A Quantum Overlay Network for Efficient Entanglement Distribution

Shahrooz Pouryousef[1], Nitish K. Panigrahy[2] and Don Towsley[1]

[1]College of Information and Computer Sciences, UMass Amherst
[2]School of Engineering & Applied Science, Yale University

***Abstract*—Distributing quantum entanglements over tances is essential for the realization of a global scale Internet. Most of the prior work and proposals a on-demand distribution of entanglements which may significant network resource under-utilization. In t we introduce Quantum Overlay Networks (QONs) fo entanglement distribution in quantum networks. Whe mand to create end-to-end user entanglements is lo can generate and store maximally entangled Bell pa pairs) at specific overlay storage nodes of the netwo during peak demands, requests can be served by p entanglement swaps either over a direct path from th or over a path using the storage nodes. We solve entanglement and storage resource allocation problem QON using a centralized optimization framework. W the performance of our proposed QON architectur wide number of network topologies under various setti extensive simulation experiments. Our results demons QONs fare well by a factor of 40% with respect to meeting surge and changing demands compared to traditional non-overlay proposals. QONs also show significant improvement in terms of average entanglement request service delay over non-overlay approaches.**

***Index Terms*—Quantum Overlay Networks, Storage, Fidelity, Quantum Network**

I. Introduction

The vision of a quantum Internet, a global network capable of transmitting quantum information, brings with it the promise of implementing quantum applications such as quantum key distribution (QKD) [6], quantum computation [10], quantum sensing [12], clock synchronisation [25], quantum-enhanced measurement networks [17], and many others [23]. Recent experiments have demonstrated successful quantum key distribution at short distances [31], [41], [37] and some are even commercially available [17].

Executing these applications relies on creating long distance quantum entanglements between end nodes in a quantum entanglement distribution network [26], [11]. Quantum entanglement is a shared state between two or more quantum bits (qubits) where the quantum state of individual qubits cannot be described independently of the others. Distributing quantum entanglement over long distances first involves creating entanglements, known as link-level entanglements, between adjacent nodes in a quantum network. Finally, end-to-end user entanglements are generated by having each node on the path connecting the end users perform a quantum operation known as an *entanglement swap* on the individual link-level entanglements. These end-to-end user entanglements can then be delivered to respective quantum applications such as QKD for consumption.

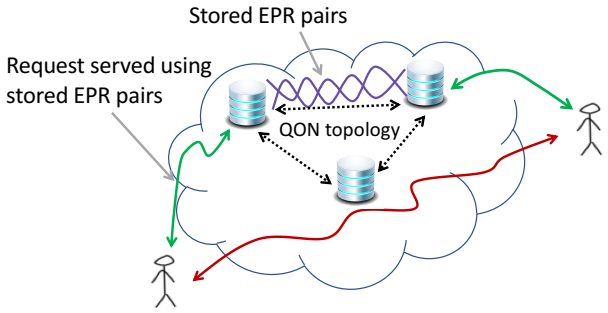


Figure 1: A pictorial depiction of serving user demands in a QON. Users can create end-to-end entanglement using either the entanglements stored at storage nodes (green arrows) or consuming link level entanglements directly from the network (red arrow).

As more and more quantum Internet based applications develop in the future, the ever-evolving nature of quantum applications will bring new challenges in managing and delivering quantum entanglement services to end users. Most studies of quantum networks have focused on developing routing protocols for entanglement generation [43], [30], [8], [40], [44] and network protocol stack design [26], [11]. Existing proposals for quantum networks generally assume static conditions and cannot serve end user demands at more than a predefined capacity at any given time. We envision time varying user traffic demands for end-to-end user entanglements in future quantum networks. Thus, situations may arise when the quantum network cannot generate entanglements at the requested rate. We pose the following research question. *What networking infrastructure will allow for handling peak traffic demands for creating end-to-end user entanglements in a quantum network?* We introduce Quantum Overlay Networks (QONs) as a solution to address this problem.

We briefly describe the QON architecture. In a QON, we envision that the network infrastructure provides entanglement

shahrooz@cs.umass.edu
nitishkumar.panigrahy@yale.edu
towsley@cs.umass.edu

storage service at some of the nodes of the network for future usage. The infrastructure may deploy a number of *storage nodes* each with a set of long-lived quantum memories as shown in Figure 1. Recent advances in quantum memory design suggest that memory lifetimes on the order of 16 seconds [15], [27] and up to one minute [19], [28] are achievable. Thus it is not unreasonable to expect quantum memory coherence times of a few minutes in 5-10 years.

However, QMs are likely to operate at cryogenic temperatures and may require isolated and costly hardware to operate. Thus, these resources will be limited and it is important to manage them efficiently in a QON. In periods of low demand, a quantum network can dedicate its resources (link-level entanglements) to creating entanglements between the storage nodes in addition to generating entanglement directly between users. During high demands, requests can be served by performing entanglement swaps either over a direct paths connecting end users (red solid arrow) or over paths using the storage nodes (green solid arrows) as shown in Figure 1.

When user demands are dynamic and stochastic, the QON resources (e.g., link-level entanglements and storage capacity) need to be manged carefully. In particular, the following key trade-off should be addressed. *How much of the network resources should be used to serve present user demands versus how much of the resources should be allocated for entanglement generation between storage nodes for future usage?* In this work, we precisely answer these questions. We formulate and solve several QON resource allocation problems with different performance objectives. These include serving peak end user demands, minimizing average request service delay, and maximizing weighted entanglement generation rate.

Apart from end-to-end entanglement generation rate, another important network performance metric is the end-to-end entanglement fidelity. Here, entanglement fidelity is a measure of the quality of a served entanglement. Due to noise in quantum channels and quantum operations, fidelity decreases with each quantum swap operation. Quantum applications may set a minimum threshold on end-to-end entanglement fidelity. End-to-end entanglement fidelity can be improved through an entanglement purification quantum operation [16]. Some end-to-end entanglements are sacrificed in the purification process, which decreases end-to-end entanglement generation rate. In this work, we apply purification on established end-to-end entanglements to satisfy the fidelity requirements of end-user applications.

Our contributions are summarized below.

- We propose and develop a QON architecture that places nodes with long-lived QMs in a quantum network to efficiently distribute quantum entanglements. To the best of our knowledge, this is the first work that presents a complete design and performance analysis of a quantum overlay network.
- We present several optimization formulations for a QON to optimize different performance objectives such as aggregate entanglement generation rate, and average entanglement request service delay.
- We evaluate the effectiveness of our proposed solution through experiments on both real-world classical networks (ex- ATT, IBM, Abilene, SURFnet) and random networks (power law and Erdos Renyi). Our results confirm that QONs can satisfy peak entanglement generation requests about 40% more than non-overlay approaches. QONs also increase the weighted entanglement generation rate, and significantly reduce the request service delay over non-overlay proposals.

The rest of the paper is organized as follows. First, we present some preliminary background information on quantum operations and the system model in Section II. In Section III, we explain different models and objectives of our proposed QON architecture. In Section IV, a thorough performance evaluation of the proposed architecture is conducted and conclusions are drawn in Section VI.

