

Adaptive Entanglement Generation for Quantum Routing

Tasdiqul Islam

University of Texas at Arlington
Arlington, TX, USA
txi7184@mavs.uta.edu

Md Arifuzzaman

Missouri University of Science and
Technology
Rolla, MO, USA
marifuzzaman@mst.edu

Engin Arslan

Meta
Menlo Park, CA, USA
enginarslan@meta.com

ABSTRACT

Entanglement generation in long distance quantum networks is a difficult process due to resource limitations and probabilistic nature of entanglement swapping. To maximize success probability, existing quantum routing algorithms employ computationally expensive solutions (e.g., linear programming) to determine which links to entangle and use for end-to-end entanglement generation. Such optimization methods however cannot meet the delay requirements of real-world quantum networks, necessitating swift yet efficient real-time optimization models. In this paper, we propose reinforcement learning (RL)-based models to determine which links to entangle and proactively swap to meet connection requests. We show that the proposed RL-based approach is 20 \times times faster compared to linear programming. Moreover, we show that one can take advantage of longevity of entanglements to (i) cache entangled link for future use and (ii) to proactively swap entanglement on high demand path segments, thereby increasing the likelihood of request success rates. Through comprehensive simulations, we demonstrate that caching unused entanglements lead to 10 – 15% improvement on the performance of the state-of-the-arts quantum routing algorithms. Complementing the caching proactive entanglement swapping leads to further enhances the request success rate by up to 52.55%.

KEYWORDS

Quantum Network, Entanglement Routing, Reinforcement Learning, End-to-End Entanglement

ACM Reference Format:

Tasdiqul Islam, Md Arifuzzaman, and Engin Arslan. 2025. Adaptive Entanglement Generation for Quantum Routing. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/nnnnnnnnnnnnnnnn>

1 INTRODUCTION

Unlike classical networks, connection establishment (i.e., entanglement generation) in long distance quantum networks [5, 12] is a difficult process that involves several probabilistic operations such as producing entangled qubit pairs for each link and swapping the entanglement over multiple links. Quantum routing algorithms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2025 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00
<https://doi.org/10.1145/nnnnnnnnnnnnnn>

aim to improve the success probability of long distance quantum communication by identifying which links to use for given set of requests considering success probabilities as well as resource limitations.

Figure 1 depicts a simple quantum network with several quantum repeaters and Entangled Photon Sources (EPS). EPSes are positioned between nodes to produce entangled qubit pairs that are then transferred to adjacent nodes via quantum links. Once the link-level entanglement is established, swapping operation is performed to extend the entanglement to distant nodes. That is, entanglement between two adjacent link can be used to create entanglement between nodes that are two hop away. Then, the generated entanglement can be used to create entanglement between nodes that are three or more hops away. As the number of hops increases, the success probability decreases as the process can fail during any one of the entanglement generation or swapping operation.

Quantum routing algorithms play an essential role for quantum networks as they optimize which links to entangle and which routes to use for given connection requests, taking uncertainties introduced by quantum noise during the entanglement swapping process into account [4, 6, 7, 15, 17–19]. Typically, they follow a two-step process to establish an end-to-end entanglement as (i) selection of links for entanglement generation and (ii) selection of entangled links to establish end-to-end entanglement through entanglement swapping. Most quantum routing algorithms rely on compute-intensive optimization methods such as integer programming to optimize link and path selection decisions. For example, REPS algorithm [19] uses Integer Linear Programming both for links and path selection. However, such compute-intensive methods fail to produce results in a timely manner even for small networks with 30–40 nodes and 10–20 requests. As an example, it takes REPS algorithm around 140 seconds to select which links to entangle and around 180 to identify which set of links to extend the entanglement end-to-end. In addition, existing quantum routing algorithms fall short to achieve high success ratio due to not utilizing generated entanglements efficiently between consecutive time slots. Therefore, it is necessary to develop fast yet highly efficient quantum routing algorithms to pave the way for large scale quantum networks.

