ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol.12 (2022) No.1, pp.283-293
DOI:10.33168/JSMS.2022.0120

A Reinforcement Learning Model for Quantum Network Data Aggregation and Analysis

Hye-jin Kim [1], S. Madhavi [2]

[1] Kookmin University, Seoul, Korea,

[2] PVP Siddhartha Institute of Technology, Andhra Pradesh, India

mukkumadhavisk@gmail.com

Abstract. Quantum Entanglement and Quantum swapping are major research areas nowadays. Remote quantum entanglement is used in many applications like secure communication, secret sharing, data aggregation, and precision sensing. In data aggregation applications, every sensor node captures data and communicates to the central node. Efficient Data aggregation depends on whether the local information or global quantum network information is used for constructing the aggregation schedules. In addition, Quantum networks suffer from lossy optical links and with limited resources such as quantum memories, edge capacities. Computation of optimal schedules deals with large quantities of data and complex time-consuming calculations. However, the quantum memories cannot hold the qubits for a longer time as the stored qubit completely decoheres an infinite amount of time. Hence, there is a necessity for finding new data aggregation scheduling protocols, which use optimal channel capacity and optimal size of memory for improving the network throughput. This paper uses a reinforcement-learning technique that considers entanglement pairing and swapping the success probability of nodes with their neighbors while finding an optimal scheduling policy. The proposed method uses local network information for constructing optimal data aggregation schedules by prior sharing the maximally entangled qubit pairs between the nodes through optimal usage of the processes, channel capacity, and memory at the intermediate nodes. Experiments show that our proposed method can maximize the network throughput.

Keywords: Reinforcement, Learning, RL, Einstein, EPR, Quantum, Network

1. Introduction

Data Aggregation is the process of aggregating data to the central node by using one or more intermediate neighbors. Data aggregation reduces the packet transmissions increasing the lifetime of the network. The base station receives the information aggregated by the nodes. In this process, entanglement with the best nearest neighbor decreases the processing time, best utilizes the available bandwidth and battery of the sensors. In this paper, we propose a method for finding new data aggregation protocols called QDARL Quantum Data Aggregation scheduling using Reinforcement learning, which uses optimal channel capacity and memory for improving the quantum network efficiency. It finds optimal data aggregation schedules for sending the captured data to a distant quantum central node using local quantum network information with minimum bottleneck issues in the network.

2. Related Studies

In (S. Madhavi, 2014) (S. Madhavi and Tai Honn Kim, 2019), authors discussed energy Efficient Genetic Inspired Scheduling for Data Aggregation. In (H. L. Yeh et al., 2011) (S. Madhavi and Tai-hoon Kim, 2008) authors discussed a secured authentication protocol for wireless sensor networks using elliptic curve cryptography, Sensors. In (Hado Van Hasselt et al., 2016) authors discussed Deep reinforcement learning with double Q-Learning. In (Bartlett, Ben, 2018) authors discussed "A distributed simulation framework for quantum networks and channels". In (H. Salarian et al., 2014) authors discussed "An energy-efficient mobile-sink path selection strategy for wireless sensor networks. In (Francesco T. et al., 2019) (H. J. Kimble, 2008) authors discussed Quantum Internet: Networking Challenges in Distributed Quantum Computing. In (Sam Morley-Short et al., 2017) authors discussed the Physical-depth architectural requirements for generating universal photonic cluster states. In (N. Alon et al., 1994) authors discussed "Routing Permutations on Graphs via Matchings". In (M. Caleffi, 2017) authors discussed Optimal Routing for Quantum Networks. In (Kober, J. et al., 2013) authors discussed Reinforcement learning in robotic. In (F. Arute, 2019) authors discussed "Quantum supremacy using a programmable superconducting processor".

3. Proposed Quantum Network Model

We defined a Quantum network as Q(V, E, C, M) where V is a set of quantum processors, E is the set of edges between the processors, C is the set of Quantum channels and links. Let M(i) denotes the quantum memories at the i-th vertex. The total transmission time T is slotted, T consists of slots $\{t_0, \dots t_{L-1}\}$ where L denotes the length of the interval. The total schedule $S = \{S_0, S_1, ..., S_{T-1}\}$, where S_i denotes the subset of nodes in V scheduled to transmit during i-th time slot t_i. Let A_{ij}, for i, j=1...n denote the set of active nodes for each cluster headi. Assign a timeslot in the schedule for each Aij. The slot time assumed smaller than the memory's coherence time.

