ESDI: Entanglement Scheduling and Distribution in the Quantum Internet

Huayue Gu, Ruozhou Yu, Zhouyu Li, Xiaojian Wang, Fangtong Zhou

Abstract—Quantum entanglement distribution between remote nodes is key to many promising quantum applications. Existing mechanisms have mainly focused on improving throughput and fidelity via entanglement routing or single-node scheduling. This paper considers entanglement scheduling and distribution among many source-destination pairs with different requests over an entire quantum network topology. Two practical scenarios are considered. When requests do not have deadlines, we seek to minimize the average completion time of the communication requests. If deadlines are specified, we seek to maximize the number of requests whose deadlines are met. Inspired by optimal scheduling disciplines in conventional single-queue scenarios, we design a general optimization framework for entanglement scheduling and distribution called ESDI, and develop a probabilistic protocol to implement the optimized solutions in a general buffered quantum network. We develop a discrete-time quantum network simulator for evaluation. Results show the superior performance of ESDI compared to existing solutions.

Keywords—*Quantum network, entanglement scheduling, entanglement routing*

I. INTRODUCTION

A quantum network enables efficient and secure quantum communication based on quantum entanglement [17]. Long-distance quantum communication is key to various novel quantum applications including quantum key distribution (QKD) [4], distributed quantum computing (DQC) [5], [6] and quantum cryptography [27]. Recently, practical quantum networks have been built around the world [7], [34], such as the DARPA quantum network [12], SECOQC Vienna QKD network [24], and the Tokyo QKD network [28].

Entanglement is the most crucial resource in a quantum network. In quantum information processing and communication, information is represented as *quantum bits* (called *qubits*), which cannot be transmitted via classical communication channels without information loss. To transmit quantum information between two arbitrary quantum-capable devices, entanglement must be leveraged for the execution of quantum protocols. The primary function of a quantum network is to distribute entanglements between nodes over long distances.

Because of its importance, entanglement distribution has received significant attention recently. Prior work has focused on entanglement routing, *i.e.*, finding paths to establish end-to-end entanglements with high throughput and/or quality [36]–[38]. In this line of research, the most common goal is to maximize network-wide throughput of entanglement distribution for all source-destination (SD) pairs, potentially with constraints on the quality of the distributed entanglements. In reality, however, each SD pair may utilize the quantum network to support specific applications, and different applications have different requirements for entanglement distribution rate and time constraints. Considering two typical quantum applications, QKD and DQC. QKD relies on a *long-term stream* of entanglements between two parties requesting

secure communications, and is not sensitive to instantaneous entanglement distribution rate, as long as sufficient entanglements are accumulated over a relatively long period of time. Meanwhile, DQC has a stringent requirement for completing communication tasks as quickly as possible, to avoid information decoherence in quantum memories.

The above motivates scheduling entanglements among SD pairs while considering different communication requirements and demands. Existing work has formulated entanglement scheduling in simple queueing scenarios, focusing on scheduling at a single quantum switch [22]. The assumption that all nodes should connect to a central quantum switch is rather strong and unrealistic. Given that real-world quantum links, such as optical fiber, can only distribute entanglements over no more than a few hundred kilometers [21], a general multi-hop network topology is realistic in building large-scale quantum networks [36]. Extending existing scheduling algorithms to this multi-hop scenario is very challenging, as scheduling in a queueing network is generally NP-hard even without any quantum-specific characteristic being considered [30].

In this paper, we study scheduling in a general quantum network, while considering SD pairs with different requests (called *commodities*) for quantum communication demands and/or deadlines. Two scenarios are specifically considered: 1) each commodity has a demand (number of entanglements) but not a deadline, in which case we seek to minimize the average completion time for all commodities' demands; 2) each commodity has a demand and a deadline, in which case we seek to maximize the number of commodities whose deadlines are met. In addition to hardness of the scheduling problem, challenges come from the probabilistic nature of quantum operations in entanglement distribution, making it impossible to obtain an exact estimation of the completion time of each commodity. Our main contributions are as follows:

- We study and formally define the entanglement scheduling and distribution problem in a general quantum network with heterogeneous quantum applications.
- We propose **ESDI**, a general framework for *entanglement scheduling and distribution*, and propose two entanglement scheduling and distribution algorithms: ESDI-O for commodities having demands but no deadlines, and ESDI-E for commodities having demands and deadlines.
- We develop practical probabilistic protocols for entanglement distribution in a buffered quantum network.
- Extensive simulation results show the superior performance of our solutions compared to state-of-the-arts.

