An Architecture for Control of Entanglement Generation Switches in Quantum Networks

Gauthier, Scarlett; Vardoyan, Gayane; Wehner, Stephanie

DOI
10.1109/TQE.2023.3320047

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Quantum Engineering

IEEE Transactions on Quantum Engineering

Received 28 July 2023; revised 17 September 2023; accepted 18 September 2023; date of publication 27 September 2023; date of current version 4 December 2023.

Digital Object Identifier 10.1109/TQE.2023.3320047

An Architecture for Control of Entanglement Generation Switches in Quantum Networks

SCARLETT GAUTHIER[1,2], GAYANE VARDOYAN[1,2,3], AND STEPHANIE WEHNER[1,2,4]

[1]QuTech, Delft University of Technology, 2628 CC Delft, The Netherlands
[2]Quantum Computer Science, Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CC Delft, The Netherlands
[3]University of Massachusetts, Amherst, MA 01003 USA
[4]Kavli Institute of Nanoscience, Delft University of Technology, 2628 CC Delft, The Netherlands

Corresponding author: Scarlett Gauthier (e-mail: s.s.gauthier@tudelft.nl).

This work was supported by European Union's Horizon Europe research and innovation program under Grant 101102140. This is an extended version of a workshop paper "A Control Architecture for Entanglement Generation Switches in Quantum Networks," QuNet '23: Proceedings of the 1st Workshop on Quantum Networks and Distributed Quantum Computing [DOI: 10.1145/3610251.3610552].

ABSTRACT Entanglement between quantum network nodes is often produced using intermediary devices—such as heralding stations—as a resource. When scaling quantum networks to many nodes, requiring a dedicated intermediary device for every pair of nodes introduces high costs. Here, we propose a cost-effective architecture to connect many quantum network nodes via a central quantum network hub called an entanglement generation switch (EGS). The EGS allows multiple quantum nodes to be connected at a fixed resource cost, by sharing the resources needed to make entanglement. We propose an algorithm called the rate control protocol, which moderates the level of competition for access to the hub's resources between sets of users. We proceed to prove a convergence theorem for rates yielded by the algorithm. To derive the algorithm we work in the framework of network utility maximization and make use of the theory of Lagrange multipliers and Lagrangian duality. Our EGS architecture lays the groundwork for developing control architectures compatible with other types of quantum network hubs as well as system models of greater complexity.

INDEX TERMS Central quantum network hub, control protocol, entanglement generation, network utility maximization (NUM), resource sharing.

I. INTRODUCTION

A quantum network enables radically new capabilities that are provably impossible to attain in any classical network [1]. Examples include applications, such as secure communication [2], [3], secure quantum computing in the cloud [4], [5], and clock synchronization [6]. Users utilize the end nodes of a network to run applications. The key to unlocking widespread roll-out of these applications is the ability to produce entanglement between these end nodes.

Prevalent methods for generating entanglement between two quantum nodes that are directly connected by a quantum communication medium (e.g., optical fibers) involve an intermediate device. A prime example is heralded entanglement generation [7], [8] in which the intermediary device is a so-called heralding station. This method of producing entanglement has successfully been demonstrated in many experimental platforms including Color Centers [9], [10], Ion Traps [11], [12], Atomic Ensembles [13], [14], and Neutral Atoms [15]. As quantum networks continue to scale, it becomes increasingly impractical to maintain direct fiber connections and dedicated heralding stations for every pair of end nodes.

To address this challenge, we propose a scalable quantum network architecture for an entanglement generation switch (EGS), a central hub equipped with a limited number of intermediate devices called resources, a switch, and a processor responsible for managing a scheduling algorithm and sending classical messages to nodes. This central hub enables multiple nodes to share the intermediate devices, significantly reducing the complexity and total resources required for large-scale deployment. While our results

apply to an EGS sharing any type of entanglement generation resource, a specific example illustrates how an EGS can operate: consider quantum network nodes that generate entanglement between them using the so-called single-click bipartite entanglement generation protocol (see, e.g., [10]). In this case, the resource(s) to be shared are the heralding station(s). Such stations consist of two input channels connected to a 50/50 beam splitter, which is then connected by two output channels to a pair of photon detectors that are each connected to a device for processing the measurement outcomes, such as a field programmable gate array (FPGA). The basic principle of the single-click protocol requires that each network node of the pair locally generates entanglement between a qubit in their local memory and a travelling photon. The photon is sent to a heralding station at which an entanglement swap is attempted on the two photons received; if the entanglement swap is successful, the qubits of the two network nodes will have become entangled. An EGS aims to share one or more heralding stations amongst many connected network nodes. These nodes will still run the single-click protocol, but be limited to using the heralding station needed in the time allocated to them by the EGS.

A crucial challenge in implementing such an architecture is the efficient allocation of the central hub's resources to different pairs of users in distinct time slots. Similar to classical networking, the allocation process should be driven by user demand for network resources. In the context of quantum networks, this translates to the demand of a user pair $(u_i,\ u_j)$ for entanglement generation at a specific rate or fidelity. Given a set of user demands, the EGS must compose a schedule for the allocation of resources in order to service those demands. In general, the total demand of users may exceed the available resources at the central hub, leading to scheduling and resource allocation challenges.

Here, we introduce the first algorithm for regulating user demand to an EGS, thereby solving this key challenge. Specifically, the algorithm takes as input a vector of rates of entanglement generation demanded by pairs of users and outputs an updated rate vector. The current set of user-originated demands is a measure of competition for EGS resources. We construct the algorithm within the network utility maximization (NUM) framework, wherein the problem of demand regulation is cast as a constrained optimization problem. To solve the problem, we derive the algorithm by using the theory of Lagrange multipliers and Lagrangian duality. These tools, respectively, enable including the constraints together with the objective of the optimization problem and solving for a parameter vector which is the unknown value of the combined problem. Regulating competition for the resources by modifying user demand makes it possible to enforce a notion of fairness in the allocation of resources and maximize resource utilization. Since the algorithm regulates competition by calculating the rates demanded by users, we call it the rate control protocol (RCP).

A. RESULTS SUMMARY

We make the following contributions.

1) We characterize (Theorem II.1) the capacity region of the EGS, which is the maximal set of rates at which users can demand entanglement generation such that there exists a scheduling policy under which, on average, the demanded rates do not exceed the delivered rates. The impact of specifying the capacity region is that it delineates which rates can feasibly be serviced by the EGS.
2) We prove (Theorem II.1) that under the maximum weight scheduling policy (Definition II.6) for resource allocation it is possible for the EGS to deliver average rates of entanglement generation that match the requested rates, for any rate vector from within the capacity region. Therefore, an EGS operated with this scheduling policy can achieve throughput optimality as long as the rates demanded by users lie within the capacity region. To prove the theorem, we use the Lyapunov stability theory of Markov chains.
3) We derive the RCP, an algorithm to regulate the rates of bipartite entanglement generation which pairs of users demand from an EGS. The RCP solves the problem of moderating user competition for EGS resources. The derivation is based on techniques from NUM and its quantum network extension (QNUM), where resource allocation in a (quantum) network is modeled as an optimization problem that can be solved using methods from convex optimization theory.
4) We prove (Theorem III.1) that the sequence of arrival rate vectors yielded by the RCP converges over time slots to an optimum value, given any feasible rate vector as initial condition. The significance of this result is that if the RCP is used to set the demand rates of entanglement generation over a series of time-slots, the set of demanded rates will approach an optimal value, as long as the initial rate vector supplied to the algorithm is feasible. The proof relies on the Lagrange multipliers and Lagrangian duality theory.
5) Finally, we supply numerical results that support our analysis.

B. RELATED WORK

A quantum network hub that can store locally at least one qubit per linked node and distributes entanglement across these links has been studied [16], [17]. We refer to such a hub as an entanglement distribution switch (EDS). This system differs from our system because the central hub has qubits and/or quantum memories, whereas our system does not. In [16], the focus is on assessing the EDS performance in terms of the rate at which it creates $n-$partite entanglements, and in [17] the possible rate/fidelity combinations of GHZ states that may be supplied by an EDS [17] are studied.

IEEE Transactions on Quantum Engineering

Maximum weight scheduling is a type of solution to the problem of resource allocation which is based on assigning resources to sets of users with the largest service backlog. A maximum weight scheduling policy was originally presented in [18] for resource allocation in classical communication networks and was adapted to the analysis of a single switch for classical networking in [19], where it was shown that under this scheduling policy the set of request arrival rates matches the request departure rates (or in other words the policy stabilizes the switch for all feasible arrival rates). In [20], the capacity region of an EDS, defined as the set of arrival rates of requests for end-to-end multipartite entanglements that stabilize the switch, is first characterized. Using the Lyapunov stability theory of Markov chains, a maximum weight scheduling policy is proposed and shown to stabilize the switch for all arrival rates within the capacity region. To summarize, in each of the classical network settings and in the EDS setting a maximum weight scheduling policy has the merit of achieving a specified performance metric. None of these results are immediately applicable to our system. We demonstrate that such a policy achieves the performance metric of throughput optimality when applied to the EGS by first characterizing the capacity region of the EGS, which has not been done before, and then proving that a maximum weight scheduling policy also achieves throughput optimality in our system.

These results on the analysis of EDS systems constitute the first analytic approaches to resource allocation by a quantum network hub. However, due to the assumption that an EDS locally controls some number of qubits per link, the system has a high technical implementation cost which may not be compatible with near-term quantum networks. Moreover, although these works assume that there is competition between multiple sets of users, the focus is purely on the capacity of the EDS system. Conversely, our analytic contributions apply to EGS quantum network hubs, which have a low technical implementation cost because the hub does not require local control of any qubits or quantum memory. Furthermore, our results extend beyond the analysis of the capacity of the EGS and we propose the RCP as a solution to the problem of moderating competition for the EGS resources.

In [21], a quantum network topology is studied where user-controlled nodes are connected through a hub known as a Qonnector. The Qonnector provides the necessary hardware for limited end nodes to execute applications in pairs or small groups. A potential configuration of the Qonnector is as an EGS. While [21] focuses on assessing the performance of certain applications in this topology, it does not address control policies for the system. In contrast, our work examines control policies for an EGS.

NUM was first introduced in [22] and has been widely used to develop and analyze control policies for classical networks [23]. It is a powerful framework for designing and analyzing communication protocols in classical networks wherein the problem of allocating resources amongst competing sets of users is cast as a constrained optimization problem. This framework was recently extended to QNUM by [24]. Therein, the authors first develop three performance metrics and use them to catalogue the utility of resource allocation in a quantum network model where each link is associated with a rate and fidelity of entanglement delivery to communicating users. This work does not immediately extend to control policies, as the resource allocations investigated are based on static numerical optimization and need to be recalculated in response to changes in the constraints or sets of users.

In classical networks, probabilistic failures, such as loss of a message during transmission or irreconcilable distortion due to transmission over a noisy channel may occur. A serious challenge introduced in the analysis of quantum networks is that in addition to the failure modes of a classical network several new probabilistic failure modes arise that are independent of the state of the network but nevertheless affect its ability to satisfy demands. An example is the probabilistic success in practical realizations of heralded entanglement generation [9], [10], [11], [12], [13], [14], [15]. Due to this failure mode, scheduling access to a resource at a certain rate does not guarantee entanglement generation at that rate, thereby complicating the analysis of scheduling.

It is important to distinguish between the concept of rates in classical network control protocols and the notion of rate in the model of a quantum network hub presented here. In classical networks, users transmit *data* at some rate and classical network control protocols, such as the transmission control protocol (TCP), regulate the rate at which users send their data [23]. In contrast, in our quantum network hub model, users demand a rate of entanglement generation. However, a significant challenge in developing a control protocol for the EGS is the difference between the rate of attempted entanglement generation and the rate at which entanglement delivery is demanded and delivered to users. Explicitly, in the RCP it is the desired rates of entanglement generation that serve as the controllable parameters moderated by the protocol.

II. PRELIMINARIES

Operation of the EGS requires interactions between the set of quantum network nodes U and the EGS processor with control over R resources. See Fig. 1(a) for an overview of the physical architecture. We delineate the process by which pairs of nodes may request [Fig. 1(d)] and receive [Fig. 1(b) and (d)] resource allocations from the processor in the following. We assume the following.