In this paper, we introduce AEG to optimize the routing in quantum networks. AEG proposes two improvements as follows: First, it introduces reinforcement learning based link selection strategy to quickly find which links to entangle. Unlike linear programming based solutions, reinforcement learning models can make prediction in the order of few seconds. Second, it innovates entanglement caching and proactive swapping methods to take advantage of unused entanglements for future requests. The rationale behind this approach is that since entanglement creation is a probabilistic process, caching unused entanglement as long as possible can increase

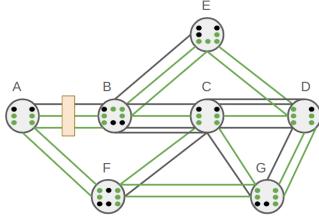


Figure 1: A simple quantum network with physical links with multiple paths between nodes and entangled photon source (EPS) in the middle of every link.

the success rate. AEG further takes advantage of unused entanglements to create multi-hop entanglements through entanglement swapping. This helps us to shorten the distance between end nodes, thereby improving the success rate of future connections.

Since AEG does not yet offer a new solution for path selection algorithm, we integrated it into quantum routing algorithm REPS [19] such that it can use linear programming for path selection. Our evaluations show that the reinforcement learning based link selection of AEG results in similar performance as REPS with 20 times faster in terms of execution time. We also show that applying our link selection approach with simply caching unused entangled qubits for several time slots (e.g., 5–10 time slots), AEG is able to outperform REPS by 20% in terms of request satisfaction rate. Taking it a step further by proactive entanglement swapping across multiple commonly used links lets AEG to attain 61% improvement over REPS.

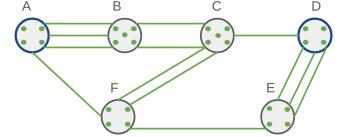
2 RELATED WORK

Various routing algorithms have been suggested for quantum communication, including Q-CAST [15], REPS [19], SEER [7], DQRA [11], and SEE [18], all aimed at maximizing throughput and reducing idle time. Most of them follow the four phase architecture discussed earlier where in the second phase they choose links for entanglement generation and works in time-slot manner. In [17] they select links for a request on its arrival as this works as online algorithm. In [14, 16] they use shortest path approach to find paths and generate entanglements for those paths.

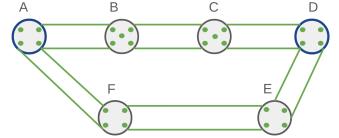
Choosing the right links for entanglement is important; to show this importance, Figure 2 compares two cases for a single request between nodes A and D. In the first case (Figure 2(a)), the selected links lead to a less reliable path. In the second case (Figure 2(b)), the chosen links form a more effective path. This example highlights how smart link selection can improve the overall performance of the network and motivates the need for better strategies for entanglement generation.

To select optimal links for generating entanglement REPS [19] and SEE [18] use Integer Linear Programming where REPS [19] tries to select links only between neighbor nodes but SEE [18] considers all-optical switching to select non neighbor nodes and tries to generate entanglement between them. The constraints for Integer Linear Programming depends on the number of requests that causes more complexity with large number of requests.

To increase success probability SEER [7] considers dividing the main path into two by selecting an intermediate trusted node. The intermediate node is chosen based on social relationships within the network. To select connection between source to the intermediate



(a) A non optimal Entanglement Generation



(b) An optimal Entanglement Generation

Figure 2: Optimal path selection for request A-D depends on which links are selected to generate entanglement

node then to destination it considers the shortest path and from that shortest path links are selected for generating entanglement. To find the shortest paths it uses a greedy algorithm which executes faster than Integer Linear Programming. Based on the successful entangled links swapping happens to establish long entanglement which follows the shortest path also. Link selection and swapping both depend on the selected shortest paths.

Deep reinforcement learning was applied to schedule requests in DQRA [11], but for routing it uses a greedy shortest path algorithm. Links are selected for entanglement from the shortest path in this work. Another Deep reinforcement learning based approach has been proposed in [13], but they do not consider multiple requests simultaneously.