3.1 Proposed Quantum data aggregation using Q-L

Each time slot t has two phases: "channel entanglement" phase and "channel entanglement swapping" phase. The "channel entanglement" phase established a shared entangled (EPR) pair. An entanglement attempt succeeds with probability p0(e) ~ η(e), where η(e) ~ e−αL(e) is the transmissibility of a lossy optical channel of length L(e) (Mihir Pant et al., 2017). In the "channel entanglement swapping" phase, at each cluster node, pairs of qubit memories are entangle swapping is attempted. We define the throughput T as the total number of e-bits in all the quantum links in the network at any instance of time t.

Hence the probability that one e-bit is established successfully across the edge e during a time slot (Mihir Pant et al., 2017) is given by: p(e) = 1 − (1 − p0)S(e). Let us also assume S(e) = S, ∀e ∈ E, which in turn gives us p(e) = p, ∀e ∈ E (Pant, Mihir et al., 2019) . A quantum path may use multiple quantum channels but in an instance to improve the network efficiency, we consider successful and best entanglement pair in the quantum path. The fidelity of entanglement decays exponentially with the distance between the nodes (Michael Siomau, 2016). The node establishes entanglement with its best neighbor. Hence, to improve this situation, we propose each node establishes W parallel channels for establishing a quantum entanglement. The edge capacity C is equal to the maximal number of entangled pairs generated between adjacent nodes. Fig. 1 shows the quantum network with W parallel channels.

3.2 Scheduling Problem Definition

Cluster heads and to which cluster head a node should establish EPR pair is discussed in Algorithm 1. Throughput depends on

1. The probability of channel entanglement success rate QE at each node i
2. The probability of the channel entanglement swapping success rate QS at each node i

For each channel w_i at each node i:

$$T = \sum_{i=1}^{n} QE_i QS_i \quad \rightarrow 1$$

If node I have indegree d_i, then QE_i is defined as

$$QE_i = \sum_{i=1}^{w_i} \sum_{j=1}^{d_i} P_j \quad \rightarrow 2$$

where, p_j denotes the channel entanglement success probability rate at the j-th 1-hop neighbour of node i, Where $p_i = e^{-\delta L}$, L = the length of the channel and δ is constant depending on the physical media. We assumed p = 0.6 to 0.9 for an in degree of 10. Assumed w = 3 and the number of qubits at each node is assumed to be 10 to 14. The probability of the channel entanglement swapping success rate QS at each node i is defined as the points achieved for swapping at state s and the total number of concurrent nodes eligible for swapping in slot t schedule s. Let $QC^{i}_{conn}(j)$ denotes the time taken for a node i to establish a quantum connection with its neighbor j. The slot duration in a schedule s is set as

$$D^{j}_{s_t} = {}^{L_1^j}_{i=1} \max(\left(QC^{i}_{conn}(j)\right). \quad \rightarrow 3$$

where the value of D is set in such a way that it is at least as great as the time required for the quantum connection/entanglement swapping L_a^b denotes the a-th hop neighbour list for a node b. Q-learning is a learning algorithm that defines an agent, a set of states, and a set of actions per state. The states are various situations in the network that the agent learns about from time to time. If those actions are fetching towards the goal then the rewards will be allocated otherwise a penalty will be awarded. In the proposed method agents follows the following three steps to find the optimal policy.

1. The agent starts in a state (s_1) and to take an action (a_1) it selects an action that gives the highest possible value by referencing Q-table
2. Calculates reward (r_1) (r_1 can be positive or negative)
3. Update q-values
4. The Q-function can be represented as a q-table with states as rows and actions as columns. Using these q-values various decisions like
   - What are the possible Action set for the current state denoted by a,
   - What is the existing State for applying the action a,
   - What is State resulting after applying the action and
   - after applying the action how many rewards will be obtained

Table 1 shows the procedure for applying the Action for Clusterization. Table 2 shows the procedure for applying Action for schedules and Table 3 shows the procedure for applying the Action for maximizing the throughput. Let “reward()” denotes the maximum reward obtained from an optimal selection policy function M which decides the optimal number of concurrent nodes eligible for swapping in slot t schedule s. For example, while deciding, a node i selects a node k as its cluster head depending on whether

1. The probability of the channel entanglement success rate QE
2. The probability of the channel entanglement swapping success rate QS

Node i and node k is maximum when compared to other 1-hop neighbors of i then reward = number of ebits on the link between i and k.

