EE/ CprE/ SE 492 - sddec23-17

Simulated Design of Quantum Networks

Biweekly Status Report

October 14 - October 25

Client: Dr. Durga Paudyal

Faculty Advisor: Dr. Durga Paudyal

Team Members:

Benjamin Amick - Network security engineer

Derrick Wright - System integration engineer

Ohik Kwon- System component designer

Steven Tompary- Network engineer

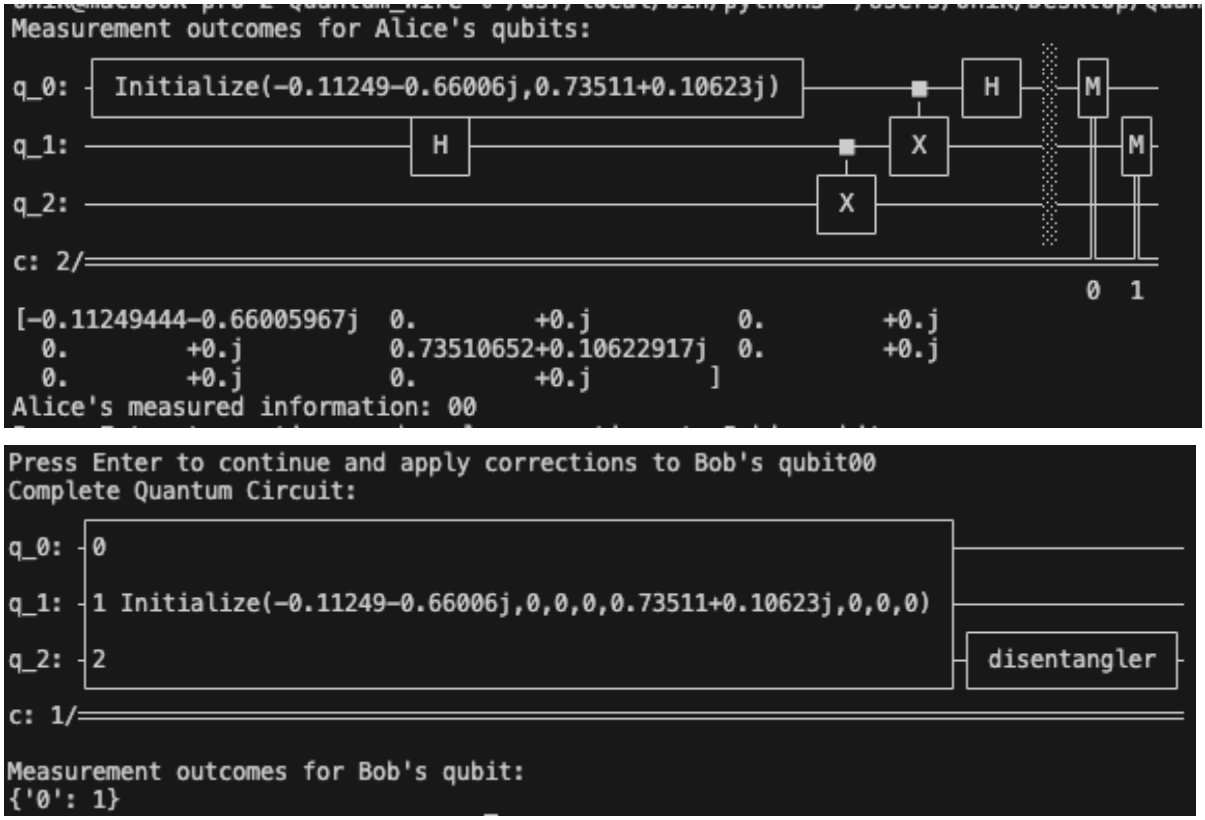
Past Week Accomplishments

We have worked on our first iteration for nearly two weeks. This version includes the creation of both quantum connections and classical connections. Most of the parts are done. Classical networks were checked that it is working stabley using TCP/IP socket. Our quantum part struggled quite a bit to handle the issues that qiskit, the quantum simulation tool, kept deleting quantum information when we tried to run separately. Now, we figured out that problem by using the initialization of the quantum circuit whenever we extract quantum information and input into our next quantum circuit. Below images are part of our classical network code and quantum simulation result for our first iteration plan. This week we will focus on integration solely, and then we will move onto the second iteration which are 1) handling multi node communication and 2) quantum wire with entanglement verification protocol.

```python
import socket
import threading

class Router:
    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def bind(self):
        self.serverSocket.bind((self.host, self.port))
        self.serverSocket.listen(5)

    def run(self):
        print ('Waiting for a connection')
        self.serverSocket.listen(5)
        (self.clientSocket, addr) = self.serverSocket.accept()
        print ('Got a connection from {}'.format(str(addr)))

        t1 = threading.Thread(target=self.recieveMessage,)
        t1.start()


    def sendMessage(self, message):
        self.clientSocket.send(message.encode('ascii'))

    def recieveMessage(self):
        while(1):
            (self.clientSocket, addr) = self.serverSocket.accept()
            message = self.clientSocket.recv(1024).decode('ascii')
            print(message)

            if message == "quit":
                self.closeSocket()
                t1.join()

    def closeSocket(self):
        self.serverSocket.close()

#
'''
USE ORDER
1 make object
2 bind()
3 run ()
4 recieve()
5 close()

'''
```

```python
import qiskit  # Ensure you have Qiskit installed
import socket
import threading

class nodeComputer:
    def __init__(self, router_host, router_port):
        self.router_host = router_host
        self.router_port = router_port
        self.serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def connect(self):