1) The EGS operates in a fixed-duration time slotted system where t_n denotes the nth time slot.
2) Timing synchronization between the processor and each node is continuously managed by classical control electronics at the physical layer.
3) Allocation of a single resource to communication session s for one time slot allows for the creation of a maximum of one entangled pair with a success probability of p_{gen}. A consistent physical model involves a

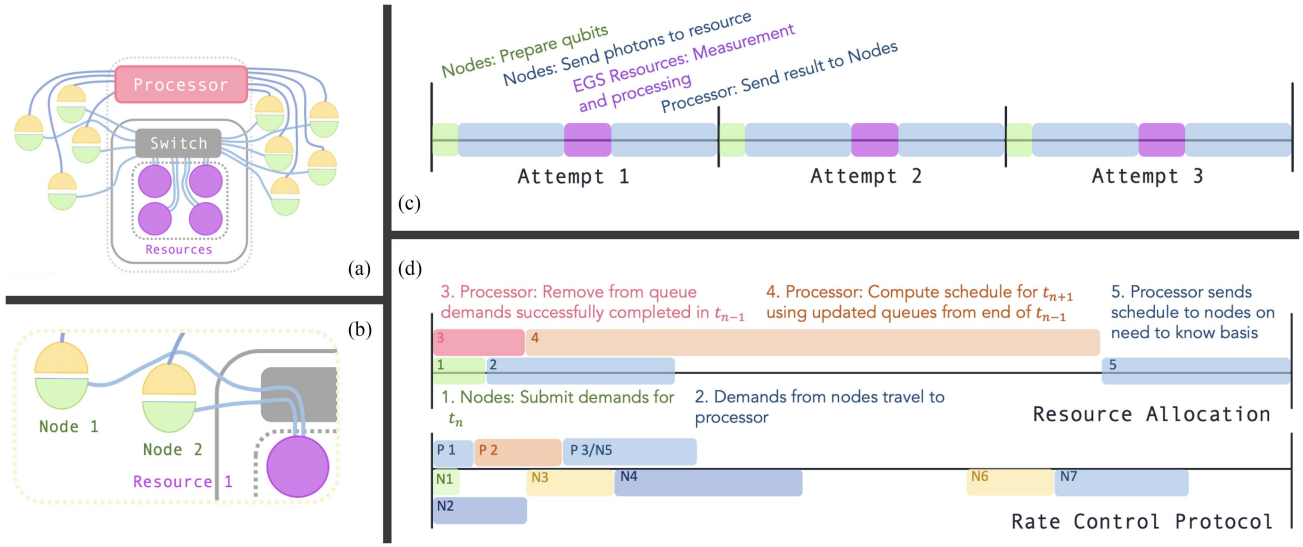

FIGURE 1. EGS Architecture. (a) EGS structure: an EGS with $R = 4$ resources connected to $N = 9$ nodes. The EGS is controlled by a classical processor and consists of a switch, resources, and physical connections. Nodes have quantum communication channels to the switch and classical communication channels to the processor. (b) Resource allocation: the switch opens connections to link nodes 1, 2, and resource 1. For example, the connections may consist of direct optical fiber paths from the nodes to the switch and from the switch to the resource, via an interface at the switch. This establishes the physical allocation of resource 1 to the communication session of nodes 1, 2 for time slot t_n. (c) Quantum communication sequence: Node-to-processor communication in time slot t_n with a batch size of three entanglement generation attempts. (d) Concurrent classical communication sequences: Nodes and the processor communicate in time slot t_n, governing resource allocation and the RCP (see Algorithm 1 for RCP details.).

TABLE 1. Inventory of Notation Introduced in Section II

Identifier	Description	Domain
U	Set of (user operated) quantum network nodes, of cardinality $\lvert U\rvert = N$	$\mathbb{N}$
R	Number of resources controlled by the EGS processor	$\mathbb{N}$
t_n	nth time-slot of the EGS system	$\mathbb{N}$
S	Set of communication sessions	$\{1, \cdots, N\}^2$
$\boldsymbol{\lambda}(t_n)$	Vector of target rates of all communication sessions at time t_n	$\mathbb{R}^{+\lvert S\rvert}$
$a_s(t_n)$	Number of demands from communication session s in time-slot t_n	$\mathbb{N}$
$\boldsymbol{q}(t_n)$	Vector of queues, with components $\big(q_s(t_n)\ \forall s\big)$	$\mathbb{N}^{\lvert S\rvert}$
$\boldsymbol{M}(t_n)$	Resource allocation schedule for time-slot t_n, with components $\big(M_s(t_n)\ \forall s\big)$	$\mathbb{N}^{\lvert S\rvert}$
x_s	Maximum number of resources that can be allocated to communication session s in any one time-slot	$\{1, \cdots, R\}$
p_{gen}	Probability a communication session allocated a resource for one time-slot successfully generates entanglement	[0, 1]
$g_s(t_n)$	Number of successfully generated entangled pairs by communication session s in t_n	$\{0, 1, \cdots, M_s(t_n)\}$
$\mathcal{C}$	Set which has the capacity region of the EGS as interior	$\mathbb{R}^+$
λ_{EGS}	Maximum total rate that can be delivered, on average, by the EGS, $\lambda_{\text{EGS}} = R \cdot p_{\text{gen}}$	$\mathbb{R}^+$
$\lambda_{\text{gen},s}^{\max}$	Maximum rate the EGS can deliver, on average, to communication session s, $\lambda_{\text{gen},s}^{\max} = x_s \cdot p_{\text{gen}}$	$\mathbb{R}^+$
$\lambda_s^{\min}$	Minimum acceptable rate of entanglement generation specified by communication session s	$\mathbb{R}^+$
λ_u	Maximum rate at which each node $u \in U$ can generate and/or make use of entanglement, across all of the sessions that it is involved in	$\mathbb{R}^+$

batched sequence of attempts, which can be terminated upon the successful creation of an entangled pair or at the end of the time slot. See Fig. 1(c) for an example quantum communication sequence compatible with heralded entanglement generation.

The classical communication sequence repeated in each time slot t_n which governs resource allocation is summarized in Fig. 1(d). In what follows we introduce and explain each step of this communication sequence. The notation introduced throughout this section is summarized in Table 1 .

A. DEMANDS FOR RESOURCE ALLOCATION FROM NODES TO THE EGS PROCESSOR

Definition II.1 (Target Rate, Communication Session): Each possible pair of nodes has the potential to require shared bipartite entanglement. To fulfill this need, a node pair

IEEE Transactions on Quantum Engineering

(u_i, u_j) requires the processor to allocate a resource. The node pair sets a *target rate* $\lambda_{(i,j)}(t_n)$ once per time slot, which represents the average number of entangled pairs per time slot they aim to generate using one or more EGS resources. A distinct pair of nodes with a nonzero target rate is referred to as a *communication session* and is associated with a unique communication session ID s. The set of communication session IDs S is defined as follows:

$$S := \big\{s = (i, j) \mid i < j \text{ and } \lambda_s(t_n) > 0 \ \forall\ (i, j) \in \{1, \ldots, N\}^2\big\} \tag{1}$$

where $N = |U|$ is the total number of network nodes with connections to the EGS.

Henceforth each pair of nodes will be identified by its communication session ID s. The target rates of all communication sessions in time-slot t_n can be written as a vector $\boldsymbol{\lambda}(t_n) \in \mathbb{R}^{|S|}$, the sth component of which is labeled by communication session ID s as $\lambda_s(t_n)$.

A rate of entanglement generation is the service demanded by each communication session from the EGS. To address the difference between the desired rate and the rate at which a communication session requires resource allocation to achieve that rate, we establish the following model for demand, which is compatible with a discrete time scheduling policy.

Definition II.2 (Demand): Demands for resources are requests made by communication session s to obtain a single entangled pair. The number of demands $a_s(t_n)$ submitted by session s at time slot t_n depends on its target rate $\lambda_s(t_n)$. If $\lambda_s(t_n) > 1$, then communication session s first submits $\lfloor \lambda_s(t_n) \rfloor$ demands. For a communication session s with $0 \le \lambda_s(t_n) \le 1$, or to account for the remaining part of the rate for any session with $\lambda_s(t_n) > 1$, each communication session randomly generates demands by sampling from a Bernoulli distribution with a mean equal to $\lambda_s(t_n) - \lfloor \lambda_s(t_n) \rfloor$, so that in general the submitted demands satisfy a (shifted) Bernoulli distribution, $a_s(t_n) \sim \text{Bernoulli}\big(\lambda_s(t_n) - \lfloor \lambda_s(t_n) \rfloor\big) + \lfloor \lambda_s(t_n) \rfloor$.

Definition II.3 (Designated Communication Node, Secondary Node): One of the nodes of every communication session is marked as the *designated communication node* for communicating the entanglement requests to the switch. The terms designated communication node and *secondary node* are used to refer to the two nodes of a communication session.

B. PROCESSING DEMANDS FOR RESOURCE ALLOCATION

Definition II.4 (Queue): When the processor receives a demand, it is added to one of $|S|$ *queues*, one for each communication session. The set of demands received by the processor by time-slot t_n and not yet satisfied is captured by the queue vector $\mathbf{q}(t_n) \in \mathbb{N}^{|S|} = (q_s(t_n)\ \forall s)$, where the component $q_s(t_n)$ is the queue of communication session s at time t_n. Each queue processes demands in a first-in–first-out order. As all demands are identical, we interchangeably use $q_s(t_n)$ to refer to both the queue length of communication session s in time slot t_n and the queue itself.

Definition II.5 [(Demand-Based) Schedule]: A resource allocation *schedule* is a vector $\boldsymbol{M}(t_{n+1}) \in \mathbb{N}^{|S|}$ calculated by the EGS processor in time slot t_n determining the assignment of the resources for time slot t_{n+1}. A single session s may be allocated the use of multiple resources, up to a maximum number x_s set by the EGS which does not exceed R, the total number of resources controlled by the EGS. For every session $s \in S$ the entry

$$M_s(t_{n+1}) \in \{0,\ 1,\ \ldots,\ x_s\} \tag{2}$$

corresponds to the number of resources assigned to s for the entire duration of time slot t_{n+1}. A *demand-based* schedule is based on the vector of all queues $\boldsymbol{q}(t_n)$, as it stands before new demands are registered in t_n, and satisfies

$$\sum_s M_s(t_{n+1}) \le \min\left(\sum_s q_s(t_n),\ R\right) \tag{3}$$

$$0 \le M_s(t_{n+1}) \le \min\big(q_s(t_n),\ x_s\big) \le R\ \forall s. \tag{4}$$

Each node of a communication session s requires a physical connection to the EGS switch. A single physical connection, such as an optical fiber, can be used for this purpose. To enable multiple connections between a node and the switch, options include the use of optical multiplexers over a single fiber or utilizing multiple fibers within a fiber bundle. The parameters $\big(x_s\ \forall s\big)$ are motivated by situations where the number of physical connections that can be dedicated to service communication session s are limited.

Definition II.6 (Maximum Weight Scheduling): The set $\mathcal{M}$ of feasible demand-based schedules at time slot t_n contains all vectors $\mathbf{M}'(t_{n+1}) \in \mathbb{N}^{|S|}$ satisfying (2)–(4). The EGS processor selects a *maximum weight schedule* $\mathbf{M}(t_{n+1}) \in \mathcal{M}$ from the feasible schedules for the following time slot by solving for

$$\mathbf{M}(t_{n+1}) \in \arg\max_{\mathbf{M}'} \sum_s q_s(t_n) M'_s(t_{n+1}). \tag{5}$$

In words, the schedule is selected from the set of feasible schedules by first solving for the subset of schedules that allocate resources to the sessions with the largest number of queued demands. If that subset contains more than one schedule, a schedule is randomly selected from the subset.

