A throughput optimal scheduling policy for a quantum switch

Vasantam, Thirupathaiah, Towsley, Don

Thirupathaiah Vasantam, Don Towsley, "A throughput optimal scheduling policy for a quantum switch," Proc. SPIE 12015, Quantum Computing, Communication, and Simulation II, 1201505 (1 March 2022); doi: 10.1117/12.2616950

Event: SPIE OPTO, 2022, San Francisco, California, United States

A Throughput Optimal Scheduling Policy for a Quantum Switch

Thirupathaiah Vasantam[a] and Don Towsley[b]

[a]Department of Computer Science, Durham University, UK
[b]College of Information & Computer Sciences, University of Massachusetts, Amherst, USA

ABSTRACT

We study a quantum switch that creates shared end-to-end entangled quantum states to multiple sets of users that are connected to it. Each user is connected to the switch via an optical link across which bipartite Bell-state entangled states are generated in each time-slot with certain probabilities, and the switch merges entanglements of links to create end-to-end entanglements for users. One qubit of an entanglement of a link is stored at the switch and the other qubit of the entanglement is stored at the user corresponding to the link. Assuming that qubits of entanglements of links decipher after one time-slot, we characterize the capacity region, which is defined as the set of arrival rates of requests for end-to-end entanglements for which there exists a scheduling policy that stabilizes the switch. We propose a Max-Weight scheduling policy and show that it stabilizes the switch for all arrival rates that lie in the capacity region. We also provide numerical results to support our analysis.

Keywords: qubit, entanglements, switch, decoherence, Max-Weight, throughput, scheduling

1. INTRODUCTION

Quantum entanglement is a key component of quantum information systems that enables applications like quantum key distribution (QKD),[1,2] quantum sensing[3] (e.g., multipartite entanglement for quantum metrology[4,5]), and distributed quantum computing.[6] These applications motivate the need for a distributed infrastructure (quantum network) that will supply high quality (fidelity) bipartite and multipartite entanglements to groups of end users;[7–11] a quantum network consists of a collection of quantum switches connected to each other through optical links. Although several network architectures have been proposed to provide high entanglement rates at high fidelity,[12–16] there is still a long road ahead in designing efficient resource allocation algorithms and their performance analysis that can guide us to implement quantum networks at full-scale in future.

In this paper we focus on design and performance analysis of efficient resource allocation algorithms for a single quantum switch that serves incoming requests for end-to-end entanglements to M different groups of users, under the assumption that the switch is connected to K users. Each user is connected to the switch via a link across which Bell-pairs are generated between user and the switch, and each of the two nodes of a link stores one qubit of an entanglement of the link in quantum memories. When enough Bell-pairs are available at the links corresponding to a group of users, the switch performs a multi-qubit measurement to provide an end-to-end entanglement to the user group. If the switch has to connect two links, it uses Bell-state measurements (BSMs) and when it must connect three or more links, it uses Greenberger-Horne-Zeilinger (GHZ) basis measurements.[17]

We consider a time-slotted system where requests arrive according to a stochastic process. Within each type, requests are stored in an infinite capacity queue and processed according to First-Come-First-Served (FCFS) service discipline. In each time-slot, every link creates at most one entanglement, which decoheres after one time-slot.[18] Hence, at most one Bell-pair is available at each link in each time-slot to serve requests. Although the expectation is that eventually quantum networks will include switches with many long coherence time quantum memories, this will not be the case in the near term. For example, first generation quantum networks are likely to use controllable optical delay line buffers[19] to store single qubit at a time.

The main objective of the switch is to allocate available Bell-pairs in each time-slot cleverly to various requests so that they are processed as quickly as possible. We ask the following research question, what is the capacity region i.e., the set

Further author information: (Send correspondence to Thirupathaiah Vasantam)
Thirupathaiah Vasantam: E-mail: thirupathaiah.vasantam@durham.ac.uk, Telephone: +44 191 33 44504
Don Towsley: E-mail: towsley@cs.umass.edu, Telephone: +1 413 545 0207

Quantum Computing, Communication, and Simulation II, edited by Philip R. Hemmer,
Alan L. Migdall, Proc. of SPIE Vol. 12015, 1201505 · © 2022 SPIE
0277-786X · doi: 10.1117/12.2616950

of arrival rates for which there exists a scheduling policy under which the Markov chain associated with queues of requests have a stationary probability distribution with finite average waiting times of requests? Can we design a scheduling policy that stabilizes the switch for all the arrival rates that belong to the capacity region? In this paper, we address these questions by characterizing this capacity region and then proposing and analyzing a Max-Weight scheduling policy that stabilizes the switch for all the arrival rates that lie in the capacity region.

Related work: A simple quantum network that connects two users by a series of repeaters was studied in the literature.[20] The focus was to compute the expected waiting time required to create an end-to-end entanglement across a path with n links, under the assumption that each link creates a link-level entanglement with certain probability and measurement operations are successful probabilistically. The analysis uses Markov chain theory to compute the waiting times of requests, but closed-form expressions were only derived for networks with at most four segments.

The analysis of a single quantum switch connected to several users was investigated in previous works.[21,22] First, the rate at which a switch creates bipartite and tripartite entanglements was analyzed, under the assumption that it has capabilities to store one qubit and two qubits per each link.[21] Later, the analysis was extended to study the switch that generates end-to-end n-partite entanglements.[22] Using Lyapunov stability theory of Markov chains, it was proved that the switch is stable if and only if the number of attached links, K, is greater than or equal to n. Linear quantum networks with multiplexing capabilities have been studied in the literature[23–25] to improve end-to-end entanglement generation rates. Quantum networks could be implemented on several physical platforms, the implementation of quantum networks with multiplexing capabilities on dual-species trapped-ion systems was investigated in previous works.[25] A major drawback of previous works[21–25] is that, entangled states for users were created whenever there are enough link-level entanglements available across links, but they did not consider queues that store requests that are waiting for their service. In our modeling and analysis of the switch, similar to previous works,[23–25] we associate probabilities with various stochastic operations that affect how a quantum switch operates.

