Learning-based Optimal Quantum Switch Scheduling

Jiatai Huang
Tsinghua University
hjt18@mails.tsinghua.edu.cn

Longbo Huang
Tsinghua University
longbohuang@tsinghua.edu.cn

ABSTRACT

In this paper, we consider the problem of optimal scheduling for quantum switches with dynamic demand and random entanglement successes. Different from prior results that often focus on (known) fixed entanglement success probabilities, we assume zero prior knowledge about the entanglement success probabilities and allow them to vary from time to time in an adversarial manner. We propose a learning-based algorithm `QSSoftMW` based on the framework developed in [1], which combines adversarial learning and Lyapunov queue analysis. We show that `QSSoftMW` is able to automatically adapt to the changing system statistics and ensure quantum switch stability.

1. INTRODUCTION

The design and control of quantum networks have received much attention recently, due to its potential to establish an interconnection foundation for future quantum computing. Among the many important problems that have been investigated, the optimization of quantum switches has been studied in several prior works, e.g., [6, 5, 3, 4].

Specifically, the quantum switch scheduling problem considers a quantum switch connecting a set of users. The switch is capable of establishing link-level entanglements between itself and the users. At every time, random requests arrive at users asking to set up entanglements with target users. By performing entanglement swapping operations, the switch turn link-level entanglements with users into end-to-end entanglements between users (in a probabilistic way). The objective of the switch is to find a scheduling policy, which determines which user pairs to connect, so as to serve as many requests as possible.

In this paper, we consider the quantum switch scheduling problem assuming that the arrival and entanglement dynamics can be *adversarial*. This setting is particularly useful for non-stationary systems without prior knowledge of the environment or having only inaccurate estimation. On the other hand, this formulation also poses new challenges compared to existing works that assume knowledge of the entanglement success probabilities, e.g., [3, 4]. In particular, the common Lyapunov analysis technique requires that at each decision making time, some form of accurate service rate estimation (either exact or in expectation) is available.

To tackle the above difficulties, we propose a learning-based scheduling policy `QSSoftMW`, which builds upon a recent framework in [1] that provides a systematic technique to incorporate adversarial learning into queueing network control. `QSSoftMW` can provably stabilize the quantum switch under mild technical assumptions. The algorithm and results in our paper shed light into designing robust quantum switch scheduling algorithms.

Copyright is held by author/owner(s).

2. PROBLEM FORMULATION

We present the problem formulation in this section. We consider a discrete-time quantum switch connecting a set of N users via quantum links, e.g., fiber optic cables, as shown in Figure 1. At each time t, entanglement requests arrive at the users. We denote by $A_{ij}(t)$ the number of requests coming to user i requesting to establish entanglement with user j at time t. We assume $0 \leq A_{ij}(t) \leq A_{\max}$ for all $1 \leq i < j \leq N$.

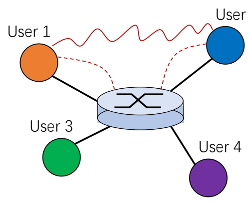

Figure 1: An example of a quantum switch connecting to four users. The switch can turn two link-level entanglements (dotted lines) into an end-to-end entanglement (solid line) via entanglement swapping.

The quantum switch can decide to generate quantum entanglement with a selected set of users at each time. We assume that both the switch and the users have enough qubits for establishing entanglements. For simplicity, we assume that if a switch decides to establish entanglement with user i, it always tries to set up the maximum number of connections. However, it is known that each attempt succeeds with certain probability at each time, and the subsequent entanglement swapping operation for end-to-end connection is also probabilistic, e.g., [4].

To capture these aspects, we denote by $\mu_{ij}(t)$ the number of entanglements successfully established between user i and user j, if the switch decides to connect them at time t via entanglement swapping over two switch-user connections. We assume $0 \leq \mu_{ij}(t) \leq \mu_{\max}$. We also assume that at each time, the switch can only choose to setup entanglements between one pair of users. As in prior works, e.g., [4, 3], we assume that each entanglement connection remains valid for a slot.