Focusing on establishing entanglement between source-destination pairs, Q-CAST [15] selects multiple redundant paths and selects links for entanglement from these paths. As this algorithm finds links for entanglement using a greedy algorithm, it also executes faster than Integer Linear Programming but does not perform similar or better than REPS [19] or SEE [18].

Since recent studies show that entanglement can last for several seconds [1, 3], in [9], authors examined the impact of caching the unused entangled links on the performance of existing algorithms. With caching mechanism for entangled links they also showed the impact of proactive swapping which reduces the average distance of the networks resulting better performance for two existing routing algorithms. Deep Reinforcement Learning was applied to select optimal pairs of nodes for proactive swapping. Next, while REPS [19] can produce near optimal performance using Integer Linear Programming (ILP), it is challenging to apply the solution for large number of requests in network with high number of nodes. In AEG we replace ILP based link selection process with reinforcement learning, as it's batch prediction capabilities to select all potential links for entanglement at once make entire process exponentially faster and scalable for networks with high traffic.

3 SYSTEM DESIGN OF AEG

The existing quantum routing algorithms generally adopt a two-step process when handling connection requests. They are (i) selecting links to generate entanglement and (ii) finding path(s) using successfully entangled links to extend the entanglement from link level

to end-to-end [15]. Since entanglement generation and swapping operations are probabilistic, quantum routing algorithms choose multiple links to entangle and swap for a given request since connection attempt on primary path may fail. Then, entangled qubit pairs are created for each link using Entanglement Photon Sources (EPS) [19] whose success rate is contingent upon the inherent probability associated with each link. From the pool of successfully entangled links, an optimal path is selected for entanglement swapping. Similar to the entanglement generation, entanglement swapping operation can also fail due to the possibility of failure in the Bell State Measurement (BSM) [10] process. Ultimately, if at least one of the selected paths is successful in creating entanglement on all links and conducting BSMs, the request is deemed to be successful.

Existing quantum routing algorithms discard unused entanglements between consecutive time slots, thus they perform entanglement generation and swapping operations in each round from scratch. However, previous studies showed that entanglement between two qubits can endure for up to ten seconds [3]. Hence, we propose to cache unused entangled links for a few time slots (e.g., 10 time slots) to better utilize resources and improve the performance of routing algorithms [9]. Moreover, state-of-the-art quantum routing solutions rely on expensive computational solutions such as linear programming for optimal link and path selection, which are not scalable especially for large scale networks due to slow nature of these optimization algorithms. Hence, we introduce reinforcement-learning based link selection model to speed up the process.

3.1 Reinforcement Learning-Based Link-Level Entanglement Generation

Limited quantum memories and probabilistic nature of entanglement generation create a significant challenge for establishing end-to-end connections in quantum network. For a given set of requests and resources, Integer Linear Programming (ILP) can be applied to select which link to entangle and swap to meet as many requests as possible. While ILP yields high performance due to transparent mathematical optimization, it can take a long time to find a solution in large scale networks. Hence, it is not suited for problems that require real-time decisions. Reinforcement Learning (RL) based algorithms have been widely used to solve complex optimization problems. A trained RL model can directly predict all potential links quickly by avoiding the cost of repeated calculations. In particular, Deep Q-learning (DQRL) is known to attain high performance and scalability with the help of deep learning based predictions. However, RL models require a clear definition of states, actions and rewards function that accurately reflect the core aspects of the networking environment and routing objectives to achieve high performance. In our problem, we define them as follows:

- *States:* The DQRL state space is consist of four components:
 - (1) *Topology:* The connectivity of the network $G^{(t)}(V, E^{(t)})$ at time slot t is calculated as a subset of the topology G . Only links that not entangled already are considered as edges. Then, we extract the adjacency matrix $A^{(t)} = (a_{i,j})_{1 \leq i,j \leq |V|}$ of $G^{(t)}(V, E^{(t)})$ where $a_{i,j}$ is the number of links between nodes i and j . This is a $N \times N$ matrix where N is the number of nodes.

