Simulation Tests of Tokyo Quantum Network

Osman Ceylan\(^1\) & Ihsan Yılmaz\(^2\)
\(^1\)Student of Department of Computer Engineering, Çanakkale 18 Mart University, Çanakkale, Turkey
\(^2\)Department of Computer Engineering, Çanakkale 18 Mart University, Çanakkale, Turkey
E-mail: \(^1\)osman.semi.ceylan@gmail.com
E-mail: \(^2\)iyilmaz@comu.edu.tr

Abstract. In this modern era where secure communication is imperative, scientists have always been in search of a new solutions for this problem. One of the best solutions discovered in this century, uses the superiorities of the quantum world. This theorem, called no cloning, mainly uses the collapse of quantum state when observing attribute of qubits. In this study, we will examine the Tokyo Quantum Network [1], which is based on the protocols derived from the BB84 [2] and B92 protocols by implementing in three quantum network simulators.

Keywords: quantum internet, quantum key distribution, quantum network, network simulation

1. Introduction

New studies are added every day in the field of quantum networking. Researchers have shown tremendous work, although studies that have applied on a large scale quantum network applications have less frequent. In this context, QKD networks are also currently possible for long-distance [3–5]. Nowadays QKD networks reach the distance of 500 km has on fibre based channels [6]. Some countries such as China [7, 8], European countries [9–11] and America [12] invests heavily in the quantum network field. It seems difficult to do such complex studies without simulation. So this reason push us to research network simulations of quantum networks. In this study, we will examine Tokyo Quantum Network more closely and run some tests with three different network simulation software named SQUANCH [13], QuNetSim [14] and SimulaQron [15]. The Tokyo Quantum Network is seen in the Figure 1.
2. Tokyo Quantum Network

The Tokyo Quantum Network [1] is one of the principles of large-scale studies in this field. Quantum network was placed in 4 districts of Tokyo, security was always at the forefront, but specific studies were also made for other qualities such as speed communication. Different protocols were tested on different hardware by placing one or more stations in Koganei, Otemachi, Hakusan and Hongo districts in Tokyo. The network has a total of 6 stations, there is a classical network between the devices with the same hardware that connects each station to each other with quantum channels and the whole station can access each other. Along with different organizations such as the Austrian Institute of Technology and Mitsubishi, different qualities of secure communication with quantum technologies were presented at these stations. Some of these stations use the simpler BB84 protocol, while some others use more complex protocols like differential phase shifted quantum key distribution (DPS-QKD) in NTT-NICT machines.

3. Simulation Software Evaluation

Although there are more simulation software available than the original, we chose the best three due to their implementation limitations on this specific work. As we mention before these software or more like frameworks are SimulaQron [15], SQUANCH [13] ve QuNetSim [14]. They are all coded in Python 3.5+.
3.1. SimulaQron

SimulaQron is the only software among them that performs socket simulation using ports. But because of this feature make framework works slower. Although it leaves a lot of work that should be done in a network to the user, SimulaQron is the most professional coded work among them.

3.2. SQUANCH

SQUANCH probably is the simplest coded among them since it do not even support multiprocessing capabilities. But it works faster on networks with few nodes because it is simple. When node count on network goes beyond 10 SQUANCH performs slower then the others in same task. So SQUANCH depends heavily on host computer raw single core performance. But SQUACNH supports simple error patterns which make him more real simulation.

3.3. QuNetSim

We found QuNetSim for user appeal then others. Using QuNetSim framework much easier then the others. The main purpose of QuNetSim is to allow the user to focus on protocol generation as its developers mentioned. While it’s not convenient for us that QuNetSim doesn’t support noise patterns, some manual operations are not inoperable. unfortunately QuNetSim not assertive in terms of speed.

3.4. Software Implementation

This section is about how we code Tokyo Quantum Network in the mentioned simulation software. As we must mention this first, we did not include NTT-NICT and IDO stations due to the lack of common protocols on simulation software.

As for start from SimulaQron, we treat all stations as a device object and add all stations in the Tokyo Quantum Network. The methods included in the SimlaQron were not used for the classical network. For this job, a process that connects each device and runs in a separate process and is also used as a server for key distribution when needed. Again, this process handles routing when needed. In addition, we have created a monitor process to transfer instant and simulation data. As we mentioned before, SimulaQron expects to user to handle the must have actions in network.

Coding in SQUANCH more less complex than the others because of data locality. But it has the same shortcomings about the must have actions. With the using queues and more threads all stations coded in SQUANCH.

We coded the network on QuNetSim with already existing functions. But we coded the needed monitor process ourself.
4. Test of the Network

In this section, we share test results of network test among frameworks. Our tests involves around Koganei-1, Koganei-2 and Otemachi-1 stations as are fidelity test of Tokyo Quantum Network. We exchanged data between the stations mentioned in our test. Before we start we run all test in high performance AMD Threadripper 3950X CPU in Linux 5+ kernel environment with using Python version 3.7. All frameworks updated to most recent version as of May 2021 and these tests did not run at same time.

![Figure 2: Successful 128-bit QKD counts among frameworks as time passes.](image1)

![Figure 3: Successful 256-bit QKD counts among frameworks as time passes.](image2)
5. Conclusion

The first we tested was total QKD counts on simulation process. If we look at the Figure 2, we see that SQUANCH is much faster. This happens most likely allocating quantum resources on a computer way faster then the others. Again this most likely because of the data locality coded in SQUANCH that lack of multiprocessing. The others performed identical. We believe the reason is they are using same backend called ProjectQ [16].