II. Technical Preliminaries

In this section, we provide a high level overview of some of the quantum operations and describe the system model that will be used throughout the paper.

A. Quantum Background

We now briefly mention some of the quantum operations that are relevant to this work.

Quantum States: A quantum bit (qubit) is fundamentally different than a classical bit. A classical bit can either be in state 0 or 1 while a qubit can be in a superposition state of state 0 and 1. Typically a qubit is represented as $|\alpha\rangle = \alpha_1|0\rangle + \alpha_2|1\rangle$. Similar to one qubit system, a two qubit quantum system can be in a superposition of $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ states and represented as $|\tilde{\alpha}\rangle = \tilde{\alpha}_1|00\rangle + \tilde{\alpha}_2|01\rangle + \tilde{\alpha}_3|10\rangle + \tilde{\alpha}_4|11\rangle$. While there exists many possible physical realization of a qubit, photonic qubits are most likely to be used for quantum communication and information transfer. In practice, these qubits can be sent through optical fiber based quantum channels using polarization, time-bin, or absence and presence of a photon based encodings [29].

Quantum Entanglement: An entangled state is a special type of multi-qubit quantum state which can not be written as the product of its individual component states. The measurement outcomes of an entangled state are correlated. A two qubit maximally entangled state shared between two parties A and B can be in one of the following four forms, also known as the *Bell pairs*: $|\psi^{\pm}_{AB}\rangle = \frac{|0_A 0_B\rangle \pm |1_A 1_B\rangle}{\sqrt{2}}$ and $|\phi^{\pm}_{AB}\rangle = \frac{|0_A 1_B\rangle \pm |1_A 0_B\rangle}{\sqrt{2}}$. We refer to Bell pair $|\psi^{+}_{AB}\rangle$ as an *EPR pair*.

Entanglement Swapping: Suppose two parties A and B share an EPR pair $|\psi^{+}_{AB}\rangle$. Also, assume that B shares another pair $|\psi^{+}_{BC}\rangle$ with C. Then B can create an EPR pair $|\psi^{+}_{AC}\rangle$ between A and C by performing a bell state measurement followed by classical communication exchange and a correction. This operation is known as *entanglement swapping*. The process can be repeated in a nested manner to create EPR pairs between distant parties.

Fidelity and Entanglement Purification: Fidelity is a widely used metric to quantify the quality of an entanglement. Due to noisy quantum channels, interaction with environment and imperfect quantum operations, the quality of an EPR pair typically decreases from its initial state. However, quantum applications may have a minimum threshold on the fidelity (F^{th}) of EPR pairs for subsequent usage. Thus, low quality EPR pairs delivered to the application may not be suitable for consumption. A quantum operation known as *Entanglement Purification* can solve this issue. Entanglement purification typically consumes two low-fidelity base EPR pairs and create one EPR pair with higher fidelity. Each purification step is probabilistic and both base pairs are lost in case of failure. This operation can be repeated in a nested manner by again purifying two newly purified EPR pair until a target fidelity is reached. These protocols are also known as *recurrence based purification protocols*. We refer to a purified EPR pair with fidelity greater than target fidelity as a high quality EPR pair.

Let F denote the fidelity of base EPR pairs that participate in purification. The average number $g(F, F^{th})$ of base EPR pairs needed to create one high quality EPR pair under a recurrence based purification protocol is given by [16],

$$g(F, F^{th}) = \prod_{k=1}^{k_{max}} \frac{2}{p_k}, \tag{1}$$

where k_{max} denotes the number of successful purification steps needed to achieve an output fidelity of F^{th} starting from an input fidelity F and p_k denotes the success probability of k^{th} purification step.

Quantum Memories: QMs can store EPR pairs for future usage. To realize long-distance entanglement storage, the photonic qubits need to be stored at intermediate nodes of a quantum network with high efficiencies. In a QM, photonic quantum states are mapped onto individual matter (non-photonic) qubits. Physical implementations of QMs can be grouped into the following five platforms: rare-earth ion-doped solids, diamond color centers, crystalline solids, alkali metal vapours, and molecules. We refer interested readers to [20], [33] for a more elaborate discussion on various proposed implementations of QMs.

B. Basic Components of QON

In this section, we describe the basic components of QON. Table I shows the notations used in this and next sections.

Network: We consider a quantum network represented by a graph $G = (V, E)$, where V is the set of nodes and E is the set of physical communication links. We define $c(u, v)$ as the capacity of edge (u, v), which denotes the average EPR pair generation rate between adjacent nodes u and v. We assume a subset of the nodes ($S \subset V$) in the network have storage capability and pairs of them can store EPR pairs for satisfying future requests. Let J denote the set of storage pairs in the network. The storage nodes can also act as normal nodes meaning that they can create EPR pairs with neighboring nodes and perform entanglement swapping. Each storage server can store a limited number of EPR pairs in its memory for future usage. The capacity of each storage server $s \in S$ is identified by B_s.

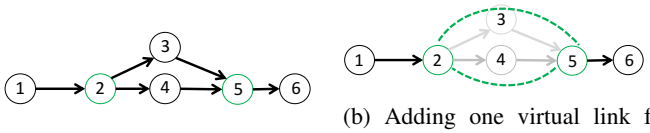
$c(u, v)$	Capacity of link (u, v) in EPRs/sec
K	Set of user pairs
Virtual link	A newly added link to the graph that connects a storage pair directly
P_N^k	Set of all paths for user pair k in G
P_S^k	Set of all paths for user pair k that use at least one virtual link in $\tilde{G}$
F_t^k	Fidelity threshold of user pair k at time interval t
$g(F_p, F^{th})$	Avg. no. of base EPR pairs needed for purification on path p to achieve fidelity threshold F^{th}
S	Set of nodes that have storing capability
J	Set of storage pairs
D_t^k	Rate of demand for request pair k at time interval t
B_s	Capacity of storage server s
F_p	Basic fidelity of path p
$w_{p,t}^k$	Rate of entanglement generation for pair k on path p at time interval t
$u_{p,t}^j$	Avg. no. of EPR pairs at storage j at the beginning of time interval t that are generated using path p
Δ	Duration of one time interval in sec
h	EPR pair lifetime (e.g., no. of time intervals) at storage servers

Table I: Notations used in this paper.

Workload: We assume the time to be discretized and consists of a set of T discrete time intervals. We denote K to be the set of source-destination pairs of users that generate entanglement requests at each time interval. Let D_t^k denote the entanglement generation request rate from source-destination pair k at time interval t. We assume there is a central controller in the network that receives these requests and orchestrates resources at different time intervals. Each request from user pair k has an application-level target fidelity threshold indicated by F_t^k.

