A Control Architecture for Entanglement Generation Switches in Quantum Networks

Gauthier, Scarlett; Vardoyan, Gayane; Wehner, Stephanie

DOI
10.1145/3610251.3610552

Publication date
2023

Document Version
Final published version

Published in
QuNet 2023 - Proceedings of the 1st Workshop on Quantum Networks and Distributed Quantum Computing

Citation (APA)
Gauthier, S., Vardoyan, G., & Wehner, S. (2023). A Control Architecture for Entanglement Generation Switches in Quantum Networks. In *QuNet 2023 - Proceedings of the 1st Workshop on Quantum Networks and Distributed Quantum Computing* (pp. 38-44). Association for Computing Machinery (ACM). https://doi.org/10.1145/3610251.3610552

A Control Architecture for Entanglement Generation Switches in Quantum Networks

Scarlett Gauthier, Gayane Vardoyan, Stephanie Wehner

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 (RCP) 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 (NUM) 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.

1 INTRODUCTION

A quantum network enables radically new capabilities that are provably impossible to attain in any classical network [26]. Examples include applications such as secure communication [7, 11], secure quantum computing in the cloud [1, 5], and clock synchronization [14]. 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 [6, 10] 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 [3, 12], Ion Traps [15, 17], Atomic Ensembles [8, 9] and Neutral Atoms [23]. 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 [12]). 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. The detectors are each connected to a device such as a Field Programmable Gate Array (FPGA), for triggering the next action of the entanglement generation sequence based on the measurement outcomes. 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

QuNet '23, September 10–14, 2023, New York, NY, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0306-5/23/09.
https://doi.org/10.1145/3610251.3610552

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).

1.1 Results Summary

We make the following contributions:

- We characterize (Theorem 2.7) 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.
- We prove (Theorem 2.7) that under the Maximum Weight Scheduling policy (Definition 2.4) 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.
- We derive the RCP, an algorithm to regulate the rates of bi-partite 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 Network Utility Maximization (NUM) and its quantum network extension (QNUM), where resource allocation in a (quantum) network is modelled as an optimization problem that can be solved using methods from convex optimization theory.
- We prove (Theorem 3.1) 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 Lagrange multipliers and Lagrangian duality theory.
- Finally, we supply numerical results that support our analysis. These results illustrate possible values of the tightness of convergence δ between the rate vectors yeilded by the RCP and the optimum, and the number of time slots $\Delta\tau$ that the RCP must run before convergence is achieved.

1.2 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 [2, 24]. 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 [24] the focus is on assessing the EDS performance in terms of the rate at which it creates $n-$partite entanglements, and in [2] the possible rate/fidelity combinations of GHZ states that may be supplied by an EDS [2] are studied.

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 backlogs of queued demands. A Maximum Weight scheduling policy was originally presented in [21] for resource allocation in classical communication networks and was adapted to the analysis of a single switch for classical networking in [18], 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 [22] the capacity region of an EDS, defined as the set of arrival rates of requests for end-to-end multi-partite 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. To do so, we first characterize the capacity region of the EGS, which has not been done before. Then, we prove 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 the the problem of moderating competition for the EGS resources.

In [19], 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 [19] 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 [13] and has been widely used to develop and analyze control policies for classical networks [20]. 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 [25]. 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 [3, 8, 9, 12, 15, 17, 23]. 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.

2 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 Figure 1 a) for an overview of the physical architecture. Below we delineate the process by which pairs of nodes may request (Figure 1 d)) and receive (Figure 1 b) and d)) resource allocations from the processor. We assume:

- the EGS operates in a fixed-duration time slotted system where t_n denotes the n^{th} time slot;
- timing synchronization between the processor and each node is continuously managed by classical control electronics at the physical layer;
- 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 *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 Figure 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 Figure 1 d). In what follows we introduce and explain each step of this communication sequence.

Definition 2.1 (Target Rate, Session). Each possible pair of nodes has the potential to require shared bipartite entanglement. To fulfill this need, a node pair (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 non-zero target rate is referred to as a *session* and associated with a unique session ID s. The set of session IDs S is defined as follows:

$$S := \left\{s = (i, j) \mid i < j \text{ and } \lambda_s(t_n) > 0, \forall\ (i, j) \in \{1, \cdots, N\}^2\right\} \quad (1)$$

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

The target rates of all sessions in t_n can be written as a vector $\boldsymbol{\lambda}(t_n) \in \mathbb{R}^{|S|}$, the s^{th} component of which is labelled by 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 2.2 (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, $a_s(t_n) \sim \text{Bernoulli}\big(\lambda_s(t_n) - \lfloor \lambda_s(t_n) \rfloor\big) + \lfloor \lambda_s(t_n) \rfloor$.