The next Figure 3 which presents the same test results as before is counts of successful QKD in whole simulation process. This 256-bit test hits hard to especially...
QuNetSim. QuNetSim could not even hold 1 QKD/sec. sometimes. While we observed SimulaQron and QuNetSim performs identical, QuNetSim is stays behind most time. The reason for this maybe QuNetSim support of extra services like routing and acknowledge checks.

The next Figure 4 represents the final key length in simulation process. We observed the same act of QuNetSim and SimulaQron in this test too. As we examined further, SQUANCH yielded expected 256-bit results while the others did not. We found this quite interesting because the CPU we used had a high core count and we expected multiprocess supported frameworks should work better. But that was not in the case. Meanwhile the single process framework SQUANCH handled far better then the other. The reason for this most likely related with pre-allocation on quantum resources.

The final Figure 5 is about error rates between sended and received message. Only SQUANCH supports errors in simulation, so we continue with it. We The total quantum channel length for this test was $45 + 24 = 79\text{ km}$. We used a special pattern to make the noise more computable. The relative result of 128-bit and 256-bit was expected.

As we investigate more Python multiprocessing and multithreading performance is far from ideal expectations, at least for simulation software that mimics quantum resources. But most of these kind of network simulation frameworks, maybe all, written by Python.

With this study, we tested the capacities and reality limits of simulation software. Such framework like the SimulaQron be able to distribute simulation targets to other computers to creates an opportunity to analysis communication. Like the QuNetSim framework, it can help us to understand more about quantum communication protocols. Such as framework the SQUANCH helps us to calculate errors in quantum memory and channels. While all these frameworks abstract most of the complexity of real quantum networks and run too much slow then a real one but we find these software are essential for who wants to built such a quantum network.

Acknowledgments

This work was supported by TUBITAK under the agreement no: 120E087.

References

[1] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka et al. “Field test of quantum key distribution in the tokyo qkd network,” Optics express, vol. 19, no. 11, pp. 10 387–10 409, 2011.

[2] C.H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner. ‘Quantum Cryptography, or Unforgeable Subway Tokens’, to appear in Advances in Cryptography: Proceedings of CRYPTO82, Plenum Press. [These CRYPTO82 Proceedings were published in 1983 and this paper was on pages267–275.]

[3] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boscheitner, T. Debuisseret, E. Diamanti, M. Dianati, J. Dynes et al. “The secqc quantum key distribution network in vienna.”, New Journal of Physics, vol. 11, no. 7, p. 075001, 2009.

[4] D. Stucki, M. Legre, F. Buntschn, B. Clausen, N. Felber, N. Gisin, L. Henzen, P. Junod, G. Litzistorf, P. Monbaron et al. “Long-term performance of the swiss quantum quantum key
distribution network in a field environment,”. New Journal of Physics, vol. 13, no. 12, p. 123001, 2011.

[5] S. Wang, W. Chen, Z.-Q. Yin, H.-W. Li, D.-Y. He, Y.-H. Li, Z. Zhou, X.-T. Song, F.-Y. Li, D. Wang et al. “Field and long-term demonstration of a wide area quantum key distribution network,”. Optics express, vol. 22, no. 18, pp. 21 739–21 756, 2014.

[6] J.-P. Chen, C. Zhang, Y. Liu, C. Jiang, W. Zhang, X.-L. Hu, J.-Y. Guan, Z.-W. Yu, H. Xu, J. Lin et al. “Sending-or-not-sending with independent lasers: Secure twin-field quantum key distribution over 509 km.”. Physical review letters, vol. 124, no. 7, p. 070501, 2020.

[7] R. Courtland. “China’s 2,000-km quantum link is almost complete,” IEEE Spectrum, vol. 53, no. 11, pp. 11–12, 2016.

[8] J. Yin, Y. Cao, Y.-H. Li, S.-K. Liao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, B. Li, H. Dai et al. “Satellite-based entanglement distribution over 1200 kilometers.”. Science, vol. 356, no. 6343, pp. 1140–1144, 2017.

[9] E. Gibney et al. “Billion-eboost for quantum tech,”. Nature, vol. 523, p. 426, 2016.

[10] E. Cartlidge. “Europe’s €1 billion quantum flagship announces grants,”. 2018.

[11] “Quantum internet alliance,”. https://quantum-internet.team, 2018.

[12] C. Monroe, M. G. Rayner, and J. Taylor. “The us national quantum initiative: From act to action.”. Science, vol. 364, no. 6439, pp. 440–442, 2019.

[13] SQUANCH. “A distributed simulation framework for quantum networks and channels,”. B. Bartlett, arXiv: 1808.07047 [quant-ph], Aug. 2018.

[14] Stephen DiAdamo, Janis N ötzle, Benjamin Zanger, Mehmet Mert Beşe. "QuNetSim: A Software Framework for QuantumNetworks, ”. http://arxiv.org/abs/2003.06397v4

[15] Axel Dahlberg1and Stephanie Wehner. "SimulaQron - A simulator for developing quantum internet software, ”. arXiv:1712.08032v2

[16] S. Steiger, Thomas Hänner, and Matthias Troyer. ”ProjectQ: An Open Source Software Frameworkfor Quantum ComputingDamian,”. arXiv:1612.08091v2