A Max-Weight scheduling policy was first introduced for resource allocation in communication networks[26] and later, this policy was adopted for the analysis of a single switch in classical networking[27] where they showed that the switch is stable for all feasible arrival rates under this policy. Although the Max-Weight policy has high implementation cost,[27] it lead to a significant progress on design and analysis of low complexity efficient scheduling algorithms in classical networking.[28] A major challenge in analyzing quantum networks is that they are more dynamic than classical networks due to the fact that several required operations to create end-to-end entanglements are probabilistic operations. Hence, both the design and analysis of scheduling policies must be modified to consider various characteristic properties of quantum networks. For example, if qubits of link-level entanglements decipher after multiple time-slots then the analysis of scheduling policies involves study of two-sided queues, in that one set of queues are used to store requests and the other set of queues are used to store qubits of link-level entanglements; analyzing two-sided queues is very difficult and they are often not needed to study classical networking problems. In this paper, we assume that qubits of entanglements decohere after one time-slot, to simplify the analysis. In a different context, a Max-Weight scheduling policy that is similar to ours was studied for networks with certain dynamic properties.[29,30] Our analysis is similar in spirit to the Lyapunov stability theory of Markov chains used in these works.[29]

Our Contributions: We make the following contributions:

- We derive necessary conditions on the request arrival rates for existence of a scheduling policy that stabilizes the switch.
- We propose a Max-Weight scheduling policy as a function of probability of successful creation of link-level entanglements and measurement operations, and dynamic queue sizes of requests. We prove that this policy stabilizes the switch for all feasible arrival rates using Lyapunov stability theory of Markov chains.
- Finally, we provide numerical results that corroborate our analysis.

The rest of the paper is organized as follows. In Section 2, we give details of the system model and then we give notation and some preliminary results in Section 3 where we also define our Max-Weight scheduling policy. In Section 4, we give necessary conditions on the request arrival rates for the stability of the switch and provide main results. We then discuss some numerical results in Section 5. Finally, we conclude in Section 6.

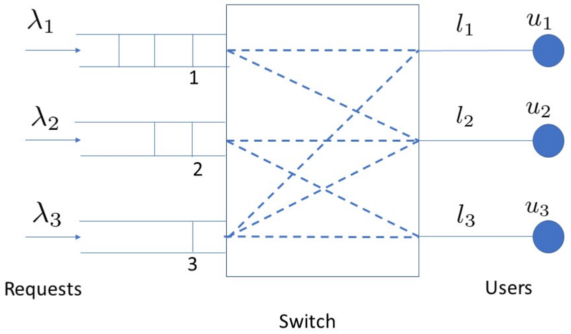


Figure 1: Switch creating end-to-end entanglements

2. SYSTEM MODEL

Consider a quantum switch that is connected to a set of K users, denoted by $\mathbb{U} \triangleq \{u_1, \cdots, u_K\}$, in a star topology with user u_i connected to the switch via link $l_i \in \mathbb{L} \triangleq \{l_1, \cdots, l_k\}$. Time is divided into fixed-length time-slots. One Bell-pair is generated across the link l_i in each time-slot with probability p_i; no entanglement is generated with probability $1 - p_i$. The Bell-pairs generated across links are referred to as link-level entanglements and the goal of the switch is to merge link-level entanglements to form end-to-end entanglements for various sets of users.

If a link-level entanglement is created between a user and the switch, then the switch stores one qubit of the entanglement in a quantum memory and the other qubit is stored at the user. The qubits of an entanglement are assumed to decoher after one time-slot. That is, if a link-level entanglement is created in time-slot n then that entanglement must be used to form an end-to-end entanglement in time-slot n, otherwise the link-level entanglement is considered to be wasted. As a result, the switch uses only one quantum memory to store qubits of entanglements generated across each link.

There are M types of requests arriving at the switch for end-to-end entanglements. A type i request is served when the switch creates an n_i-qubit end-to-end entanglement shared among a set of users denoted by $\mathcal{R}_i \triangleq \{u_{i1}, \cdots, u_{in_i}\}$. Let $\mathcal{L}_i$ denote the set of links whose Bell-pairs are required to serve a type i request, and let $\mathcal{X}_j$ denote the set of types of requests that require a link-level entanglement of link j. The process of creating an end-to-end entanglement shared among users $\mathcal{R}_i$ involves two steps; successful generation of Bell-pairs across links $\mathcal{L}_i$ followed by a successful entanglement swapping operation performed on the qubits stored at the switch, which is assumed to occur with probability q_i.

Let $A_i(n)$ denote the number of type i requests that arrive in time-slot n and the process $\{A_i(n)\}$ is an independent and identically distributed (i.i.d.) process with $\mathbb{E}[A_i(n)] = \lambda_i$ where λ_i is the rate at which type i requests arrive in each time-slot. The switch stores requests of each type in an infinite capacity queue and processes them in First-Come-First-Served (FCFS) basis. The main aim of the switch is to make scheduling decisions on how to allocate available link-level entanglements to different types of requests so as to serve requests with finite waiting times.

