a shell script to run the network messaging stress tests in order to derive the network properties. We tested and recorded the send time, receive time, and message size for every message sent over the network buffer using two versions of our source code. The two versions are blocking and nonblocking which use different MPI Receive functions (i.e. MPI_Recv and MPI_Irecv). We tested message sizes 1, 2, 4, 8, …, 4096 bytes.

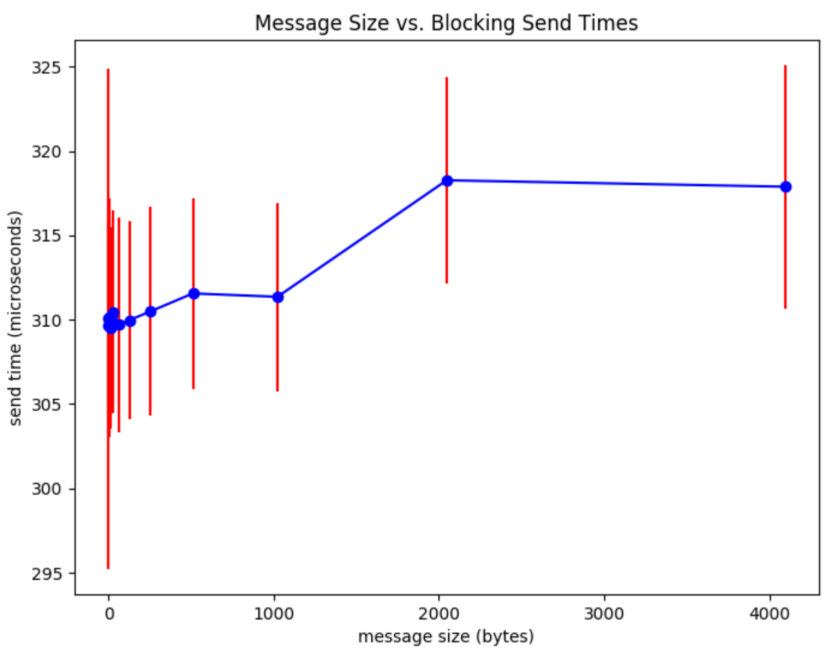
For each message size we performed the test 160 times and recorded all results into a single csv file. We then wrote a python script to parse the csv file and calculated the average and standard deviation for send and receive times over the 160 tests for every message size.

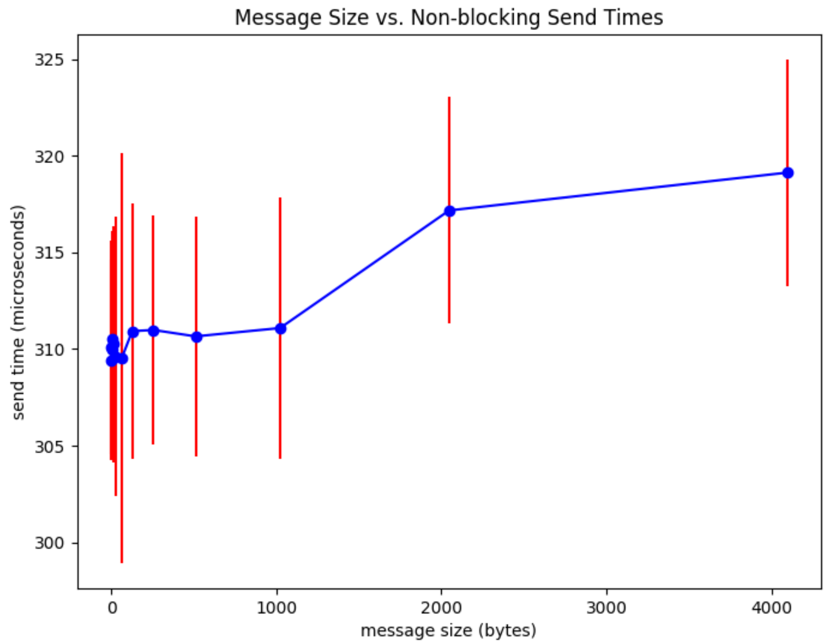
We then graphed the results to determine the message size at which we began to see an upward linear trend for send and receive times of both versions of the source code. This message size represents to the network buffer size. Based off the network buffer size found, we fit a line of best fit to all points after this buffer size to determine the slope of the line for both send and receive. The slope in this case represents the inverse of the network bandwidth. We averaged all points before the network buffer size point to determine the latency of send and receive too.

Results




Message Size vs. Non-blocking Recv Times
recv time (microseconds)
message size (bytes)
400
420
440
460
480
500
520
0
1000
2000
3000
4000


Message Size vs. Blocking Send Times
send time (microseconds)
message size (bytes)
295
300
305
310
315
320
325
0
1000
2000
3000
4000



	Blocking	**Non-blocking**
Send Latency (microseconds)	310.15	310.20
Receive Latency (microseconds)	434.07	431.32
Send Buffer Size (bytes)	1024	1024
Receive Buffer Size (bytes)	256	256
Send Bandwidth (bytes/microseconds)	557.41	420.33
Receive Bandwidth (bytes/microseconds)	47.66	44.93

Discussion

From the graphs of non-blocking and blocking above, we concluded:

- For messages smaller than 1024 bytes to send and messages smaller than 256 bytes to receive, latency (start-up time) dominates transfer time because the start up time for the connection takes longer.
- For messages larger than 1024 bytes to send and messages larger than 256 bytes to receive, bandwidth dominates transfer time because the message size exceeds the network buffer size.
- The network buffer of the senders is bigger than the network buffer of the receivers. The network buffer size of both senders is estimated to be 1000 bytes whereas the network buffer size of both receivers is estimated to be 256 bytes
- For the senders only, the more spread out the send times will be for the message size that does not exceed the buffer size