Paths and Virtual Links: We assume that each user pair $k = (src, dst)$ has a set of predefined paths between them. Same holds true for storage pairs $j = (j_1, j_2)$. We denote the set of paths that connect two user (storage) pairs through the network as P_N^k (P_N^j). We now add a parallel link between a storage pair for each path that connects them and refer to these new links as *virtual* links[1] (Figure 2b). Let $\tilde{G} = (V, \tilde{E})$ denote the virtual graph created by adding virtual links between storage node pairs. We define a virtual path to be a path that uses at least one virtual link. For each parallel virtual link, one needs to know the corresponding network path that created it. This

[1]The notion of a single virtual link was first introduced in [34].



(a) Original graph

(b) Adding one virtual link for each path between storage servers

Figure 2: Adding parallel virtual links between storage pair $(2, 5)$ to construct virtual paths for user pair $(1, 6)$.

will be helpful in tracking the fidelities associated with the EPR pairs stored at the corresponding storage servers. Hence, we associate an identifier for each virtual path.

For example, in Figure 2, consider the virtual path $p_1 = \{1, 2, 5, 6\}$ that was created using network sub-path $s_1 = \{2, 3, 5\}$. Consider another virtual path $p_2 = \{1, 2, 5, 6\}$ that was created using network sub-path $s_2 = \{2, 4, 5\}$. We treat p_1 and p_2 differently and do not count them as duplicates. Let $f(\cdot)$ be a function that outputs the complete path corresponding to a virtual path. In our example: $f(p_1) = \{1, 2, 3, 5, 6\}$ and $f(p_2) = \{1, 2, 4, 5, 6\}$. Let P_S^k (P_S^j) denote the set of virtual paths between user (storage) pair k (j) in $\tilde{G}$.

Fidelity of stored EPR pairs: We need to keep track of the fidelity of stored EPR pairs at each pair of storage servers. Since each storage pair $j \in J$ may use different paths to generate EPR pairs, the fidelity of the stored EPR pairs at each storage pair can be different. For each storage pair and for each path between them, we define a variable $u_{p,t}^j$ to identify the number of EPR pairs generated over path p that are stored at storage pair j at time t. All EPR pairs generated over path p have the same basic fidelity. We assume that entanglement purification is always performed by the end-users and storage nodes do not perform any purification.

Storage Lifetimes: We assume the qubits of an EPR pair decohere after being stored for h time intervals and, if not used are thrown out. In Section III, we consider formulations for resource allocation in a QON for different values of h. In particular, we are interested in two extremes: (i) $h = 1$ (unit storage lifetimes) (ii) $h \geq |T|$ (infinite storage lifetimes).

III. QON: Models and Objectives

We now introduce the related models and objectives for resource allocation in QONs.

A. Handling demand spikes

One of the goals of this work is to evaluate the robustness of QON in handling sudden demand fluctuations for generating end-user entanglements. Our goal is to serve the demands of all user pairs during each time interval while satisfying their fidelity requirements. We first consider the case that EPR pairs can stay in storage forever and can be used in any time in the future, $h \geq |T|$. We then consider the case that EPR pairs are valid for only one time interval, $h = 1$. The third case where EPR pairs can be stored at storage servers for multiple time intervals, $1 < h < |T|$, is relegated to Appendix VIII-A.

1) $h \geq |T|$ case: We now consider the case where EPR pairs can be stored at storage servers forever. Our decision variables $w_{p,t}^k$ denote the rate of entanglement generation for user pair k using path p during time interval t. Below, we cast the demand satisfaction problem as a constraint feasibility problem and explain its constraints.

$$\max_{w_{p,t}^k} \quad 1$$

subject to $\quad \forall t \in T:$

$$u_{p_s,t}^j = u_{p_s,t-1}^j - \sum_{\substack{k \in \{K \cup \{J-j\}\} \\ p \in P_S^k | p_s \subset f(p)}} w_{p,t-1}^k g(F_p, F_{t-1}^k)\Delta + w_{p_s,t-1}^j \Delta \qquad j \in J, p_s \in P_N^j \cup P_S^j \tag{2}$$

$$\sum_{\substack{k \in \{K \cup \{J-j\}\} \\ p \in P_S^k | p_s \subset f(p)}} w_{p,t}^k g(F_p, F_t^k)\Delta \leq u_{p_s,t}^j \qquad j \in J, p_s \in P_N^j \cup P_S^j \tag{3}$$

$$\sum_{p \in P_N^k \cup P_S^k} w_{p,t}^k = D_t^k \quad k \in K \tag{4}$$

$$\sum_{\substack{k \in \{K \cup J\} \\ p \in P_N^k \cup P_S^k | (u,v) \in p}} w_{p,t}^k g(F_p, F_t^k) \leq c(u,v) \qquad (u,v) \in E \tag{5}$$

$$\sum_{\substack{s_2 \in S \\ j=(s,s_2) \in J \\ p_s \in P_N^j \cup P_S^j}} u_{p_s,t}^j \leq B_s \quad s \in S \tag{6}$$

$$w_{p,t}^k \geq 0 \quad k \in \{K \cup J\}, p \in P_N^k \cup P_S^k \tag{7}$$

$$u_{p_s,t}^j \geq 0 \quad \forall j \in J, \forall p_s \in P_N^j \cup P_S^j \tag{8}$$

Here, (2) captures the evolution of each storage pair j. $u_{p_s,t}^j$ is equal to $u_{p_s,t-1}^j$ minus the average number of EPR pairs served from it to satisfy user demands and end-to-end purification in the previous time interval, plus the average number of EPR pairs stored in it using path p_s in the previous time interval. We repeat this constraint for all storage pairs and for all paths between them across all time intervals. Function f returns a complete path $\tilde{p}_s$ corresponding to a virtual path p as explained in Section II.

Constraint (3) ensures that the average number of purified EPR pairs that are served from one storage pair during time interval t should be less than or equal to what has been available in it at the beginning of the interval. Similar to the previous constraint, we repeat this constraint for all storage pairs and for each path between them that is being used for entanglement generation. For a storage pair j and a given path

p_s between them, the condition on the summation term in this constraint captures all paths (end-user pair paths + other inter-storage pair paths) that have p_s as their sub-path.

Constraint (4) ensures that all the demands for each user pair should be satisfied. Constraint (5) ensures that the average number of EPR pairs served or stored using an edge should not be more than its capacity. Constraint (6) enforces that the storage capacity of each storage node must not be violated. Note that, we have set the value of $w^j_{p,t}$ for $t = 0$ to zero, i.e. no EPR pairs are stored at the storage servers at the beginning of first time interval[2]. In addition, the value of $w^k_{p,t}$ for $t = 0$ and for any user pair k with $p \in P^k_S$ is set to zero. This means that we can not serve any EPR pair from the storage servers at the first time interval since nothing has been stored at the storage servers yet. The function $g(F_p, F^k_t)$ in constraints (2), (3), and (5) returns the average number of base EPR pairs with fidelity F_p that needs to be sacrificed in order to create one high quality EPR pair of fidelity at least F^k_t. The function $g(F_p, F^k_t)$ can be computed using Equation (1).