In Figure 1, we show a quantum switch that connects to three users where user u_i is connected to the switch via link l_i. There are three types of requests arriving in the system, each type of request seeks a creation of shared entanglement for a set of users. For type i requests, λ_i denotes the average number of requests arriving in each time-slot. From Figure 1, the set of users and links associated with different types of requests are: $\mathcal{R}_1 = \{u_1, u_2\}$, $\mathcal{R}_2 = \{u_2, u_3\}$, $\mathcal{R}_3 = \{u_1, u_2, u_3\}$, $\mathcal{L}_1 = \{l_1, l_2\}$, $\mathcal{L}_2 = \{l_2, l_3\}$, and $\mathcal{L}_3 = \{l_1, l_2, l_3\}$. There is a competition among different types of requests to use available link-level entanglements in each time-slot. For example, entanglements of link l_1 are used to serve both type 1 and type 3 requests.

In this paper, we address the following question. For a switch with K links and given system parameters $\mathbf{p} = [p_1, \cdots, p_K]$ and $\mathbf{q} = [q_1, \cdots, q_M]$, what is the capacity region of request rates that is defined as the set of request rates $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_M]$ for which there exists a scheduling policy that stabilizes the switch? In the next section, we define a Max-Weight scheduling policy, which is shown to stabilize the switch for all feasible arrival rates.

3. NOTATION AND PRELIMINARY RESULTS

We write vectors as bold-faced letters in the rest of the paper. We denote the number of type i requests that are waiting for service at the beginning of time-slot n by $Q_i(n)$. The number of link-level entanglements generated across link l_j in time-slot n is written as $T_j(n)$, where, $T_j(n) = 1$ with probability p_j and $T_j(n) = 0$ with probability $1 - p_j$. The states of the vector $\mathbf{T}(n) = (T_j(n))$ belong to the set $\mathcal{A}$, defined as

$$\mathcal{A} \triangleq \{\mathbf{a} = (a_1, \cdots, a_K) : a_i \in \{0, 1\}, 1 \le i \le K\}.$$

In time-slot n, since qubits of link-level entanglements decoher after one time-slot, the number of Bell-pairs available at link l_j is $T_j(n)$. When $T_j(n) = 1$, only one request can use it. The switch needs to decide how to process various requests using available link-level entanglements of links, in such a way that the average waiting times of requests are finite. Next we define a notion called matching, that is used in the process of allocating available link-level entanglements to requests.

DEFINITION 1. *Matching: We call $\boldsymbol{\pi} = [\pi_1, \cdots, \pi_M]$ a matching if $\pi_i \in \{0, 1\}$ and for each link l_j ($1 \le j \le K$), we have*

$$\sum_{i \in \mathcal{X}_j} \pi_i \le 1. \tag{1}$$

Furthermore, the vector $\boldsymbol{\pi}$ satisfies the condition that if $\pi_r = 0$ ($1 \le r \le M$) in $\boldsymbol{\pi}$, then the vector $\boldsymbol{\pi}^*$ obtained from $\boldsymbol{\pi}$ by replacing the r^{th} element $\pi_r = 0$ with $\pi_r = 1$ violates the condition (1).

Condition (1) guarantees that each link-level entanglement is assigned to at most one request. If the scheduler selects a matching $\boldsymbol{\pi}$ to decide which requests to be served in a time-slot, if $\pi_i = 1$, then the switch attempts to serve a type i request by performing a swapping operation on qubits of related links $\mathcal{L}_i$. Let $\mathcal{M}$ be the set defined as

$$\mathcal{M} \triangleq \{\boldsymbol{\pi} : \pi_i \in \{0, 1\}, \sum_{i \in \mathcal{X}_j} \pi_i \le 1, \forall l_j\}.$$

.

In time-slot n, the switch selects a matching from the set $\mathcal{M}$ based on $\mathbf{Q}(n) = (Q_i(n))$ and $\mathbf{T}(n) = (T_i(n))$. Intuitively, the switch should allocate available link-level entanglements to types of requests that have large queues, which guides us to define a Max-Weight scheduling policy for given quantum switch. Suppose that $\mathbf{W}(n) \in \mathcal{M}$ is the matching to be used in time-slot n, then we denote $r_i(\mathbf{T}(n), \mathbf{W}(n))$ to be the probability that a type i request is successfully served given that it is selected for service. To serve a type i request, first, the switch should make a decision to perform a relevant swapping operation, which happens if all the links in $\mathcal{L}_i$ have Bell-pairs and the selected matching $\mathbf{W}(n)$ satisfies $W_i(n) = 1$. Second, the subsequent swapping operation must succeed, which happens with probability q_i. As a result,

$$r_i(\mathbf{T}(n), \mathbf{W}(n)) = q_i I_{\{W_i(n)=1\}} I_{\{T_j(n)>0, \forall l_j \in \mathcal{L}_i\}}, \tag{2}$$

where, $I_{\{B\}}$ is the indicator function of the event B.

Next, we define the Max-Weight scheduling policy of interest below.

DEFINITION 2. *Max-Weight Scheduling: In time-slot n, the switch selects the matching $\mathbf{W}(n)$ computed as follows:*

$$\mathbf{W}(n) = \arg\max_{\boldsymbol{\pi} \in \mathcal{M}} \sum_{i=1}^{M} r_i(\mathbf{T}(n), \boldsymbol{\pi}) Q_i(n). \tag{3}$$

From (3), it is clear that $\mathbf{W}(n)$ is chosen to maximize the weighted sum of queue sizes of requests over the set $\mathcal{M}$ with weights corresponding to success probabilities of serving requests. This helps us to avoid congested queues.

