2024 43rd International Symposium on Reliable Distributed Systems (SRDS)

PhD Forum: Evaluating and Designing Routing Protocols for Reliable Distributed Quantum Systems

Huzaifah Nadeem
University of Pittsburgh
Pittsburgh, PA, USA
h.nadeem@pitt.edu

***Abstract*—As quantum computers and quantum networks become more capable, it allows for the possibility to use multiple quantum computers as part of a distributed system. Such a distributed system will have applications such as running quantum algorithms in a distributed fashion [7], leadership elections without communication [7], clock synchronisation [10], and a cryptographic protocol called quantum key distribution [1], etc. However, to create a reliable system, it is important that the underlying quantum network can reliably support the necessary communication protocols between quantum nodes. To support reliable communication in a quantum network, quantum routing algorithms – which are algorithms that plan out how to send qubits and over which subset of nodes in the network – is an active area of research. However, various design choices and algorithms exist that are developed without a realistic simulated environment that can provide rich data that can further help develop, improve, and test these algorithms. Such an environment is important because general-purpose algorithms need to take into account detailed information about how the qubit changes over the communication process – as that requirement can vary for different applications – rather than simpler probability based loss models existing algorithms were developed with. Using existing lower-level tools, we are creating a tool will help evaluate and design routing protocol by providing a variety of models – from simple to complex – to support various stages of development, as well as the ability to generate useful data to evaluate how the protocols perform in a realistic quantum network.**

I. BACKGROUND

A. Quantum Computing

Quantum computing is different from digital computing in a few ways. A quantum computer uses a logical quantum bit or qubit, for short. These logical qubits are typically represented as being in the general form as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

$|0\rangle$ and $|1\rangle$ are analogous to the two values of a bit in digital computers. The general state of a qubit essentially implies that a qubit is in a superposition of the two states, $|0\rangle$ and $|1\rangle$. When a special operation called a measurement is performed, the qubit's state will collapse to either the state $|0\rangle$ or $|1\rangle$ with probabilities $|\alpha^2|$ and $|\beta^2|$, respectively [12]. This means that during a calculation, a qubit remains in such a state of superposition and any operations being performed on the qubit will change the values of α and β, with certain constraints such as $|\alpha^2| + |\beta^2| = 1$. The operations are performed using quantum gates which are analogous to digital logic gates. A few quantum gates can be used in various combinations to create more complicated operations [12]. While the physical realization of a digital bit is almost always the flow of current in electronic systems and light beams in optical systems, the physical realization of a qubit can be done in various different ways with their own pros and cons. While a common way is to use photons [9] – as photons are easier to transmit between computers e.g. through optic fibre cables – there are other possibilities too such as using trapped ions [3], Helium's electrons [14], and molecular magnets [11] among others. However, this detail is abstracted away at the level of quantum computers.

B. Quantum Networking

There are different kinds of quantum networks in terms of the underlying technology that is being used. For example, one common way to create quantum networks is to use photons to represent the qubit over the network [9]. In order for a quantum computer to send some data to another computer, it can directly send a qubit to the other computer, such as in the "tell-and-go" approach in [19]. However, there are a few issues that are not the case for digital computers. First, the quality of the state of the qubit decreases exponentially over the length of the connection [13]. Because of this, modern quantum networks are generally limited to being under 100 km [18]. There have been attempts at using satellites, such as in [18], to cover larger distances close to 1200 km. However, that is still much shorter than what digital networks can achieve. Another major difference is that a qubit's state cannot be cloned [17]. Therefore, if a qubit is lost over a channel, there is no way to send another copy. Therefore, techniques similar to packet switching cannot be used for qubits, where approaches such as store-and-forward by nodes on the network are used.

Because of such considerations, directly sending qubits over a channel is not the main way quantum networks work. Instead, a qubit's state is teleported to the other quantum computer, such as in [19, 15, 13, 16]. Note that we are not teleporting the qubit itself but rather its state. As part of the teleportation process, a measurement operation has to be performed on the original qubit (i.e. the 'data qubit'), due to which the original qubit's state collapses, and so we are not violating the no-cloning theorem. However, the teleportation process requires some setting up for it to work. To facilitate

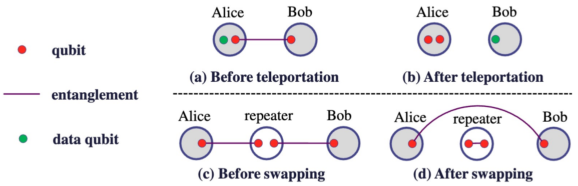
2575-8462/24/$31.00 ©2024 IEEE
DOI 10.1109/SRDS64841.2024.00041

teleportation of the data qubit's state, a special pair of qubits – called an EPR pair – is used. These qubits are entangled with each other, which means that there is a mathematical correlation between the two qubits' state. These EPR pairs can be considered a resource that is needed for the teleportation to take place. An example is shown in Fig. 1(a) and 1(b). By utilizing the red entangled pair, i.e. the EPR pair, the green data qubit was teleported from Alice to Bob.