Depending on the outcome of the attempts, some of the entanglement requests will be served. We denote by $Q_{ij}(t)$ the number of requests that are still pending at the end of the t-th time slot. These unserved requests will be buffered in the request queues and wait for service. Thus, we see that $Q_{ij}(t)$ evolves according to

$$Q_{ij}(t) = \max[Q_{ij}(t-1) - \mu_{ij}(t), 0] + A_{ij}(t), \quad \forall\, t. \tag{1}$$

The goal of the switch is to look for a scheduling policy, which determines to serve one user-pair each time, with the objective of stabilizing the request queues, formally defined as:

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i<j} \mathbb{E}\{Q_{ij}(t)\} < \infty. \tag{2}$$

One important feature of our formulation is that it does not assume stationarity of both the arrival and service amounts, i.e., the values $\{A_{ij}(t)\}_{t=1}^{\infty}$ and $\{\mu_{ij}(t)\}_{t=1}^{\infty}$ are chosen by an adversary (the environment) at $t = 0$, before any scheduling decision being made, and they are only assumed to be bounded and can vary from time to time in an arbitrary manner. This is very different from existing quantum switch scheduling works, e.g., [6, 5, 3, 4]. Such a setting is not uncommon in practice, e.g., due to unknown interference or inaccurate estimation of the entanglement success probabilities.

Notations. $[N] = \{1, 2, \cdots, N\}$ denotes the set of first N positive integers. We denote the N dimensional probability simplex by $\Delta^{[N]} \triangleq \{\mathbf{x} \in \mathbb{R}^N : \sum_{i=1}^N x_i = 1, x_i \geq 0\ \forall i \in [N]\}$. For any $0 \leq \beta \leq 1/N$, $\Delta^{[N],\beta}$ denotes the set $\{\mathbf{x} \in \Delta^{[N]} : x_i \geq \beta\ \forall i \in [N]\}$. We denote $K = \binom{N}{2}$ for all feasible user pairs for entanglement and $[K] = \{1, 2, \cdots, K\}$. We also abuse the notation $\mathbf{Q}_t$ as a K-dimension vector to denote the backlog sizes for all user pairs at the end of the t-th time slot. Similarly, we use $\mathbf{A}_t$ and $\vec{\mu}_t$ to represent the vectors of entanglement request arrivals and successful services at time t, respectively.

3. ALGORITHM

In this section, we present our algorithm `QSSoftMW`, which is a restated version of the `SoftMW` algorithm developed in our recent work [1] that combines adversarial learning and Lyapunov analysis.

The high level idea of the algorithm design is as follows. In the classical `MaxWeight` scheduling [2], where the values of $\{\mu_{ij}(t)\}$ need to be known beforehand, at each time t, the scheduler picks a pair of users (i, j) such that the term $Q_{ij}(t-1)\mu_{ij}(t)$ is the maximum among all user pairs. The fact that $Q_{ij}(t-1)\mu_{ij}(t)$ are maximized at each time step allows one to upper-bound `MaxWeight`'s quadratic Lyapunov function value by comparing it with any queue-length unaware randomized policy in a Lyapunov drift analysis framework [2]. Then, by choosing an appropriate stationary and randomized stabilizing policy, the inequalities on Lyapunov function values can solve to a queue length upper-bound, and queue stability is thus established.

In our setting, $\mu_{ij}(t)$ can be observed only at the end of the t-th time slot, i.e., after the decision has been made. Moreover, only the service rate of the user pair selected at this time slot is observable, and the service rates of other user pairs remain unknown to the scheduler (i.e., bandit feedback). To handle this markedly different setting, we adopt an adversarial MAB algorithm into the scheduling process. Specifically, we use the algorithm proposed in [1], which can generate scheduling decisions under the bandit feedback, and guarantee that, the term $\sum_{t=1}^T Q_{ij}(t-1)\mu_{ij}(t)$, i.e., the cumulative sum of the queue-length-service-rate-products, is either larger than or sufficiently close to any queue-length unaware randomized policy (called reference policy) in expectation. This reference random policy is even allowed to be time-varying, as long as the time-variation of the probability to serve each user pair is not too large. This weaker form of $\sum_{t=1}^T Q_{ij}(t-1)\mu_{ij}(t)$ lower-bound allows one to recover the remaining steps in a common Lyapunov-drift based average queue length analysis (see next section for details).

The formal pseudo-code of the algorithm is given in Algorithm 1 below, which is a restated version of the `SoftMW` algorithm developed in [1]. In Section 4, we provide the formal average queue length bounds and technical assumptions for Algorithm 1.