2) $h = 1$ case: The problem formulation for the case when stored EPR pairs have unit storage lifetimes is similar to the infinite lifetime scenario except for the constraint that tracks the evolution of average number of stored EPR pairs across storage node pairs. The formulation is as follows:

$$\begin{aligned} \max_{w^k_{p,t}} \quad & 1 \\ \text{subject to} \quad & \forall t \in T : \\ & w^j_{p_s,t} = w^j_{p_s,t-1}\Delta - \\ & \sum_{\substack{k\in\{K\cup\{J-j\}\} \\ p\in P^k_S | p_s \subset f(p)}} w^k_{p,t-1} g(F_p, F^k_{t-1})\Delta \\ & \qquad\qquad j \in J \,\&\, p_s \in P^j_N \cup P^j_S \qquad (9) \\ & \text{and constraints} \quad (3), (4), (5), (6), (7), (8) \end{aligned}$$

Note that we do not have the variable $u^j_{p_s,t-1}$ in constraint (9) anymore.

B. Maximizing Weighted Entanglement Generation Rate

Entanglement Generation Rate (EGR) provided by a quantum network is another important performance metric and a subject of great interest in recent quantum network proposals. We now investigate a scenario where a population of end-user pairs get served by the quantum network, possibly at different rates. In a quantum network, some end user pairs may have different priority or reward when they get served. Thus, the user pairs can compete for access to QON resources. Let α^k_t denote the weight associated with user pair k at time interval t. Our goal is to solve the resource allocation problem in a QON that maximizes the aggregate weighted EGR across all user pairs and all time intervals. The optimization formulation for $h \geq |T|$ is presented below.

[2]Even though here we focus on the formulation for finite $|T|$, one can easily modify our formulation for the case when T is periodic by setting $u^j_{p,0} = u^j_{p,|T|}$

$$\begin{aligned} \max_{w^k_{p,t}} \quad & \frac{1}{|T|} \sum_{t\in T} \sum_{k\in K} \sum_{p\in P^k_N \cup P^k_S} \alpha^k_t * w^k_{p,t} \qquad (10) \\ \text{subject to} \quad & \forall t \in T : \\ & \text{constraints} \quad (2), (3), (5), (6), (7), (8) \end{aligned}$$

Note that, the constraints for this formulation are similar to the formulation mentioned in Section III-A except that we do not have a notion of end-user demands. Instead we are interested in maximizing aggregate weighted EGR. The formulation for $h = 1$ scenario can be described similar to Section III-A2.

C. Minimizing request service delay

Similar to the classical overlays, QONs can also be used to minimize the aggregate request service delay while satisfying user request requirements. The key idea is that the EPR pairs at the storage nodes are readily available for consumption. Thus, when requests are served using EPR pairs from storage nodes, one needs to perform lesser number of entanglement swap operations compared to the case when they are served using the network which in turn decreases the end-to-end request service delay. We present the formulation as follows.

$$\begin{aligned} \min_{w^k_{p,t}} \quad & \sum_{t\in T} \sum_{k\in K} \sum_{p\in P^k_{N\cup S}} w^k_{p,t}(|p| - 1) \qquad (11) \\ \text{subject to} \quad & \forall t \in T : \\ & \text{Constraints} \quad (2), (3), (4), (5), (6), (7), (8) \end{aligned}$$

Here, $|p|$ is the length of path p. When $p \in P^k_S$, the path length greatly reduces due to the notion of virtual links.

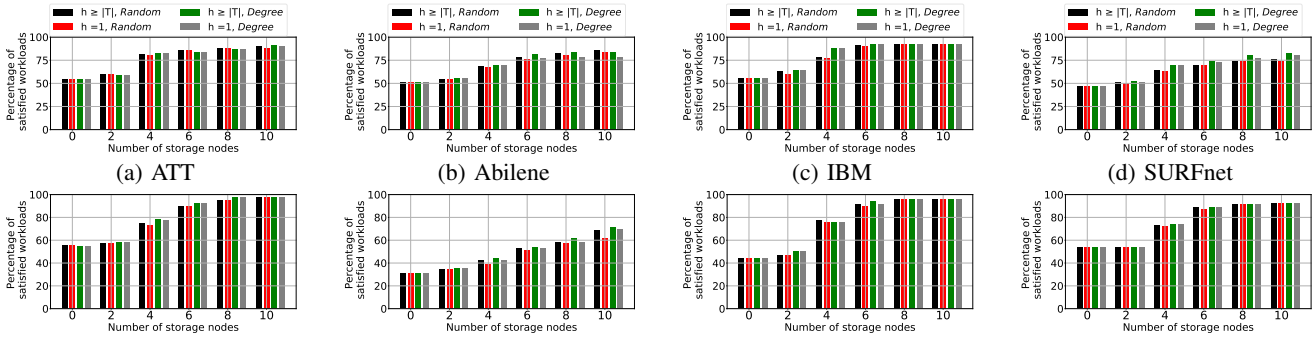
D. Complexity Analysis

The computational complexity of our problem formulation can be analyzed as follows. First of all, the decision variables $w^k_{p,t}$, are continuous. The constraints and objectives mentioned in all previous formulations are linear. Thus all of our optimization formulations are linear programs. Assume $|K|$, $|P|$, and $|E|$ represents the number of user pairs, the number of paths used for each user pair, and number of edges in the network respectively. Our optimization problem has at most $|K| * |P| * |T|$ number of variables and the number of constraints are $\mathcal{O}(|T|*|J|*|P|*|K|)^3 + \mathcal{O}(|T|*(|K|+|E|)*|P|))$.

IV. Performance Evaluation

In this section, we experimentally evaluate QONs to answer the following questions: (1) Do QONs help to handle demand spikes from users and help to serve more EPR pairs in the network? (2) Do QONs significantly drive down entanglement request service delays by storing and moving EPR pairs closer to users? (3) How do different strategies for storage node selection in the network affect the answers to two previous questions?

We conduct the experiments using the IBM CPLEX solver [21].



(a) ATT (b) Abilene (c) IBM (d) SURFnet

(e) $G(50, 0.1)$ (f) $G(50, 0.05)$ (g) $PA(50, 2)$ (h) $PA(50, 3)$

Figure 3: Percentage of satisfied workloads in real (3a,3b, and 3h) and random (3e,3f, and 3g) topologies with different number of storage nodes. B_s for all servers is $12,000$, Δ is 20 seconds, and the fidelity threshold for requests is 0.8.

Param	Δ	$\|T\|$	$c(u,v)$	Link Fidelity	B_s
Value	20s	10	Unif$[200, 1400]$	Unif$[0.96, 0.99]$	12000

Table II: Parameters used in our experiments.

Topologies	$\|V\|$	$\|E\|$	Avg. Degree	Diameter
ATT	25	112	4.4	5
Abilene	12	30	2.5	5
IBM	17	46	2.7	6
SURFnet	50	68	2.7	11
$G(50, 0.0.5)$	50	67	2.8	9.3
$G(50, 0.1)$	50	123	4.9	5.3
$PA(50, 2)$	50	96	3.8	4.9
$PA(50, 3)$	50	141	5.6	4

Table III: Properties of network topologies used in Simulations.