Organization: §II introduces the background. §III presents our quantum network model. §IV proposes a general framework for multi-commodity entanglement scheduling and distribution. §V and §VI propose algorithms for scheduling commodities without and with deadlines, respectively. §VII presents our protocol design. §VIII and §IX present simulation results and conclusion, respectively.

Gu, Yu, Li, Wang and Zhou ({hgu5, ryu5, zli85, xwang244, fzhou}@ncsu.edu) are all with NC State University, Raleigh, NC 27606, USA.

II. Background and Related Work

Quantum communication transfers quantum states from one place to another. A quantum network enables long-distance quantum communication between arbitrary end points [32].

Early work mainly focused on idealized network topologies including square-grid [23], ring and sphere topologies [29]. Unfortunately, these ideal network designs are far from realistic due to physical and geographical limitations. Considering a general network topology, recent works have paid much attention to entanglement routing for multiple SD pairs [8]. Shi *et al.* [31] proposed two algorithms, Q-PATH and Q-CAST, for entanglement routing to maximize the throughput (entanglement distribution rate). Zhao *et al.* [37] designed a redundant entanglement provisioning and selection (REPS) algorithm to maximize throughput for multiple SD pairs in a circuit-switched, multi-hop quantum network. Zeng *et al.* [36] proposed an integer programming-based solution with branch-and-price to maximize throughput. Farahbakhsh *et al.* [13] presented opportunistic routing with theoretical analysis.

None of the above studies have considered the scheduling of entanglement distribution in a general quantum network. Panigrahy *et al.* [22] developed a max-weight scheduling policy to stabilize queues in a star-shaped quantum network. Pouryousef *et al.* [26] studied the link entanglement and storage resource allocation problem in a quantum overlay network. These works focus on scheduling in a specialized, single-repeater quantum network. Li *et al.* [18] proposed an effective routing scheme for multiple requests from SD pairs, but they only focus on the fair sharing of entanglements without considering the characteristics of requests.

Meanwhile, most research in entanglement routing considers a *bufferless quantum network*, where entanglement generation and swapping must be completed in one time slot. Otherwise, the intermediate entanglements will be discarded. With recent advances in quantum memories with long coherence times, quantum networks with buffers have been shown to improve quantum network performance and throughput [8], [14]. Dai *et al.* [8] first studied the optimal remote entanglement distribution (ORED) protocol in a buffered quantum network for a single SD pair and provided an upper throughput bound for any entanglement routing protocols. Dai *et al.* [10] also analyzed the queuing delay for a single channel in a quantum network. However, the ORED protocol only considers one SD pair in a quantum network and may not be practical for large-scale operations. This paper considers multiple SD pairs with heterogeneous quantum communication requests, and designs a comprehensive framework to satisfy their demands.

III. Quantum Network Model

In this section, we present preliminaries and modeling of a quantum network. All notations are summarized in Table I.

A. Quantum Information Basics

We consider the prevalent discrete-variable binary state quantum systems. A qubit differs from a classical binary bit represented by either 0 or 1, and can be in a superposition of two basis states. Let $|0\rangle$ and $|1\rangle$ be the two single-qubit basis states. Here $|\cdot\rangle$ is called a *ket* in Dirac notation denoting a column vector representing the (pure) state of a qubit. A qubit is written as $|b\rangle = \alpha|0\rangle + \beta|1\rangle$, with complex numbers α, β as *amplitudes* of the two basis states, satisfying $|\alpha|^2 + |\beta|^2 = 1$. Measuring this qubit yields a classical bit of either 0 with a probability of $|\alpha|^2$ or 1 with probability $|\beta|^2$.

