QDNS: An Event Driven Dynamic Quantum Network Simulator

Osman Ceylan[1] **& İhsan 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 the last decade, we have seen an increase in the studies of scientists using the superiority of the quantum world compared to the classical. Using no-cloning, one of these advantages, we witness few quantum networks have been established in China, Europe and America in this period of time. But since building a quantum network is a very costly academic work, not every researcher can build one or access one. Even otherwise, simulation will be needed to build such a real quantum network. With our quantum network simulator, QDNS, we think it will help fill this gap.

Keywords: quantum internet, simulation, quantum key distribution

1. Introduction

Quantum Dynamic Network Simulator (QDNS) is a event driven quantum network simulation framework written in Python. QDNS allows users to program any node in the network and develop quantum network protocols over a dynamic and uncertain environment.

2. Ring Based Quantum Network Topology

In this study, the quantum Internet is modelled as a ring-based quantum network consisting of quantum nodes, repeaters and channels . Since the quantum network will be long distance, there should be quantum repeaters at certain intervals between the nodes. Each nodes consist quantum devices and classic controlled agent devices. These quantum devices will be able to create qubits, apply some quantum transformations and have acceptable error rates. In addition to this, classic controlled agent devices can do classical work on all nodes. All this is shown in Figure 1 below.

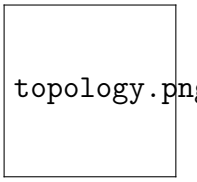


Figure 1: An example of nodes and channels in a ring-based network topology.

There is a central node, as can be seen in Figure 1. This is the zero ring. Other nodes are opened ring-by-ring on it. Each node in a ring is connected by quantum channels with the nearest nodes up to the degree of the ring. Classical channels between all nodes in the network can be arranged as desired. Let's examine how communication takes place on the proposed topology. For this purpose, let's consider a cross-section of the mesh as in Figure 2.

Figure 2: Portion of the network.

The communication between Alice and Bob over the proposed ring-based quantum network takes place as follows.

- Alice's Key Generation
- Generation of Sign as Key
- Alice Inherits Sign as Key
- Alice Prepares the Message
- Bob Handles Message
- Decryption of Message
- Return of Message Pieces

a-) Alice's Key Generation

The communication is start with generic quantum key distribution process. Alice and Bob runs an quantum key generation protocol between themselves. After a successful run of protocol both party generates the S_{AB} key. If this protocol ends with no success Alice prepares and sends an error report to center node. Table 1 is holds current knowledge of nodes.

Table 1: Knowledge table after step 1.

Alice	Bob	0	1	2	C
S_{AB}	S_{AB}	-	-	-	-

b-) Generation of Sign as Key

Alice now needs keys for encryption other than generated key. Alice request signs from the connected sub-level nodes. Sub-level nodes must holds an generated sign from center node. If these sub-level nodes grade low enough to hold original sign then they request a generated sign from center node. Else they inherit the signs from one of connected sub-level nodes. In this example "0", "1" and "2" named nodes grade is 1 which is low enough to hold original signs from center node. So they contact to center node for getting these sign. After successful quantum key distribution process between "0" to center and "1" to center S_{C0} and S_{C1} keys are generated. Center node generates unique M_0 and M_1 signs then encrypts them with related keys from QKD protocol. Center node sends $V_{C0} = S_{CO}(M_0)$ and $V_{C1} = S_{C1}(M_1)$ information to related nodes.

Table 2: Knowledge table after step 2.

Alice	Bob	0	1	2	C
S_{AB}	S_{AB}	-	-	-	-
-	-	V_{C0}	V_{C1}	-	M_0, M_1

c-) Alice Inherits Sign as Key

In this stage Alice inherit encrypted sign as keys from connected low-level nodes. These signs was generated in previous step. With the success of QKD protocol, inherit communication process happens.

Table 3: Knowledge table after step 3.

