Connecting two quantum cities: understanding the requirements for long-distance quantum links

Daily supervisor: Janice van Dam (j.vandam-3@tudelft.nl)
Project supervisor: Stephanie Wehner

In quantum communication, two remote parties share or exchange quantum resources in the form of qubits to perform various applications. Sharing qubits between distant parties is done using photons. Sharing photonic qubits is difficult: photons get lost easily and we have to deal with the no-cloning theorem. The further two parties are apart, the harder it is to transmit the qubits between them.
The two nodes involved can be either a quantum computer or a photonic client – a 'quantum-light' device which is cheaper than a full quantum computer but has less capabilities. This gives us three distinct scenarios:

- Client – client connections, allowing for quantum key distribution [1]
- Client – quantum computer connections, allowing for blind quantum computing [2]
- Quantum computer – quantum computer connections, allowing for quantum coordination games [3]

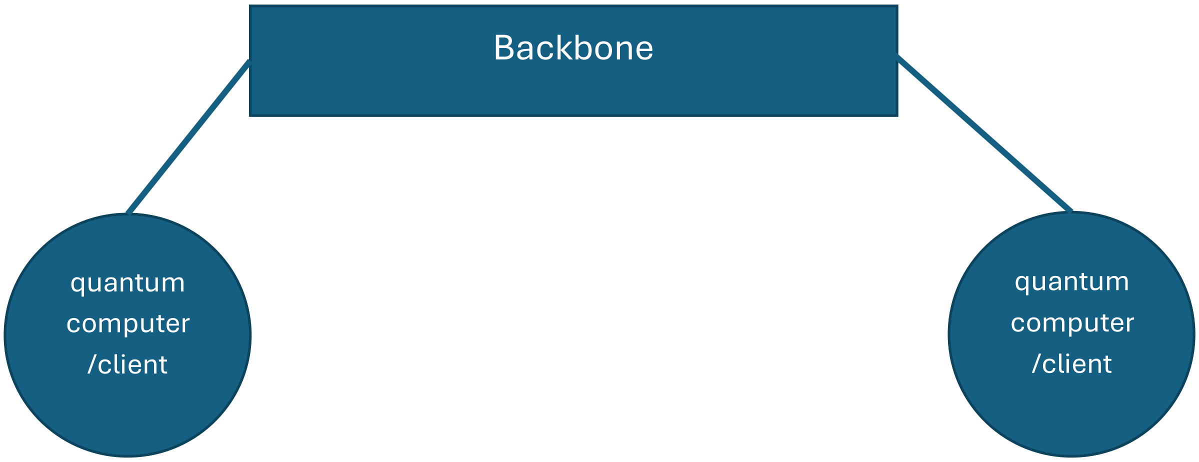

Figure 1: Setup sketch. Two end nodes that can be either a quantum computer or photonic client, connected by a backbone long-distance link.

Each of these scenarios has a certain target that we need to hit to be able to use them, putting requirements on the quantum link between the nodes.
In this project, we will work on putting up a simulation of these scenarios using the quantum network simulator NetSquid [4] in Python. We will then work on integrating the simulation with optimization tools to optimize the link between the nodes to find the minimal requirements to hit the targets for each scenario. In addition to this, we can ask various questions such as:

- How do the requirements for each scenario compare?
- How do the requirements change if we can 'buffer' entanglement?
- What are the requirements to do all of the applications in one setup?
- How does this scale with certain network parameters and the hardware capabilities of the end nodes?

Learning Goals:

- Obtain a good understanding of applications in the scenarios;
- Formulate research directions, either from the examples listed above, or we can come up with interesting (extra) questions to investigate!
- Create simulation code using specific software packages and pre-existing code snippets;
- Run optimization of the simulation on a high-performance computing cluster;
- Analyze the results to find the requirements for the different scenarios.

Output:

- An answer to the formulated research question(s);
- Effecitve communication of the results, both verbally and written.

Requirements:

At the start of the project, you should be familiar with Python and have basic knowledge of quantum mechanics.

Experience with git and Linux is appreciated but not required. Ideally, one might have followed the course 'Quantum communication and cryptography'.

References

[1] Bennett, C. H., & Brassard, G. (2014). Quantum cryptography: Public key distribution and coin tossing. *Theoretical computer science*, *560*, 7-11

[2] Broadbent A, Fitzsimons J and Kashefi E 2009 Universal blind quantum computation 2009 50th Annual IEEE Symp. on Foundations of Computer Science (IEEE) pp 517–26

[3] Kim, G., & Nho, E. W. (2019). A review of quantum games. *Journal of Young Investigators*, *37*(2).

[4] The Network Simulator for Quantum Information using Discrete events. NetSquid. https://netsquid.org/