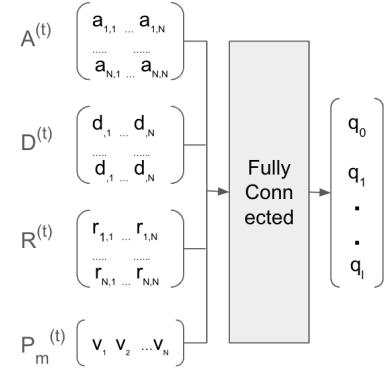


Figure 3: A sample input and output of the Deep Q-Learning. It takes network topology, distances among nodes, requests and entanglement edges as input and output a list of potential links for entanglement generation

- (2) *Link Distance:* Distance of links impacts successful entanglement generation. Thus, we consider the distances between links in our state and represent them as adjacency matrix $D^{(t)} = (d_{i,j})_{1 \leq i,j \leq |V|}$ where $d_{i,j}$ is the distance between nodes i and j .
- (3) *Requests:* The new requests at time slot t along with the remaining requests from previous time slots are represented a $N \times N$ matrix which is denoted as $R^{(t)} = (r_{s,d})_{1 \leq s,d \leq |V|}$ where $r_{s,d}$ is the number of requests between source and destination nodes.
- (4) *Edges:* The m^{th} pair (n_1, n_2) from the set of edges in the topology that will be chosen to create entanglement is represented by a binary vector $P_m^{(t)}$, where each element v_i is either 1 (if i belongs to (n_1, n_2)) or 0 otherwise.

The above four components make up the input state for m^{th} pair $S_m^{(t)} = [A^{(t)}; D^{(t)}; R^{(t)}; P_m^{(t)}]$

- *Output Actions:* The Deep Q neural network outputs a Q-value vector for each link in each time slot $Q_m^{(t)} = [q_0^{(t)}, q_1^{(t)}, \dots, q_l^{(t)}]$ of length $l+1$ where l is the maximum link capacity between two nodes and the action $a_m^{(t)} = \text{argmax}_{i \in 0,1} (q_i^{(t)})$ is the number of links we'll try to generate entanglement for the m^{th} pair or if $a_m^{(t)}$ is greater than maximum number of links between the pair then will try to entangle all the links. Figure 3 shows an example of the input and output.
- *Rewards:* We define rewards and penalty as follows.
 - (1) We assign a positive reward ($R(s, a) > 0$) for an action (i.e., choosing a link to entangle) if a selected link is used for a request either directly or as part of link segments that is explained in Section 3.2.
 - (2) We assign a negative reward ($(R(s, a) < 0)$) to an action if selected link expires (i.e., lifespan ends) without being used for a connection request. This penalty ensures that the model will try avoid choosing links that are not likely to be used in the future.

Model Training: Instead of using the traditional train-and-evaluate method, we design the algorithm as an online optimizer with both exploration and exploitation phases [8]. Exploration

phase helps the model to stay up-to-date with evolving networking dynamics. The exploitation phase, on the other hand, aims take advantage of current model weights to predicts optimal links to entangle in any specific time slot. In our implementation, we maintain two versions of Q network for model training, a prediction network and a target network. The Q-values are obtained from the prediction network which is trained using the Stochastic Gradient Descent algorithm [2]. The target network's weights are updated periodically with those of the prediction network, typically at every few hundred time steps. The target network is exclusively utilized for forecasting future Q-values. While this process is expected to repeat indefinitely, we used a model with fixed number of iterations to compare results with the state-of-the-arts solutions.

3.2 Proactive Entanglement Swapping

Existing routing algorithms discard the unused entanglements which necessitates new entanglement generation in each time slot. Taking advantage of longevity of entanglement, we aim to maximize the success of connection establishment by caching unused entangled links. Since entanglements cannot be cached forever, we mark the entanglement generation time for each unused link to keep track of their remaining lifespan. If a cached entanglement is not utilized within its lifetime, it is destroyed and the associated memory is freed.