When the processor receives a demand it is added to one of $|S|$ queues, one for each session. Each queue processes demands in first in first out order.

Definition 2.3 ((Demand-Based) Schedule). A resource allocation *schedule* is a vector $\mathbf{M}(t_{n+1}) \in \mathbb{N}^{|S|}$ calculated by the EGS processor in t_n determining the assignment of the resources for 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

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

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, $\mathbf{q}(t_n) \in \mathbb{N}^{|S|}$, as it stands before new demands are registered in t_n and satisfies,

$$\sum_s M_s(t_{n+1}) \le \min\big(\sum_s q_s(t_n),\ R\big), \quad (3)$$

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

Definition 2.4 (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), (3), and (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

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

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 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

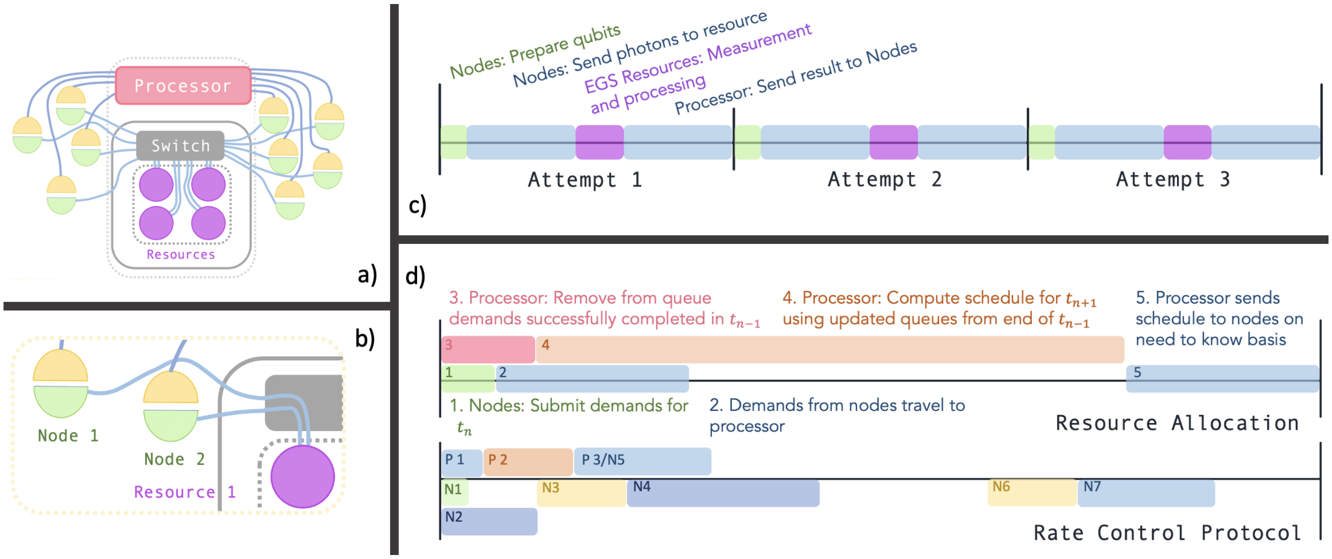
Processor
Switch
Resources
a)
Node 1
Node 2
Resource 1
b)
Nodes: Prepare qubits
Nodes: Send photons to resource
EGS Resources: Measurement and processing
Processor: Send result to Nodes
Attempt 1
Attempt 2
Attempt 3
c)
d)
3. Processor: Remove from queue demands successfully completed in t_{n-1}
4. Processor: Compute schedule for t_{n+1} using updated queues from end of t_{n-1}
5. Processor sends schedule to nodes on need to know basis
3
4
1
2
5
1. Nodes: Submit demands for t_n
2. Demands from nodes travel to processor
Resource Allocation
P 1
P 2
P 3/N5
N1
N3
N4
N6
N7
N2
Rate Control Protocol

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.)

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. Note that this number of successfully generated entangled pairs is a sample from a binomial random variable where the number of trials is the number of resources allocated to s, $M_s(t_n)$, 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 2.5 (Supportable rate). The arrival rate vector $\boldsymbol{\lambda}(t_n) \in \mathbb{R}^{+|S|} = \big(\lambda_s(t_n)\ \forall\ s\big)^{\mathrm{T}}$ is *supportable* if there exists a schedule under which,

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

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 2.6 (Capacity Region). The *capacity region* of the EGS is the set of arrival rate vectors that are supportable by the EGS.