TABLE I: Notation Table

Parameters	Description
$G = (V, E)$	quantum network with nodes V and links E
U	the set of SD pairs
c_e, p_e, q_v	capacity, ebit generation and swapping success probabilities
s_i, t_i	the source and destination of the ith SD pair
Z_i	the set of commodities in the ith SD pair
z_j^i	the jth commodity in the ith SD pair
d_j^i, a_j^i, δ_j^i	the demand, arrival time and deadline of the commodity z_j^i
P_s, P_c	the priority list for SD pairs and commodities
P_c^i	the set of commodities belong to SD pair i
$P_c^i[l]$	the l commodities in list P_c^i
κ	the scheduling length
Θ_j^i, Δ_j^i	the remaining demand and time slots of z_j^i
$\mathcal{S}_T^\tau$	the network state after Phase-τ at time T
Z_T	the set of active commodities at time T
Π_T^τ	available ebits numbers after Phase-τ at time T
$\mathcal{M}_{m:n}, \mathcal{D}_{k:n}^{m:n}, \mathcal{R}_{m:n}$	Input buffer & output buffer & receiving buffer

Variables	Description
$\eta_{s_i t_i}$	the entanglement distribution rate of $s_i{:}t_i$
$f_{m:n}^{m:k}$	amounts of entangled qubit pairs of $m{:}k$ would be distributed to $m{:}n$ after swapping.
$g_{m:n}$	amounts of ebits would be generated along physical link $m{:}n$ divided by the capacity c_{mn}

A two-qubit system is described by a superposition of four basis states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$. Let $|b_1 b_2\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$, such that $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. If a simultaneous measurement is taken on these two qubits, it will yield either 00 with probability $|\alpha_{00}|^2$, 01 with probability $|\alpha_{01}|^2$, 10 with probability $|\alpha_{10}|^2$, or 11 with probability $|\alpha_{11}|^2$. Now, consider a special class of two-qubit states: the four orthogonal Bell states written as $|\Phi^\pm\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle)$, and $|\Psi^\pm\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)$. Two qubits jointly in a Bell state are considered an EPR pair or Bell pair. Each Bell pair is a *maximally entangled* state because it is the superposition of only two complementary states out of the four basis states, and the two qubits are perfectly correlated. Bell pairs form the basis of two-party quantum communications: if Alice and Bob each holds one qubit in a Bell pair, they can use it to exchange quantum information by local operations and classical communication. An entangled qubit pair is called an *ebit*.

B. Entanglement Generation and Swapping

Entanglements can be distributed over long distances via the generation and swapping of entangled photons.

Entanglement generation is the process of generating an entangled pair of photons that are separated by a physical distance, and each photon in the pair is sent to one of two end nodes through a quantum-capable channel, such as an optical fiber. The most common entanglement generation operation is *spontaneous parametric down-conversion (SPDC)*, where a pump laser is shot at a nonlinear crystal and probabilistically generates entangled photon pairs at a high rate. However, entangled photons will get lost during transmission because of channel attenuation or other environmental factors, and the success probability of entanglement generation will decrease exponentially as distance increases [25]. For instance, with a typical single-mode fiber loss of 0.2dB/km [23], a 10000-pair-per-second entanglement source would only be able to successfully distribute 1 pair per second on average over 200km of distance. We consider ebits generated and distributed along a physical channel as *elementary ebits*.
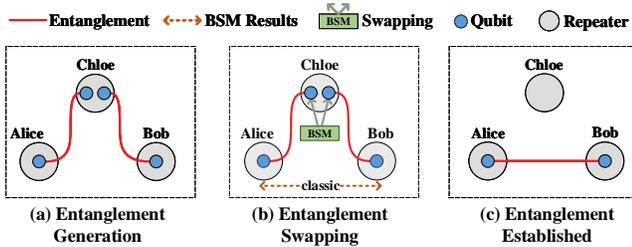
Fig. 1: Entanglement Generation and Swapping

Entanglement swapping is a key process in quantum repeaters to compensate for photon losses in long-distance entanglement generation. In Fig. 1, assume Alice and Bob both perform entanglement generation with Chloe, resulting in one ebit between Alice and Chloe and another between Bob and Chloe. In order to get an end-to-end ebit between Alice and Bob, Chloe will first entangle the two local qubits, and then perform a Bell State Measurement (BSM) to decide which of the four Bell states her local qubits are in. She then sends the result to either Alice or Bob, who applies a unitary operation to the local qubit to turn Alice's and Bob's qubits into a Bell pair *without physical interaction*. Note that because of BSM, this process destroys the two qubits at Chloe regardless of whether the swapping succeeds. If there are multiple repeaters between Alice and Bob, this process can be repeated recursively until an end-to-end ebit is formed. The established long-distance end-to-end ebit can then be used for quantum communication without involving any intermediate nodes.