In addition to caching unused entanglements at the link-level, we also conduct entanglement swapping across neighboring links to extend the entanglement from single link to multiple links (i.e., segments). These segments can then be used to meet future requests. Entanglement swapping over multiple links reduces the average path length, thereby increasing the probability of successful entanglement for future requests. Additionally, it creates additional routes between source and destination nodes, which increases the success probability of connection requests. On the other hand, the capacity of quantum repeaters are limited, thus, it is important to create entangled segments judiciously to avoid running out of quantum memory which could leave insufficient resources for future time slots. That is, since proactive entanglement swapping cannot guarantee a perfect prediction, it is important to reserve some links to meet “unexpected” requests in upcoming time slots. Therefore, we implemented another DQRL method to predict which segments to proactively establish entanglement such that the success rate of future requests can be improved. We define its state, actions, and rewards as follows:

- *Environment*: We incorporated our caching and proactive entanglement swapping approach into an existing quantum routing algorithm REPS [19]. REPS uses ILP to find optimal paths for a given set of connection requests. Hence, the routing algorithm REPS acts as the environment for the Q-learning model.
- *States*: DQRL state space is consist of three components:
 - (1) *Topology*: The connectivity of the network $G^{(t)}(V, E^{(t)})$ at time slot t is calculated as a subset of the topology G . Only entangled links are considered as edges. Then, we extract the adjacency matrix $A^{(t)} = (a_{i,j})_{1 \leq i,j \leq |V|}$ of $G^{(t)}(V, E^{(t)})$ where $a_{i,j}$ is the number of entangled links

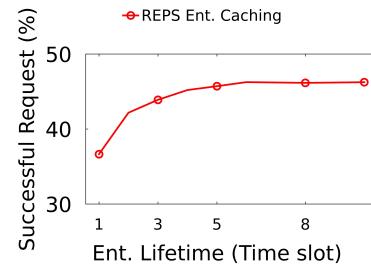


Figure 4: Impact of entanglement lifetime on the performance of AEG.

between nodes i and j . This is a $N \times N$ matrix where N is the number of nodes.

- (2) *Requests*: The new requests at time slot t along with the remaining requests from previous time slots are represented a $N \times N$ matrix which is denoted as $R^{(t)} = (r_{s,d})_{1 \leq s,d \leq |V|}$ where $r_{s,d}$ is the number of requests between source and destination nodes.
- (3) *Segments*: The m^{th} pair (n_1, n_2) that will be chosen to create proactive entanglement is represented by a binary vector $P_m^{(t)}$, where each element v_i is either 1 (if i belongs to (n_1, n_2)) or 0 otherwise.

The above three components make up the input state for m^{th} pair $S_m^{(t)} = [G^{(t)}; R^{(t)}; P_m^{(t)}]$

- *Output Actions*: The Deep Q neural network outputs a Q-value vector for each segment in each time slot $Q_m^{(t)} = [q_0^{(t)}, q_1^{(t)}]$ of length 2 and the action $a_m^{(t)} = \text{argmax}_{i \in 0,1} (q_i^{(t)})$ is applied on the m^{th} pair.

Model Training: The model training is similar to our previous exploration/exploitation model for link selection for entanglement generation. Deep Neural Network is evaluated for every segment to determine whether or not to swap the entanglement. Ideally, these decisions (i.e., actions) should be executed sequentially such that state changes caused by one action (e.g., reduction of available link capacity) can be reflected in the evaluation of another action. However, sequential execution introduces two challenges. First, evaluation order of actions will affect the performance since available capacity will decrease as more actions are selected, giving advantage to actions that are evaluated earlier. Second, it takes long time (up to 30 seconds) to evaluate DNN model sequentially for all possible actions. Hence, we evaluate actions in parallel, then choose the ones with the highest Q values as long as there are enough resources. However, selecting segments with largest Q-values may limit the models ability to explore the solution space by choosing the same segments all the time. Thus, we utilize exploration together with exploitation to ensure that we can explore new solutions while taking advantage of discovered solutions.