Network topologies: In our evaluation, we use the topologies of four real networks including the Dutch SURFnet network, taken from the Internet topology zoo [24], ATT and IBM network topologies taken from [36], Abilene [2], and different randomly generated topologies based on preferential attachment model (power law graph) and *Erdos Renyi* graph [5]. We use $G(n, p)$ to refer to the *Erdos Renyi* graphs where n is the number of nodes and p is the probability for edge creation. We use $PA(n, m)$ to refer to the preferential attachment model where n is the number of nodes and m is the number of edges to attach from a new node to existing nodes. We use python *networkx* library [18] to generate random topologies.

Table II shows the value of different parameters used in our experiments. Table III specifies the characteristics of the topologies. We average the degree and the diameter over 5 randomly generated topologies for $G(n, p)$ and $PA(n, m)$ where $p = 0.1, 0.05$ and $m = 2, 3$. In all topologies, unless mentioned, we consider at most one shortest path (based on hop count) between each pair of users, each pair of storage servers, and each user and each storage server.

Purification and Swapping: We assume all noisy mixed entangled states in the network are Werner states [42]. In our experiments we use the recurrence based purification scheme as explained in Section II. In particular, we use the DEJMPS protocol [13] for purification. The values of p_k and k_{max} in (1) can be determined from the results presented in [13]. When a node performs an (noise-free) entanglement swap operation between two EPR-pairs with fidelities F_1 and F_2, the fidelity of the resulting state is $\frac{1}{4} + \frac{3}{4} * (\frac{4F_1-1}{3})(\frac{4F_2-1}{3})$ [7]. We compute the basic fidelity of a path in the same way.

Storage selection schemes: We select the storage servers in the network based on two schemes. In *Random* scheme, we select storage nodes randomly. In *Degree* scheme, nodes with the higher degrees are selected first as storage servers.

Workload: We consider 6 user pairs for each network topology and generate the demands to create entanglements between the user pairs for each time interval using the spike model of *tgem* library [32]. The spike model takes the number of user pairs, number of time intervals, number of user pairs that would have a spike in their demands, and a mean value for the spike as inputs and generates the demands between each pair of users for each time interval. In our experiments, at each time interval, three user pairs can have a spike in their demands. A workload includes demands from all user pairs and their fidelity requirements over T time intervals. Since each topology has a different capacity for serving entanglements, we normalize the generated workloads of each topology based on the capacity of that topology. We first compute the capacity of each topology as the maximum EPR rate that it can generate with a fidelity threshold 0.75 to different sets of 6 user pairs. If the capacity of topology, with this definition, is c, we set the mean value of each spike to $c/3$ as we have three spikes in each time interval in our workload generation module.

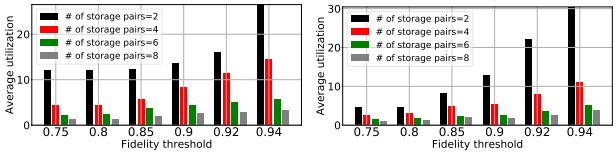
A. Handling demand spikes

We first evaluate the robustness of QONs in handling unexpected spikes in user demands. We say a workload is satisfied if the network can serve the demands of all users in all time intervals while fulfilling their fidelity requirements. For each network, we measure the percentage of 200 generated

workloads that the network can satisfy and plot them as a function of number of storage nodes in the network.

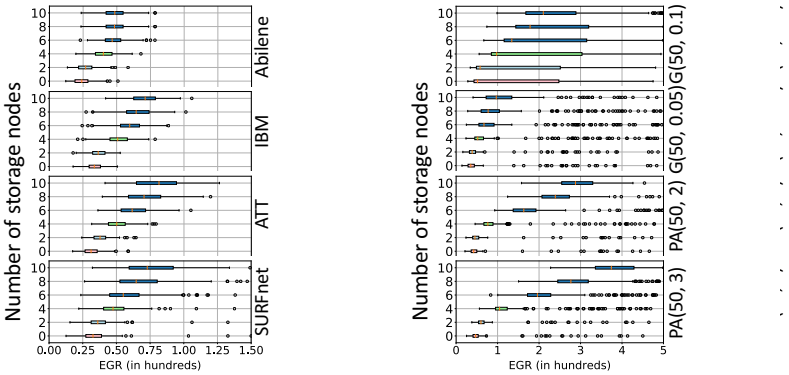
Figure 3 shows the percentage of satisfied workloads among all workloads for real and randomly generated topologies. In each sub-figure, the first keyword in the legend indicates the value of h that represents the EPR pair storage lifetime and the second keyword represents the scheme for storage node selection. The threshold for fidelity requirement of all requests at all time intervals in this experiment is set to 0.8 and the values of Δ and B_s (for each storage server s) are 20 seconds and 12,000 EPR pairs respectively.

A couple observations are noteworthy. As expect number of satisfied workloads increases with the nu storage nodes in the network and the highest value obs around 92% with 10 storage nodes in most topologie is almost no difference between infinite time-intervals pair lifetime ($h \geq |T|$) and one-time-interval lifetime ($h = 1$) except for the case when the number of storag is low in the network. In addition, the percentage of workloads varies across different topologies. Topologi a higher number of edges can handle more demand s the paths would use disjoint links with a higher pro in these networks. Two storage selection schemes different results for different topologies. However, cases, choosing storage nodes using *Degree* scheme (red and black bars) outperforms the random scheme.



(a) ATT (b) $PA(50, 2)$

Figure 4: Storage server utilization under different fidelity requirements for requests for *Degree* storage selection scheme, $B_s = 12,000$ and $h \geq |T|$ case.



(b) Random topologies

Figure 5: Maximizing EGR in different topologies. Storage selection scheme is *Degree* and $h \geq |T|$. The fidelity threshold is 0.8.

B. Storage utilization

We now evaluate the utilization of storage nodes in the network under different target fidelity requirements for end-user requests. For each storage server s, if the highest number of EPR pairs that is stored at it among all time intervals of a workload is v and the capacity of the storage is B_s, we define the utilization of that storage to be $\frac{v}{B_s} * 100$. For each workload, we measure the utilization for all storage servers in the network and compute the average. For each generated workload, we change the fidelity requirement of the requests for 6 user pairs in the network and measure the storage server utilization. We use the previous experiment setup for this experiment as well. We conduct this experiment with ATT and $PA(50, 2)$ routing topologies.

Figure 4 shows the average storage utilization across different number of storage servers in the network. The storage servers are chosen using *Degree* scheme. As might be expected utilization decreases as the number of servers increases as the load gets spread over more servers. By increasing the fidelity requirement of requests, the utilization is increased as we need more EPR pairs for the purification scheme ans more EPR pairs are stored at storages for this purpose.

C. Maximizing Weighted Entanglement Generation Rate

In this experiment, we evaluate the maximum rate that a QON can serve to a set of users. We assume the set of users for all time intervals is fixed but the weight for each user is changing. We set the weight of each user pair at each time interval from the range $[0, 1]$. We assume the fidelity threshold for all delivered EPR pairs is equal and is 0.8. We conduct this experiment with the *Degree* scheme for storage node selection in the network.