By the end of t_n, the schedule for t_{n+1} has been computed by the processor and broadcast to the nodes. If the schedule allocates use of a resource to communication session s for t_{n+1}, the users of s utilize the allocated resource to make a batch of entanglement generation attempts over the duration of t_{n+1}. The demand at the front of queue s is only marked as served once both a resource has been allocated and the users of s have successfully generated entanglement. Hence the dynamics of each queue are given by

$$q_s(t_{n+1}) = [q_s(t_n) + a_s(t_n) - g_s(t_n)]^+\ \forall s \tag{6}$$

where $[z]^+ = \max(z, 0)$, and $g_s(t_n)$ is the number of successfully generated entangled pairs by s during t_n. In words, for every subsequent time slot, the demands that arrived in the previous time slot are added to the queue and those that were scheduled and successfully resulted in the generation of an entangled pair are removed from the queue. The updated queue is always of nonnegative length since the number of successfully generated entangled pairs is a sample of a binomial random variable where the number of trials is the number of resources allocated to s, $M_s(t_n)\big(\leq q_s(t_n)\big)$, and the trial success probability is p_{gen}

$$g_s(t_n) \sim \text{Bin}\big(M_s(t_n),\ p_{\text{gen}}\big).$$

Definition II.7 (Supportable Rate): The arrival rate vector $\boldsymbol{\lambda}(t_n) \in \mathbb{R}^{+|S|} = \big(\lambda_s(t_n)\ \forall\ s\big)^{\text{T}}$ is *supportable* if there exists a schedule under which

$$\lim_{Q\to\infty} \lim_{n\to\infty} \text{P}\big(|\boldsymbol{q}(t_n)| \geq Q\big) = 0 \tag{7}$$

where $|\boldsymbol{q}(t_n)| := \sum_s |q_s(t_n)|$ is the sum of the number of demands in the queue of each session in time slot t_n. That is, $\boldsymbol{\lambda}(t_n)$ is supportable if the probability that the total queue length becomes infinite is zero.

Definition II.8 (Capacity Region): The *capacity region* of an EGS is the set of arrival rate vectors that are supportable by the EGS. For each rate vector $\boldsymbol{\lambda}$ in the capacity region, there exists some scheduling routine such that an EGS operating under that scheduling algorithm can support the rate vector $\boldsymbol{\lambda}$.

If the rate vector $\boldsymbol{\lambda}$ falls outside the capacity region, the EGS cannot support it under any scheduling algorithm, leading to unpredictable performance. The goal of moderating the rate vector through the RCP is twofold: first, to keep it within the capacity region, and second, to maximize resource utilization by saturating the capacity region, thus fully leveraging the potential of the EGS to facilitate entanglement generation.

Theorem II.1 (Capacity Region): Let x_s be the maximum number of resources that can be allocated to a session s per time slot. For each resource, p_{gen} is the probability that a communication session allocated the resource for one time slot will successfully create an entangled pair. The capacity region of an EGS with R resources is the set of rate vectors $\boldsymbol{\lambda} \in \text{Int}\mathcal{C}$, where $\mathcal{C}$ is defined as

$$\mathcal{C} = \left\{ \boldsymbol{\lambda} : \boldsymbol{\lambda} \geq \mathbf{0},\ \sum_s \lambda_s \leq \lambda_{\text{EGS}},\ \text{and } \lambda_s \leq \lambda_{gen,s}^{\max}\ \forall s \in S \right\}. \tag{8}$$

$\lambda_{\text{EGS}} = R \cdot p_{\text{gen}}$ and $\lambda_{\text{gen},s}^{\max} = x_s \cdot p_{\text{gen}}$. Moreover, maximum weight scheduling (Definition II.6) is throughput optimal and supports any rate vector $\boldsymbol{\lambda} \in \text{Int}\mathcal{C}$. For proof, see Section V-A2.

The first requirement of $\mathcal{C}$ states that all request rate vectors must be positive, meaning every component of the rate vector must be positive or zero ($\boldsymbol{\lambda} \geq \mathbf{0} \Leftrightarrow \lambda_s \geq 0\ \forall s \in S$). The second requirement enforces that the total rate of entanglement requested from the EGS $\sum_s \lambda_s$, cannot exceed the total average service rate of the EGS $R \cdot p_{\text{gen}}$. The final requirement states that the request rate λ_s of any communication session s must not exceed the maximum average service rate that can be allocated to the communication session $x_s \cdot p_{\text{gen}}$.

C. CONSTRAINTS

We assume that there are two types of constraints on the sequence of target rates set by a session. The first is a minimum rate of entanglement generation $\lambda_s^{\min}$; below this rate, session s cannot obtain sufficient entangled pairs within a short enough period of time in order to enable its target application. The second constraint $\lambda_u\ \forall u \in U$ is an upper limit on the rate at which each node u can generate and/or make use of entanglement across all of the sessions that it is involved in. This parameter can capture a range of technical limitations of the quantum nodes, including a limited rate of entanglement generation or a limited speed of writing generated entanglement to memory, hence temporarily decreasing the availability of the node for engaging in further entanglement generation immediately following the successful production of a pair.

D. MATHEMATICAL BACKGROUND

In this section, we briefly summarize and motivate some relevant mathematical techniques from optimization theory. See [25] for a thorough coverage of the topic. The method of Lagrange multipliers is a tool for solving constrained optimization problems. Consider the problem of maximizing a continuously differentiable objective function $F(\boldsymbol{z}) : \mathbb{R}^n \mapsto \mathbb{R}$ over the domain $Z \subseteq \mathbb{R}^n$ subject to a set of m inequality constraint functions $\{g_m(\boldsymbol{z}) \leq 0\}_m$. This is referred to as the *primal problem*

$$\max_{z\in Z} F(\boldsymbol{z}) \tag{9}$$

subject to

$$g_m(\boldsymbol{z}) \leq 0\ \forall m. \tag{10}$$

The problem is said to be *feasible* if there exists a vector $\boldsymbol{z}$ for which (10) is satisfied, meaning the constraints of the problem are satisfied. A local maximizer $\boldsymbol{z}^*$ of the function F over the domain Z satisfies the optimality condition [25]

$$\nabla_{z_n} F(\boldsymbol{z}^*)^T (\boldsymbol{z} - \boldsymbol{z}^*) \leq 0\ \forall \boldsymbol{z} \in Z. \tag{11}$$

In the method of Lagrange multipliers, one introduces a nonnegative vector $\boldsymbol{p} = (p_m\ \forall m) \in \mathbb{R}^{+m}$ and the Lagrangian function

$$L(\boldsymbol{z}, \boldsymbol{p}) = F(\boldsymbol{z}) - \sum_m p_m g_m(\boldsymbol{z}).$$

The following theorem motivates a method of solving the primal problem by searching for vectors $\boldsymbol{z}^*$ that

IEEE Transactions on Quantum Engineering

satisfy $\nabla_{z_n} L(z^*, \boldsymbol{p}) = 0$, for some vector of Lagrange multipliers $\boldsymbol{p}$.

Theorem II.2 (Karush–Kuhn–Tucker (KKT) Conditions [25]): Let z^* be a local maximizer of the function F over the domain Z. Suppose that the functions F and $\{g_m(z)\}_m$ are continuously differentiable and that the set of constraint gradients $\{\nabla_{z_n} g_m(z^*) | g_m(z^*) = 0\}_m$ is linearly independent. Then, there exists a unique Lagrange multiplier vector $\boldsymbol{p}^*$ with components p_m^* such that

$$\nabla_{z_n} L(z^*, \boldsymbol{p}^*) = 0 \tag{12}$$

$$g_m(z^*) \leq 0 \ \forall m \tag{13}$$

$$p_m^* \geq 0 \ \forall m \tag{14}$$

$$p_m^* \ g_m(z^*) = 0 \ \forall m. \tag{15}$$

In some cases, it may be computationally challenging to solve the system of (12)–(15). However, in these cases it may be possible to solve a related problem, known as the *dual problem*

$$\inf_{\boldsymbol{p} \geq \boldsymbol{0}} D(\boldsymbol{p}) \tag{16}$$

where the dual function $D(\boldsymbol{p})$ is defined as

$$D(\boldsymbol{p}) := \sup_{z \in Z} L(z, \boldsymbol{p}). \tag{17}$$

Searching for a solution to the dual problem is particularly useful when there is no *duality gap*, meaning that the value of the optimal solution $D(\boldsymbol{p}^*)$ achieved with a solution $\boldsymbol{p}^*$ to the dual problem is equal to the value of the optimal solution $F(z^*)$ achieved with a solution z^* to the primal problem. The following theorem lays out conditions under which there is no duality gap and the existence of a vector of Lagrange multipliers is guaranteed.

Theorem II.3 (Strong Duality Theorem [25]): Suppose that the primal problem is feasible and its optimal value $F(z^*)$ is finite. Furthermore, suppose the set Z is a convex subset of $\mathbb{R}^n$, the function $F(z)$ is concave over Z, and the functions $\{g_m(z)\}_m$ are convex over Z. In addition, suppose there exists a vector $\bar{z} \in Z$ such that $g_m(\bar{z}) < 0 \ \forall m$. Then, there is no duality gap and there exists at least one vector of Lagrange multipliers.

The assumption in the Strong Duality Theorem II.3 that the feasible region of the domain Z must contain an interior point is commonly known as the Slater constraint qualification [23].

III. RCP ALGORITHM

An algorithm moderating competition for EGS resources enables the possibility of introducing a notion of fairness in how resources are allocated amongst competing communication sessions and ensuring that the resources are fully utilized. We consider a situation where the rate vector produced by any such algorithm is constrained by the maximum service rate of the switch, as described by the capacity region $\mathcal{C}$, as well as the node or user level constraints described by $\lambda_u \ \forall u$ and $\lambda_s^{\min} \ \forall s$. In the framework of NUM, we pose an optimization problem where each communication session s is associated with a utility function $f_s(\lambda_s(t_n)) : \mathbb{R} \mapsto \mathbb{R}$, which encodes the benefit s derives from the rate vector $\boldsymbol{\lambda}(t_n)$. We apply the theory of Lagrange multipliers and Lagrangian duality (see [25] for detailed coverage) to formulate and analyze the optimization problem. We then derive the RCP (Algorithm 1) as the solution to this problem.

The primal problem is to maximize the aggregate utility or the total benefit that users derive from the EGS by maximizing the sum of the utility functions, including the constraints by the use of Lagrange multipliers. The dual problem is to determine an optimal vector of Lagrange multipliers. In the case where there is no duality gap [25], a solution to the dual problem is equivalent to a solution of the primal problem. The vector of Lagrange multipliers $\boldsymbol{p}(t_{n+1}) = \left(p_c(t_n), p_u(t_n) \ \forall u\right) \in \mathbb{R}^{+(1+N)}$, with components for the processor and each node, is denoted as the price vector in our algorithm and serves as a measure of the competition for resources amongst the communication sessions. Define $S(u) := \{s : u \in s\} \subseteq S$ to be the subset of communication sessions in which node u participates. In each communication session one node is designated to communicate demand to the switch and the other node is secondary (see Definition II.3). Note that $u \in s \Leftrightarrow s \in S(u)$. The feasible rate region of the communication session s is

$$\Lambda_s := \left\{\lambda_s : \lambda_s^{\min} \leq \ \lambda_s \leq \lambda_{\text{gen},s}^{\max}\right\} \ \forall s \tag{18}$$

and the feasible region for a rate vector $\boldsymbol{\lambda}$ is

$$\Lambda = \bigcup_s \Lambda_s. \tag{19}$$

We make the following two assumptions on the utility function f_s of each communication session s.

- *A1:* On the interval $\Lambda_s = [\lambda_s^{\min}, \lambda_{\text{gen},s}^{\max}]$ the utility functions f_s are increasing, strictly concave, and twice continuously differentiable.
- *A2:* The curvatures of all f_s are bounded away from zero on Λ_s. For some constant $\alpha_s > 0$

$$-f_s''(\lambda_s) \geq \frac{1}{\alpha_s} > 0 \ \forall \ \lambda_s \in \Lambda_s.$$

To ensure feasibility and satisfy the Slater constraint qualification [25], in addition to assumptions A1 and A2 it is necessary that the rate vector with components equal to the minimal rates of each communication session is an interior point of the constraint set

$$\sum_s \lambda_s^{\min} < \lambda_{\text{EGS}} \tag{20}$$

$$\sum_{s \in S(u)} \lambda_s^{\min} < \lambda_u \ \forall u. \tag{21}$$

Algorithm 1: Rate Control Protocol (RCP).

Processor's Algorithm: At times $t_n = 1,\ 2, \ldots,$ the processor:

1) receives rates $\lambda_s(t_n)$ from all communication sessions $s \in S$;
2) computes a new central price,

$$p_c(t_{n+1}) = \left[\frac{1}{\lambda_{\text{EGS}}} \sum_s q_s(t_n) + \theta_c \left(\sum_s \lambda_s(t_n) - \lambda_{\text{EGS}} \right) \right]^+ \quad (22)$$

where θ_c is a constant step-size for the central price;
3) broadcasts the new central price $p_c(t_{n+1})$ to all communication sessions $s \in S$.