4 EVALUATIONS

To evaluate the performance of RL-based link selection and proactive entanglement swapping solutions, we integrated AEG into quantum routing algorithm, REPS [19]. REPS uses two separate ILPs for link selection and path selection operations. We replaced

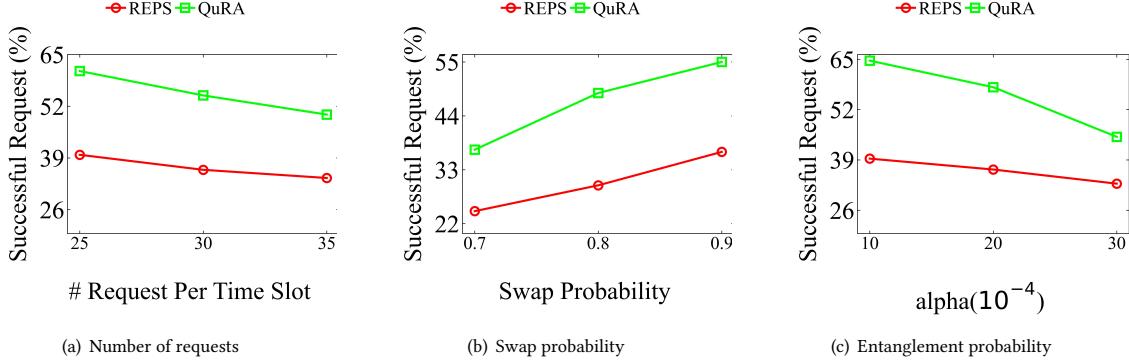


Figure 5: Performance comparison of REPS and AEG for varying request count, swap probability, and entanglement probability values.

the first ILP in REPS with AEG’s RL based link selection algorithm. Then, we incorporated proactive entanglement swapping solution into REPS’s path selection algorithm. As a result, proactive entanglement generation receives the output of RL-based link selection algorithm (i.e., the network with a set of successfully entangled links) as input and produces output (a new network with a combination of entangled segments and links). Its output is then consumed by the second ILP of REPS algorithm to choose the optimal paths for given connections.

4.1 Evaluation Methodology

To construct the network, we employed the Waxman model and adhered to the simulation methodology outlined in [7, 15, 19]. In accordance with these simulation parameters, we placed 50 quantum nodes within a rectangular area measuring 2000 km x 4000 km. Each link has a randomly generated qubit transmission capacity between 3-7 and each node has randomly generated quantum memory capacity between 10-14. The probability for entanglement generation was formulated as $P(u, v) = e^{-\alpha \cdot l(u, v)}$, where $l(u, v)$ represents the Euclidean distance between two nodes. A typical value for the parameter α was set to 0.002. With $\alpha = 0.002$ and $l(u, v) = 100\text{km}$ the value of $P(u, v) = e^{-\alpha \cdot l(u, v)}$ is 0.819. The success probability for entanglement swapping was configured to 0.9. Our simulations, on average, ran for 200,000 time slots, and the results are averaged over 10 trials for each outcome. In Q-value updating phase of Q-learning, we use learning rate of $\beta = 0.1$ and discount factor of $\gamma = 0.95$.

4.2 Evaluation Results

4.2.1 Entanglement Lifetime. The impact of entanglement caching strategy is dependent on the duration of entanglement. If an entanglement can be sustained for only one time slot, no entangled link can be retained in the cache. However, if entanglement can persist longer, then caching can enhance the performance. We illustrate how entanglement duration impacts the performance of quantum routing in Figure 4. When entanglement is sustained for two time slots, the performance improves by 7%. The improvement rate increases to 19.21% when the entanglement lifetime is set to six or more time slots. It is evident, entanglement caching leads to much

better performance, and the success rate plateaus at six time slots, indicating that all cached entangled links are used within six time slots. Given that the entanglement lifetime can last for 10 seconds [3] and most quantum routing algorithms consider each time slot as one (or less) second, we set the default entanglement lifetime as 10 time slots.

