Assignment 2 Part 2

Group: Julia Garant, Rebecca Wallace, Conner Sax

Q1) Design a network application protocol for this application and justify your design decision. In your protocol design, you should address the following. [35 points]

1. The communication patterns of the network application.
Legend: S = Job Seeker, C = Job Creator

This is going to be a stateful protocol since there is information which is going to be stored on both sides.

```
Connection Opened
S: My IP;UID is _____ // IP will be each seeker/creators UID
C: My IP;UID is _____ // creator will store the IP of the seeker
S: I am offering _____ service // creator will also store the
service this seeker gives
C:  I have/don’t have corresponding job for you _____ // if the
job isn't available then a not available code will be returned
and the connection will be close
S: I accept/deny _____ job // when the seeker denies the job
will stay available for other seekers that can also complete
that same job
C: job data sent
S: received job data
C: Connection closed

Job seeker has completed job
Connection Opened
S: My IP;UID is  _____
C: My IP;UID is  _____, waiting for return status of job
S: Job completed with code 0
```

S: result data (may have to break this up more)

C: Connection closed // if there are more segments to the job or there is another job the seeker can do then the seeker will be asked if it wants to do the new job before the connection is closed.

2. Define the protocol design goals.

- Clear design outline
  - Having a clear outline of the protocol before implementation to allow a clear focus on the task at hand and will provide a direction of action

- Simple but efficient
  - Keeping a small number of elements with each focused on one task will reduce complexity

- Minimum connection maintenance
  - The environment we create should stay minimal in relation to all connections and messages (request & response) created

- Limit redundancy
  - Assure messages are short and to the point to limit overflow

- Bounded
  - Adheres to system memory requirements

3. Define the message format, structure and semantics

Seeker and creator messages are based the same:
identification [int], availability [Boolean]
Eg.

Seeker sends message: I am offering these ____INT____
Creator responds: I have/don’t have corresponding job for you ___BOOL___

The job seeker will send out a request message to the job creator when it is ready for a job, then the job creator will send a response message with details of the job if one is being assigned.

4. Design the communication rules.

- Each node can only play one role (must be either a seeker or creator, never both)
- Every job seeker connects to one job creator at a time, that is it is a 1:1 ratio
- Job seekers do not connect to other job seekers
- Jobs can either be assigned right away, or the seeker can go into a queue waiting for the job to be assigned
- The job size can affect the number of job seekers required
- When a seeker completes its job, the status and results must be sent to the creator
- Seekers have the ability to reject jobs
- A seeker can have multiple jobs assigned to it, but only one active at a time
- The job can be terminated at any time by both the seeker and creator

- Every message field must contain the identifier, availability status, and message field

Q2) Argue the need for a new application layer protocol for this network application instead of using existing standard protocols (e.g. HTTP, SMTP, WebSocket, etc.) [10 points]

Creating a new application layer protocol for this network application is beneficial since we can design for the exact communication between job seeker and job creator. If we were to use a protocol such as HTTP we would not be able to retain any information since a protocol like HTTP is a stateless protocol. However, we would still need a new protocol even for a stateful protocol such as SMTP. This is because SMTP is designed with one goal and that is to send emails between two clients. For our protocol there will be multiple states which will not align with any existing protocol. Finally, building a new protocol makes the most sense since its communication patterns will be unique over other stateful protocols.

Q3) Provide the implication source code of your network application protocol with sufficient test cases based on the design goals, message philosophy (format, structure, semantics), and Communication rule.[25 points]

https://github.com/connersax/CompNetAssn2

Julia:

I worked on completing question 1.2, 1.3, and 1.4 with Rebecca. We focused on being as clear as possible with our answers while also making sure we had a solid understanding for this project. We were all three working together through calls while answering the questions to make sure everything was cohesive. Conner and I later worked together on the implementation for question 3. I worked on the job_seeker and Conner worked on the job_creator. Working in Python gave a nice memory refresh for the language.

Rebecca:

All group members worked together to help complete the second part of the assignment. Tools such as Google Docs, Skype, GitHub and Discord were used so we could all work together on the assignment at the same time from different locations. Julia and I were able to collaborate on 1.2, 1.3, and 1.4 to assure our answer is clear and valid.

Conner:

I worked on the majority of the implementation. Julia and I worked together with job_seeker and job_creator, I went through everything and made sure there were no bugs with our protocol. I created conditions and reviewed message formats to make sure responses were being received correctly on both ends (client and server). I also answered questions 1.1 and question 2. Question 1.1 was helpful in our implementation as well as I clearly set out what we wanted it to look like. Question 2 tied our protocol together covering all our bases.