**COMP-3670: Assignment 2**

1. a) The communication pattern we will be using will be server-to-client. We chose this pattern because the job seekers won’t connect to each other they will only connect to the job creators. Since a job seeker will only connect to one job creator at a time and a node can only take on one role at a time, this resembles a server-to-client pattern which is why we are going to implement this type of network. The network application will have one group of end systems as the job seekers and the other group of end systems as the job creators. The job seekers will offer their services to the job creators and the job creators will give them jobs if they have any.

b) The protocol design that we are going to use is TCP since it is a more reliable connection. For our network application we want the job seeker to connect to the job creator and maintain that connection until the job seeker replies to the job creator and once it is done it will terminate the connection. TCP offers a stable and reliable connection since it has its “handshake” method which will guarantee that the two end systems will stay connected until the task is done or rejected. It also offers error checking and will send the packets in order which will ensure that all the data is being transmitted properly and notify us if not. If all the data was not sent accordingly then we can implement a solution to fix it.

c) Since there are only two types of nodes, job-seeker and job-creator, they will communicate with each other using the following message format:

**Job-Creator Message:**

method|sp|value|cr|lf|

Connection |sp|value|cr|lf|

The job-creator’s message contains two lines. The first line specifies the method which can be one of the following:

* LEAVE: this is when the job seeker connects to the job-creator but the job-creator does not have any jobs for it so it tells the job-seeker to leave and come back later. When this method is called the value will be nothing
* ADD: this will be called when the job-creator has a job and wants the job-seeker to perform one of the methods. When this method is called the value will be a list of numbers separated by a space to perform the operation on(ex. ADD 1 1 1 => the answer will be 3)

The second line specifies whether it is connected to the job-seeker or not and can terminate the connection when it wants. The connection value can be one of the following:

* Open: this is when there is a connection between the job-creator and job-seeker
* Close: this is when the job-creator wants to terminate the connection with the job-seeker

**Job-Seeker Message:**

Status|sp|value|cr|lf

Connection:|sp|value|cr|lf|

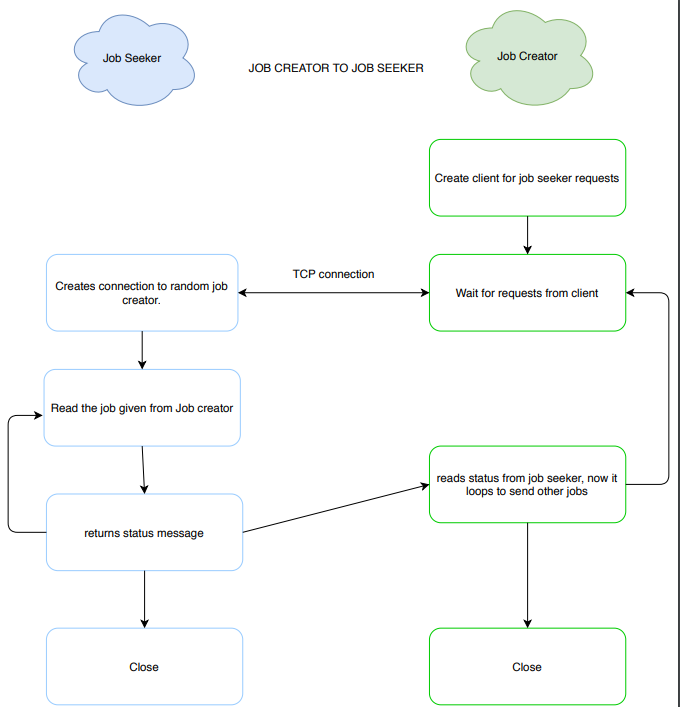
|cr|lf|

(data…)

The job-seeker’s message contains 3 lines and the entity body. The first line is the status code which is what it will return to the job-creator to notify it of the result of the request. The status could be one of the following:

* 100 OK: this is when the job was completed successfully
* 200 INVALID REQUEST: this is when the job-creator sends a job that cannot be understood or is sent the wrong data by the job-seeker
* 300 REJECTED: this is when the job-seeker rejects the job

The second line “connection” is the same as the job-creator’s “connection” and specifies whether it is connected to the job-creator or not and can terminate the connection when it wants. The values are the same as well and can be either “open” or “close”. Lastly, the entity body contains the actual answer or the data of the message.

d) 

Our network is going to be a half-duplex because we want one node or end-system to send a message at a time. It is also going to be stateless so no node will store information or know about previous connections of other nodes. Each job-seeker will be on a loop and go through the list of job-creators and connect to one of them at a time. Once they are connected the job-creator will notify the job-seeker if they have a task to do, in that case, the job-seeker can either do the job and return the status or it can reject the job and disconnect from the job-creator and go to the next one. If the job-creator does not have any tasks, then it will notify the job-seeker and the job-seeker will disconnect from the job-creator and notify it in the return status and go to the next one. Once job-creator and job-seeker connect, they will stay connected until one of them decides to break the connection.

1. Our new application layer protocol is designed for this network, so it transmits only the information needed which makes it cleaner and easier to communicate. Although the standard protocols may work for this network, they still include other useless information that won’t benefit any aspect of the network such as date or language.