Algorithm 1: `QSSoftMW` (**Q**uantum **S**witch **Soft** **M**ax**W**eight)

Input: One-step arrival upper-bound $A_{\max} > 0$ and service upper-bound $\mu_{\max} > 0$, number of users N, problem instance smoothness parameter $\delta > 0$

Output: A sequence of user pairs to establish entanglments at each time step $(i_1, j_1), (i_2, j_2), \ldots \in [N]^2$

$M \leftarrow \max\{A_{\max}, \mu_{\max}\}$

$K \leftarrow \binom{N}{2} = N(N-1)/2$ `// number of user pairs`

Denote by $\mathcal{A}$ the set of all K feasible user pairs, choose a bijection ϕ from $[K]$ to $\mathcal{A}$

$\Psi(\mathbf{x}) \triangleq \sum_{i=1}^K (x_i \ln x_i - x_i)$

$\mathbf{x}_1 = \mathbf{1}/K \in \Delta^{[K]}$

for $t = 1, 2, \ldots$ **do**

  $\beta_t \leftarrow t^{-3}/K$

  $\eta_t = \left(t^{-(\frac{1}{4}-\frac{\delta}{2})} M \sqrt{86M^2K^6t^{\frac{3}{2}} + \sum_{s=0}^{t-1}\|\mathbf{Q}_s\|_2^2}\right)^{-1}$

  $\mathbf{e}_t = \mathbf{Q}_{t-1}/\|\mathbf{Q}_{t-1}\|_1$

  $\gamma_t = M\eta_t\|\mathbf{Q}_{t-1}\|_1 = \|\mathbf{Q}_{t-1}\|_1 \left(t^{-(\frac{1}{4}-\frac{\delta}{2})} \sqrt{86M^2K^6t^{\frac{3}{2}} + \sum_{s=0}^{t-1}\|\mathbf{Q}_s\|_2^2}\right)^{-1}$

  $\mathbf{p}_t \leftarrow (1-\gamma_t)\mathbf{x}_t + \gamma_t \mathbf{e}_t$

  Sample $a_t \sim \mathbf{p}_t$

  $(i_t, j_t) \leftarrow \phi(a_t)$

  Setup entanglements between i_t and j_t, observe the actual successful entanglements $\mu_{i_t,j_t}(t)$

  $\widetilde{\mathbf{g}}_t \leftarrow \begin{cases} \frac{Q_{i_t,j_t}(t-1)\mu_{i_t,j_t}(t)}{p_{t,a_t}} & \text{the } a_t\text{-th coordinate} \\ 0 & \text{the other } K-1 \text{ coordinates} \end{cases}$

  $\mathbf{x}_{t+1} \leftarrow \arg\min_{\mathbf{x}' \in \triangle^{[K]},\beta_t} \langle -\eta_t \widetilde{\mathbf{g}}_t, \mathbf{x}' \rangle + D_\Psi(\mathbf{x}', \mathbf{x}_t)$

4. PERFORMANCE ANALYSIS

Results and technical assumptions in this section are all from our recent work [1] (the `SoftMW` algorithm in the paper). Readers may refer to [1] for more detailed discussion. As we briefly described in Section 3, the analysis of `QSSoftMW` (Algorithm 1) is mainly two-fold. Firstly, the embedded MAB algorithm guarantees that (when inspected in the Lyapunov-drift analysis framework), `QSSoftMW` is no worse than any reference random policy whose time-variation satisfies some smoothness condition. The stability of `QSSoftMW` then largely reduces to the stability of the reference random policy, for which we found a sufficient condition, generalizing the common depiction for the capacity region in stationary queueing systems. Below, we specifically give these two categories of technical assumptions for `QSSoftMW` to provide stability guarantees.

Assumption 1 (Piecewise Stabilizability [1]). *There exist $C_W \ge 0$, $\epsilon > 0$, $\vec{\theta}_1, \vec{\theta}_2, \cdots \in \Delta^{[K]}$ and a partition of $\mathbb{N}_+$ into intervals $W_0, W_1, \cdots$, such that for any $T \ge 1$ we have*

$$\sum_{i:\min_{t\in W_i} t<T} (|W_i|-1)^2 \le C_W T \tag{3}$$

and for any $\tau \ge 0$ and $i < j \in [N]$ we have

$$\frac{1}{|W_\tau|}\sum_{t\in W_\tau} \theta_{ij}(t)\mu_{ij}(t) \ge \epsilon + \frac{1}{|W_\tau|}\sum_{t\in W_\tau} A_{ij}(t). \tag{4}$$

