Group: Jack Mcleod 104743223 Steven Shlimoon 104832637, Ivan Beric (104 831 658)

q1)

1) Seeker establishes a connection with a 3 way handshake. Creator sends a message and Seeker receives it. Seeker sends a message and Creator receives it. Creator then sends a close message.

2) Our goal in using TCP is to make sure that our data is able to be sent reliably back and forth between multiple job creators and senders without loss of data.

3) Job seekers will establish a connection with the job creator**.** The job creator sends a message asking to send a job. The job seeker then receives the job and messages back that the job is complete or declines the job. The creator receives the completion message or non acceptance message and once all jobs are complete sends a close message.

4) Job seekers must establish a connection with a three way handshake. Job seeker nodes can not connect directly to each other. Job creators must send a specified number of jobs to seekers before closing. Job seekers can accept or decline a job. Job creators receive information of a job's completion or denial and end when the required amount of jobs is complete.

q2)

It would be better not to use a standard protocol because a standard protocol is not self-scalable, meaning that it becomes difficult for this application to broaden its audience and grow a larger network. This application is meant to communicate to an un-set amount of multiple devices, so a standard protocol would not be as efficient in growing the amount of devices to be communicated to.

q3)

CURRENT ITERATION TEST CASES:

CASE 0:

Start a Job Creator 'Ca', start a Job Seeker 'Sa', connection is terminated by Sa

CASE 1:

Start a Job Creator 'Ca', start a Job Seeker 'Sa', connection is terminated by Ca

CASE 2:

Start a Job Creator 'Ca' and create a Job 'CaJ1',
start a Job Seeker 'Sa', Sa connects to Ca and rejects job offer from Ca,
connection is maintained, Sa accepts job offer from Ca,
connection is maintained, Sa rejects job offer from Ca,
connection is terminated

ITERATION 2 TEST CASES:

CASE 3:

Start a Job Creator 'Ca' and create Jobs 'CaJ1' and 'CaJ2,
Start another Job Creator 'Cb' and create a Job 'CbJ1',
Start a job seeker 'Sa', Sa connects to Ca and accepts job CaJ1,
connection is maintained, Sa accepts job CaJ2, connection is maintained,
Sa connects to Cb and accepts job CbJ1, connection is terminated

CASE 4:

Start a Job Creator 'Ca' and create a Job 'CaJ1',
start Job Seekers 'Sa1', 'Sa2' and 'Sa3',
Sa1, Sa2 and Sa3 connect to Ca and all accept job CaJ1,
Sa1 completes job CaJ1, connection is terminated with Sa1 and Sa2 but
maintained with Sa1, connection is terminated by Sa1