If $\mathbf{W}(n)$ is the matching selected in time-slot n, then the number of entanglement swapping operations performed to serve type i requests is equal to $\min(Q_i(n), W_i(n))$. Next, we show how the process $\{\mathbf{Q}(n)\}$ evolves with time. Suppose that $Z_i(n) \in \{0, 1\}$ denotes whether an entanglement swapping operation performed on qubits of links $\mathcal{L}_i$ in time-slot

n succeeds or not. Variable $\mathbf{Z}(n)$ satisfies $Z_i(n) = 1$ if the entanglement swapping operation succeeds and $Z_i(n) = 0$, otherwise. Now define $D_i(n) \in \{0, 1\}$ to be the number of type $i$requests served in time-slot n. Then we have

$$D_i(n) = Z_i(n) I_{\{W_i(n)>0\}} I_{\{Q_i(n)>0\}} I_{\{T_j(n)>0, \forall l_j \in \mathcal{L}_i\}}. \tag{4}$$

The process $\{\mathbf{Q}(n)\}$ is a Markov chain that evolves according to the following relation,

$$\mathbf{Q}(n+1) = \mathbf{Q}(n) - \mathbf{D}(n) + \mathbf{A}(n), \tag{5}$$

where, $\mathbf{A}(n) = (A_i(n))$ and $\mathbf{D}(n) = (D_i(n))$. Note that the newly arrived requests in time-slot n, $\mathbf{A}(n)$, are not used to compute $\mathbf{D}(n)$, but rather they are used to determine $\mathbf{D}(n+1)$.

Our goal is to find necessary conditions on $\boldsymbol{\lambda}$ for existence of a scheduling policy under which the switch is stable, that there exists a stationary probability distribution for queue sizes of requests with finite average queues. We will show that our Max-Weight policy, stabilizes the switch for all arrival rates belonging to the capacity region as defined below.

DEFINITION 3. *Capacity Region: The set of request rates $\boldsymbol{\lambda}$ for which there exists a scheduling policy that stabilizes the switch.*

A scheduling policy is said to be throughput optimal if it stabilizes the switch for all arrival rates belonging to the capacity region. In the following remark, we will recall results on the analysis of a switch in classical networking, and then discuss how classical and quantum switch differ in the way they operate.

Remark 1. *In classical networking, a switch forwards packets from input ports to output ports, under the condition that in each time-slot, an input port forwards at most one packet to only one output port and an output port receives at most one packet from only one input port. Suppose that λ_{ij} denotes the average number of arriving packets per time-slot at the input port i to be transferred to the output port j. Define Λ' as*

$$\Lambda' = \{\mathbf{a} = [a_{ij}] : \sum_j a_{ij} \leq 1 \textit{ and } \sum_l a_{lm} \leq 1, \forall i, m\}.$$

Let $\mathcal{M}'$ denote the set of matchings used in classical networking defined as

$$\mathcal{M}' \triangleq \{\boldsymbol{\pi} = [\pi_{ij}] : \sum_j \pi_{ij} = 1 \textit{ and } \sum_l \pi_{lm} = 1, \forall i, m\}.$$

It was shown that if the switch selects the matching $W(n)$ computed according to the following Max-Weight scheduling policy then the switch is stable if $\boldsymbol{\lambda}$ lies inside Λ',[27]

$$\mathbf{W}(n) = \arg \max_{\pi \in \mathcal{M}'} \sum_{ij} \pi_{ij} Q_{ij}(n).$$

Furthermore, if $\boldsymbol{\lambda} \notin \Lambda'$, then no scheduling policy can stabilize the switch. We can view the quantum switch as the device with M input ports and K output ports, where each input port is associated with an application that generates requests for end-to-end entanglements and each output port is associated with a link. In every time-slot, the input port i is either matched to output ports $\mathcal{L}_i$ or not matched to any output port depending on whether $W_i(n) = 1$ or not. Furthermore, each output port is matched to at most one input port since each link has at most one link-level entanglement. If the input port i is matched to output ports, then it means that the switch has decided to serve a type i request.

In the next section, we will derive necessary conditions on $\boldsymbol{\lambda}$ for achieving the stability of the switch and show that the proposed Max-Weight scheduling policy achieves the stability of the switch for all arrival rates that lie inside the capacity region.

4. MAIN RESULTS

In this section, we present necessary conditions on arrival rates $\boldsymbol{\lambda}$ to achieve stability of the switch and prove that the Max-Weight policy stabilizes the switch for all feasible arrival rates using Lyapunov stability theory of Markov chains. We omit proofs of theorems presented in this section and provide them in an online report.

Next we provide a formal definition of stability of the switch.

DEFINITION 4. *Stability of the switch: We say that the quantum switch is stable if the sequence $\{\mathbf{Q}(n)\}$ converges in distribution to a random vector $\mathbf{Q}(\infty)$ with $\mathbb{E}[\mathbf{Q}(\infty)] < \infty$ for all initial states $\mathbf{Q}(0)$.*

In our proofs we use the condition that the process $\{\mathbf{Q}(n)\}$ is an irreducible Markov chain. The process $\{\mathbf{Q}(n)\}$ is an irreducible Markov chain under a scheduling policy if the following two conditions are satisfied. These are:

C_1 : If $\lambda_i > 0$, then there exists $\boldsymbol{\pi} \in \mathcal{M}$ such that $r_i(\mathbf{T}(n), \boldsymbol{\pi}) > 0$ for some $\mathbf{T}(n)$.