Figure 5 shows a box plot of EGR as a function of the number of storage nodes in the network with 6 user pairs over 400 generated workloads of 10 consecutive time intervals for different topologies. Unsurprisingly, EGR is an increasing function of the number of storage nodes. However, the results for EGR in each topology depends on the average degree and number of edges in the network that can effect the paths used by each user pair. With more edges in the network, it is more likely that different paths would use different links in comparison to the case that the network has a smaller degree and number of edges. When multiple user pairs use a set of paths that share a link, all user pairs are limited to the capacity of that link. In our evaluation, we found that in 50 percentage of workloads, in real topologies, an edge is being used by an average of 2.1 user pairs. However, this value is 1.2 for random topologies.

In Figure 6, we check the effect of increasing the number of paths between each user pair and storage servers on the EGR of the network. The y-axis is the average EGR over 200 workloads with each workload including 6 user pairs each having a different weight value for 10 time intervals. We have plotted the results for only the $h \geq |T|$ case. In this experiment, we select four storage nodes using *Degree* scheme and the fidelity threshold for all served EPR pairs

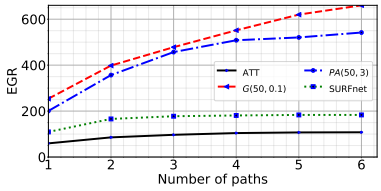


Figure 6: EGR as a function of number of paths.

is 0.8. Topologies with larger degrees and number of nodes benefit from increasing the number of paths. As explained in the previous paragraph, having more edges also provide more disjoint links between selected paths.

D. EGR as a function of fidelity threshold

In another experiment, we measure EGR in different topologies as a function of the fidelity of the delivered EPR pairs. By increasing the fidelity threshold for delivered EPR pairs, we expect more resources will be consumed for purification and that the EGR will decrease. We assume there are four storage servers selected using *Random* and *Degree* schemes in each network and there are 6 user pairs in the network. The capacity of each storage server is $12,000$ EPR pairs and the edge capacity and fidelity is chosen as the first experiment (exp IV-A). In this experiment, we assume $h \geq |T|$.

Figure 7 shows EGR as a function of fidelity threshold for different topologies. The higher the fidelity threshold, the more resources used for purification and hence lower the EGR. We find that using *Degree* scheme for storage node selection outperforms the *Random* scheme. One reason is that when nodes with higher degrees are selected as storage servers, more users can connect to them via disjoint paths and when not disjoint, there are fewer shared links. We omit the results for the case $h = 1$ as there is little difference from the results for the case $h \geq |T|$. The reason is that the infinite lifetime extreme would outperform the one-time interval extreme only when the weight of users in the next two or more time intervals is larger than the weight of users in the current time interval. Since we set the weight of users at each time interval randomly, this is unlikely to happen.

E. Reducing request service delay

In this section, we evaluate how service delays are reduced using a QON. We measure the delay of an end-to-end request in the network as the number of entanglement swaps required to establish the end-to-end entanglement.

Figure 8 shows a box plot of the request service delays (number of entanglement swaps) in our real and random topologies as a function of number of storage nodes in the network. As expected, serving requests from storage servers reduces delays. SURFnet has the largest service delays (subfigure 8d). This topology has the largest network diameter among all topologies (Table III). With 6 or more storage nodes in the network in most topologies, we only need 2 entanglement swap to serve the request. This can happen when each user in the user pair only needs to be connected to one storage server and 2 entanglement swaps are required to establish the end-to-end connection.

V. Related Work

In this Section, we discuss the state-of-the-art related to classical overlays and entanglement distribution in quantum networks.

A. Classical Overlay Networks

Broadly, almost all overlay networks designed for classical internet-based services fall into the following three categories: caching overlay [14], routing overlay [3], [4] and security overlay [22]. We refer the interested reader to [1] for a comprehensive discussion on different classical overlay network architectures. Inspired by classical overlays, in this work, we propose an overlay network architecture to efficiently distribute quantum entanglements between end users by utilizing an underlying quantum network. Our proposed architecture resembles more to a classical routing overlay in spirit where multiple overlay paths are constructed between same end users for routing communication.

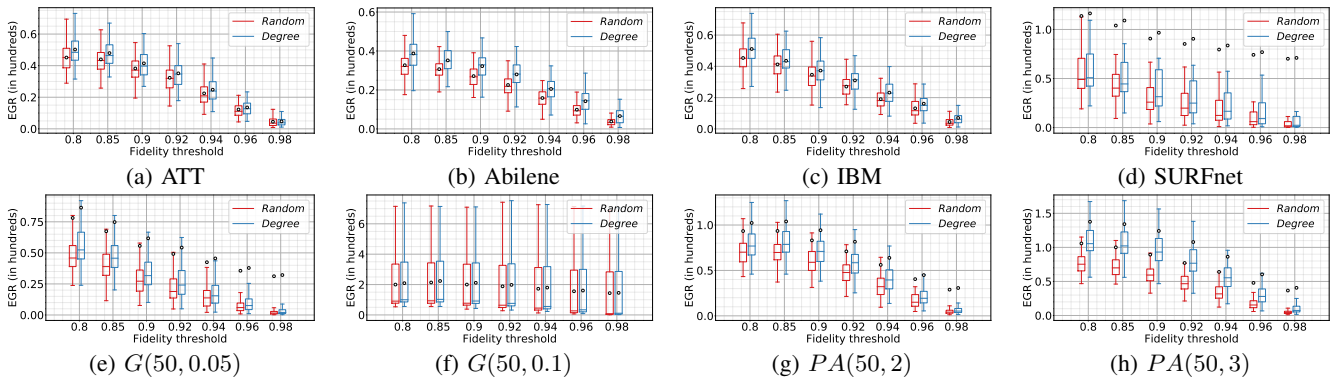
B. Entanglement Distribution in Quantum Networks

A large and growing body of literature in quantum routing have investigated the problem of efficiently distributing long distance entanglements between end users using a network of quantum nodes. For example, Van Meter et al. [38] considered entanglement distribution in a linear quantum network. Pant et al. [30] presented a multi-path routing protocol to distribute entanglement in a grid network. Van Meter et al. [39] explored several link cost metrics and used Dijkstra's algorithm to compute the optimal routing path in a generic quantum network. Shi et al. [35] developed a nonlinear path cost based routing algorithm for a generic quantum network. Authors in [40], [44] applied entanglement purification and developed fidelity aware routing protocols for general quantum networks.

In a separate line of work [34], [9], authors considered the notion of virtual links in a quantum network by assuming preshared entanglement among non-adjacent nodes and developed routing protocols. We outline the main differences between those approaches and our work. [34] focused on specific network topologies such as ring and sphere networks while we derive results for a general quantum network. In [34] and [9], authors assumed that the virtual links were created in the background and were not concerned about its creation process. In our model we use the underlay quantum network to create multiple entanglements that can be stored in overlay storage nodes. Also, we consider the effect of end-to-end entanglement purification which is ignored in these work.