Network Node u's Algorithm: At times $t_n = 1,\ 2, \ldots,$ network node u:

1) marks the subset of communication sessions $\text{COMM}(u) \subseteq S(u)$ involving node u for which it is the designated communication node;
2) receives from every secondary node u' the price $p_{u'}(t_n)$ for each communication session $s = (u, u') \in \text{COMM}(u)$;
3) computes a new node price

$$p_u(t_{n+1}) = \left[\frac{1}{\lambda_u} \sum_{s \in S(u)} q_s(t_n) + \theta_u \left(\sum_{s \in S(u)} \lambda_s(t_n) - \lambda_u \right) \right]^+ \forall\, u \quad (23)$$

where $\theta_u\ \forall u$ is a constant step-size for each node, which may be fixed or differ from node to node;
4) communicates the new price $p_u(t_{n+1})$ to the communication node from every communication session $s \in S(u) \setminus \text{COMM}(u)$ in which u is a secondary node;
5) receives from the switch the central price $p_c(t_{n+1})$;
6) computes the new rate for every communication session $s \in \text{COMM}(u)$

$$\lambda_s(t_{n+1}) = \left[\left(\frac{\mathrm{d}f_s}{\mathrm{d}\lambda_s} \right)^{-1} \left(\boldsymbol{p}(t_{n+1}) \right) \right]_{\lambda_s^{\min}}^{\lambda_{\text{gen},s}^{\max}} \quad (24)$$

where $[z]_m^M = \max\left(\min(z,\ M), m\right)$ and $\boldsymbol{p}(t_i) = \left(p_c(t_i),\ p_u(t_i)\ \forall u\right)$ is the vector of prices pertaining to time slot t_i;
7) communicates the new rate $\lambda_s(t_{n+1})$ to the EGS processor, for every communication session $s \in \text{COMM}(u)$.

A. DERIVATION

Formally, the RCP yields rate vectors which solve the *Primal Problem*

$$\max_{\boldsymbol{\lambda} \in \Lambda} F(\boldsymbol{\lambda}) := \sum_s f_s(\lambda_s) \quad (25)$$

subject to

$$\sum_s \lambda_s \le \lambda_{\text{EGS}} \quad (26)$$

$$\sum_{s \in S(u)} \lambda_s \le \lambda_u\ \forall u. \quad (27)$$

The Lagrangian, which includes the constraints (26) and (27) with a vector of Lagrange multipliers $\boldsymbol{p} = (p_c,\ p_u\ \forall u) \ge \boldsymbol{0}$ together with the objective function (25), is given by

$$L(\boldsymbol{\lambda}, \boldsymbol{p}) = \sum_s f_s(\lambda_s) - p_c \left(\sum_s \lambda_s - \lambda_{\text{EGS}} \right) - \sum_u p_u \left(\sum_{s \in S(u)} \lambda_s - \lambda_u \right). \quad (28)$$

We identify that the problem is separable in the communication sessions S, and rewrite the Lagrangian in separable form

$$L(\boldsymbol{\lambda}, \boldsymbol{p}) = \sum_s l_s(\lambda_s) + p_c \lambda_{\text{EGS}} + \sum_u p_u \lambda_u \quad (29)$$

where $l_s(\lambda_s)$ is defined as

$$l_s(\lambda_s) := f_s(\lambda_s) - \lambda_s p_c - \lambda_s \sum_{u \in s} p_u$$

and we make use of the equivalence

$$\sum_u p_u \sum_{s \in S(u)} \lambda_s = \sum_s \lambda_s \sum_{u \in s} p_u.$$

A rate vector $\boldsymbol{\lambda}^*$ is a local maximum of (25) if it satisfies the optimality condition [25]

$$\nabla_{\lambda_s} F(\boldsymbol{\lambda}^*)^T (\boldsymbol{\lambda} - \boldsymbol{\lambda}^*) \le 0\ \forall \boldsymbol{\lambda} \in\ \Lambda. \quad (30)$$

If moreover $F(\boldsymbol{\lambda})$ is concave over Λ, then (30) is also sufficient for $\boldsymbol{\lambda}^*$ to maximize $F(\boldsymbol{\lambda})$ over Λ [25] (it is also a global maximum).

To obtain a $\boldsymbol{\lambda}^*$ satisfying both the optimality condition (30) and the constraints (26) and (27) we set the gradient with respect to rate of each communication session of the Lagrangian to zero

$$\nabla_{\lambda_s} L = \sum_s \frac{\mathrm{d}l_s(\lambda_s)}{\mathrm{d}\lambda_s} = 0.$$

The maximization in the primal problem (25) is constrained to the feasible rate region defined by (18) and (19). To restrict solutions to the problem domain, any $\tilde{\boldsymbol{\lambda}}^* \notin \Lambda$ is projected componentwise so that $\tilde{\lambda}_s^* \mapsto \lambda_S^* \in \Lambda_s\ \forall s$. With the assumptions in (20) and (21) there exists at least one set of Lagrange

multipliers [25]. In terms of a given vector of Lagrange multipliers $\boldsymbol{p}$, an optimal rate vector $\boldsymbol{\lambda}^*$ satisfies

$$\lambda_s^* = \left[\left(\frac{\mathrm{d}f_s}{\mathrm{d}\lambda_s}\right)^{-1}(\mathbf{p})\right]_{\lambda_s^{\min}}^{\lambda_{\mathrm{gen},s}^{\max}} \forall s \tag{31}$$

where $[z]_m^M = \max\Big(\min\big(z,\ M\big), m\Big)$. To obtain a $\boldsymbol{\lambda}^*$, it remains to obtain a vector of Lagrange multipliers.

An optimal vector $\boldsymbol{p}^*$ of Lagrange multipliers is a solution to the *Dual Problem.*

Select $\boldsymbol{p} = (p_c,\ p_u\ \forall\, u)$ so as to achieve

$$\inf_{\boldsymbol{p}\geq\mathbf{0}} D(\mathbf{p}) \tag{32}$$

where the dual-objective function $D(\mathbf{p})$ is defined as

$$D(\mathbf{p}) = \sup_{\boldsymbol{\lambda}\in\Lambda} L(\boldsymbol{\lambda}, \boldsymbol{p}). \tag{33}$$

With assumptions A1, A2 and (20), (21), the problem satisfies the Slater constraint qualification and has no *duality gap* [25], meaning a solution to the dual problem is also a solution to the primal problem. Define $\boldsymbol{\lambda}^*$ to be a rate vector that maximizes $L(\boldsymbol{\lambda},\ \boldsymbol{p})$. A vector of Lagrange multipliers $\boldsymbol{p}^*$ is an optimal solution to the dual problem if it satisfies the optimality condition

$$\nabla_p D(\boldsymbol{p}^*)^T(\boldsymbol{p} - \boldsymbol{p^*}) \geq 0\ \forall \boldsymbol{p} \geq \mathbf{0}. \tag{34}$$

Gradient projection is a type of algorithm where in order to solve an optimization problem, such as the dual problem (32) with respect to a vector $\boldsymbol{p}$, one starts by selecting some initial vector $\boldsymbol{p}(0)$ and iteratively adjusting $\boldsymbol{p}(t_n) \mapsto \boldsymbol{p}(t_{n+1})$ by making steps in the opposite direction of the gradient of the objective function. We introduce a vector of step-sizes $\boldsymbol{\theta} = (\theta_c, \theta_u\ \forall u) \in \mathbb{R}^{1+N}$. The components of $\nabla_p D(\boldsymbol{p})$ are

$$\frac{\partial D(\boldsymbol{p})}{\partial p_c} = -\left(\sum_s \lambda_s^* - \lambda_{\mathrm{EGS}}\right) \tag{35}$$

$$\frac{\partial D(\boldsymbol{p})}{\partial p_u} = -\left(\sum_{s\in S(u)} \lambda_s^* - \lambda_u\right)\ \forall u. \tag{36}$$

An implementation of the gradient projection algorithm is to iteratively adjust the Lagrange multipliers according to

$$p_c(t_{n+1}) = \left[p_c(t_n) + \theta_c\left(\sum_s \lambda_s^*(t_n) - \lambda_{\mathrm{EGS}}\right)\right]^+ \tag{37}$$

$$p_u(t_{n+1}) = \left[p_u(t_n) + \theta_u\left(\sum_{s\in S(u)} \lambda_s^*(t_n) - \lambda_u\right)\right]^+,\ \forall u \tag{38}$$

where $\lambda_s^*(t_n) = \lambda_s^*\big(\boldsymbol{p}(t_n)\big)$ is given by inputting the vector of Lagrange multipliers in (31). An implementation of the algorithm necessitates identifying parameters in the system that correspond to the components of the vector of Lagrange multipliers. We note that the centralized price $p_c(t_n)$ and the user prices $p_u(t_n)\ \forall u$, have, respectively, the same dynamics as the total queue lengths and the sum total of the session queue lengths in which user u participates (6). Therefore, we make the following identifications:

$$p_c(t_n) \leftrightarrow \frac{1}{\lambda_{\mathrm{EGS}}}\sum_s q_s(t_n)$$

$$p_u(t_n) \leftrightarrow \frac{1}{\lambda_u}\sum_{s\in S(u)} q_s(t_n)\ \forall u.$$

Note that these identifications are not unique, since the only strict criteria on the identification is that the queue dynamics generated by (6) match the dynamics of (37) and (38), whereas the scaling is arbitrary. For more information on the interpretation of Lagrange multipliers as prices in communication networks, see [22], [23].

B. CONVERGENCE

The RCP is a gradient projection algorithm with constant step-sizes from the vector $\boldsymbol{\theta} \in \mathbb{R}^{1+N} = (\theta_c, \theta_u\ \forall u)$. Establishing that the algorithm converges is crucial to ensure that it yields solutions that effectively address the problem it is designed to solve. To establish convergence, we follow a similar treatment, as in [26].

Theorem III.1 (RCP Convergence): Suppose assumptions A1 and A2 and the constraints (20) and (21) are satisfied and each of the the step-sizes $\theta_r \in \{\theta_c, \theta_u\ \forall u\}$ satisfies $\theta_r \in (0, 2/\overline{\alpha}|S|)$, where $\overline{\alpha} = \max_{s\in S}\alpha_s$ with α_s the curvature bound of assumption A2 and $|S|$ is the number of communication sessions. Then, starting from any initial rate $\boldsymbol{\lambda}(0) \in \Lambda$ and price $\boldsymbol{p}(0) \geq \mathbf{0}$ vectors, every accumulation point $\big(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{p}}\big)$ of the sequence over time slots $\{\big(\boldsymbol{\lambda}(t_n), \boldsymbol{p}(t_n)\big)\}_{t_n}$ generated by the RCP is primal-dual optimal. For proof, see Section V-A3.

IV. CASE STUDY

To illustrate the use of the RCP we associate a log utility function with each session

$$f_s(\lambda_s) = \log(\lambda_s)\ \forall s \in S. \tag{39}$$

Log utility functions are suitable when throughput is the target performance metric, and a set of sessions all employing log utility functions will have the property of proportional fairness. In such a system, if the proportion by which one session rate changes is positive, there is at least one other session for which the proportional change is negative [23]. For compatibility with Theorem III.1 note that log utility functions satisfy A1, and A2 is satisfied with $\alpha_s = (\lambda_{\mathrm{gen},s}^{\max})^2\ \forall s$.