C_2 : If $\mathbf{Q}(n) \neq 0$, then there exists a matching $\boldsymbol{\pi}$ such that $r_i(\mathbf{T}(n), \boldsymbol{\pi}) > 0$ for a given state of $\mathbf{T}(n)$ for some i with $Q_i(n) > 0$, in this case the scheduling policy of interest must select a matching $\mathbf{W}(n)$ such that $r_j(\mathbf{T}(n), \mathbf{W}(n)) > 0$ for some j with $Q_j(n) > 0$.

If a scheduling policy satisfies conditions C_1 and C_2, then the process $\{\mathbf{Q}(n)\}$ is guaranteed to reach the empty state starting from any initial state. From (2) and (3), it is evident that the two conditions, C_1 and C_2, are satisfied under our Max-Weight scheduling policy. Hence, the process $\{\mathbf{Q}(n)\}$ is an irreducible Markov chain.

If the switch is stable under any scheduling policy then the request arrival rate coincides with the request departure rate in the stationary regime since we have

$$\boldsymbol{\lambda} = \lim_{n\to\infty} \frac{\sum_{j=1}^{n} \mathbf{D}(j)}{n}, \qquad a.s. \tag{6}$$

Next, we derive necessary conditions on $\boldsymbol{\lambda}$ for existence of a scheduling policy that stabilizes the switch. If the switch is stable under a scheduling policy, then we denote $\mathbf{X}(\infty)$ to be the random vector with stationary probability distribution of $\mathbf{X}(n)$. Let $c_{a,\boldsymbol{\pi}}$ be defined as

$$c_{a,\boldsymbol{\pi}} = \mathbb{P}(\min(\mathbf{W}(\infty), \mathbf{Q}(\infty)) = \boldsymbol{\pi},\ \mathbf{Q}(\infty) \neq \mathbf{0} | \mathbf{T}(\infty) = \mathbf{a}),$$

where $\min(\mathbf{W}(\infty), \mathbf{Q}(\infty)) = (\min(W_i(\infty), Q_i(\infty)))$ indicates the number of entanglement swapping operations performed for each type of requests in time-slot n, and $c_{a,\boldsymbol{\pi}}$ denotes the stationary probability that $\min(\mathbf{W}(\infty), \mathbf{Q}(\infty)) = \boldsymbol{\pi}$ and $\mathbf{Q}(\infty) \neq \mathbf{0}$ given that $\mathbf{T}(\infty)) = \mathbf{a}$. Note that the process $\{\mathbf{T}(n)\}$ is a stationary process with the property that $\mathbb{P}(\mathbf{T}(n) = 1) = p_i$ and $\mathbb{P}(\mathbf{T}(n) = 0) = 1 - p_i$.

THEOREM 4.1. *If there exists a scheduling policy that stabilizes the switch with $\{\mathbf{Q}(n)\}$ being an irreducible Markov chain, then using (6) we show that $\boldsymbol{\lambda}$ satisfies*

$$\lambda = \sum_{\{\mathbf{a}\in\mathcal{A}, \mathbf{a}\neq\mathbf{0}\}} \mathbb{P}(\mathbf{T}(n) = \mathbf{a}) \sum_{\boldsymbol{\pi}\in\mathcal{M}} c_{\mathbf{a},\boldsymbol{\pi}} \mathbf{r}(\mathbf{a}, \boldsymbol{\pi}), \tag{7}$$

where, $\mathbf{r}(\mathbf{a}, \boldsymbol{\pi}) = (r_i(\mathbf{a}, \boldsymbol{\pi}))$, $c_{\mathbf{a},\boldsymbol{\pi}} > 0$, and $\sum_{\boldsymbol{\pi}} c_{\mathbf{a},\boldsymbol{\pi}} < 1$ for all $\mathbf{a} \in \mathcal{A}$ with $\mathbf{a} \neq \mathbf{0}$.

From (7), we can write

$$\begin{aligned}
\lambda = & \sum_{\{\mathbf{a}\in\mathcal{A}, \mathbf{a}\neq\mathbf{0}\}} \mathbb{P}(\mathbf{T}(n) = \mathbf{a}) \sum_{\boldsymbol{\sigma}\in\mathcal{M}} \sum_{\boldsymbol{\pi}\in\mathcal{M}} \mathbb{P}(\mathbf{W}(\infty) = \boldsymbol{\sigma}, \min(\mathbf{W}(\infty), \mathbf{Q}(\infty)) = \boldsymbol{\pi},\ \mathbf{Q}(\infty) \neq \mathbf{0} | \mathbf{T}(\infty) = \mathbf{a}) \mathbf{r}(\mathbf{a}, \boldsymbol{\pi}) \\
\leq & \sum_{\{\mathbf{a}\in\mathcal{A}, \mathbf{a}\neq\mathbf{0}\}} \mathbb{P}(\mathbf{T}(n) = \mathbf{a}) \sum_{\boldsymbol{\sigma}\in\mathcal{M}} \sum_{\boldsymbol{\pi}\in\mathcal{M}} \mathbb{P}(\mathbf{W}(\infty) = \boldsymbol{\sigma}, \min(\mathbf{W}(\infty), \mathbf{Q}(\infty)) = \boldsymbol{\pi},\ \mathbf{Q}(\infty) \neq \mathbf{0} | \mathbf{T}(\infty) = \mathbf{a}) \mathbf{r}(\mathbf{a}, \boldsymbol{\sigma}) \\
= & \sum_{\{\mathbf{a}\in\mathcal{A}, \mathbf{a}\neq\mathbf{0}\}} \mathbb{P}(\mathbf{T}(n) = \mathbf{a}) \sum_{\boldsymbol{\sigma}\in\mathcal{M}} \mathbb{P}(\mathbf{W}(\infty) = \boldsymbol{\sigma},\ \mathbf{Q}(\infty) \neq \mathbf{0} | \mathbf{T}(\infty) = \mathbf{a}) \mathbf{r}(\mathbf{a}, \boldsymbol{\sigma}) \\
= & \sum_{\{\mathbf{a}\in\mathcal{A}, \mathbf{a}\neq\mathbf{0}\}} \mathbb{P}(\mathbf{T}(n) = \mathbf{a}) \sum_{\boldsymbol{\sigma}\in\mathcal{M}} b_{\mathbf{a},\boldsymbol{\sigma}} \mathbf{r}(\mathbf{a}, \boldsymbol{\sigma}),
\end{aligned} \tag{8}$$