Algorithm 1 is repeated for all the nodes in the network. A node having a maximum number of successful entanglements is called the best node. All the neighbors of this best node are said to be in one cluster and the best node is called the cluster head. The procedure for finding the best successful entanglement pair is as follows in Algorithm 1.

- Step 1. Best Successful entanglement
- For k=1 to n number of nodes in the network
- For I = 1 to number of 1-hop neighbors of k
- For j=1 1 to w_i
- To establish entanglement pair with node I on jth channel
- Compare Max_entaglement _success_probability with a probability of
- Channel entanglement success rate QEik
- Compare Max_entaglement _swapping__success_probability and the
- probability of the channel entanglement swapping success rate QSik
- store maximum value in Max_entaglement _success_ probabilityk
- store maximum value in Max_entaglement _swapping_success_ probabilityk
- Arrange the nodes in the priority of Max_entaglement _swapping _
- success_probabilityk and second priority of Max_entaglement _success _
- probability k into list denoted by List_Scuuess_entaglement k
- Find node p, which is listed first in the list List_Scuuess _ entanglement tk
- Except with p, k deactivates all the other entanglements
- p is the best neighbor with which k can make an engagement Bestneighbor[k]=p
- Step 2. Reconstruct the network with the best neighbors as the new neighbors and the entanglements as edge
- Step 3. Return modified Graph G
- Stop

Algorithm 1. Best successful entanglement pair for node

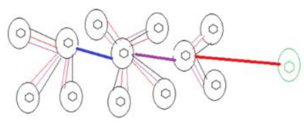

Fig. 1: The quantum network with W parallel channels

Table 1.Action for Cauterization

State action pair	Q(S1,a1)
Action	Action a1 Find the best successful entanglement pair between a node k and its 1-hop neighbors
Reward	The probability of the channel entanglement success rate QE and the probability of the channel entanglement swapping success rate QS at node k's 1-hop neighbor i is maximum when compared to other 1-hop neighbors of k then reward = number of ebits on the link between i and k
State for applying the action	State S1 Initial Network state Q(S1) with vertices V and edges E
The state resulting after applying the action	Entanglement is the success between k and node i and An agent will Update the q-table with nodes and its cluster head, its entanglement pairing, number of ebits, link capacity, number of qubits available Clusterization The base station selects the details of the cluster heads and records the list and proceeds for preparing a schedule for them

Table 2. Action for schedules

State action pair	Q(s2,a2)
Action	Action a1 The total number of ebits on quantum channel
Reward	The total number of ebits on the quantum channel
State for applying the action	State S2 Reconstruct graph(V,E) where E denotes entanglement pairs as edges and corresponding nodes with entanglement as vertices V The node having the highest 1-hop neighbors is called cluster head
state resulting after applying the action	Find interfering cluster head list and allocate slot I to the nodes which are entangled with their corresponding cluster heads Find noninterfering cluster head list and allocate slot i+1 to the nodes which are entangled with their corresponding cluster heads

Table 3. Action for maximizing the throughput

State action pair	Q(s3,a3)
Action	the throughput of the network
Reward	Action a3 improve Q-values by training the model for Nmax time
State for applying action	State S3 Q(s) is updated to achieve the optimal scheduling policy
The state resulting after applying action	Q-value from the Q matrix can be used for making a scheduling decision for maximizing the throughput