Another important technique that is needed for the teleportation process to send a qubit's state between nodes that are not directly connected, is what is called a 'swap' operation. If a node in a network shares a pair each with two of its neighbours, it can perform a swap operation which would result in its neighbours sharing a pair afterwards. For example, in Fig. 1(c) and 1(d), after the 'repeater' node performs the swap operation, nodes Alice and Bob end up sharing an EPR pair. Using a sequence of these swap operations, an end-to-end EPR pair can be formed which is shared by the source and the destination node in the network. Once this end-to-end pair is shared, the source can use its EPR qubit, and the data qubit, together to perform teleportation procedure which would result in 2 digital bits. These digital bits are sent over a digital network to the destination node which then, based on the values of these bits, performs a certain operation which gives it the state of the data qubit. This is how quantum networks send data [19, 15, 13, 16, 7].

Akin to digital networks, a quantum network stack has been developed [6], which consists of a transport layer, a network layer, a link layer, and a physical layer. The transport layer provides reliable transmission of qubits to the application. The network layer is responsible for establishing *end-to-end EPR pairs*. The link layer provides EPR pairs to the network layer. Lastly, the physical layer is actually responsible for generating these EPR pairs. We are concerned with the network layer of the network stack and concerns such as what the underlying network technology is, e.g. using photons [9], is of concern to lower layers.

The details of how the EPR pairs are distributed over the network, and the sequence of swap operations are what quantum routing algorithms are concerned with. There are complications such as the fact that EPR pairs have very short lifetimes, that there is a metric called fidelity associated with qubits (essentially a metric to denote its quality), and that quantum operations are probabilistic. These issues introduce various trade-offs, and different routing approaches can be used depending on what the end goal is.



(a) Before teleportation (b) After teleportation

(c) Before swapping (d) After swapping

Fig. 1. Example showing teleportation. Reproduced from [16].

II. PLANNED CONTRIBUTION AND RELEVANCE

A distributed quantum system requires research in areas such as development of quantum computers, quantum networks, and distributed quantum applications. Our work focuses on routing algorithms for quantum networks. Routing algorithms for quantum networks are algorithms that are used to distribute and use EPR pairs to create end-to-end EPR pairs so that qubits can be teleported. Routing algorithms can take into account factors such as the network topology, the demands between different source-destination pairs, the fidelity requirements for an end-to-end EPR pair, any network contention, existence and usage of storage nodes (i.e. special nodes that can hold an EPR pair longer than other nodes), time constraints and scheduling, etc, as in [13, 16, 8].

Our goal is to create a tool to help develop and evaluate routing protocols for a quantum network. The evaluation can be in terms of various metrics that can be tailored to a specific type of a distributed system. With our tool, the developers for these routing algorithms would be able to use a variety of models to specify what their assumptions about the network are, and test the algorithm accordingly. Since these algorithms are sometimes developed in an iterative fashion, initially they can use simpler models, such as probabilistic loss models, as a proof-of-concept, and later on the algorithm can be further developed and evaluated using a more realistic model, such as one where the distances between nodes are used to calculate qubit loss and noise. The tool will allow for the developers to write their code without the learning curve of using lower-level tools such as NetSquid [4].

III. STATE OF THE ART AND PROBLEM DOMAIN

Multiple quantum routing algorithms exist – such as the algorithms in [13], [16], [8], and [2]; however, each of them has been developed and evaluated with different assumptions about the underlying network technology, structure and the model of the network, and traffic demands, etc. This makes it challenging to compare the algorithms with each other. Moreover, certain assumptions they are making are not always realistic e.g. [13] ignores whether or not a qubit's state will decohere as it goes through a certain path, and it evaluates the algorithms on an infinitely large grid. [16] ignores a qubit's priority so it is possible that in a realistic situation a qubit is ignored and is not transmitted in a certain time window. It is also important to realize that what counts as realistic may change as the technology develops and therefore, having a tool such as ours is useful to evaluate and update the protocols with the technology. Therefore, having a tool to evaluate these algorithms in a realistic simulated scenario would be useful for the overall development of reliable quantum distributed systems.

Some of these algorithms such as [16] also developed their own simulators but in all cases that we have encountered, the simulators are tailored to a specific proof-of-concept evaluation, and lack lower-level models to specify important details of the networks. For instance, it is important to specify what noise model is used when a qubit travels through a quantum channel, how a qubit's state is affected while it is being stored in a quantum storage, and how an operation affects the state, etc. [16] also assumes a specific structure to the routing algorithm which makes it hard to use for diverging ideas in how a routing algorithm would look like. Therefore, existing simulators would require major modifications to fit our goals.

