III Measurement (We take this part as optional for COMP429)

Our measurement method involves analyzing code to find what are possible sources of delay that are irrelevant to network bandwidth, measuring average one-round ping-pong message time delay, and estimating network bandwidth.

First, upon each connection between client and server, we will send a special message that contains only one byte information (this differs from ping-pong message in size so that we can treat this special message as speed testing message). Second, based on our analysis, the most of overhead are added by recv() and send(). Since send() is non-blocking and recv() is blocking, they have different time of delays. We measure send() at client side and recv() at server side.

Our reasoning behind this choice is that server will only read from client if there are something can be read, therefore, after the if statement checks that there are something can be read from this client, the server will then recv() from client. We can then measure the delay caused by recv() function call by calling gettimeofday() before and after function calls without having transmission time between client and server (this is the very reason that we don't measure recv() at client side because recv() is blocking and the delay will include both recv() delay and transmission delay which is bandwidth dependent). We cannot measure send() at server side because then we have to send twice which is in efficient (one send()for measuring its delay and another ont to send the data back to the client). We measure send() at the client side because we assume the send() function call itself should have roughly equal delay time on server and client sides (same assumption on recv()). Because in our design, our server do not wait upon receiving and send back immediately we assume no other independent delay is significant in our interest. Thus we hypothesize the independent delay most likely consists 2 send() calls and 2 recv() calls.

We calculate the delay by measuring time differences. independent_delay = ((endtime->tv_sec) * 1000000L + (endtime->tv_usec))*2 + ((starttime1.tv_sec - starttime1.tv_sec)* 1000000L + (starttime1.tv_usec - starttime1.tv_usec)) * 2; Here, endtime is struct contains the delay of recv() on server side, starttime1.tv_sec - starttime1.tv_sec and starttime1.tv_usec - starttime1.tv_usec measures send() delay in seconds and microseconds respectively. Then in ping-pong message, we measure COUNT times total delay and take the average. We can calculate bandwidth by bandwidth = (size * 2 * 8) / average_delay