4.2.2 Request Count. Figure 5(a) illustrates the impact of the number of new requests. As the number of requests increase, we observe a decrease in the performance in terms of serving the requests. AEG outperforms REPS in term of serving requests by 52.55%. This can be attributed to AEG’s entanglement caching and proactive swapping solutions as it helps us to reduce the number of links to entangle and swap at the time of request.

4.2.3 Swap Probability. The success of quantum entanglement routing algorithms is significantly influenced by the probability of successfully swapping entangled links. As depicted in Figure 5(b), an increase in the swap probability correlates with improved performance for REPS algorithm. AEG leads up to 54% improvement over the original REPS algorithm. As AEG includes proactive entanglement swapping and entanglement caching on top of RL based link selection for entanglement, this finding emphasizes the impact of our proactive entanglement swapping strategy, particularly in scenarios with higher swap probabilities.

4.2.4 Entanglement Generation Probability. The probability of successful entanglement generation affects the performance of quantum routing algorithm. Figure 5(c) shows that as the success rate of entanglement reduces (entanglement success probability has inverse relation to α), the rate of successful requests also decreases. AEG improves the performance over REPS by up to 52.38%. The performance difference is higher when entanglement generation probability is higher since entanglement caching and proactive swapping are able to cache more entangled links and create more entangled segments.

4.3 Ablation Study

Figure 6 shows the impact of entanglement caching and proactive entanglement swapping for AEG. We observe that RL-based link selection for entanglement generation yields similar performance to

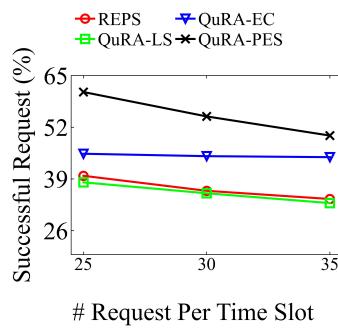


Figure 6: Performance impact of link selection, entanglement caching and proactive entanglement swapping in AEG.

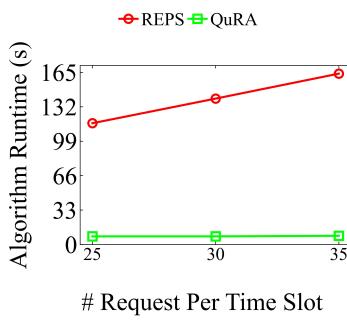


Figure 7: Execution time for REPS and AEG for link selection phase. While REPS relies on Integer Linear Programming, AEG takes advantage of Reinforcement Learning to lower the execution time without sacrificing the performance.

REPS. Entanglement caching with RL based link selection improves the performance by 18.34%. We get the best performance combining RL based link selection with caching and proactive swapping. While entanglement caching improves the performance by 15–20% over REPS, Deep Q-learning based proactive entanglement swapping increases the request success rate by up to 52.55%, demonstrating the effectiveness of proactively swapping entanglement over frequently accessed links.

4.3.1 Execution Time. REPS uses ILP to obtain near optimal performance [19]. In Figure 5, we show that by replacing ILP with Deep RL, AEG attains a similar performance with significantly lower execution time. The reinforcement learning approach reduces the runtime to select links for entanglement as much as 20 times than the ILP approach in REPS. We observe that with increasing request per time slot, the runtime remains almost constant in our solution compared to linear increase with ILP.

5 CONCLUSION

In this paper, we introduced a fast and scalable link selection mechanism for large quantum networks. We applied reinforcement learning (RL) to select potential links for entanglement generation and swapping. RL reduces the execution time by more than 20 \times compared to integer linear programming based solution. Additionally, by taking advantage of entanglement caching and proactive swapping, we can improve the performance of routing algorithms by

around 50%. The evaluation results shows that RL is able to reduce execution time without compromising the performance, which makes it more practical and feasible for real-world quantum networks.