Although the convergence theorem only guarantees asymptotic convergence of the sequence $\{\big(\boldsymbol{\lambda}(t_n), \boldsymbol{p}(t_n)\big)\}_{t_n}$ to an optimal rate-price pair $\big(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{p}}\big)$, in any realization of an EGS one expects that the convergence time $\Delta\tau$, the number of time slots that the RCP must run before convergence is attained, is finite. In addition, it is practically relevant to

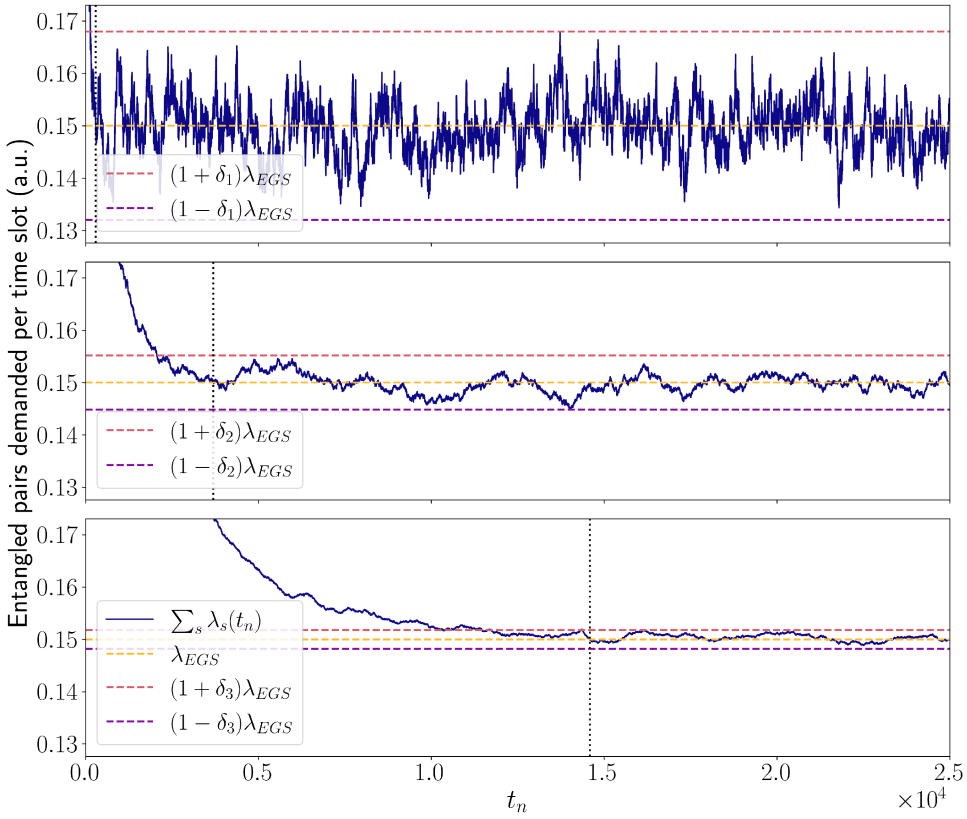

FIGURE 2. RCP drives the sum of the demanded rates of entanglement generation across all communication sessions $\sum_s \lambda_s(t_n)$ to converge with respect to the sequence of time slots to the maximum average entanglement generation rate of the EGS λ_{EGS}. The EGS has $R = 3$ resources, the probability of entanglement generation is $p_{gen} = 0.05$, and the EGS is connected to (top) $N = 20$, (middle) $N = 50$, and (bottom) $N = 100$ nodes. The total number of communication sessions served are $|S| = 19$, 123, and 495 in the top, middle, and bottom plots, respectively. Black dotted lines indicate the convergence times $\Delta\tau$. The observed values for the tightness of convergence δ are $\delta_1 = 0.12$, $\delta_2 = 0.035$ and $\delta_3 = 0.012$. Step-sizes ($\theta_c, \theta_u\ \forall u$) were all $1/(40 \cdot \lambda_{EGS})$.

characterize the tightness of convergence δ, or the maximum size of fluctuations about the optima.

If an EGS is connected to N nodes, there are $|S|_{\max} = \binom{N}{2}$ possible sessions. We assume that in a real network not all pairs of users require shared entanglement. In Fig. 2, we numerically investigate the convergence time and tightness of convergence, $(\Delta\tau, \delta)$, for an EGS with $R = 3$ resources and $p_{gen} = 0.05$ connected to $N = 20$, 50, and 100 users, where the number of sessions is restricted to $|S| = 0.1 \cdot |S|_{\max}$ by randomly sampling 10% of the possible sessions. In these simulations, we set $x_s = 1\ \forall s$, and average over 1000 independent runs of the simulation, each using the same set of sessions.

The reported convergence times $\Delta\tau$ are the number of time slots that occur before the sum of demand rates $\sum_s \Sigma\lambda_s(t_n)$ first crosses the optimal value λ_{EGS}. Reporting of the tightness of convergence δ is based on the maximum size of fluctuations of $\sum_s \Sigma\lambda_s(t_n)$ about λ_{EGS} following $\Delta\tau$. As the number of sessions hosted by an EGS increases, we observe a tradeoff between $\Delta\tau$ and δ. When the number of sessions is lower, $\Delta\tau$ is shorter but δ is larger. We have performed additional simulations which indicate that increasing the step size used in the RCP can be used to trade larger δ for somewhat shorter $\Delta\tau$.

If constraint changes occur slowly compared to $\Delta\tau$, Theorem III.1 implies that the RCP should reestablish convergence to a new optimal rate and price vector pair, $(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{p}}) \mapsto (\hat{\boldsymbol{\lambda}}', \hat{\boldsymbol{p}}')$. In a real EGS system, it is possible that the number of available resources will not be static in time, as resources may require periodic downtime for calibration. The effect of a change in the number of resources $R \mapsto R'$ changes the maximum service rate $\lambda_{EGS} = R \cdot p_{gen} \mapsto \lambda'_{EGS} = R' \cdot p_{gen}$. To validate the robustness of the algorithm against such constraint changes we simulate EGS systems originally equipped with $R = 3$ resource nodes, where after every 10 000 time-slots one of the resources may either be taken offline for calibration or an offline resource may be returned to service. Fig. 3 demonstrates that the RCP successfully reestablishes convergence of $\Sigma_s\lambda_s(t_n)$ about λ'_{EGS} following these constraint changes in an EGS system connected to $N = 50$ nodes, serving $|S| = 123$ communication sessions.

In Fig. 3, we record the sequence of convergence times $\{\Delta\tau\}$ after each constraint change as the first time-steps, where $\sum_s \lambda_s(t_n)$ crosses λ'_{EGS}. To calculate the tightness of

IEEE Transactions on Quantum Engineering

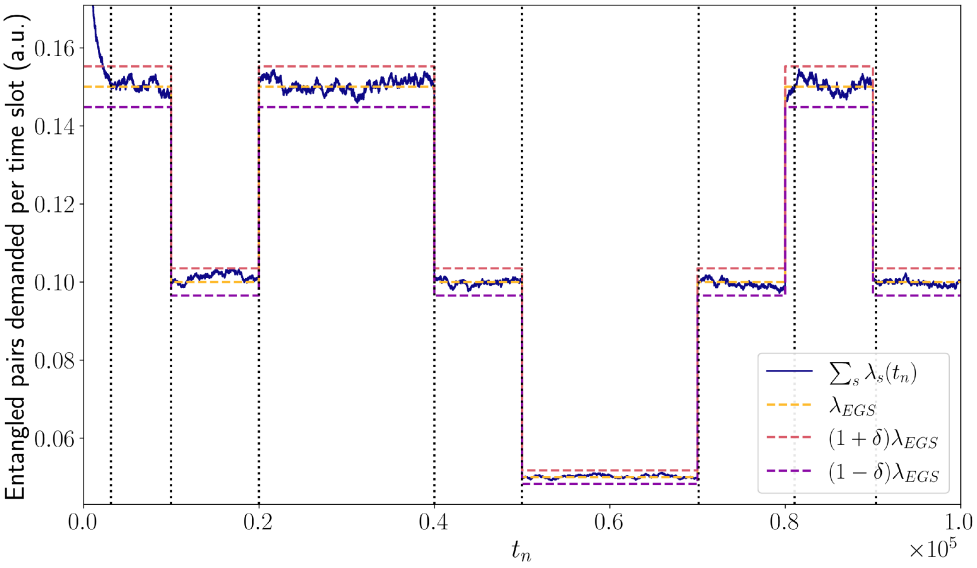

FIGURE 3. In response to changes in the number of resources available at the EGS ($R \rightarrow R'$), the RCP drives the sum of the demanded rates of entanglement generation across all communication sessions $\sum_s \Sigma\lambda_s(t_n)$ to converge with respect to the sequence of time slots to the updated maximum average entanglement generation rate of the EGS, $\lambda_{EGS} = R' \cdot p_{gen}$. In simulation, an EGS connected to $N = 50$ nodes, serving $|S| = 123$ communication sessions, is originally equipped with $R = 3$ resources. After every 10 000 time-slots, one of the resources may either be taken offline for calibration or an offline node may be returned to service. Black dashed lines indicate the convergence, $\Delta\tau$ calculated for every R' (initially R). We observe and overall tightness of convergence of $\delta = 0.035$, identical to that observed in Fig. 3 for the EGS operated with fixed $R = 3$ and with the same N, $|S|$. Step-sizes ($\theta_c, \theta_u \; \forall u$) were all $1/(10 \cdot \lambda_{EGS})$.

convergence δ, we first calculate the sequence of $\{\delta'\}$, the size of the maximum fluctuations about λ'_{EGS} following each $\Delta\tau'$ and set $\delta = \max(\{\delta'\})$. Notably, every subsequent $\Delta\tau' < \Delta\tau$ and the achieved δ is equal to that observed when there are no changes to the constraint set in Fig. 2 (middle plot, δ_2) for an EGS with the same number of nodes, serving the same number of communication sessions. Additional simulations of EGS systems connected to various numbers of nodes ranging from 10 to 100, with random changes to the number of resources after every 10 000 time-steps, suggest that the data in Fig. 3 is representative. Specifically, in each case investigated the absolute relative difference

$$\frac{|\delta - \tilde{\delta}|}{\tilde{\delta}} < 1$$

between the achieved tightness of convergence when there are (δ) and are not ($\tilde{\delta}$) changes to the constraints is less than 1.

The constraints $\{\lambda_u\}_u$ on the capabilities of nodes appear in (23), and therefore affect both the prices calculated by the nodes and the rates set by communication sessions in (24). Since these constraints limit the total rate at which a node can submit demands summed across all of the communication sessions in which it participates, it is expected that uniform settings of $\{\lambda_u\}_u$ yield rate vectors under the RCP, where $\{\lambda_s(t_n)\}_s$ are approximately uniform. In contrast, if the node constraints are nonuniform amongst the nodes, it is expected that the RCP yields rate vectors with larger differences between the rates set by each communication session. In Fig. 4, we investigate the effect of different settings for these constraints by plotting the difference between the average maximum $\max_s\{\lambda_s(t_n)\}_s$ and minimum $\min_s\{\lambda_s(t_n)\}_s$ communication session rates yielded by the RCP for two different settings of the constraints. In the first setting, node

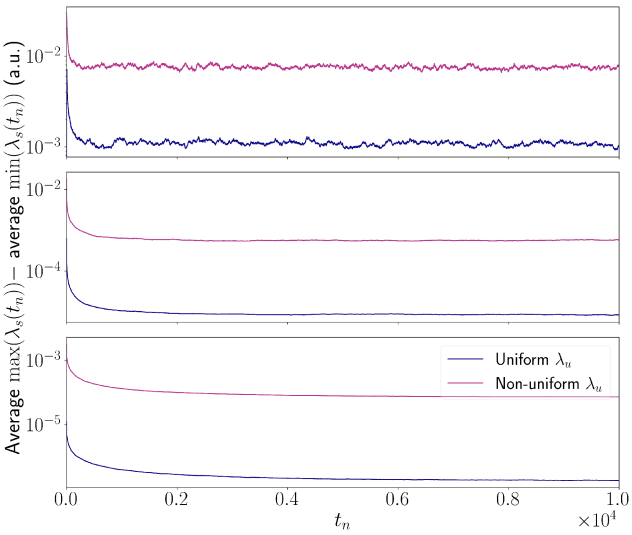

FIGURE 4. Differences between the average maximum rate and average minimum rate requested by any communication session in time-slot t_n, for an EGS connected to (top) $N = 20$, (middle) $N = 50$, and (bottom) $N = 100$ nodes serving $|S| = 19$, 123, and 495 communication sessions, respectively. As described in the main text, nodes are either associated with a uniform and effectively unrestricted set of capabilities or a nonuniform and more restricted set of capabilities. Step-sizes ($\theta_c, \theta_u \; \forall u$) were all $1/(40 \cdot \lambda_{EGS})$.

constraints are set uniformly as $\lambda_u = \big((|S|-1)/2\big) \cdot p_{\text{gen}}\ \forall u$ so that in practice the algorithm functions as if the network node constraints have been removed. In the other setting, there are three possible constraint values: 1) a quarter of the nodes sampled at random have $\lambda_u = 1.5 \cdot p_{\text{gen}}$; 2) half of the nodes have $\lambda_u = p_{\text{gen}}$; and 3) a quarter of the nodes have $\lambda_u = 0.5 \cdot p_{\text{gen}}$. Fig. 4 confirms that the difference between the average maximum rate and the average minimum rate requested by any session at time-step t_n is one or more orders of magnitude larger when nodes are associated with the nonuniform constraint set. The uniform node constraint setting led to communication sessions updating their rates of demand submission to be nearly uniform across all communication sessions.