4. Simulation Results

The proposed a QDARL using reinforcement learning is implemented using qiskit. We assumed a dense network with nodes spreading in a 200x200 sq unit area where the distance between the nodes is $\frac{50}{\sqrt{n}}$ units. γ is a discount factor and can range from 0.0 to 1.0. The initial energy of all sensor nodes is 0.5 J. Each node has a qubit pool with qubits equal to the degree of the network. The number of parallel channels W is set to half a degree of the network. We implemented the method for testing the network performance using the important parameters like n number of nodes with n=200,400,600,800, d indegree of the network with d=5, 8, QS, probability of the channel entanglement swapping success rate at each node, Learning rate α € [0, 1] and number of bits per packet = 1000. We tested the proposed method for a network with indegree 5 and indegree 8. We observed the impact of learning rate on q-values which indirectly affects the throughput, channel efficiency. We assumed that each node can have up to w channels where w=half the indegree of the node. For the denser networks, the throughput increases when compared to the networks with less number of in degree. When compared with indegree 5 the result are maximized with in-network having indegree 8. But our proposed method could not obtain good results with indegree more than 8 for a network of size more than 800 in an area of 200x200 sq units. This may be due to noise leading to the failure of establishing the links. Fig. 2 shows the Learning rate Vs Q-Values. As the Learning rate increases (alpha value) agent knowledge about the better actions increases and the Q-values increase. An increase in the Q-value automatically increases the throughput of the system.

The network throughput is increased with the adoption of Q-learning as its optimal policy. All the nodes in the network are trained over 27000 iterations. The maximum throughput is seen for n=800 and degree = 8 at a learning rate = 1.0. Fig. 2(a) shows that when the agent learning rate is 0.1, the RL agent executes on an average for degree 5 with n=200,400,600,800 number of nodes approximately 8000,12900,16000,23000 steps to make the optimal strategy for the first time and converges after 150 iterations. Fig. 3(a) shows that when the agent learning rate increases to 1.0, the steps to learn the best strategy on an average for degree 5 with n=200,400,600,800 number of nodes approximately is about 3000, 000,6700,10000, and it converges to a stable state after 30 iterations.

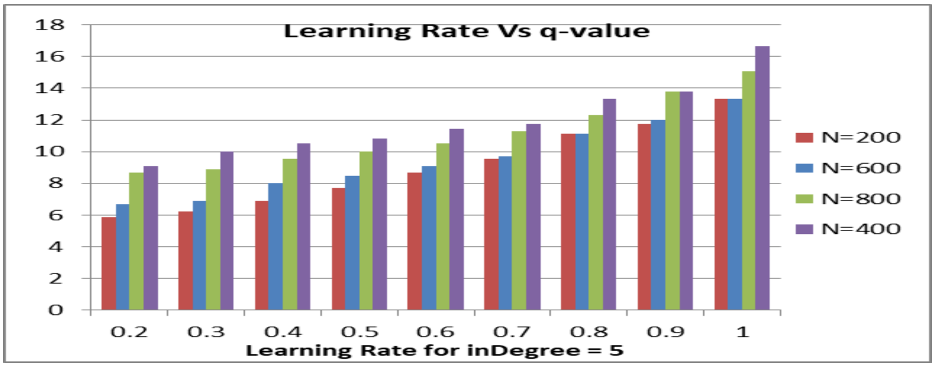

Fig. 2(a) Learning rate Vs Q-Values for indegree = 5

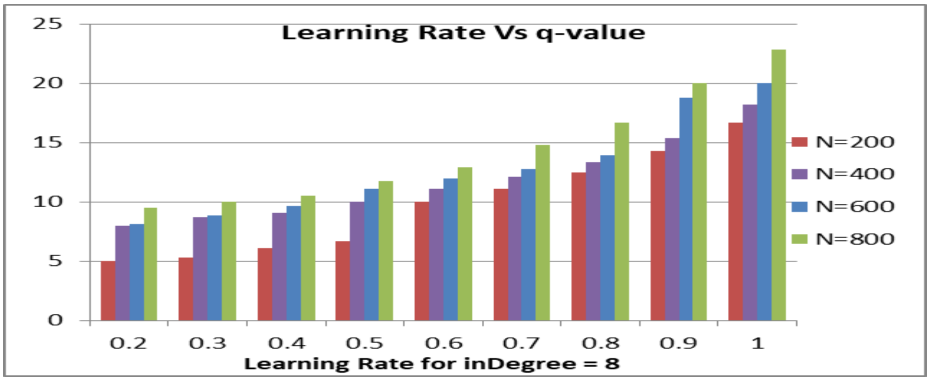

Fig. 2(b) Learning rate Vs Q-Values for indegree = 8.