C. Quantum Network Model

We consider a quantum network consisting of quantum repeaters interconnected by lossy links. The physical topology is denoted by an undirected graph $G = (V, E)$, where node $v \in V$ denotes a quantum repeater and link $e \in E$ is a physical link. For conciseness, we use $m{:}n$ to interchangeably denote an *unordered* node pair $\{m, n\}$ for $m, n \in V$; note that $m{:}n = n{:}m$ since they are unordered. Each node $v \in V$ is associated with swapping success probability $q_v \in (0, 1]$. Each link $e \in E$ consists of c_e quantum channels where c_e is called its *capacity*, and success probability $p_e \in (0, 1]$ for entanglement generation along each channel.

We consider a time-slotted quantum network with discrete time $\mathbf{T} = \{1, 2, 3, \dots\}$ following existing work [31], [37]. Each time slot $T \in \mathbf{T}$ corresponds to the possible generation and distribution of one ebit along a quantum channel, as well as the completion of one round of swapping at all nodes. Each time slot is divided into three phases described below:

1) For any edge $m{:}n \in E$, node m and node n attempt to generate an ebit along each quantum channel with the success probability of p_{mn};
2) For node pair $m{:}n$ and intermediate node k where ebits are available between both $m{:}k$ and $k{:}n$, node k can attempt to perform entanglement swapping with success probability of q_k to establish ebits between m and n by using pairs of ebits between $m{:}k$ and $k{:}n$.
3) For SD pair $s_i{:}t_i \in U$, established ebits are distributed to each commodity to be defined in §III-D for quantum communications.

To optimize network performance, we assume a central network controller oversees the entanglement establishment process across the entire network. The controller communicates with nodes via classical communication, collects infor-

mation from network nodes and SD pairs, and makes global entanglement scheduling and routing decisions.

D. Commodities and Demands

Consider a set of SD pairs in a quantum network requesting end-to-end ebits for communication. Following conventional networking terminology, each request is a *commodity*, denoted by $z_j^i \in Z_i$, where Z_i is the set of all commodities belonging to SD pair i. Each commodity is described by a tuple, $z_j^i = (d_j^i, a_j^i, \delta_j^i)$, where d_j^i denotes the number of requested end-to-end ebits (the demand), a_j^i denotes the arrival time, and δ_j^i denotes the deadline for finishing the demand. If a commodity does not have a deadline but wants to be completed as quickly as possible, we define $\delta_j^i = \infty$. We assume commodities arrive over time, and the network controller has no knowledge of future commodities before they arrive.

Given the time-slotted model, the network state evolves over time. For ease of description, let $\mathcal{S}_T^\tau$ denote the network state after the Phase-τ at time slot T, for $\tau \in \{0, 1, 2\}$; $\mathcal{S}_T^0$ denotes the state at the beginning of time slot T. Each $\mathcal{S}_T^\tau = (G, Z_T, \Pi_T^\tau)$, where $Z_T = \bigcup_i Z_T^i$ denotes the set of commodities that are **active** at time T, *i.e.*, they arrived on or before time T with unfinished demands and unexpired deadlines at time T; $\Pi_T^\tau : V \times V \to \mathbb{Z}^*$ denotes the *number of ebits* available between arbitrary node pair $m{:}n$ after Phase-τ at time slot T. We use Θ_j^i to denote the unfinished remaining demand of $z_j^i \in Z_T^i$ at time T. The value of Π_T^τ depends on the ebits successfully generated after each Phase-1, the ebits successfully swapped after each Phase-2, and the network model (whether buffers exist; see §VII).