where $b_{\mathbf{a},\boldsymbol{\sigma}} = \mathbb{P}(\mathbf{W}(\infty) = \boldsymbol{\sigma},\ \mathbf{Q}(\infty) \neq \mathbf{0} | \mathbf{T}(\infty) = \mathbf{a})$.

Using Theorem 4.1 and (8), we characterize the capacity region as follows.

DEFINITION 5. *Capacity region: The capacity region is defined as*

$$\Lambda \triangleq \left\{ \boldsymbol{\lambda} : \exists\{b_{\mathbf{a},\boldsymbol{\pi}}, \mathbf{a} \in \mathcal{A}, \boldsymbol{\pi} \in \mathcal{M}\} \textit{ s.t. } \boldsymbol{\lambda} \leq \sum_{\{\mathbf{a}\in\mathbb{A},\mathbf{a}\neq\mathbf{0}\}} \mathbb{P}(\mathbf{T}(n) = \mathbf{a}) \sum_{\boldsymbol{\pi}} b_{\mathbf{a},\boldsymbol{\pi}} \mathbf{r}(\mathbf{a},\boldsymbol{\pi}),\ b_{\mathbf{a},\boldsymbol{\pi}} > 0,\ \sum_{\boldsymbol{\pi}} b_{\mathbf{a},\boldsymbol{\pi}} < 1, \forall \mathbf{a} \right\}. \tag{9}$$

If $\lambda \notin \mathbf{\Lambda}$, then the switch cannot be stabilized under any scheduling policy as this would contradict the results of Theorem 4.1.

Next, we prove that the Max-Weight scheduling policy stabilizes the switch for all arrival rates in the capacity region. For this, we apply a Lyapunov stability theorem of Markov chains [26, Theorem 3.1], using the following Lyapunov function

$$V(\mathbf{Q}(n)) = \sum_{i=1}^{M} Q_i(n)^2.$$

It suffices to show that

$$\mathbb{E}\left[V(\mathbf{Q}(n+1)) - V(\mathbf{Q}(n)) | \mathbf{Q}(n)\right] \leq -\epsilon \|\mathbf{Q}(n)\|, \tag{10}$$

for sufficiently large $\|\mathbf{Q}(n)\|$, where $\|\mathbf{Q}(n)\| = \sqrt{\sum_{i=1}^{M} Q_i(n)^2}$, and $\epsilon > 0$. Finally, we state the main result on the stability of the switch under our Max-Weight scheduling policy in the following theorem.

THEOREM 4.2. *If $\boldsymbol{\lambda} \in \Lambda$ and $\mathbb{E}\left[A_i^2(n)\right] < \infty$ for all $1 \leq i \leq M$, then the Max-Weight scheduling policy defined in Definition 1 stabilizes the switch.*

5. NUMERICAL RESULTS

In this section, we provide numerical results that support our analysis. We simulate the switch shown in Figure 1 to understand the behavior of the process $\{\overline{\mathbf{Q}}(n)\}$ for various parameters, where $\overline{\mathbf{Q}}(n) = \frac{\sum_{i=1}^{M} Q_i(n)}{M}$.

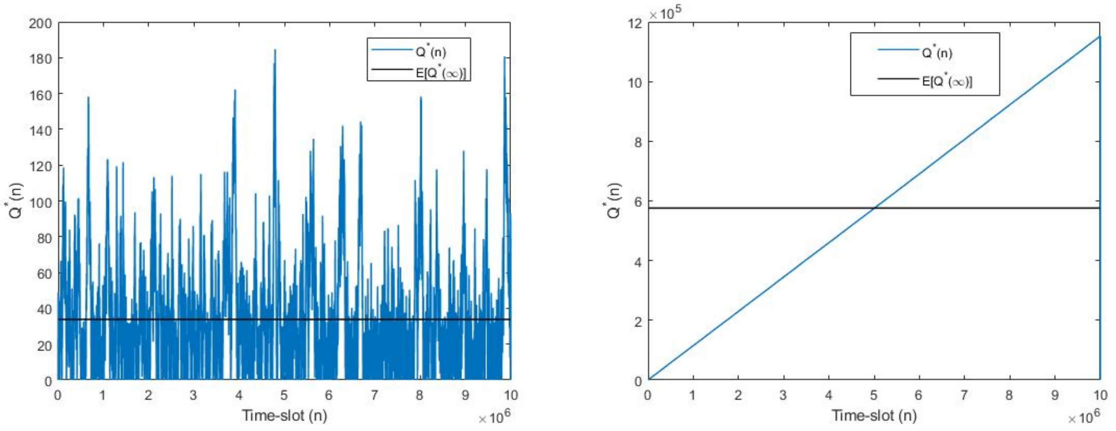
In Figure 2, we plot $\overline{\mathbf{Q}}(n)$ as a function of n for $\mathbf{p}$ =[0.7 0.8 0.6], and $\mathbf{q}$ =[0.9 0.8 0.7]. In Figure 2a, for λ =[0.35 0.2 0.15], we observe that switch is stable for the considered parameters and the stationary average queue size denoted by $\mathbb{E}[\overline{\mathbf{Q}}(\infty)]$ is finite as shown in the figure, where $\mathbb{E}[\overline{\mathbf{Q}}(\infty)] = \frac{\sum_{n=1}^{N} \overline{\mathbf{Q}}(n)}{N}$ with $N = 10^7$. In Figure 2b, we study the switch assuming higher request arrival rates than the arrival rates considered in Figure 2a. For λ =[0.45 0.35 0.25], we observe that $\overline{\mathbf{Q}}(n)$ increases monotonically with n as shown in Figure 2b implying that the switch is unstable and $\mathbb{E}[\overline{\mathbf{Q}}(\infty)]$ is very large.