REFERENCES

- [1] HP Bartling, MH Abobeih, B Pingault, MJ Degen, SJH Loenen, CE Bradley, J Randall, M Markham, DJ Twitchen, and TH Taminiau. 2022. Entanglement of spin-pair qubits with intrinsic dephasing times exceeding a minute. *Physical Review X* 12, 1 (2022), 011048.
- [2] Léon Bottou et al. 1991. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nimes* 91, 8 (1991), 12.
- [3] Conor E Bradley, Joe Randall, Mohamed H Abobeih, RC Berrevoets, MJ Degen, Michiel A Bakker, Matthew Markham, DJ Twitchen, and Tim H Taminiau. 2019. A ten-qubit solid-state spin register with quantum memory up to one minute. *Physical Review X* 9, 3 (2019), 031045.
- [4] Lutong Chen, Kaiping Xue, Jian Li, Zhonghui Li, Ruidong Li, Nenghai Yu, Qibin Sun, and Jun Lu. 2024. Redp: Reliable entanglement distribution protocol design for large-scale quantum networks. *IEEE Journal on Selected Areas in Communications* (2024).
- [5] L-M Duan, Mikhail D Lukin, J Ignacio Cirac, and Peter Zoller. 2001. Long-distance quantum communication with atomic ensembles and linear optics. *Nature* 414, 6862 (2001), 413–418.
- [6] Ali Farahbakhsh and Chen Feng. 2022. Opportunistic routing in quantum networks. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 490–499.
- [7] Shao-Min Huang, Ming-Huang Chien, Cheng-Yang Cheng, Jian-Jhih Kuo, and Li-Hsing Yang. 2022. Socially-aware Concurrent Entanglement Routing with Path Decomposition in Quantum Networks. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 3664–3669.
- [8] Shin Ishii, Wako Yoshida, and Junichiro Yoshimoto. 2002. Control of exploitation-exploration meta-parameter in reinforcement learning. *Neural networks* 15, 4–6 (2002), 665–687.
- [9] Tasdiqul Islam, Md Arifuzzaman, and Engin Arslan. 2024. Reinforcement Learning Based Proactive Entanglement Swapping for Quantum Networks. In *2024 International Conference on Quantum Communications, Networking, and Computing (QCNC)*. IEEE, 135–142.
- [10] Yoon-Ho Kim, Sergei P Kulik, and Yanhua Shih. 2001. Quantum teleportation of a polarization state with a complete Bell state measurement. *Physical Review Letters* 86, 7 (2001), 1370.
- [11] Linh Le, Tu N Nguyen, Ahyoung Lee, and Braulio Dumba. 2022. Entanglement routing for quantum networks: a deep reinforcement learning approach. In *ICC 2022-IEEE International Conference on Communications*.
- [12] Sreraman Muralidharan, Linshu Li, Jungsang Kim, Norbert Lütkenhaus, Mikhail D Lukin, and Liang Jiang. 2016. Optimal architectures for long distance quantum communication. *Scientific reports* 6, 1 (2016), 20463.
- [13] Jan Roik, Karol Bartkiewicz, Antonín Černoch, and Karel Lemr. 2024. Routing in quantum communication networks using reinforcement machine learning. *Quantum Information Processing* 23, 3 (2024), 89.
- [14] Sara Santos, Francisco A Monteiro, Bruno C Coutinho, and Yasser Omar. 2023. Shortest path finding in quantum networks with quasi-linear complexity. *IEEE Access* 11 (2023), 7180–7194.
- [15] Shouqian Shi and Chen Qian. 2020. 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*. 62–75.
- [16] Rodney Van Meter, Takahiko Satoh, Thaddeus D Ladd, William J Munro, and Kae Nemoto. 2013. Path selection for quantum repeater networks. *Networking Science* 3 (2013), 82–95.
- [17] Lan Yang, Yangming Zhao, Hongli Xu, and Chunming Qiao. 2022. Online entanglement routing in quantum networks. In *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.
- [18] Gongming Zhao, Jingzhou Wang, Yangming Zhao, Hongli Xu, and Chunming Qiao. 2022. Segmented entanglement establishment for throughput maximization in quantum networks. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 45–55.
- [19] Yangming Zhao and Chunming Qiao. 2021. Redundant entanglement provisioning and selection for throughput maximization in quantum networks. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 1–10.