Alice	Bob	0	1	2	C
S_{AB}	S_{AB}	-	-	-	-
-	-	V_{C0}	V_{C1}	-	M_0, M_1
V_{C0}, V_{C1}	-	-	-	-	-

d-) Alice Prepares the Message

In this step Alice processes the K message she wants to send. Alice then selects an partition method B from the partition method pool which designed for the topology. Alice then divides the message as $K = K_0 + K_1 + ... + K_n$ with the selected method. n is equal to degree of the Alice in topology. Alice encrypts every piece of the message with encrypted sign as keys that she was got and translates them into $F = V(K)$. Alice then prepares package P containing partition method, connected low-level node identifications and message F. Alice encrypts the P package with S_{AB} and translate P into $P' = S_{AB}(P)$. Lastly Alice sends P' to Bob.

Table 4: Knowledge table after step 4.

Alice	Bob	0	1	2	C
S_{AB}	S_{AB}	-	-	-	-
-	-	V_{C0}	V_{C1}	-	M_0, M_1
V_{C0}, V_{C1}	-	-	-	-	-
P, K	P'	-	-	-	-

e-) Bob Handles Message

Bob runs successful QKD protocol with each of connected nodes from low-level ring after Bob decrypts P' package and extract P information. Bob sends them every piece of message F with using generated keys. If node identification in package P matches with one of node from Bob's connected nodes, Bob sends related piece of message to them. In this example Bob sends $F0$ piece to node "2" and $F1$ piece to node "1". Bob also sends them identification of nodes with related piece because target nodes needs to verify if they are bequeath of sign.

Table 5: Knowledge table after step 5.

Alice	Bob	0	1	2	C
S_{AB}	S_{AB}	-	-	-	-
-	-	V_{C0}	V_{C1}	-	M_0, M_1
V_{C0}, V_{C1}	-	-	-	-	-
P, K	P'	-	-	-	-
-	-	-	$F0$, node1	$F1$, node0	-

f-) Decryption of Message

The low-level nodes who got message from Bob tries decrypts the message. If related node identification matches with themselves they decrypts piece of message F_n with its encrypted sign as key V_n and extracts the piece of real message K_n. If match process returns false then sends them to low-level nodes until any identification match happens. At the worst case scenario piece of message reaches center node and center node decrypts the piece of massage. Since center node only one sign provider of network he knows all signs exits on whole network.

Table 6: Knowledge table after step 6.

Alice	Bob	0	1	2	C
S_{AB}	S_{AB}	-	-	-	-
-	-	V_{C0}	V_{C1}	-	M_0, M_1
V_{C0}, V_{C1}	-	-	-	-	-
P, K	P'	-	-	-	-
-	-	-	F_0, node1	F_1, node0	-
-	-	-	K_0	-	K_1

g-) Return of Message Pieces

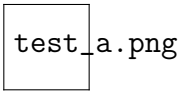
In this final step every piece of message that decrypted returns to Bob. Bob easily gather all pieces he got since he know partition method B and reaches the K message. Alice and Bob agrees upon a successful communication and reports the whole process to center node. As we demonstrated an example communication; all protocol about is splitting and encrypting message while other party tries to recover the message with the helps from other nodes.

3. Implementation of Ring-Based Quantum Network with QKD and QDS on SQUANCH

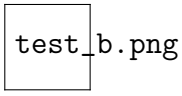
SQUANCH is simple quantum node and channel simulation framework written in Python. This framework allows us to simulate quantum networks. While SQUANCH unfortunately allows only few nodes for reasonable simulation, it sure runs fast among the other tools. We coded the same network in Figure 3 with SQUACNH and run the exact same example we gave previous section.

Table 7: Knowledge table after step 7.

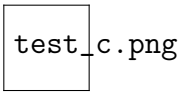
Alice	Bob	0	1	2	C
S_{AB}	S_{AB}	-	-	-	-
-	-	V_{C0}	V_{C1}	-	M_0, M_1
V_{C0}, V_{C1}	-	-	-	-	-
P, K	P'	-	-	-	-
-	-	-	F_0, node1	F_1, node0	-
-	-	-	K_0	-	K_1
-	K	-	-	-	



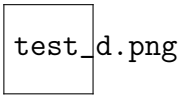
(a) Simulation consume time.



(b) Center node sign as key process counts.



(c) Error rate in different distances.



(d) QKD protocol count over communication count.

Figure 3: Various test results, Intel Core i7-10750H CPU, Python 3.7 on Linux.

4. Conclusion

In this article, we study a ring topology-based quantum network with QKD and QDS for the quantum internet. We implemented the purposed network in the SQUANCH network simulator.