        self.serverSocket = socket.socket()
        self.serverSocket.connect((self.router_host, self.router_port))

        t1 = threading.Thread(target=self.recieveMessage,)
        t1.start()

    def recieveMessage(self):
        message = self.serverSocket.recv(1024).decode('ascii')
        print(message)
        if message == "quit":
            self.closeSocket()

    def sendMessage(self, message):
        self.serverSocket.send(message.encode('ascii'))
        if message == "quit":
            self.closeSocket()
            #t1.join()

    def closeSocket(self):
        self.serverSocket.close()
```

[figure: Classical network code, Router (left) and Node (right)]



-Quantum information successfully teleported from Alice to Bob.
-Next, we will try to implement entanglement verification code.

[figure: Quantum wire simulation results]

- **Ben** - Worked with Steven to build classical network protocol and finished to make our first primitive classical network.
- **Ohik** - Worked to simulate quantum networks by parts. Evaluated that it is now fully working.

- **Steven** - Worked with Steven to build classical network protocol and finished to make our first primitive classical network.

- **Derrick** - Made visualization movies and images for preparing the final presentation.

Resources

Our git repository
https://github.com/Kcops11/SeniorDesignQuantum17

Books we are reading

- Quantum Computation and Quantum Information, Michael A. Nielson

Articles we found this week and reading

- Github Qiskit Community Tutorials
- When Entanglement meets Classical Communications: Quantum Teleportation for the Quantum Internet, IEEE Transactions on Communication, 2020, 10.1109/TCOMM.2020.2978071
- The controlled SWAP test for determining quantum entanglement, Quantum Science and Technology, 2021, *https://doi.org/10.1088/2058-9565/abe458*

Pending Issues

- There are no pending issues for this week since we all agreed on detailed functionalities of our first iteration network.
- We will proceed to the integration phase. That will be our main focus for a couple weeks.

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours
Benjamin Amick	Worked on classical network coding	6	66
Derrick Wright	Made visualization tool for presentation	6	66
Ohik Kwon	Worked on quantum teleportation simulation coding	6	66
Steven Tompary	Worked on classical network coding	6	66

Plans for Coming Week

- Continue to work on the communication between router and nodes then ensure that they can communicate quantum instructions.
- After the integration part, we will focus on 1) handling multi node communication and 2) quantum wire with entanglement verification protocol.