Theorem 2.7 (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 $\boldsymbol{\lambda} \in \text{Int}C$, where C is defined as:*

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

$\lambda_{\text{EGS}} = R \cdot p_{\text{gen}}$ and $\lambda_{\text{gen},s}^{\max} = x_s \cdot p_{\text{gen}}$. Moreover, maximum weight scheduling (Definition 2.4) is throughput optimal and supports any rate vector $\boldsymbol{\lambda} \in \text{Int}C$. For proof, see the Appendices.

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.

3 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 C, as well as the node or user level constraints described by λ_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 [4] 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 [4], a solution to the dual problem is equivalent to a solution of the primal problem. The vector of Lagrange multipliers $\boldsymbol{p}(t_{n+1}) = \big(p_c(t_n), p_u(t_n)\ \forall u\big) \in \mathbb{R}^{+(1+N)}$, 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. Note that $u \in s \Leftrightarrow s \in S(u)$. The feasible rate region of communication session s is,

$$\Lambda_s := \{\lambda_s : \lambda_s^{\min} \leq\ \lambda_s \leq \lambda_{\text{gen},s}^{\max}\}\ \forall\ s, \tag{9}$$

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

$$\Lambda = \bigcup_s \Lambda_s. \tag{10}$$

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 [4], 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,

$$\sum_s \lambda_s^{\min} < \lambda_{\text{EGS}}; \tag{11}$$

$$\sum_{s \in S(u)} \lambda_s^{\min} < \lambda_u\ \forall\ u. \tag{12}$$

Algorithm 1: Rate Control Protocol (RCP)

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

(1) receives rates $\lambda_s(t_n)$ from all sessions $s \in S$;

(2) computes a new central price,

$$p_c(t_{n+1}) = \Big[\frac{1}{\lambda_{\text{EGS}}} \sum_s q_s(t_n) + \theta_c\big(\sum_s \lambda_s(t_n) - \lambda_{\text{EGS}}\big)\Big]^+, \tag{13}$$

where θ_c is a constant step size;

(3) broadcasts the new central price $p_c(t_{n+1})$ to all sessions $s \in S$.

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

(1) marks the subset of 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 session $s = (u, u') \in \text{COMM}(u)$;

(3) computes a new node price,

$$p_u(t_{n+1}) = \Big[\frac{1}{\lambda_u} \sum_{s \in S(u)} q_s(t_n) + \theta_u\big(\sum_{s \in S(u)} \lambda_s(t_n) - \lambda_u\big)\Big]^+,\ \forall\ u, \tag{14}$$

where θ_u is a constant step-size;

(4) communicates the new price $p_u(t_{n+1})$ to the communication node from every 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 session $s \in \text{COMM}(u)$,

$$\lambda_s(t_{n+1}) = \left[\Big(\frac{\mathrm{d}f_s}{\mathrm{d}\lambda_s}\Big)^{-1}\big(\boldsymbol{p}(t_{n+1})\big)\right]_{\lambda_s^{\min}}^{\lambda_{\text{gen},s}^{\max}}, \tag{15}$$

where $[z]_m^M = \max\big(\min(z,\ M), m\big)$ and $\boldsymbol{p}(t_i) = \big(p_c(t_i),\ p_u(t_i)\ \forall\ u\big)$ 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 session $s \in \text{COMM}(u)$.

The RCP is a gradient projection algorithm with constant step-sizes over the closed convex set Λ. To establish convergence we follow a similar treatment as in [16].

Theorem 3.1 (RCP Convergence). *Suppose assumptions A1 and A2 and the constraints (11, 12) 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 $\boldsymbol{p}(0) \geq \boldsymbol{0}$ vectors, every accumulation point $\big(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{p}}\big)$ of the sequence over time slots $\{\big(\boldsymbol{\lambda}(t_n), \boldsymbol{p}(t_n)\big)\}$ generated by the RCP is primal-dual optimal. Refer to the Appendices for proof.*

