Simulated Design of Quantum networks

Senior Design team sddec 23-17

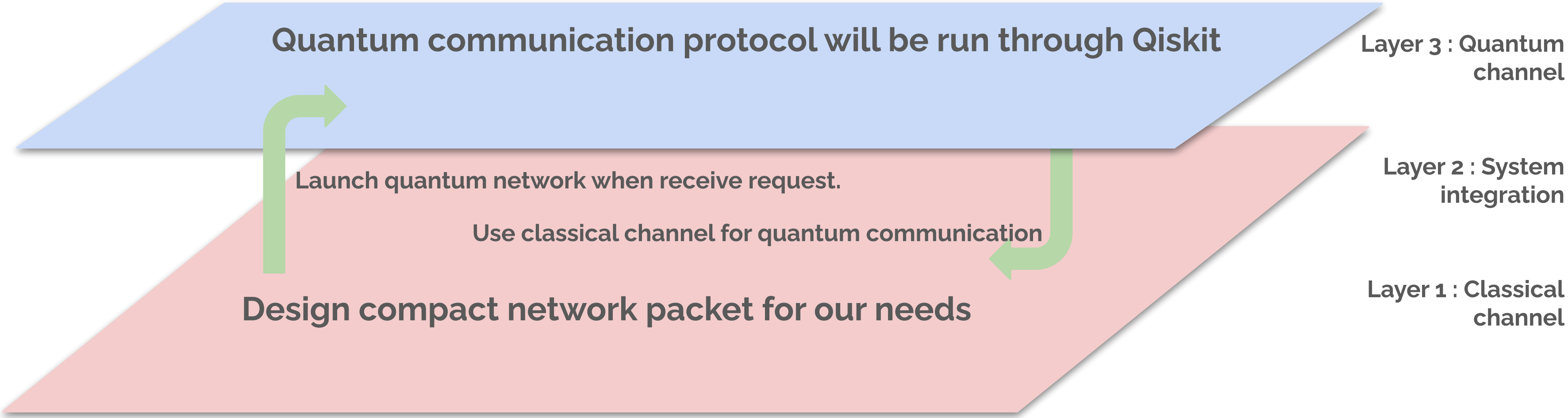
Benjamin Amick (Team leader / Network engineer) CYBE
Ohik Kwon (Quantum information system engineer) EE
Steven Tompary (Network engineer) CYBE
Derrick Wright (System integration engineer) SE

Team leader email : bcamick@iastate.edu

Contents

- Brief Quantum Network design explanation
- Progress on a quantum network
- Progress on a classical network
- Progress on a system integration
- The next goal for our network | technical challenges
    - Quantum
    - Classical
    - System Integration

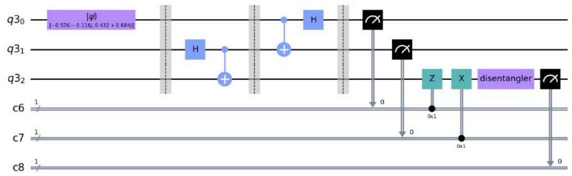
Quantum Network Design explanation



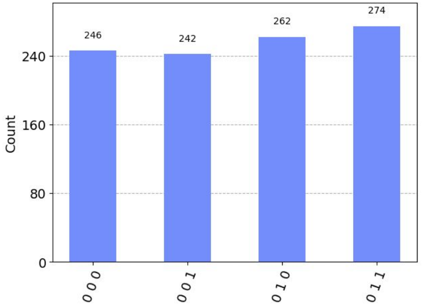
Progress on a quantum network

- We want to transport Qbit without break!
  - Break Qbit -> loose information
- Use well known Quantum Teleportation Protocol
  - Proposed by Charles H. Bennett, in 1993
- Implemented in Qiskit and verified it is working.
- Entangled state (Bell state) is required for the communication.

```
#add a new classical register to see the result
cr_result = ClassicalRegister(1)
qc.add_register(cr_result)
qc.measure(2,2)
qc.draw(output='mpl')
```




```
backend = BasicAer.get_backend('qasm_simulator')
counts = execute(qc, backend, shots=1024).result().get_counts()
plot_histogram(counts)
```



Above plot should be closed to 0.25% 0.25% 0.25% 0.25%

Progress on a quantum network

- How can we check the entanglement?
  - Use controlled SWAP gate protocol.
  - Proposed in “*The controlled SWAP test for determining quantum entanglement.”, 2021, Quantum Sci. Technol. 6*
  - It is not the most efficient, but feasible way to verify entanglement status.
  - Now implementing.

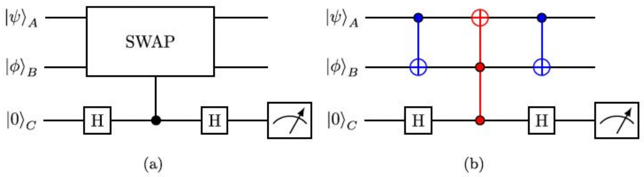


Figure 1. The quantum circuit for an equivalency SWAP test on the two states $|\psi\rangle$ and $|\phi\rangle$. H is a Hadamard gate from equation (7). (a) The SWAP gate swaps all qubits in the test states on the condition that the control qubit is in state $|1\rangle$. (b) Shows the SWAP gate broken down into individual gates for the one-qubit test state case. The central gate, shown in red, is a Toffoli gate from equation (9) and the two gates either side in blue are CNOT gates from equation (8), where the crossed circles are controlled on the dots. The final CNOT gate—not necessary for the test outcome—returns the system to its initial state in the case of equivalent states.