V. DISCUSSION

We have presented the first control architecture for an EGS. The architecture is tailored to a simple system model. As a natural extension of this work, a refined version of the control architecture can be developed to suit a more versatile physical model. In the following discussion, we explore considerations for the development of a second generation control architecture.

In this work, we assume a demand model in which user generated demands are fully parameterized by a desired rate of entanglement generation. Specifically, every communication session s sets $\lambda_s(t_n)$, updated once per time-slot and specifies the constraint parameter $\lambda_s^{\min}$ which defines the minimum rate of entanglement generation the communication session must receive in order to enable some target application. While this model is mathematically simple, it may not fully address real application requirements on a physical quantum network. Real applications may require the simultaneous existence of a number of entangled pairs, each with some minimum fidelity and it is possible that applications need such packets of pairs to be supplied periodically over a longer application run-time. In the future, it may therefore be relevant to consider a demand model wherein communication sessions submit demands for packets of entanglement generation. A packet would be fully specified by the desired number of entangled pairs, a minimum fidelity for the pairs, some maximum window of time between the generation time of the first and last entangled pair of the request, and possibly some rate at which the demand with the preceding parameters should be repeatedly fulfilled.

The discussed model assumes that user controlled nodes can engage in multiple entanglement generation tasks in parallel. We do not impose restrictions on simultaneously scheduling communication sessions. Hence, it is possible for communication sessions s and s' with node $u \in s, s'$ to be scheduled simultaneously. In addition, we consider the option of assigning multiple resource nodes to a single communication session in any time-slot. Therefore, we consider nodes with an unrestricted number of qubits and independent physical connections to the EGS. A subtlety we do not address here is that allocating multiple resources to a single communication session may require temporal multiplexing in the scheduling of individual entanglement generation attempts, especially when the multiple qubits of a single node are coupled to the physical connection via a single output. Furthermore, for nodes consisting of a single quantum processor, it may not be possible to calibrate the node to simultaneously engage in entanglement generation attempts with multiple partner nodes, even if the node has unlimited qubits. To capture this physical feature, it will be interesting to include the restriction of scheduling only nonoverlapping communication sessions in the design of scheduling routines for future EGS control architectures.

The control architecture for an EGS relies on precise timing synchronization. Our model assumes that at both the control and physical layers, all communication sessions can adhere to the time slots defined by the EGS processor. Tight synchronization of timing is possible at the physical layer, which controls the quantum devices and coordinates the exact timing of entanglement generation attempts. However, tight timing synchronization of any type of classical communication may be a considerable challenge in any real world application. In particular, such coordination is a serious challenge if there are nonuniform communication times between any of the nodes and the EGS or between any of the node pairs. To reduce the timing requirements and possibly make the control architecture delineated here executable on a real-world system, it is possible to consider the processor of the EGS simulating the actions of the nodes. To do so, the processor would locally run the RCP and simulate the generation of demands originating from the user operated nodes by simply adding demands to the queues based on the rates output by the RCP. Such an approach trades the difficulty of timing synchronization for the requirement of increased power of the classical processor at the EGS. To reduce the need for timing synchronization, a second generation architecture may be designed which does not rely on fixed, centrally defined time slots.

A. PROOFS

1) OUTLINE OF GOALS TO PROVE

In this section, we will prove two theorems to establish the results quoted in the main body of the article. The results are as follows.

1) The capacity region of the EGS is the set of demand arrival rate vectors fully contained in the set $\mathcal{C}$ (8) and maximum weight scheduling (Definition II.6) supports any rate vector from within $\mathcal{C}$ (Theorem II.1). To establish the capacity region, we first prove a proposition stating that any rate vector $\boldsymbol{\lambda} \notin \mathcal{C}$ necessarily results in divergent queues. We then prove a second proposition establishing at once that any rate vector $\boldsymbol{\lambda} \in \text{Int}\mathcal{C}$ is supportable under some scheduling algorithm and that maximum weight scheduling is such a scheduling

algorithm. Therefore, we also demonstrate that maximum weight scheduling is throughput optimal.

2) The RCP, Algorithm 1, results in the calculation of a sequence of rate and price vector pairs $\big(\boldsymbol{\lambda}(t_n), \boldsymbol{p}(t_n)\big)$ which converge to optimal solutions $\big(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{p}}\big)$ of the primal and dual problems, defined in Section III (Theorem III.1).

2) PROOF OF THEOREM II.1

First, it is to be shown that no rate vector $\boldsymbol{\lambda} \notin \mathcal{C}$ of an EGS with R resources is supportable under any scheduling algorithm.

Proposition V.1: If $\boldsymbol{\lambda} \notin \mathcal{C}$, no scheduling algorithm can support $\boldsymbol{\lambda}$.

Proof: There are three cases where $\boldsymbol{\lambda} \notin \mathcal{C}$

1) $\sum_s \lambda_s > R \cdot p_{\text{gen}}$;
2) $\lambda_{s^*} > x_{s^*} \cdot p_{\text{gen}}$ for some $s^* \in S$;
3) $\boldsymbol{\lambda}$ is not nonnegative ($\exists\ \lambda_{s^*} < 0$ for at least some $s^* \in S$).

In the third case, the node pair corresponding to session s^* has set a nonphysical rate and the rate must be changed. The proof for case (2) is very similar to case (1) and equations from the first case are reused or modified to complete the proof of case (2). The main strategy of the proof relies on Definition II.7; a rate vector $\boldsymbol{\lambda} \notin \mathcal{C}$ is not supportable if $\boldsymbol{\lambda}$ causes the queue lengths at the EGS processor to diverge with probability 1, regardless of scheduling algorithm. To prove the proposition in each case, it serves to calculate the total queue length.

Proposition V.1 (1): Suppose $\sum_s \lambda_s > R \cdot p_{\text{gen}}$. Then, $\exists\ \epsilon > 0$ such that

$$\sum_s \lambda_s \geq R \cdot p_{\text{gen}} + \epsilon. \tag{40}$$

Assume that the initial length of each queue is finite. The sum of queue lengths at time step t_{n+1}, $\sum_s q_s(t_{n+1})$ is

$$\begin{aligned}\sum_s q_s(t_{n+1}) &= \sum_s \big[q_s(t_n) + a_s(t_n) - g_s(t_n)\big]^+ \\ &\geq \sum_s \big(q_s(t_n) + a_s(t_n) - g_s(t_n)\big) \\ &\geq \sum_s \left(q_s(t_1) + \sum_{t_i=t_1}^{t_n} \big(a_s(t_i) - g_s(t_i)\big)\right)\end{aligned} \tag{41}$$

where $a_s(t_i)$ is the integer number of demands submitted by communication session s at time step t_i and $g_s(t_i)$ is the integer number of successfully generated entangled pairs between the nodes corresponding to communication session s in time step t_i. The final inequality in (41) follows from the previous inequality by repeated application of (6). By the strong law of large numbers

$$\lim_{n\to\infty} \frac{1}{t_n} \sum_{t_i=t_1}^{t_n} a_s(t_i) = \lambda_s\ \forall s \in S, \text{ with probability 1.} \tag{42}$$

Recall that the number of successfully generated entangled pairs between the nodes corresponding to communication session s at time t_i is a sample from a binomial random process where the number of trials is set by $M_s(t_i)$ and the trial success probability is p_{gen}

$$g_s(t_i) \sim \text{Bin}\big(M_s(t_i),\ p_{\text{gen}}\big).$$

By the strong law of large numbers

$$\lim_{n\to\infty} \frac{1}{t_n} \sum_{t_i=t_1}^{t_n} g_s(t_i) = M_s(t_n) \cdot p_{\text{gen}}, \text{ with probability 1.} \tag{43}$$

Since each feasible schedule satisfies $\sum_s M_s(t_i) \leq R$, it follows from (43) that

$$\begin{aligned}\lim_{n\to\infty} \frac{1}{t_n} \sum_{t_i=t_1}^{t_n} \sum_s g_s(t_i) &= \sum_s \lim_{n\to\infty} \frac{1}{t_n} \sum_{t_i=t_1}^{t_n} g_s(t_i) \\ &\leq R \cdot p_{\text{gen}}\end{aligned} \tag{44}$$

where we use the distribution property of limits, which is possible because the individual limits (43) exist. Finally, by assumption (40) and (41), (42) and (44)

$$\begin{aligned}&\lim_{n\to\infty} \frac{1}{t_n} \sum_s q_s(t_{n+1}) \\ &\quad\geq \lim_{n\to\infty} \frac{1}{t_n} \sum_s q_s(t_1) \\ &\qquad + \lim_{n\to\infty} \frac{1}{t_n} \sum_{t_i=t_1}^{t_n} \left(\sum_s a_s(t_i) - \sum_s g_s(t_i)\right) \\ &\quad\geq \sum_s \lambda_s - R \cdot p_{\text{gen}} \\ &\quad\geq R \cdot p_{\text{gen}} + \epsilon - R \cdot p_{\text{gen}} \\ &\quad\geq \epsilon.\end{aligned} \tag{45}$$

Therefore, with probability 1, $\sum_s q_s(t_n) \to \infty$ as $n \to \infty$, so $\boldsymbol{\lambda}$ is not supportable, regardless of scheduling algorithm.

Proposition V.1 (2): Suppose that $\lambda_{s^*} > x_{s^*} \cdot p_{\text{gen}}$ for some $s^* \in S$. Then, $\exists\ \epsilon > 0$ such that

$$\lambda_{s^*} \geq x_{s^*} \cdot p_{\text{gen}} + \epsilon. \tag{46}$$

In this case, we show that $\boldsymbol{\lambda}$ is not supportable by proving that the queue $q_{s^*}(t_i)$ of demands associated with communication session s^* diverges for large t_i. Recall (43) and note $M_s(t_i) \leq x_s\ \forall s\ \forall t_i$. This inequality describes that a maximum of x_s heralding stations can be allocated any communication session s in t_i. With this restriction, (43) becomes

$$\lim_{n\to\infty} \frac{1}{t_n} \sum_{t_i=t_1}^{t_n} g_s(t_i) \leq x_s \cdot p_{\text{gen}}\ \forall s. \tag{47}$$

Combining assumption (46) using (42) and (47), and making repeated use of (6)

$$\begin{aligned}\lim_{n\to\infty}\frac{1}{t_n}q_{s^*}(t_{n+1}) &\geq \lim_{n\to\infty}\frac{1}{t_n}q_{s^*}(t_1)\\ &+\lim_{n\to\infty}\frac{1}{t_n}\sum_{t_i=t_1}^{t_n}\big(a_{s^*}(t_i)-g_{s^*}(t_i)\big)\\ &\geq \lambda_{s^*}-x_{s^*}\cdot p_{\text{gen}}\\ &\geq x_{s^*}\cdot p_{\text{gen}}+\epsilon-x_{s^*}\cdot p_{\text{gen}}\\ &\geq \epsilon. \end{aligned}\tag{48}$$

Therefore, with probability 1, $q_{s^*}(t_{n+1})\to\infty$ as $n\to\infty$. Hence $\boldsymbol{\lambda}$ is not supportable. ■

Proposition V.1 proved that rate vectors $\boldsymbol{\lambda}\notin\mathcal{C}$ are not in the capacity region of the EGS. To finish proving $\mathcal{C}$ is the capacity region of the EGS (Theorem II.1), it is necessary to prove that any rate vector $\boldsymbol{\lambda}\in\mathcal{C}$ is supportable under some scheduling algorithm. To do so, we prove that the specific scheduling algorithm of maximum weight scheduling (Definition II.6) supports all arrival rate vectors fully contained in $\mathcal{C}$.

Proposition V.2: Maximum weight scheduling can support any arrival rate vector $\boldsymbol{\lambda}$ for which $\exists\ \epsilon>0$ such that $(1+\epsilon)\boldsymbol{\lambda}\in\mathcal{C}$.