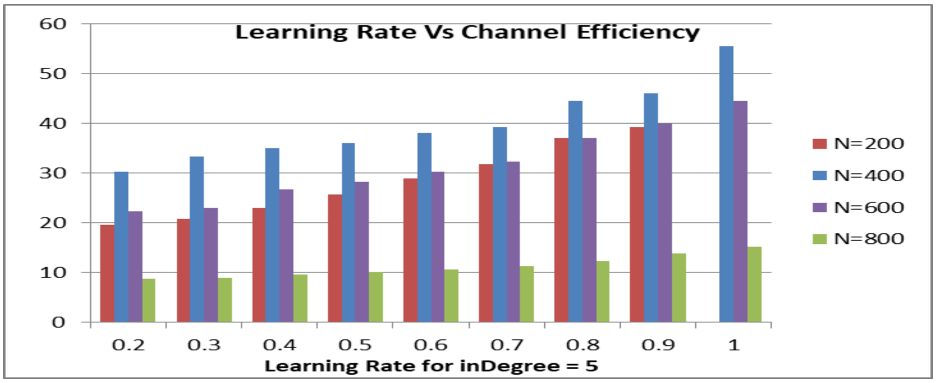

Fig. 3(a) Learning rate Vs Channel Efficiency for indegree = 5

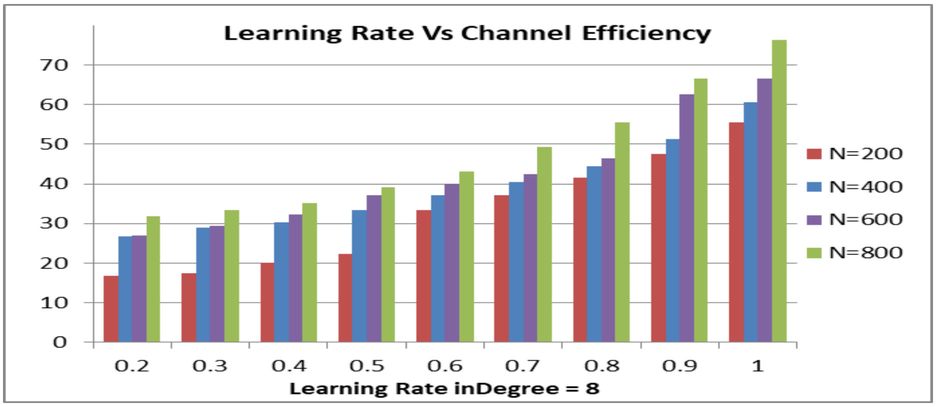

Fig. 3(b) Learning rate Vs Channel Efficiency for indegree = 8.

Fig. 3(b) shows that when the agent learning rate is 0.1, the RL agent executes on an average for degree 8 with n=200,400,600,800 number of nodes approximately 10000,16000,22600,27000 steps to make the optimal strategy for the first time and converges to a stable state after 60 iterations and Fig. 4 shows that when the agent learning rate is 1.0, the RL agent executes on an average for degree 8 with n=200,400,600,800 number of nodes approximately 4000,6000,12260, 17000 steps to make the optimal strategy and it converges to a stable state after 25 iterations.

Fig. 4 shows how the channel capacity is utilized best with the increase in the learning rate. Since the nodes can choose the best neighbour as their cluster head giving priority to the link capacity too, the channel utilization also increases.

5. Conclusions

Q-learning is a reinforcement learning algorithm, which reaches the state of convergence through several continuous iterations to obtain the maximum action value matrix Q(st, at) besides making its computational complexity relatively low: without a model, the agent learns through an iterative process and estimates the q-values.