4 CASE STUDY

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

$$f_s(\lambda_s) = \log(\lambda_s)\ \forall\ s \in S. \tag{16}$$

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

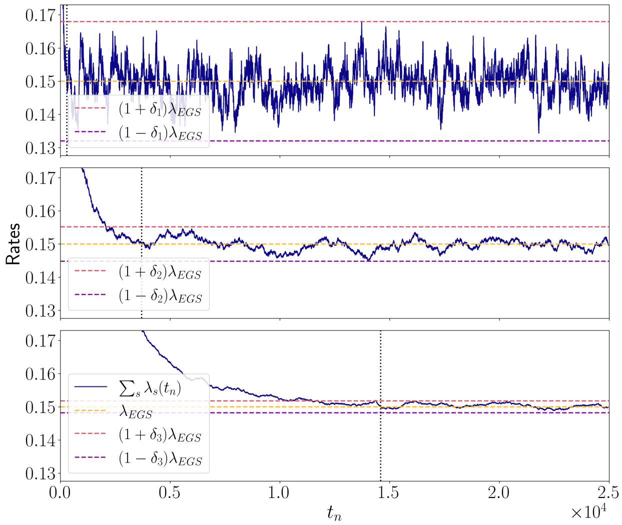

Figure 2: The RCP drives convergence of $\sum_s \lambda_s(t_n)$ to λ_{EGS} for an EGS with $R = 3$ resources, $p_{\text{gen}} = 0.05$, connected to $N = 20$ (top), $N = 50$ (middle) and $N = 100$ (bottom) nodes. Black dotted lines indicate $\Delta\tau$. Observed δ are $\delta_1 = 0.12$, $\delta_2 = 0.035$ and $\delta_3 = 0.012$. Step-sizes $(\theta_c, \theta_u\ \forall u)$ were all $1/(40 \cdot \lambda_{\text{EGS}})$.

is positive, there is at least one other session for which the proportional change is negative. For compatibility with Theorem 3.1 note that log utility functions satisfy A1 and A2 is satisfied with $\alpha_s = (\lambda^{\max}_{\text{gen},s})^2\ \forall s$.

Although the convergence theorem only guarantees asymptotic convergence of the sequence over time slots $\{(\boldsymbol{\lambda}(t_n), \boldsymbol{p}(t_n))\}$ to an optimal rate-price pair $(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{p}})$, 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 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 Figure 2 we numerically investigate $(\Delta\tau, \delta)$ for an EGS with $R = 3$ resources and $p_{\text{gen}} = 0.05$ 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 $\Delta\tau$ are the number of time slots that occur before the sum of demand rates $\left(\sum_s \lambda_s(t_n)\right)$ first crosses the optimal value λ_{EGS}. Reporting of δ is based on the maximum size of fluctuations of $\sum_s \lambda_s(t_n)$ about λ_{EGS} following $\Delta\tau$. As the number of sessions hosted by an EGS increases, we observe a trade-off between the $\Delta\tau$ 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 $\Delta\tau$.

5 OPEN QUESTIONS

We have presented the first control architecture for an EGS. The architecture is tailored to a simple system model, hence a natural corollary to this work is to create a refined version of the control architecture that will be compatible with a more versatile physical model. In particular it would be interesting to study sources of delay such as heterogeneous connection lengths between nodes and the EGS and to extend the definition of demand to allow pairs of users with variable quantum network node capabilities to demand packets of a number of entangled pairs, delivered at some desired rate.

REFERENCES

[1] P. Arrighi and L. Salvail. 2006. Blind Quantum Computation. *International Journal of Quantum Information* 04, 05 (2006), 883–898.

[2] G. Avis, F. Rozpędek, and S. Wehner. 2022. Analysis of Multipartite Entanglement Distribution using a Central Quantum-Network Node. (2022).

[3] H. Bernien, B. Hensen, W. Pfaff, G. Koolstra, M. S. Blok, L. Robledo, T. H. Taminiau, M. Markham, D. J. Twitchen, L. Childress, and R. Hanson. 2013. Heralded entanglement between solid-state qubits separated by three metres. *Nature* 497, 7447 (April 2013), 86–90.

[4] D. P. Bertsekas. 1999. *Nonlinear Programming, Second Edition.* Athena Scientific, Belmont, Massachusetts, USA.

[5] A. Broadbent, J. Fitzsimons, and E. Kashefi. 2009. Universal Blind Quantum Computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, Atlanta, Georgia, USA, 517–526. https://doi.org/10.1109/focs.2009.36

[6] C. Cabrillo, J. I. Cirac, P. García-Fernández, and P. Zoller. 1999. Creation of entangled states of distant atoms by interference. *Phys. Rev. A* 59 (Feb. 1999), 1025–1033. Issue 2.

[7] B. H. Charles and G. Brassard. 2014. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science* 560 (Dec. 2014), 7–11.

[8] C. W. Chou, H. de Riedmatten, D. Felinto, S. V. Polyakov, S. J. van Enk, and H. J. Kimble. 2005. Measurement-induced entanglement for excitation stored in remote atomic ensembles. *Nature* 438, 7069 (Dec. 2005), 828–832.