Modeling a queue vector as a Markov chain is a standard tool in queuing theory [23]. This approach makes it possible to take advantage of the many strong analytic results on the behavior of Markov chains, which can then be used to make statements about the queue vector. The vector $\boldsymbol{q}(t_n)=\big(q_s(t_n)\ \forall s\big)$ of queued demands from each communication session maintained in the processor at t_n can be modeled as a Markov chain, with transitions given by (6). An irreducible Markov chain has the property that any state i of the chain is reachable from any other state j. A positive recurrent Markov chain has the property that from any state i, the expectation value of the time it will take to revisit any other state j is finite. A queue vector, with specified dynamics, that can be modeled as an irreducible Markov chain with the property of positive recurrence will not diverge (i.e., is guaranteed to remain a finite queue) [23]. The dynamics of such a queue vector are fixed by the arrival rate vector and the scheduling routine, therefore if a queue vector can be modeled as a positive recurrent Markov chain, the arrival rate vector is supportable by the scheduling routine. To prove Proposition V.2, we demonstrate that the queue vector is an irreducible Markov chain and use the Foster–Lyapunov theorem to prove that whenever $\boldsymbol{\lambda}$ lies strictly within $\mathcal{C}$ the Markov chain is also positive recurrent. An equivalent statement is that all rate vectors lying strictly within $\mathcal{C}$ are supportable by some scheduling algorithm.

Theorem V.1 (Foster–Lyapunov Theorem [23]): Let $\{X_k\}$ be an irreducible Markov chain with a state space $\mathcal{S}$. Suppose that there exists a function $V:\mathcal{S}\to\mathbb{R}^+$ and a finite set $\mathcal{B}\subseteq\mathcal{S}$ satisfying the following conditions.

1) $\mathbb{E}[V(X_{k+1})-V(X_k)|X_k=x]\leq-\epsilon$ if $x\in\mathcal{B}^c$, for some $\epsilon>0$.
2) $\mathbb{E}[V(X_{k+1})-V(X_k)|X_k=x]\leq A$ if $x\in\mathcal{B}$, for some $A<\infty$.

Then the Markov chain $\{X_k\}$ is positive recurrent.

Proof of Proposition V.2: First, we establish that the queue vector $\boldsymbol{q}(t_i)\ \forall t_i$ is an irreducible Markov chain. The queue vector $\boldsymbol{q}(t_i)$ is a Markov chain with state space

$$\mathcal{S}=\{\boldsymbol{q}:\ \boldsymbol{q}\text{ is reachable from }\mathbf{0}\text{ under the given scheduling algorithm}\}.$$

Assume that $\boldsymbol{q}(t_1)$ is finite and $\boldsymbol{q}(t_1)\in\mathcal{S}$. It follows from the definition of $\mathcal{S}$ that $\boldsymbol{q}(t_i)\in\mathcal{S}\ \forall t_i$ if $\boldsymbol{q}(t_1)\in\mathcal{S}$. Irreducibility of $\boldsymbol{q}(t_i)\ \forall t_i$ requires that any state $\boldsymbol{q}(t_j)$ is reachable from any other state $\boldsymbol{q}(t_i)$. By the definition of the state space $\mathcal{S}$, it suffices to demonstrate that from $\boldsymbol{q}(t_i)$, the Markov chain can always return to $\mathbf{0}$. Under maximum weight scheduling (Definition II.6), the processor always serves $k(t_i)$ demands per time-slot, where

$$k(t_i)=\max\{k\ :\ k\leq R\text{ and }k\leq\sum_s\ \min\big(|q_s(t_i)|,\ x_s\big)\}$$

where $|q_s(t_i)|$ is the number of demands in the queue for session s in time-slot t_i and x_s is the maximum number of resource modules that can be allocated communication session s per time-slot. Hence when $\boldsymbol{q}(t_i)$ is nonzero, at least one demand and up to R demands are served per time-slot. Therefore, from any $\boldsymbol{q}(t_i)\in\mathcal{S}$, $\boldsymbol{q}(t_{i+l})=\mathbf{0}$ is reachable from $\boldsymbol{q}(t_i)$ in $l\in\{\lceil\frac{|\boldsymbol{q}(t_i)|}{R}\rceil,\lceil\frac{|\boldsymbol{q}(t_i)|}{R}\rceil+1,\ldots,\ |\boldsymbol{q}(t_i)|\}$ time steps, where $|\boldsymbol{q}(t_i)|:=\sum_s\ |q_s(t_i)|$. Since any other $\boldsymbol{q}(t_j)\in\mathcal{S}$ is then reachable from $\mathbf{0}$, it follows that $\boldsymbol{q}(t_i)$ is irreducible. To prove that $\boldsymbol{\lambda}$ is supportable, it suffices to demonstrate that $\boldsymbol{q}(t_i)$ is positive recurrent.

Define the Lyapunov function

$$L\big(\boldsymbol{q}(t_i)\big)=\frac{1}{2}\sum_s q_s^2(t_i).\tag{49}$$

To apply the the Foster–Lyapunov theorem (V.1), the key quantity is the drift of $L\big(\boldsymbol{q}(t_i)\big)$. Using the queue update dynamics (6), the drift can be expanded as

$$\begin{aligned}&L\big(\boldsymbol{q}(t_{i+1})\big)-L\big(\boldsymbol{q}(t_i)\big)\\ &=\frac{1}{2}\sum_s\Big(\big[q_s(t_i)+a_s(t_i)-g_s(t_i)\big]^+\Big)^2-\frac{1}{2}\sum_s q_s^2(t_i)\\ &\leq\frac{1}{2}\sum_s\big(q_s(t_i)+a_s(t_i)-g_s(t_i)\big)^2-\frac{1}{2}\sum_s q_s^2(t_i)\\ &=\frac{1}{2}\sum_s\big(a_s(t_i)-g_s(t)\big)^2\\ &\quad+\sum_s q_s(t_i)\big(a_s(t_i)-g_s(t_i)\big).\end{aligned}\tag{50}$$

Taking the conditional expectation of the Lyapunov drift with respect to the randomness of arrivals and the probabilistic

success of scheduled demands

$$\begin{aligned}&\mathbb{E}\big[L(\boldsymbol{q}(t_{i+1})) - L(\boldsymbol{q}(t_i)) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big] \\ &\quad\le \frac{1}{2}\sum_s \mathbb{E}\big[\big(a_s(t_i) - g_s(t_i)\big)^2 \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big] \\ &\qquad+ \sum_s \mathbb{E}\big[q_s(t_i)\big(a_s(t_i) - g_s(t_i)\big) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big] \qquad (51)\end{aligned}$$

where $\tilde{\boldsymbol{q}} \in \mathcal{S}$ is a particular queue vector.

Using $\big(a_s - g_s\big)^2 \le a_s^2 + g_s{}^2$ and the linearity of expectation, the first term of the conditional expectation can be rewritten

$$\begin{aligned}&\mathbb{E}\left[\sum_s \big(a_s(t_i) - g_s(t_i)\big)^2 \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\right] \\ &\quad\le \sum_s \mathbb{E}\big[a_s^2(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big] + \sum_s \mathbb{E}\big[g_s{}^2(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big].\end{aligned} \qquad (52)$$

Recall that $g_s(t_i) \le M_s(t_i) \le x_s \ \forall s \ \forall t_i$. Hence

$$\sum_s \mathbb{E}\big[g_s{}^2(t) \mid \boldsymbol{q}(t) = \tilde{\boldsymbol{q}}\big] \le \sum_s x_s^2. \qquad (53)$$

Define the variance in the arrivals to the queue of session s, $\sigma_s^2 := \mathrm{Var}[a_s(t_i)]$. Then, noting that the arrivals are independent of the state of the queues, using the definition of variance and $\mathbb{E}[a_s(t_i)] = \lambda_s$

$$\mathbb{E}[a_s^2(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}] = \mathbb{E}[a_s^2(t_i)] = \sigma_s^2 + \lambda_s^2. \qquad (54)$$

Together (53) and (54) bound the first term of (51)

$$\begin{aligned}&\frac{1}{2}\sum_s \mathbb{E}\big[\big(a_s(t_i) - g_s(t_i)\big)^2 \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big] \\ &\qquad\le \frac{1}{2}\sum_s \big(\sigma_s^2 + \lambda_s^2 + x_s^2\big) =: B.\end{aligned}$$

Then (51) is

$$\begin{aligned}&\mathbb{E}\big[L(\boldsymbol{q}(t_{i+1})) - L(\boldsymbol{q}(t_i)) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big] \\ &\quad\le B + \sum_s \mathbb{E}\big[q_s(t_i)\big(a_s(t_i) - g_s(t_i)\big) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big] \\ &\quad= B + \sum_s \tilde{q}_s\big(\lambda_s - \mathbb{E}\big[g_s(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big]\big). \qquad (55)\end{aligned}$$

Recall that the conditional expectation of the Lyapunov drift is taken with respect to the randomness of the arrival processes as well as the success of scheduled demands. The schedule selected for a given time-slot depends on the queues, but the success of any scheduled demand does not. The conditional expectation of pair production for communication session s can be rewritten as

$$\mathbb{E}[g_s(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}] = p_{\text{gen}} \cdot \mathbb{E}[M_s(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}]. \qquad (56)$$

Recall that $\boldsymbol{M}$ denotes the schedule decided under the maximum weight scheduling policy, Definition II.6. Allow $\tilde{\boldsymbol{M}}$ to denote a schedule that is decided by any other scheduling policy. It follows from Definition II.6 that

$$\sum_s \tilde{q}_s \cdot \mathbb{E}[M_s(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}] \ge \sum_s \tilde{q}_s \cdot \mathbb{E}[\tilde{M}_s(t_i) | \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}]. \qquad (57)$$

Consider a scheduling policy $\tilde{\boldsymbol{M}}$ which schedules each session at a rate of $\frac{\lambda_s + \epsilon}{p_{\text{gen}}}$ (this is possible since, by assumption, $(1+\epsilon)\boldsymbol{\lambda} \in \mathcal{C}$). Such a scheduling policy is aware of the demand arrival rates to each queue but is not demand based (i.e., it does not use queue information in deciding the schedule). Hence

$$\begin{aligned}\sum_s \tilde{q}_s \cdot \mathbb{E}[\tilde{M}_s(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}] &= \sum_s \tilde{q}_s \cdot \mathbb{E}[\tilde{M}_s(t_i)] \\ &= \sum_s \tilde{q}_s\left(\frac{\lambda_s + \epsilon}{p_{\text{gen}}}\right). \qquad (58)\end{aligned}$$

Combining (56)–(58), the conditional expectation of the Lyapunov drift is bounded by

$$\begin{aligned}&\mathbb{E}\big[L(\boldsymbol{q}(t_{i+1})) - L(\boldsymbol{q}(t_i)) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big] \\ &\quad\le B + \sum_s \tilde{q}_s\lambda_s - \sum_s \tilde{q}_s \cdot p_{\text{gen}} \cdot \mathbb{E}\big[M_s(t_i) \mid \boldsymbol{q}(t_i) = \tilde{\boldsymbol{q}}\big]\big) \\ &\quad\le B + \sum_s \tilde{q}_s\lambda_s - p_{\text{gen}} \cdot \sum_s \tilde{q}_s \cdot \mathbb{E}[\tilde{M}_s(t_i)] \\ &\quad= B - \epsilon \cdot \sum_s \tilde{q}_s. \qquad (59)\end{aligned}$$

Application of the Foster–Lyapunov theorem completes the proof. ■

3) PROOF OF THEOREM III.1

The proof of this theorem is closely inspired by the proof of an analogous theorem in [26]. To begin, we establish basic properties of the dual function which follow from assumption A1.

Lemma V.1: Under assumption A1 the dual objective function $D(\boldsymbol{p})$ (33) is convex, lower bounded, and continuously differentiable.

For each session $s \in S$, and price vector $\boldsymbol{p} \ge 0$, define the quantity $\beta_s(\boldsymbol{p}) : \mathbb{R}^{1+N} \mapsto \mathbb{R}^+$ as follows:

$$\beta_s(\boldsymbol{p}) := \begin{cases} \frac{1}{-f_s''\left(\lambda_s^*(\boldsymbol{p})\right)}, & \text{if } f_s'(\lambda_{\text{gen},s}^{\max}) \le p_s \le f_s'(\lambda_s^{\min}) \\ 0, & \text{otherwise} \end{cases} \qquad (60)$$

where $p_s := p_c + \sum_{u \in s} p_u$ and $\lambda_s^*(\boldsymbol{p})$ is the unique maximizer of (33).