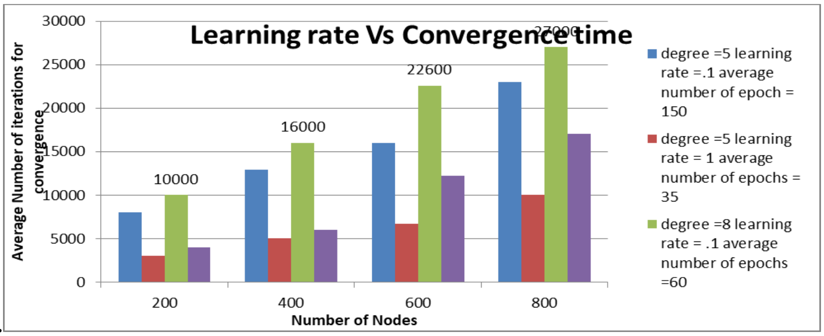

Fig. 4 Agent learning rate and Average Number of iterations for convergence

From a state Q[s, a] it will apply possible actions denoted by “a” and after applying an action fetches reward/punishments depending on the result of the action. Through this q-table agent guesses the best action. The proposed method finds optimal data aggregation schedules to maximize system throughput over N time slots. There are many traditional methods for finding the best schedules for a quantum network. But to our knowledge, this is the first time to use RL for finding data aggregation schedules using multiple channels and considering the probability of entanglement success and swapping probabilities for pairing with neighbor’s nodes for aggregating the data. In a future study, we will extend this work for other performance indicators such as delay, engagement losses due to noise as reward points in RL.

References

S. Madhavi. (2014). Secured Data Aggregation Scheduling in Ubiquitous Quantum Sensor Networks. *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, 4(1), 17-30.

H. L. Yeh, T. H. Chen, P.C. Liu, T. H. Kim, and H. W. Wei. (2011). A secured authentication protocol for wireless sensor networks using elliptic curve cryptography, *Sensors*, 11(5), 4767–4779.

Hado Van Hasselt, Arthur Guez, and David Silver. (2016). Deep reinforcement learning with double Q-Learning. *In 30th AAAI Conference on Artificial Intelligence*, AAAI. arXiv:1509.06461

Bartlett, Ben. (2018). A distributed simulation framework for quantum networks and channels. arXiv:1808.07047 v1.

S. Madhavi and Tai Honn Kim. (2016). Energy Efficient Genetic Inspired Scheduling for Data Aggregation. *International Journal of Control and Automation*, Journal of Control and Automation, 9(3), 227-234.

H. Salarian, K.-W. Chin, and F. Naghdy. (2014). An energy-efficient mobile-sink path selection strategy for wireless sensor networks, *IEEE Transactions on Vehicular Technology*, 63(5), 2407–2419.

S. Madhavi and Tai-hoon Kim. (2008). An Intrusion Detecting System in Mobile AdHoc Networks. *International Journal of Security and its Applications*, 2(3), 1-11.

Francesco Tafuri Francesco Saverio Cataliotti Stefano Gherardini Angela Sara Cacciapuoti, Marcello Caleffi and Giuseppe Bianchi. (2019). Quantum Internet: Networking Challenges in Distributed Quantum Computing, arXiv:1810.08421v2.

Sam Morley-Short, Sara Bartolucci, Mercedes Gimeno-Segovia, Pet Shadbolt, Hugo Cable, Terry Rudolph. (2017). Physical-depth architectural requirements for generating universal photonic cluster states, Quantum Science and Technology, arXiv:1706.07325.

N. Alon, F. R. K. Chung, and R. L. Graham. (1994). Routing Permutations on Graphs via Matchings, *SIAM Journal on Discrete Mathematics*, 7(3), 513–530. DOI: 10. 1137 /s0895480192236628.

M. Caleffi. (2017). Optimal Routing for Quantum Networks, *IEEE Access*, 5, 22299-22312.

Kober, J., Bagnell, J. A. and Peters, J. (2013). Reinforcement learning in robotics: A survey, The International Journal of Robotics Research, p.0278364913495721.

F. Arute et al. (2019). Quantum supremacy using a programmable superconducting processor, *Nature* 574(7779), 505–510. DOI: 10.1038/s41586-019-1666-5.

H. J. Kimble. (2008). The Quantum Internet. *Nature*, 453, 1023–1030.

Y. Yin, S. Aihua, G. Min, X. Yueshen, and W. Shuoping. (2016). QoS prediction for Web service recommendation with network location-aware neighbour selection, *Int. J. Softw. Eng. Knowl. Eng.*, 26(4), 611_632.

Michael Siomau (2016), Quantum Entanglement Percolation, arXiv:1602.06152v2.

Mihir Pant, Hari Krovi, Don Towsley, Leandros, Tassiulas,Liang Jiang, Prithwish Basu, Dirk Englund,Saikat Guha, (2017), Routing entanglement in the quantum internet, arXiv :1708.07142.