In Figure 3, we study how the average queue sizes of requests change with $\boldsymbol{\gamma}$ for the parameters λ =[0.35 0.2 0.15], $\mathbf{p}$ =[γ γ γ], and $\mathbf{q}$ =[0.9 0.8 0.7]. We plot $\mathbb{E}[\overline{\mathbf{Q}}(\infty)]$ as a function of γ in Figure 3a for $\gamma \in [0.5, 0.95]$. We observe that the switch is unstable when $\gamma < 0.75$ due to the fact that there are not enough link-level entanglements available in each time-slot to serve requests stored in queues. The average queue sizes of requests decrease with link-level entanglement generation rate γ. In Figure 3b, for $\gamma \geq 0.75$, we observe that the average queue sizes of requests are small and decrease with γ. Our numerical results support the importance of characterizing the capacity region of the switch for given $\mathbf{p}$ and $\mathbf{q}$.

6. CONCLUSIONS

In this paper, we have investigated stability properties of a quantum switch that provide insights into performance of the switch. We proposed a Max-Weight scheduling policy that takes into account for differences in various parameters so as to achieve good performance. We also prooved that the proposed policy stabilizes the switch for all feasible arrival rates. Although our policy has high implementation cost due to the fact that it requires the switch to search over all possible matchings to find the best matching in each time-slot, it provides insights into how to design low complexity scheduling algorithms and also, its performance acts as a benchmark to the performance of other policies.

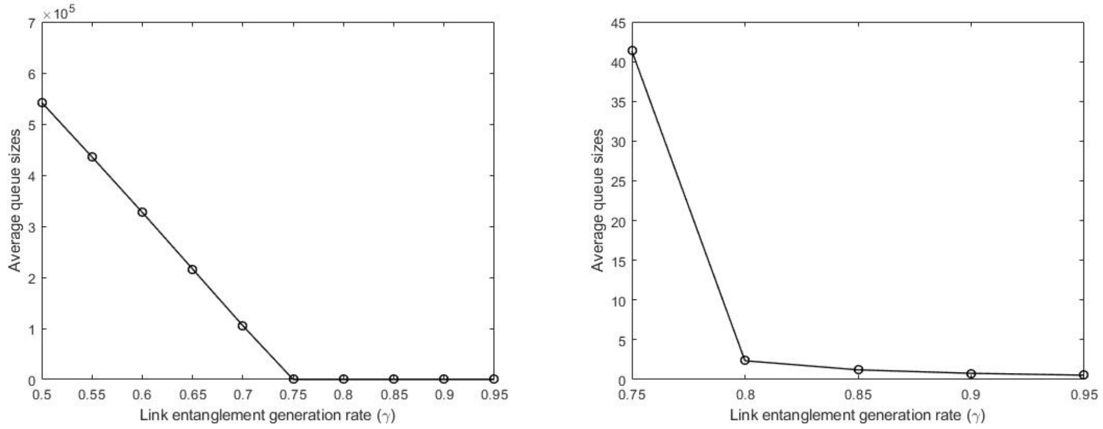
We plan to address several important problems in future work. We would like to investigate the design and analysis of scheduling algorithms that have low implementation costs. We also plan to study the case where Bell-pairs take more than one time-slot to decoher. Finally, it is of interest to analyze scheduling algorithms for distribution of entangled states over quantum networks and also consider the effect of entanglement purification procedures into the design of scheduling algorithms.



(a) Stable switch

(b) Unstable switch

Figure 2: Evolution of queue sizes



(a) For $\gamma \in [0.5, 0.95]$

(b) For $\gamma \geq 0.75$

Figure 3: Average queue sizes versus γ

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grants CNS-1955834 and ERC-1941583.

REFERENCES

[1] Bennett, C. H. and Brassard, G., "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.* **560**, 7–11 (2014).

[2] Ekert, A. K., "Quantum cryptography based on bell's theorem," *Phys. Rev. Lett.* **67**, 661–663 (Aug 1991).

[3] Eldredge, Z., Foss-Feig, M., Gross, J. A., Rolston, S. L., and Gorshkov, A. V., "Optimal and secure measurement protocols for quantum sensor networks," *Phys. Rev. A* **97**, 042337 (Apr 2018).

[4] Giovannetti, V., Lloyd, S., and Maccone, L., "Advances in quantum metrology," *Nature Photonics* **5**, 222–229 (mar 2011).

[5] Xia, Y., Li, W., Clark, W., Hart, D., Zhuang, Q., and Zhang, Z., "Demonstration of a Reconfigurable Entangled Radiofrequency-Photonic Sensor Network," *Phys. Rev. Lett.* **124**(15), 150502 (2020).