Intuitively speaking, Assumption 1 states that we can partition the infinite time-horizon into finite-length intervals. The interval lengths may not be the same, as long as the growth rate is within $\mathcal{O}(\sqrt{T})$. Moreover, there exists a queue-length unaware scheduling policy, under which the probabilities to serve each user pair are encoded in the sequence $\{\vec{\theta}_t : t \ge 1\}$, such that for each user pair i, j, when we run this policy, in each time interval W_τ in the partition, the average service rate (the $\frac{1}{|W_\tau|}\sum_{t\in W_\tau}\theta_{ij}(t)\mu_{ij}(t)$ term in Eq. (4)) is at least ϵ more than the average arrival rate (the $\frac{1}{|W_\tau|}\sum_{t\in W_\tau} A_{ij}(t)$ term in Eq. (4)).
Remark. Assumption 1 is indeed a sufficient stability condition for the above randomized policy induced by $\{\vec{\theta}_t : t \ge 1\}$ (refer to Proposition 5.5 of [1] for details). Below we call this policy *the reference randomized policy.*

Assumption 2 (Reference Policy Stationarity [1]). *For the reference policy encoded by $\{\vec{\theta}_t\}$ in Assumption 1, there exist some $\delta > 0$ and $C_V > 0$ such that*

$$\sum_{t=1}^{T-1} \|\vec{\theta}_{t+1} - \vec{\theta}_t\|_1 \le C_V T^{\frac{1}{2}-\delta}$$

for any $T \ge 1$.

Intuitively speaking, Assumption 2 says that the sequence $\{\vec{\theta}_t\}$ (and hence the environment) does not change too fast (the sum of l_1-distances between the probability vector successive time steps is $\mathcal{O}(T^{\frac{1}{2}-\delta})$).

Now we are able to present the theoretical average queue length bound (and hence stability) result of `QSSoftMW` (Algorithm 1), as the following Theorem 1 which is a direct corollary of Theorem 5.2 of [1]:

Theorem 1. *For quantum switch scheduling problem instances satisfying Assumptions 1 and 2, `QSSoftMW` (Algorithm 1) guarantees*

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\|\mathbf{Q}_t\|_1\right] \le \frac{N^2M^2 + 2C_W(N^2M^2 + \epsilon N^2 M)}{\epsilon} + o(1)$$

where $M = \max\{A_{\max}, \mu_{\max}\}$. In particular, the system is stable.

Proof. `QSSoftMW` is a special instance of the `SoftMW` in [1] with the number of available actions being $K = \binom{N}{2}$. Thus, we can apply Theorem 5.2 of [1] for `QSSoftMW`, and the claimed bound is a direct corollary. □

We see that the queueing bound scales as $O(\frac{1}{\epsilon})$, which is similar to that under the `MaxWeight` [2] algorithm for settings where the success probabilities are stationary and known. This shows that by incorporating a proper adversarial learning mechanism, one can obtain a queueing bound that is of the same order (with respect to ϵ).

5. CONCLUSION

We propose a learning-based algorithm `QSSoftMW` for quantum switch scheduling, based on the framework developed in [1]. `QSSoftMW` combines adversarial learning and Lyapunov queue analysis, and is able to achieve queue stability under unknown and adversarial entanglement success probabilities.

6. REFERENCES

[1] J. Huang, L. Golubchik, and L. Huang. Queue scheduling with adversarial bandit learning. *arXiv preprint arXiv:2303.01745*, 2023.

[2] L. T. L. Georgiadis, M. J. Neely. Resource allocation and cross-layer control in wireless networks. In *Foundations and Trends in Networking, Vol. 1, no. 1, pp. 1-144*, 2006.

[3] D. T. N. Panigrahy, T. Vasantam and L. Tassiulas. On the capacity region of a quantum switch with entanglement purification. *arXiv preprint arXiv:2212.01463*, 2022.

[4] V. V. P. Promponas and L. Tassiulas. Full exploitation of limited memory in quantum entanglement switching. *arXiv preprint arXiv:2304.10602*, 2023.

[5] T. V. D. Towsley. Stability analysis of a quantum network with max-weight scheduling. In *arXiv:2106.00831*, 2021.

[6] D. T. W. Dai, A. Rinaldi. The capacity region of entanglement switching: Stability and zero latency. In *IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2022.