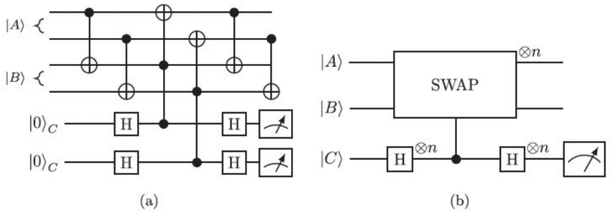


Figure 2. The quantum circuit used to carry out an SWAP test for entanglement on test state $|A\rangle$ and copy state $|B\rangle$. H denotes a Hadamard gate. Initially, $|C\rangle = |0\rangle_C^n$. (a) Shows the SWAP gate broken down into individual gates in the case of a two-qubit test state, composed of CNOT gates and Toffoli gates. The final two CNOT gates are to return the test and copy states to their original states (in some cases) and so are optional. (b) Shows the circuit for an n-qubit test state in compact form.

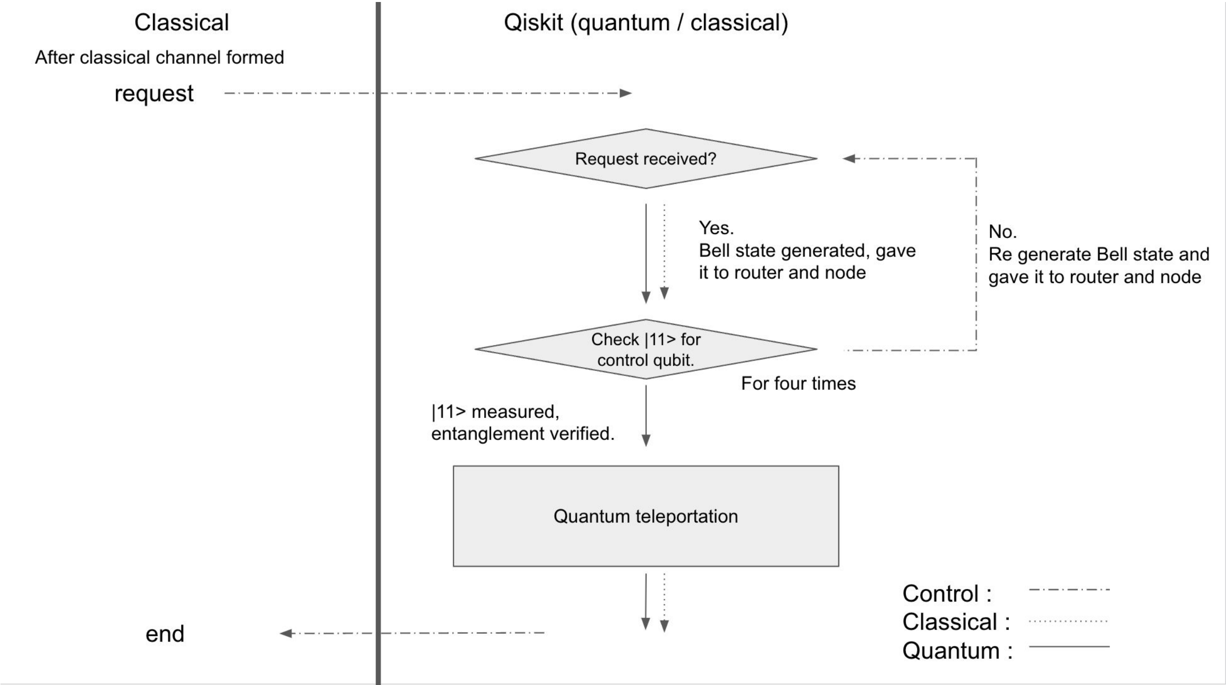
with the $\pm$s being $+$ in the case that $A_{00}A_{11} > A_{01}A_{10}$ and $-$ if $A_{00}A_{11} < A_{01}A_{10}$. If the system is in a product state then $C_2 = 0$; applying this to the above equation gives

$$|\Psi\rangle = |A\rangle_A |A\rangle_B |00\rangle_C \quad (11)$$

and so the control state is $|00\rangle_C$ with certainty. Any measurement of $|11\rangle$ for the control qubits therefore proves a non-zero concurrence, and evidences the presence of entanglement [7] in state $|A\rangle$.

Note that if the test state is a product state, and only then, the final state is the same as the initial state. In this case, the test is non-destructive and so the output state can be used as an input state in the next test iteration.

Progress on a quantum network



Progress on a system integration

Primary system integration is still in preparation phase. Focus right now is on facilitating communication between router and cluster computer.

E.g: Custom packet is designed but needs to be formalized in code, router then needs to make use of this packet to communicate via socket, to an instance of cluster computer.

This is the major goal for us right now. Next steps include communicating quantum information and then tasking the cluster computer to do Shor's algorithm.

Progress on a classical network

ID - 3 bits	Packet type - 1 bit	Data Length - 12 bits
Data 4079 bits		

This is our outline for our classical network packet. We have included some information required by the router for determining type and length which will be critical for ordering and error correction.

The next goal for our network | Technical challenges

Quantum

Implementation of quantum algorithm

Check the availability of Quantum encoder

Move onto hardware circuit design

Integration

Modify cluster computer to utilize custom packet and socket. Return proper messages to router. Communicate quantum information, entanglement, solving Shor's algorithm.

Classical

Implementation of threads and TCP Sockets

Thank you !