The goal of the network controller is to meet as many deadlines of commodities as possible, and complete commodities without deadlines as fast as possible, by making real-time decisions on: 1) how to generate ebits over physical links, 2) how to swap ebits at intermediate repeaters, and 3) how to distribute end-to-end ebits to commodities.

IV. MULTI-COMMODITY ENTANGLEMENT SCHEDULING AND DISTRIBUTION: A GENERAL FRAMEWORK

Consider a commodity arrives with a demand and/or a deadline. The goal of entanglement scheduling and distribution is to deliver as many end-to-end ebits as possible so that the commodity can finish as quickly as possible and hopefully within its deadline. However, as the overall network entanglement generation rate is bounded by the capacities and further discounted by probabilities of generation and swapping, the primary challenge would be resource contention between multiple commodities with different demands and deadlines. In the worst case, a fair sharing network may simply lead to all commodities missing deadlines, while some may have succeeded if prioritization between commodities is applied.

We propose *network-wide entanglement scheduling*, combined with *entanglement distribution*, to deal with network resource contention. Below, we first motivate entanglement scheduling via prioritization, and then define a general framework for optimizing entanglement scheduling and distribution with different scheduling (prioritization) disciplines.

A. A Motivating Example

As shown in Fig. 2, consider A, B and D all connected to a quantum repeater C each with a capacity of 2, and two commodities A-B and A-D arrive at the same time. For simplicity, assume entanglement generation and swapping
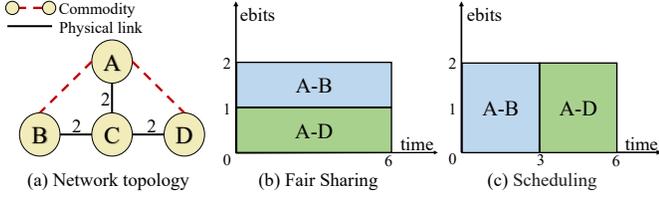
Fig. 2: Motivating example. (a) Network topology with two commodities A-B and A-D and capacity of each link is 2; (b) fair sharing; (c) scheduling with A-B prioritized over A-D.

are deterministic, *i.e.*, all success probabilities are 1. Both commodities have the same demands of 6 end-to-end ebits. In a fair-sharing quantum network, entanglements generated at link A-C are equally shared for swapping to generate A-B and A-D ebits respectively, each at a rate of 1 ebit per time slot. Both commodities finish in 6 time slots, with an *average completion time* of 6. Now assume scheduling (prioritization) is employed, where A-B is given priority over A-D. By utilizing all capacities on links A-C and B-C, commodity A-B can finish within 3 time slots by generating 2 end-to-end ebits per time slot, while commodity A-D's completion time remains the same. The *average completion time* is $\frac{3+6}{2} = 4.5$, yielding a 25% improvement. If commodity A-B has a deadline of 4 time slots and commodity A-D has a deadline of 6 time slots, only A-D can meet its deadline under fair-sharing, while both commodities can meet their deadlines under prioritization, yielding an 100% improvement of *deadline satisfaction ratio*.

The example shows the significance of scheduling in serving time-sensitive communication demands. Such demands widely exist in near-term quantum applications, due to short coherence time in quantum computers, fast changing object environments in quantum sensing, bursty QKD demands, etc. We also highlight that the same motivation has been observed and utilized in classical communication such as data center networks, where scheduling of traffic flows can significantly reduce end-to-end latency and improve user experience [2], [35]. The unique characteristics of quantum networking, however, have posed significant challenges in direct application of classical flow scheduling disciplines to quantum networks.

B. Problem Statement

Considering the *probabilistic nature* of quantum operations, in this subsection, we formally define the *entanglement scheduling and distribution (ESDI)* problem.