[6] Broadbent, A., Fitzsimons, J., and Kashefi, E., "Universal blind quantum computation," in [*Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS '09)*], *Annual Symposium on Foundations of Computer Science*, 517–526, Institute of Electrical and Electronics Engineers (IEEE), United States (2009).

[7] Pirandola, S., "End-to-end capacities of a quantum communication network," *Communications Physics* **2** (May 2019).

[8] Pant, M., Krovi, H., Towsley, D., Tassiulas, L., Jiang, L., Basu, P., Englund, D., and Guha, S., "Routing entanglement in the quantum internet," *npj Quantum Information* **5** (Dec. 2019).

[9] Dahlberg, A., Skrzypczyk, M., Coopmans, T., Wubben, L., Rozpundefineddek, F., Pompili, M., Stolk, A., Pawełczak, P., Knegjens, R., de Oliveira Filho, J., Hanson, R., and Wehner, S., "A link layer protocol for quantum networks," in [*Proceedings of the ACM Special Interest Group on Data Communication*], *SIGCOMM '19*, 159–173, Association for Computing Machinery, New York, NY, USA (2019).

[10] Van Meter, R., [*Quantum Networking*], vol. 9781848215375, Wiley Blackwell (June 2014).

[11] Bhaskar, M. K., Riedinger, R., Machielse, B., Levonian, D. S., Nguyen, C. T., Knall, E. N., Park, H., Englund, D., Lončar, M., Sukachev, D. D., and et al., "Experimental demonstration of memory-enhanced quantum communication," *Nature* **580**, 60–64 (Mar 2020).

[12] Lee, Y., Bersin, E., Dahlberg, A., Wehner, S., and Englund, D., "A quantum router architecture for high-fidelity entanglement flows in quantum networks," (2020).

[13] Li, R., Petit, L., Franke, D., Dehollain, J., Helsen, J., Steudtner, M., Thomas, N., Wehner, S., Vandersypen, L., and Veldhorst, M., "A crossbar network for silicon quantum dot qubits," *Science Advances* **4** (July 2018).

[14] Armstrong, S., Morizur, J.-F., Janousek, J., Hage, B., Treps, N., Lam, P. K., and Bachor, H.-A., "Programmable multimode quantum networks," *Nature Communications* **3** (Jan 2012).

[15] Herbauts, I., Blauensteiner, B., Poppe, A., Jennewein, T., and Hübel, H., "Demonstration of active routing of entanglement in a multi-user network," *Opt. Express* **21**, 29013–29024 (Nov 2013).

[16] Hall, M. A., Altepeter, J. B., and Kumar, P., "Ultrafast switching of photonic entanglement," *Phys. Rev. Lett.* **106**, 053901 (Feb 2011).

[17] Nielsen, M. A. and Chuang, I. L., [*Quantum Computation and Quantum Information*], Cambridge University Press (2000).

[18] Li, B., Coopmans, T., and Elkouss, D., "Efficient optimization of cut-offs in quantum repeater chains," *Proceedings - IEEE International Conference on Quantum Computing and Engineering, QCE 2020*, 158–168 (2020).

[19] Burmeister, E. F., Mack, J. P., Poulsen, H. N., Klamkin, J., Coldren, L. A., Blumenthal, D. J., and Bowers, J. E., "Soa gate array recirculating buffer with fiber delay loop," *Opt. Express* **16**, 8451–8456 (Jun 2008).

[20] Shchukin, E., Schmidt, F., and van Loock, P., "Waiting time in quantum repeaters with probabilistic entanglement swapping," *Phys. Rev. A* **100**, 032322 (Sep 2019).

[21] Vardoyan, G., Guha, S., Nain, P., and Towsley, D., "On the Capacity Region of Bipartite and Tripartite Entanglement Switching," in [*Performance 2020 - 38th IFIP International Symposium on Computer Performance, Modeling, Measurements and Evaluation*], 1–6 (Nov. 2020).

[22] Nain, P., Vardoyan, G., Guha, S., and Towsley, D., "On the analysis of a multipartite entanglement distribution switch," *Proc. ACM Meas. Anal. Comput. Syst.* **4** (June 2020).

[23] Guha, S., Krovi, H., Fuchs, C. A., Dutton, Z., Slater, J. A., Simon, C., and Tittel, W., "Rate-loss analysis of an efficient quantum repeater architecture," *Phys. Rev. A* **92**, 022357 (Aug 2015).

[24] Dhara, P., Patil, A., Krovi, H., and Guha, S., "Sub-exponential rate versus distance with time multiplexed quantum repeaters," (2021).

[25] Dhara, P., Linke, N. M., Waks, E., Guha, S., and Seshadreesan, K. P., "Multiplexed quantum repeaters based on dual-species trapped-ion systems," (2021).

[26] Tassiulas, L. and Ephremides, A., "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in [*29th IEEE Conference on Decision and Control*], 2130–2132 vol.4 (1990).

[27] McKeown, N., Mekkittikul, A., Anantharam, V., and Walrand, J., "Achieving 100% throughput in an input-queued switch," *IEEE Transactions on Communications* **47**(8), 1260–1267 (1999).

[28] Srikant, R. and Ying, L., [*Communication Networks: An Optimization, Control and Stochastic Networks Perspective*], Cambridge University Press (2014).

[29] Tassiulas, L., "Scheduling and performance limits of networks with constantly changing topology," *IEEE Transactions on Information Theory* **43**(3), 1067–1073 (1997).

[30] Andrews, M., Kumaran, K., Ramanan, K., Stolyar, A., Vijayakumar, R., and Whiting, P., "Scheduling in a queuing system with asynchronously varying service rates," *Probability in the Engineering and Informational Sciences* **18**(2), 191–217 (2004).