NetSquid [4] is lower-level simulator that can be used to develop protocols for the physical and link layer. Therefore, it includes realistic networking, memory, and quantum processor models to realistically change the qubit's state and introduce noise as it is being used in the network. However, NetSquid lacks higher level abstractions to make it easy to use for quantum routing algorithms.

IV. ORIGINALITY AND RELATED WORK

Work on developing a quantum network stack and been done and well received such as in [6]. Moreover, physicists and engineers have been working on reliable physical layer and link level protocols and systems. Simulators for various purposes have also been developed. However, all these simulators are created to serve very specific purposes. For instance, some of the simulators are designed to verify whether and how well a routing algorithm works in a given situation, however, they are not designed to be modular and are quite static.

Our tool would have the capability to use a wide variety of inputs such as using any network topology, using a number of lower level models from [4] for quantum memories, processors, and channels, what metrics are to be measured, etc. NetSquid provides a good backend for our tool since the lower level models that it provides are important for routing algorithms. However, since NetSquid is a lower-level simulator, it is complicated to use in the context of routing algorithms. Our tool will help the developers to develop and test their routing algorithms with all the lower-level functionality of NetSquid.

V. RESEARCH QUESTIONS AND MAIN CHALLENGES

The main research question is whether having such a tool can provide more data towards creating reliable routing protocols. Once we have the tool ready, it will contribute towards understanding how to design routing algorithms that will perform well in real quantum networks. As a first step towards that, we will re-evaluate existing algorithms under more realistic network settings and with respect to metrics they did not consider.

To start with, the algorithms in [13] have been written in our tool while the algorithms in [16] are being developed. Using these implementations, we would like to answer questions such as, but not limited to:

- How does [13] work on non-grid topologies? This is important to understand since it does claim that the algorithm should work on any topology. However, it has only been evaluated on a grid. Moreover, the algorithm seems to be made specifically for grid with same edge lengths. We would like to test it with different assumptions.
- What is the effect of the average degree of a node for the QPASS algorithm in [16]? The algorithm uses a heuristic based approach to route qubits and having nodes with high degree but less reliability, in terms of the noise they introduce, might make the algorithm's performance worse.
- What is average consumption of EPR pairs for [16] and [13]. There should be some cost associated with generating EPR pairs and taking that into account would be important for cost-effective quantum networks and systems.
- What is the average fidelity of paths that [16] and [13] are routing over? Fidelity requirements can vary for different applications and they do not take that into account.

Running the algorithms in [13] in a more realistic simulation have yielded interesting results and imply more consideration should be given to the details we are putting at the center of our tool. As an example, the performance results in that work were generated using a numerical approach. By using our tool, it was more clear that certain details were missed such as how big the underlying network actually is. Their results were based on the assumption of having an infinitely large grid network which is not practical. We confirmed their results with some caveats using our tool.

Some of the challenges are that there are quite a few design decisions that can be taken to make it more suitable for a certain type of algorithms. However, to keep it useful for as many algorithms as possible, the interface to plug in an algorithm into the tool has to be very generalized. We expect to provide examples and classes that can be used and extended to make the process as simple as possible but it would be challenge to keep things simple while supporting a wide variety of algorithms.

VI. RESEARCH METHODOLOGY AND OVERALL APPROACH

The tool will use NetSquid [4] for its backend and the front end is being developed in Python. The relevant Octave functions and models from [5] will also be supported. NetSquid will provide us with its extensive functions and models of the underlying quantum networks. This includes models such as for how a qubit decoheres or loses its quality as it travels through a channel, how a qubit is affected by noise while it is being held in a quantum memory, and how a quantum operation might introduce noise into the qubits state. These lower level functions will be used by the tool and it will provide a higher level interface for the user. This means that a user would write the algorithm that calls functions such as "send_qubit()" and the tool will use NetSquid's objects to forward a qubit to the specified node while applying the appropriate noise to it, as specified in the network configuration, and the qubit will be put in the receiving node's quantum memory. If the user were to directly do this on NetSquid, they would have to set up objects for the channels that receives and forwards a qubit using a noise model, as well as setting up callback functions regarding what to do with a received qubit.

We will also write extensive documentation, which existing simulators tend to miss, and make the tool completely open-source and free to use and modify.

REFERENCES

[1] Charles H. Bennett and Gilles Brassard. “Quantum cryptography: Public key distribution and coin tossing”. In: *Theoretical Computer Science* 560 (2014). Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84, pp. 7–11. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2014.05.025. URL: https://www.sciencedirect.com/science/article/pii/S0304397514004241.

[2] Kaushik Chakraborty et al. *Distributed Routing in a Quantum Internet*. 2019. eprint: arXiv:1907.11630.