VI. Conclusion and Future Work

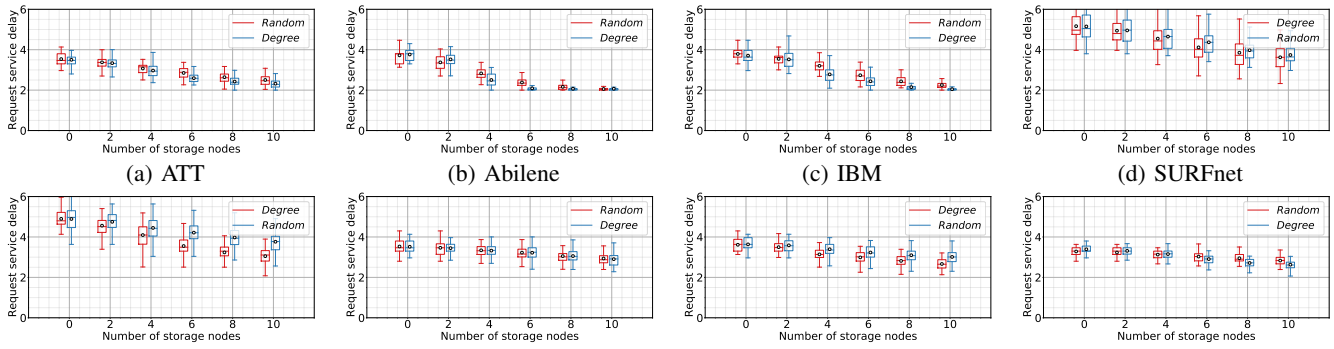
In this paper, we presented the design, formal analysis, and evaluation of QONs. Although the concept of overlay networks is a widely embraced direction in classical networks, to our knowledge, this is the first work to present a full-fledged design and problem formulation of QONs. Our underlying



(a) ATT (b) Abilene (c) IBM (d) SURFnet

(e) $G(50, 0.05)$ (f) $G(50, 0.1)$ (g) $PA(50, 2)$ (h) $PA(50, 3)$

Figure 7: EGR in real (top) and random (below) topologies as a function of fidelity threshold using 4 storages nodes selected using *Random* and *Degree* approaches. For this figure, we have $h \geq |T|$ and $\Delta = 20$ seconds and $B_s = 12000$.



(a) ATT (b) Abilene (c) IBM (d) SURFnet

(e) $G(50, 0.05)$ (f) $G(50, 0.1)$ (g) $PA(50, 2)$ (h) $PA(50, 3)$

Figure 8: Minimizing entanglement service delay by reducing entanglement swapping using storage servers in real and random topologies. Fidelity threshold is 0.8, $h \geq |T|$,$B_s = 12000$.

contributions include the problem formulation for resource allocation in QONs with different performance objectives and demonstrating the potential for significantly increasing the entanglement generation rate and handling end-user entanglement demand spikes. Below we discuss some interesting open problems and challenges associated with QONs.

- **Joint storage node placement and performance optimization:** In this work, we assumed the locations of storage nodes were given as an input to the optimization problem. However, one can solve the joint problem of placing the storage nodes and maximizing a certain performance objective.
- **Overlay Network Topology:** We assumed the overlay storage network topology to be a complete graph. It would be interesting to explore the effect of other topologies on overlay performance.
- **Link-level Purification:** The focus of this work has been on an architecture where purification is performed only at the end users. One can also consider an alternate architecture, where both link-level and end-to-end purification are performed at the intermediate nodes and end users respectively.

VII. Acknowledgements

This research was supported by the NSF Engineering Research Center for Quantum Networks (CQN), awarded under cooperative agreement number 1941583 and by NSF grant CNS-1955834. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

VIII. Appendix

A. $1 \leq h \leq |T|$ *case*

In this section, we explain the case that EPR pairs can be stored at storage servers for multi-time-intervals (for the case $1 \leq h \leq |T|$). We add one more dimension h for the age of delivered or consumed EPR pairs for purification to variables $w^k_{p,t}$ and $u^j_{p,t}$. In our formulation, we use function π to indicate different scheme in selecting EPR pairs from the storage. Possible candidate polices are as (1) *oldest-first scheme:* the oldest EPR pairs that have been stored at storage server would be used first, (2) *newest-first scheme:* use the freshest EPR pairs first, and (3) *randomly selected scheme:* EPR pairs would randomly be selected. The problem formulation would be as follows. Note that we will sum constraints (3), (4), (5), (6), (7), (8) on dimension h as well. We have skipped this for brevity.

$$\max_{w^k_{p,t}} \quad 1$$

subject to $\quad \forall t \in T :$

$$u^{j,h}_{p_s,t} = \begin{cases} 0, & \text{if} \quad h > t \\ u^{j,h-1}_{p_s,t-1} - \\ \quad \pi(\sum_{k\in\{K_t\cup\{J-j\}\}\&p\in P^k_S | p_s \in f(p)} \\ \quad w^{k,h-1}_{p,t-1} g(F_p, F^k_{t-1}))\Delta_{t-1}, \\ \quad \text{if } h \neq t, \\ w^{j,t-1}_{p_s,t-1}, \text{ if } h == t \end{cases}$$

and constraints $\quad (3), (4), (5), (6), (7), (8)$

REFERENCES

[1] Overlay Networks: An Akamai Perspective. *Advanced Content Delivery, Streaming, and Cloud Services*, pages 305–328, 2014.

[2] Abilene. Yin Zhang's Abilene.. [Online]. http://www.cs.utexas.edu/yzhang/research/AbileneTM/, 2020. [Online; accessed 2-Nov-2021].

[3] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. *SIGOPS Oper. Syst. Rev.*, 35(5):131–145, oct 2001.

[4] Konstantin Andreev, Bruce M. Maggs, Adam Meyerson, and Ramesh K. Sitaraman. Designing overlay multicast networks for streaming. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, page 149–158, New York, NY, USA, 2003. Association for Computing Machinery.

[5] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[6] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *arXiv preprint arXiv:2003.06557*, 2020.

[7] H-J Briegel, Wolfgang Dür, Juan I Cirac, and Peter Zoller. Quantum repeaters: the role of imperfect local operations in quantum communication. *Physical Review Letters*, 81(26):5932, 1998.

[8] Kaushik Chakraborty, David Elkouss, Bruno Rijsman, and Stephanie Wehner. Entanglement distribution in a quantum network: A multicommodity flow-based approach. *IEEE Transactions on Quantum Engineering*, 1:1–21, 2020.

[9] Kaushik Chakraborty, Filip Rozpedek, Axel Dahlberg, and Stephanie Wehner. Distributed Routing in a Quantum Internet. 2019.

[10] J Ignacio Cirac, AK Ekert, Susana F Huelga, and Chiara Macchiavello. Distributed quantum computation over noisy channels. *Physical Review A*, 59(6):4249, 1999.

[11] Axel Dahlberg, Matthew Skrzypczyk, Tim Coopmans, Leon Wubben, Filip Rozpędek, Matteo Pompili, Arian Stolk, Przemysław Pawełczak, Robert Knegjens, Julio de Oliveira Filho, et al. A link layer protocol for quantum networks. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 159–173. 2019.