For any price vector $\boldsymbol{p} \ge \boldsymbol{0}$, define the matrix $B(\boldsymbol{p}) = \mathrm{diag}\big(\beta_s(\boldsymbol{p}),\ s \in S\big)$ to be the $|S| \times |S|$ matrix with diagonal elements $\beta_s(\boldsymbol{p})$. Note that from Assumption A2, for all $\boldsymbol{p} \ge \boldsymbol{0}$

$$0 \le \beta_s(\boldsymbol{p}) \le \alpha_s < \infty. \qquad (61)$$

Define the user-session mapping matrix R to be the $N \times |S|$ matrix whose $(u,\ s)$th entry is given by

$$R_u^{\ s} = \begin{cases} 1, & \text{if } u \in s \text{ or equivalently } s \in S(u) \\ 0, & \text{otherwise.} \end{cases} \tag{62}$$

The augmented session mapping matrix $\tilde{R}$ is the $\big(1+N\big) \times |S|$ matrix whose $(r,\ s)$th entry is

$$\tilde{R}_r^{\ s} = \begin{cases} 1, & \text{if } r = 1 \\ R_{r-1}^{\ \ s}, & r \neq 1. \end{cases} \tag{63}$$

Lemma V.2: Under Assumption A1, where it exists, the Hessian of the dual function D is given by

$$\nabla^2 D(\boldsymbol{p}) = \tilde{R} B(\boldsymbol{p}) \tilde{R}^{\mathrm{T}}. \tag{64}$$

Proof: Let $\nabla_p \lambda^*$ denote the $|S| \times (1+N)$ Jacobian matrix whose $(s,\ r)$th element is $\big(\partial \lambda_s^* / \partial p_r\big)(\boldsymbol{p}),\ r \in (c,\ u\ \forall u)$. As a consequence of the inverse function theorem [27] and (31), when it exists, we have

$$\frac{\partial \lambda_s^*}{\partial p_r} = \begin{cases} \dfrac{\tilde{R}_r^{\ s}}{f_s'' \big(\lambda_s^*(\boldsymbol{p})\big)}, & \text{if } f_s'(\lambda_{\mathrm{gen},s}^{\max}) < p_s < f_s'(\lambda_s^{\min}) \\ 0, & \text{otherwise} \end{cases} \tag{65}$$

where $r \in (c,\ u\ \forall u)$. Using (60) we can write

$$\nabla_p \lambda^* = -B(\boldsymbol{p}) \tilde{R}^{\mathrm{T}}. \tag{66}$$

From (35) and (36), $\nabla D(\boldsymbol{p}) = c - \tilde{R}\lambda$, where $c := (\lambda_{\mathrm{EGS}}, \overline{\lambda}_u\ \forall u)$. Therefore

$$\nabla^2 D(\boldsymbol{p}) = -\tilde{R} \nabla_p \boldsymbol{\lambda} = \tilde{R} B(\boldsymbol{p}) \tilde{R}^{\mathrm{T}}.$$

■

Lemma V.3: Under Assumptions A1 and A2, the gradient of the dual function $\nabla D(\boldsymbol{p})$ (33), (35), and (36) is Lipschitz continuous with Lipschitz constant $L = \max_{s \in S} \beta_s(p) \cdot\ |S|$.

We use the following theorem to prove Lemma V.3,

Theorem V.2 (Rudin, 9.19 [27]): Suppose $\mathbf{f}$ maps a convex open set $E \subset \mathbb{R}^n$ into $\mathbb{R}^m$, $\mathbf{f}$ is differentiable in E, and there is a real number M such that

$$||\mathbf{f}'(\mathbf{x})|| \leq M$$

for every $\mathbf{x} \in E$. Then

$$|\mathbf{f}(\mathbf{b}) - \mathbf{f}(\mathbf{a})| \leq M |\mathbf{b} - \mathbf{a}|$$

for all $\mathbf{a} \in E,\ \mathbf{b} \in E$.

Proof of Lemma V.3: From Lemma V.2, the Hessian of the dual function is the $\big(1+N\big) \times \big(1+N\big)$ matrix $\nabla^2 D(\boldsymbol{p}) = \tilde{R} B(\boldsymbol{p}) \tilde{R}^{\mathrm{T}}$. It is simple to explicitly determine the (r, r')th entry of $\nabla^2 D(\boldsymbol{p})$. By matrix multiplication, $B(\boldsymbol{p}) \tilde{R}^{\mathrm{T}}$ is the $|S| \times \big(1+N\big)$ matrix whose $(s,\ r)$th entry is

$$\big(B(\boldsymbol{p}) \tilde{R}^{\mathrm{T}}\big)_s^{\ r} = \begin{cases} \beta_s(\boldsymbol{p}), & \text{if } r = 1 \text{ or } \big(r > 1 \text{ and } s \in S(r-1)\big) \\ 0, & \text{otherwise.} \end{cases} \tag{67}$$

By matrix multiplication we calculate the $(r,\ r')$th entry of $\nabla^2 D(\boldsymbol{p})$ as

$$\begin{aligned} \big(\nabla^2 D(\boldsymbol{p})\big)_r^{\ r'} &= \sum_s R_r^{\ s} \big(B(\boldsymbol{p}) R^{\mathrm{T}}\big)_s^{\ r'} \\ &= \begin{cases} \sum_s \beta_s, & r = r' = 1 \\ \sum_{s \in S(r'-1)} \beta_s, & r = 1 \text{ and } r' > 1 \\ \sum_{s \in S(r-1)} \beta_s, & r > 1 \text{ and } r' = 1 \\ \sum_{s \in S(r-1) \cap S(r'-1)} \beta_s, & r > 1 \text{ and } r' > 1. \end{cases} \end{aligned} \tag{68}$$

Using the definition of the operator norm [[27], Definition 9.6 (c)] we bound the norm of the Hessian of the dual function

$$||\nabla^2 D(\boldsymbol{p})|| \leq \max_{s \in S} \beta_s \cdot\ |S|. \tag{69}$$

The result of the lemma then follows by application of Theorem V.2. Proof of Theorem III.1 is assured by the following theorem, which follows from the descent lemma of convex optimization theory [28]. ■

Theorem V.3 ([28]): Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function and let X be a closed convex set. Assume ∇f satisfies the Lipschitz condition with Lipschitz constant L and consider the gradient projection iteration

$$x_{k+1} = P_x\big(x_k - \gamma \nabla f(x_k)\big)$$

with a constant step-size γ in the range $\big(0,\ \frac{2}{L}\big)$. Then every limit point $\overline{x}$ of the generated sequence $\{x_k\}$ satisfies the optimality condition $\nabla f(\overline{x})^{\mathrm{T}}(x - \overline{x}) \geq 0\ \forall x \in X$. ■

REFERENCES

[1] S. Wehner, D. Elkouss, and R. Hanson, "Quantum Internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, 2018, Art. no. eaam9288, doi: 10.1126/science.aam9288.

[2] B. H. Charles and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.*, vol. 560, pp. 7–17, Dec. 2014, doi: 10.1016/j.tcs.2011.08.039.

[3] A. K. Ekert, "Quantum cryptography based on Bell's theorem," Phys. Rev. Lett., vol. 67, pp. 661–663, Aug. 1991, doi: 10.1103/PhysRevLett.67.661.

[4] P. Arrighi and L. Salvail, "Blind quantum computation," *Int. J. Quantum Inf.*, vol. 04, no. 05, pp. 883–898, 2006, doi: 10.1142/S0219749906002171.

[5] A. Broadbent, J. Fitzsimons, and E. Kashefi, "Universal blind quantum computation," in *Proc. 50th Annu. IEEE Symp. Foundations Comput. Sci.*, 2009, pp. 517–526, doi: 10.1109/FOCS.2009.36.

[6] P. Kómár et al., "A quantum network of clocks," *Nature Phys.*, vol. 10, no. 8, pp. 582–587, Jun. 2014, doi: 10.1038/nphys3000.

[7] C. Cabrillo, J. I. Cirac, P. García-Fernández, and P. Zoller, "Creation of entangled states of distant atoms by interference," *Phys. Rev. A*, vol. 59, pp. 1025–1033, Feb. 1999, doi: 10.1103/PhysRevA.59.1025.

[8] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, "Long-distance quantum communication with atomic ensembles and linear optics," *Nature*, vol. 414, no. 6862, pp. 413–418, Nov. 2001, doi: 10.1038/35106500.

[9] H. Bernien et al., "Heralded entanglement between solid-state qubits separated by three metres," *Nature*, vol. 497, no. 7447, pp. 86–90, Apr. 2013, doi: 10.1038/nature12016.

[10] P. C. Humphreys et al., "Deterministic delivery of remote entanglement on a quantum network," *Nature*, vol. 558, no. 7709, pp. 268–273, Jun. 2018, doi: 10.1038/s41586-018-0200-5.

[11] P. Maunz, D. L. Moehring, S. Olmschenk, K. C. Younge, D. N. Matsukevich, and C. Monroe, "Quantum interference of photon pairs from two remote trapped atomic ions," *Nature Phys.*, vol. 3, no. 8, pp. 538–541, 2007, doi: 10.1038/nphys644.

[12] V. Krutyanskiy et al., "Entanglement of trapped-ion qubits separated by 230 meters," *Phys. Rev. Lett.*, vol. 130, Feb. 2023, Art. no. 050803, doi: 10.1103/PhysRevLett.130.050803.

[13] C. W. Chou, H. D. Riedmatten, D. Felinto, S. V. Polyakov, S. J. V. Enk, and H. J. Kimble, "Measurement-induced entanglement for excitation stored in remote atomic ensembles," *Nature*, vol. 438, no. 7069, pp. 828–832, Dec. 2005, doi: 10.1038/nature04353.

[14] C. W. Chou et al., "Functional quantum nodes for entanglement distribution over scalable quantum networks," *Science*, vol. 316, no. 5829, pp. 1316–1320, Jun. 2007, doi: 10.1126/science.1140300.

[15] T. V. Leent et al., "Entangling single atoms over 33 km telecom fibre," *Nature*, vol. 607, no. 7917, pp. 69–73, Jul. 2022, doi: 10.1038/s41586-022-04764-4.

[16] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the stochastic analysis of a quantum entanglement distribution switch," *IEEE Trans. Quantum Eng.*, vol. 2, 2021, Art. no. 4101016, doi: 10.1109/TQE.2021.3058058.

[17] G. Avis, F. Rozpdek, and S. Wehner, "Analysis of multipartite entanglement distribution using a central quantum-network node," 2022, *arXiv:2203.05517*, doi: 10.1103/PhysRevA.107.012609.

[18] L. Tassiulas and A. Ephremides, "Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992, doi: 10.1109/9.182479.

[19] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999, doi: 10.1109/26.780463.

[20] T. Vasantam and D. Towsley, "A throughput optimal scheduling policy for a quantum switch," in *Quantum Computing, Communication, and Simulation II*, vol. 12015, P. R. Hemmer and A. L. Migdall, Eds., San Fransisco, CA, USA: SPIE. 2022, pp. 14–23, doi: 10.1117/12.2616950.

[21] R. Yehia, S. Neves, E. Diamanti, and I. Kerenidis, "Quantum city: Simulation of a practical near-term metropolitan quantum network," 2022, *arXiv:2211.01190*, doi: 10.48550/arXiv.2211.01190.

[22] F. P. Kelly, A. K. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, Feb. 1998, doi: 10.1057/palgrave.jors.2600523.

[23] R. Srikant and L. Ying, *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective*. Cambridge, U.K.: Cambridge Univ. Press, 2014, doi: 10.1017/CBO9781139565844.

[24] G. Vardoyan and S. Wehner, "Quantum network utility maximization," 2022, *arXiv:2210.08135*, doi: 10.48550/arXiv.2210.08135.

[25] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Nashua, NH, USA: Athena Scientific, 1999. [Online]. Available: http://www.athenasc.com/nonlinbook.html

[26] S. H. Low and D. E. Lapsley, "Optimization flow control. I. Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999, doi: 10.1109/90.811451.

[27] W. Rudin, *Principles of Mathematical Analysis, 3rd ed*. New York, NY, USA: McGraw-Hill Inc., 1976. [Online]. Available: https://www.mheducation.com/highered/product/principles-mathematical-analysis-rudin/M9780070542358.html

[28] D. P. Bertsekas, *Convex Optimization Algorithms*. Belmont, MA, USA: Athena Scientific, 2015. [Online]. Available: http://www.athenasc.com/convexalgorithms.html