We will discuss about security on the suggested topology. While all nodes must run QKD protocol for any communication but we always thought QKD is penetrable protocol. The first aim is about how to always check for observers.

In the suggested topology first two ring of network should locally close to center nodes and always supervise. We prefer other rings expand exponential in distance since we wanted to maximize security for long distance. The nodes of first two ring should able to hold original signs from center node and their count in their ring also should lower than any ring in topology.

The other topic would be about generation of signs. Center node should never generate same sign from past. Sign holder nodes should throw old signs as new request made. Center node only approver of sign other then sign holder.

The other subject we need to discuss about is partition method of messages. We imagine dynamic and huge pool for partition methods. Center node should handle all syncing these pool over whole topology. In addition to classical complexity we prefer a

dummy message to start a new communication between nodes.

The topology in our work may appear to much centralized but in the any communication other party nodes never gets the complete of message. Even in worst case scenario where center node decrypts all piece of message center node do not know partition method.

Attacker must penetrate QKD protocol and doing this for all channels of a target node. But being success with this method also drops if attacker tries to listen a high-level ring node since that node have much higher channel count. Attacker should perfectly listen $n+1$ quantum channel for a success penetration. Also success depends on attacker knows message partition method.

Acknowledgments

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

References

[1] SQUACH *Github*: https://github.com/att-innovate/squanch.

[2] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. Dynes et al. *"The secoqc quantum key distribution network in vienna,"*. New Journal of Physics, vol. 11, no. 7, p. 075001, 2009.

[3] 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.

[4] D. Stucki, M. Legre, F. Buntschu, 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] D. Lago-Rivera, S. Grandi, J. V. Rakonjac , A.Seri, H. Riedmatten. *"Telecom-heralded entanglement between multimode solid-state quantum memories, "*. Nature, 594, p.37–40, 2021.

[8] R. Courtland. *"China's 2,000-km quantum link is almost complete,"* IEEE Spectrum, vol. 53, no. 11, pp. 11–12, 2016.

[9] 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.

[10] E. Gibney et al. *"Billion-eboost for quantum tech,"*. Nature, vol. 523, p. 426, 2016.

[11] E. Cartlidge. *"Europe's e1 billion quantum flagship announces grants,"*. 2018.

[12] *"Quantum internet alliance,"* https://quantum-internet.team, 2018.

[13] C. Monroe, M. G. Raymer, and J. Taylor. *"The us national quantum initiative: From act to action,"*. Science, vol. 364, no. 6439, pp. 440–442, 2019.

[14] S. K. Joshi, J. Pienaar, T. C. Ralph, L. Cacciapuoti, W. McCutcheon, J. Rarity, D. Giggenbach, J. G. Lim, V. Makarov, I. Fuentes et al. *"Space quest mission proposal: experimentally testing decoherence due to gravity,"*. New Journal of Physics, vol. 20, no. 6, p. 063016, 2018.

[15] S.-K. Liao, H.-L. Yong, C. Liu, G.-L. Shentu, et al. *"Long-distance free-space quantum key distribution in daylight towards inter-satellite communication,"*. Nature Photonics 11, 509 (2017).

[16] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, et al. *"Satellite-to-ground quantum key distribution,"*. Nature 549, 43 (2017).

[17] H. Takenaka, A. Carrasco-Casado, M. Fujiwara, M. Kitamura, M. Sasaki, and M. Toyoshima. *"Satellite to ground quantum-limited communication using a 50-kg class microsatellite,"*. Nature Photonics 11, 502 (2017).

[18] J.-G. Ren, P. Xu, H.-L. Yong, L. Zhang, S.-K. Liao, J. Yin, W.-Y. Liu, W.-Q. Cai, M. Yang, L. Li, et al. *"Ground-to-satellite quantum teleportation,"*. Nature 549, 70 (2017).

[19] S.-K. Liao, W.-Q. Cai, J. Handsteiner, B. Liu, et al. *"Satellite-relayed intercontinental quantum network,"*. Physical Review Letters 120, 030501 (2018).

[20] L. Calderaro, C. Agnesi, D. Dequal, F. Vedovato, et al. *"Towards quantum communication from global navigation satellite system,"*. Quantum Science and Technology 4, 015012 (2018).