[9] C. W. Chou, J. Laurat, H. Deng, K. S. Choi, H. de Riedmatten, D. Felinto, and H. J. Kimble. 2007. Functional Quantum Nodes for Entanglement Distribution over Scalable Quantum Networks. *Science* 316, 5829 (June 2007), 1316–1320. https://doi.org/10.1126/science.1140300

[10] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller. 2001. Long-distance quantum communication with atomic ensembles and linear optics. *Nature* 414, 6862 (Nov. 2001), 413–418.

[11] A. K. Ekert. 1991. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.* 67 (Aug. 1991), 661–663. Issue 6. https://doi.org/10.1103/PhysRevLett.67.661

[12] P. C. Humphreys, N. Kalb, J. P. J. Morits, R. N. Schouten, R. F. L. Vermeulen, D. J. Twitchen, M. Markham, and R. Hanson. 2018. Deterministic delivery of remote entanglement on a quantum network. *Nature* 558, 7709 (June 2018), 268–273. https://doi.org/10.1038/s41586-018-0200-5

[13] F.P. Kelly, A.K. Maulloo, and D. Tan. 1998. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society* 49 (Feb. 1998).

[14] P. Kómár, E. M. Kessler, M. Bishof, L. Jiang, A. S. Sørensen, J. Ye, and M. D. Lukin. 2014. A quantum network of clocks. *Nature Physics* 10, 8 (June 2014), 582–587.

[15] V. Krutyanskiy, M. Galli, V. Krcmarsky, S. Baier, D. A. Fioretto, Y. Pu, A. Mazloom, P. Sekatski, M. Canteri, M. Teller, J. Schupp, J. Bate, M. Meraner, N. Sangouard, B. P. Lanyon, and T. E. Northup. 2023. Entanglement of Trapped-Ion Qubits Separated by 230 Meters. *Phys. Rev. Lett.* 130 (Feb. 2023), 050803. Issue 5. https://doi.org/10.1103/PhysRevLett.130.050803

[16] S.H. Low and D.E. Lapsley. 1999. Optimization flow control. I. Basic algorithm and convergence. *IEEE/ACM Transactions on Networking* 7, 6 (1999), 861–874.

[17] P. Maunz, D. L. Moehring, S. Olmschenk, K. C. Younge, D. N. Matsukevich, and C. Monroe. 2007. Quantum interference of photon pairs from two remote trapped atomic ions. *Nature Physics* 3, 8 (2007), 538–541.

[18] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand. 1999. Achieving 100% Throughput in an Input-Queued Switch. *IEEE Transactions on Communications* 47, 8 (Aug. 1999), 1260–1267.

[19] Y. Raja, N. Simon, D. Eleni, and K. Iordanis. 2022. Quantum City: simulation of a practical near-term metropolitan quantum network. (2022). https://arxiv.org/abs/2211.01190

[20] R. Srikant and Lei Ying. 2014. *Communication Networks: An Optimization, Control, and Stochastic Nerworks Perspective.* Cambridge University Press, Cambridge, UK.

[21] L. Tassiulas and A. Ephremides. 1992. Stability Properties of Constrained Queuing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks. *IEEE Trans. Automat. Control* 37, 12 (Dec. 1992), 1936–1948.

[22] V. Thirupathaiah and T. Don. 2022. A throughput optimal scheduling policy for a quantum switch. In *Quantum Computing, Communication, and Simulation II*, Philip R. Hemmer and Alan L. Migdall (Eds.). SPIE, San Fransisco, California, USA, 1201505.

[23] T. van Leent, M. Bock, F. Fertig, R. Garthoff, S. Eppelt, Y. Zhou, P. Malik, M. Seubert, T. Bauer, W. Rosenfeld, W. Zhang, C. Becher, and H. Weinfurter. 2022. Entangling single atoms over 33 km telecom fibre. *Nature* 607, 7917 (July 2022), 69–73.

[24] G. Vardoyan, S. Guha, P. Nain, and D. Towsley. 2021. On the Stochastic Analysis of a Quantum Entanglement Distribution Switch. *IEEE Transactions on Quantum Engineering* 2 (2021), 1–16.

[25] G. Vardoyan and S. Wehner. 2022. Quantum Network Utility Maximization. (2022). https://arxiv.org/abs/2210.08135

[26] S. Wehner, D. Elkouss, and R. Hanson. 2018. Quantum internet: A vision for the road ahead. *Science* 362, 6412 (2018).