[12] G Mauro D'Ariano, Paoloplacido Lo Presti, and Matteo GA Paris. Using entanglement improves the precision of quantum measurements. *Physical review letters*, 87(27):270404, 2001.

[13] David Deutsch, Artur Ekert, Richard Jozsa, Chiara Macchiavello, Sandu Popescu, and Anna Sanpera. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Phys. Rev. Lett.*, 77:2818–2821, Sep 1996.

[14] John Dilley, Bruce M Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. Globally distributed content delivery. 10 1998.

[15] YO Dudin, L Li, and A Kuzmich. Light storage on the time scale of a minute. *Physical Review A*, 87(3):031801, 2013.

[16] W. Dür, H. J. Briegel, J. I. Cirac, and P. Zoller. Quantum repeaters based on entanglement purification. *Physical Review A - Atomic, Molecular, and Optical Physics*, 59(1):169–181, 1999.

[17] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum-enhanced measurements: beating the standard quantum limit. *Science*, 306(5700):1330–1336, 2004.

[18] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[19] Georg Heinze, Christian Hubrich, and Thomas Halfmann. Stopped light and image storage by electromagnetically induced transparency up to the regime of one minute. *Physical review letters*, 111(3):033601, 2013.

[20] Khabat Heshami, Duncan G England, Peter C Humphreys, Philip J Bustard, Victor M Acosta, Joshua Nunn, and Benjamin J Sussman. Quantum memories: emerging applications and recent advances. *Journal of modern optics*, 63(20):2005–2028, 2016.

[21] IBM. IBM [Online]. https://www.ibm.com/academic, 2022. [Online; accessed 2-Jun-2022].

[22] Angelos D. Keromytis, Vishal Misra, and Dan Rubenstein. Sos: Secure overlay services. SIGCOMM '02, page 61–72, New York, NY, USA, 2002. Association for Computing Machinery.

[23] H Jeff Kimble. The quantum internet. *Nature*, 453(7198):1023–1030, 2008.

[24] Simon Knight, Hung X Nguyen, Nickolas Falkner, Rhys Bowden, and Matthew Roughan. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, 29(9):1765–1775, 2011.

[25] Peter Komar, Eric M Kessler, Michael Bishof, Liang Jiang, Anders S Sørensen, Jun Ye, and Mikhail D Lukin. A quantum network of clocks. *Nature Physics*, 10(8):582–587, 2014.

[26] Seth Lloyd, Jeffrey H Shapiro, Franco NC Wong, Prem Kumar, Selim M Shahriar, and Horace P Yuen. Infrastructure for the quantum internet. *ACM SIGCOMM Computer Communication Review*, 34(5):9–20, 2004.

[27] Jevon J Longdell, Elliot Fraval, Matthew J Sellars, and Neil B Manson. Stopped light with storage times greater than one second using electromagnetically induced transparency in a solid. *Physical review letters*, 95(6):063601, 2005.

[28] Yu Ma, You-Zhi Ma, Zong-Quan Zhou, Chuan-Feng Li, and Guang-Can Guo. One-hour coherent optical storage in an atomic frequency comb memory. *Nature communications*, 12(1):1–6, 2021.

[29] Klaus Mattle, Harald Weinfurter, Paul G Kwiat, and Anton Zeilinger. Dense coding in experimental quantum communication. *Physical Review Letters*, 76(25):4656, 1996.

[30] Mihir Pant, Hari Krovi, Don Towsley, Leandros Tassiulas, Liang Jiang, Prithwish Basu, Dirk Englund, and Saikat Guha. Routing entanglement in the quantum internet. *npj Quantum Information*, 5(1):1–9, 2019.

[31] Momtchil Peev, Christoph Pacher, Romain Alléaume, Claudio Barreiro, Jan Bouda, W Boxleitner, Thierry Debuisschert, Eleni Diamanti, Mehrdad Dianati, JF Dynes, et al. The secoqc quantum key distribution network in vienna. *New Journal of Physics*, 11(7):075001, 2009.

[32] Python. TMgen API[Online]. https://tmgen.readthedocs.io/en/latest/api.html#tmgen.models.spike_tm, 2022. [Online; accessed 2-Jun-2022].

[33] Nicolas Sangouard, Christoph Simon, Hugues De Riedmatten, and Nicolas Gisin. Quantum repeaters based on atomic ensembles and linear optics. *Reviews of Modern Physics*, 83(1):33, 2011.

[34] Eddie Schoute, Laura Mancinska, Tanvirul Islam, Iordanis Kerenidis, and Stephanie Wehner. Shortcuts to quantum network routing. pages 1–45, 2016.

[35] Shouqian Shi and Chen Qian. Concurrent entanglement routing for quantum networks: Model and designs. SIGCOMM '20, page 62–75, New York, NY, USA, 2020. Association for Computing Machinery.

[36] Teavar source code. Traffic Engineering Applying Value at Risk tool source code [Online]. https://github.com/manyaghobadi/teavar, 2020. [Online; accessed 2-Nov-2020].

[37] Damien Stucki, Matthieu Legre, Francois Buntschu, B Clausen, Nadine Felber, Nicolas Gisin, Luca Henzen, Pascal Junod, Gérald Litzistorf, Patrick Monbaron, et al. Long-term performance of the swissquantum quantum key distribution network in a field environment. *New Journal of Physics*, 13(12):123001, 2011.

[38] Rodney Van Meter, Thaddeus D. Ladd, W. J. Munro, and Kae Nemoto. System design for a long-line quantum repeater. *IEEE/ACM Trans. Netw.*, 17(3):1002–1013, jun 2009.

[39] Rodney Van Meter, Takahiko Satoh, Thaddeus D Ladd, William J Munro, and Kae Nemoto. Path selection for quantum repeater networks. *Networking Science*, 3(1):82–95, 2013.

[40] Michelle Victora, Stefan Krastanov, Alexander Sanchez de la Cerda, Steven Willis, and Prineha Narang. Purification and entanglement routing on quantum networks. *arXiv preprint arXiv:2011.11644*, 2020.

[41] Shuang Wang, Wei Chen, Zhen-Qiang Yin, Hong-Wei Li, De-Yong He, Yu-Hu Li, Zheng Zhou, Xiao-Tian Song, Fang-Yi Li, Dong Wang, et al. Field and long-term demonstration of a wide area quantum key distribution network. *Optics express*, 22(18):21739–21756, 2014.

[42] Reinhard F Werner. Quantum states with einstein-podolsky-rosen correlations admitting a hidden-variable model. *Physical Review A*, 40(8):4277, 1989.

[43] Yangming Zhao and Chunming Qiao. Redundant entanglement provisioning and selection for throughput maximization in quantum networks. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021.

[44] Yangming Zhao, Gongming Zhao, and Chunming Qiao. E2e fidelity aware routing and purification for throughput maximization in quantum networks. In *Proceedings of the IEEE INFOCOM*, 2022.