[3] J. I. Cirac and P. Zoller. “Quantum Computations with Cold Trapped Ions”. In: *Phys. Rev. Lett.* 74 (20 May 1995), pp. 4091–4094. DOI: 10.1103/PhysRevLett.74.4091. URL: https://link.aps.org/doi/10.1103/PhysRevLett.74.4091.

[4] Tim Coopmans et al. “NetSquid, a NETwork Simulator for QUantum Information using Discrete events”. In: *Communications Physics* 4.1 (July 2021), p. 164. ISSN: 2399-3650. DOI: 10.1038/s42005-021-00647-8. URL: https://doi.org/10.1038/s42005-021-00647-8.

[5] Toby Cubitt. *Quantum Information Package*. URL: https://www.dr-qubit.org/matlab.html.

[6] Axel Dahlberg et al. “A link layer protocol for quantum networks”. In: *Proceedings of the ACM Special Interest Group on Data Communication*. SIGCOMM ’19. Beijing, China: Association for Computing Machinery, 2019, pp. 159–173. ISBN: 9781450359566. DOI: 10.1145/3341302.3342070. URL: https://doi.org/10.1145/3341302.3342070.

[7] Vasil S. Denchev and Gopal Pandurangan. “Distributed quantum computing: a new frontier in distributed systems or science fiction?” In: *SIGACT News* 39.3 (Sept. 2008), pp. 77–95. ISSN: 0163-5700. DOI: 10.1145/1412700.1412718. URL: https://doi.org/10.1145/1412700.1412718.

[8] Ali Farahbakhsh and Chen Feng. “Opportunistic Routing in Quantum Networks”. In: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. 2022, pp. 490–499. DOI: 10.1109/INFOCOM48880.2022.9796816.

[9] E. Knill, R. Laflamme, and G. J. Milburn. “A scheme for efficient quantum computation with linear optics”. In: *Nature* 409.6816 (Jan. 2001), pp. 46–52. ISSN: 1476-4687. DOI: 10.1038/35051009. URL: https://doi.org/10.1038/35051009.

[10] P. Kómár et al. “A quantum network of clocks”. In: *Nature Physics* 10.8 (Aug. 2014), pp. 582–587. ISSN: 1745-2481. DOI: 10.1038/nphys3000. URL: https://doi.org/10.1038/nphys3000.

[11] Michael N. Leuenberger and Daniel Loss. “Quantum computing in molecular magnets”. In: *Nature* 410.6830 (Apr. 2001), pp. 789–793. ISSN: 1476-4687. DOI: 10.1038/35071024. URL: https://doi.org/10.1038/35071024.

[12] N. David Mermin. *Quantum Computer Science*. Cambridge, England: Cambridge University Press, Aug. 2007.

[13] Mihir Pant et al. *Routing entanglement in the Quantum internet*. Mar. 2019. URL: https://www.nature.com/articles/s41534-019-0139-x.

[14] P. M. Platzman and M. I. Dykman. “Quantum Computing with Electrons Floating on Liquid Helium”. In: *Science* 284.5422 (1999), pp. 1967–1969. DOI: 10.1126/science.284.5422.1967. eprint: https://www.science.org/doi/pdf/10.1126/science.284.5422.1967. URL: https://www.science.org/doi/abs/10.1126/science.284.5422.1967.

[15] Shahrooz Pouryousef, Nitish K. Panigrahy, and Don Towsley. “A Quantum Overlay Network for Efficient Entanglement Distribution”. In: *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. 2023, pp. 1–10. DOI: 10.1109/INFOCOM53939.2023.10228944.

[16] Shouqian Shi and Chen Qian. “Concurrent Entanglement Routing for Quantum Networks: Model and Designs”. In: *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*. SIGCOMM ’20. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 62–75. ISBN: 9781450379557. DOI: 10.1145/3387514.3405853. URL: https://doi.org/10.1145/3387514.3405853.

[17] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned”. In: *Nature* 299.5886 (Oct. 1982), pp. 802–803. ISSN: 1476-4687. DOI: 10.1038/299802a0. URL: https://doi.org/10.1038/299802a0.

[18] Juan Yin et al. “Satellite-based entanglement distribution over 1200 kilometers”. In: *Science* 356.6343 (2017), pp. 1140–1144. DOI: 10.1126/science.aan3211. eprint: https://www.science.org/doi/pdf/10.1126/science.aan3211. URL: https://www.science.org/doi/abs/10.1126/science.aan3211.

[19] Yangming Zhao and Chunming Qiao. “Distributed Transport Protocols for Quantum Data Networks”. In: *IEEE/ACM Transactions on Networking* 31.6 (2023), pp. 2777–2792. DOI: 10.1109/TNET.2023.3262547.