Definition 1. *Given a quantum network G and commodities $Z = \bigcup_i Z_i$, a solution to the **entanglement scheduling and distribution (ESDI)** problem consists of three algorithms, $(\mathcal{A}_{\mathsf{gen}}, \mathcal{A}_{\mathsf{swap}}, \mathcal{A}_{\mathsf{dis}})$ to perform the following tasks respectively:*
- *$\mathcal{A}_{\mathsf{gen}}(\mathcal{S}_T^0)$: In Phase-1 at time T, decide the number of ebits to attempt along physical link $e \in E$;*
- *$\mathcal{A}_{\mathsf{swap}}(\mathcal{S}_T^1)$: In Phase-2 at time T, given the number of ebits between node pairs $m{:}k$ and $k{:}n$ respectively, decide how many ebit pairs are used to swap for node pair $m{:}n$, for $\forall m, k, n \in V$; and*
- *$\mathcal{A}_{\mathsf{dis}}(\mathcal{S}_T^2)$: In Phase-3 at time T, given the number of ebits between each SD pair $s{:}t \in U$, decide how many ebits are distributed to each commodity $z_j^i \in Z_i$.*

We note that the above defined problem incorporates all existing entanglement routing or distribution methods as solutions [9], [31], [36]–[38], thus enabling fair comparison between existing and new solutions. Meanwhile, we emphasize

that each algorithm in $\{\mathcal{A}_{\mathsf{gen}}, \mathcal{A}_{\mathsf{swap}}, \mathcal{A}_{\mathsf{dis}}\}$ can be either *deterministic* or *probabilistic*. Despite this, the inputs to $\mathcal{A}_{\mathsf{swap}}$ and $\mathcal{A}_{\mathsf{dis}}$ are *always probabilistic* regardless of previous phases' outputs, due to the probabilistic nature of the actual quantum operations (generation and swapping) in the first and second phases. This probabilistic nature makes the ESDI problem intrinsically challenging.

C. Multi-Commodity Remote Entanglement Distribution

While existing work has tried to design specific algorithms for specific objectives (such as maximizing total entanglement distribution rate (EDR) or ensuring fairness), our first goal is to design a general optimization framework for ESDI that can incorporate flexible objectives and constraints. In the following, we first extend the single-commodity Optimal Remote Entanglement Distribution (ORED) formulation in [9] to a multi-commodity formulation (MRED), serving as the *backbone* of our optimization framework.

Define variables $\mathcal{F} = \{f_{m:n}^{m:k} \geq 0 \,|\, m, k, n \in V\} \cup \{g_{m:n} \in [0, 1] \,|\, (m{:}n) \in E\}$. Here $f_{m:n}^{m:k}$ represents the number of ebits between $m{:}k$ that would be contributed to swapping with ebits between $k{:}n$ at node k; $g_{m:n}$ represents the number of ebits that would be attempted to be generated along physical link $m{:}n \in E$, divided by the link capacity c_{mn}. Then, the **MRED** formulation is defined as follows:

$$\text{(MRED)} \quad \text{find } \mathcal{F} \tag{1}$$
$$\text{s.t.} \quad f_{m:n}^{m:k} = f_{m:n}^{k:n}, \quad \forall k, m, n \in V; \tag{1a}$$
$$I(m{:}n) = \Omega(m{:}n), \; \forall m, n \in V, m{:}n \notin U; \tag{1b}$$
$$I(s{:}t) \geq \Omega(s{:}t), \; \forall s{:}t \in U. \tag{1c}$$

Two auxiliary functions $I(m{:}n)$ and $\Omega(m{:}n)$ are defined as:

$$I(m{:}n) \triangleq \mathbb{1}_{m:n} p_{mn} c_{mn} g_{m:n} + \sum_{k \in N \setminus \{m,n\}} \frac{q_k}{2} \left(f_{m:n}^{m:k} + f_{m:n}^{k:n} \right); \tag{1d}$$

$$\Omega(m{:}n) \triangleq \sum_{k \in N \setminus \{m,n\}} \left(f_{m:k}^{m:n} + f_{k:n}^{m:n} \right), \tag{1e}$$

where $\mathbb{1}_{m:n} = 1$ if $m{:}n \in E$ and 0 otherwise. Here $I(m{:}n)$ denotes the *input* (established) ebits between $m{:}n$, and $\Omega(m{:}n)$ denotes the *output* (consumed) $m{:}n$-ebits for swapping to generate ebits between all other node pairs.

Explanation: Program (1) defines a feasibility problem: finding appropriate variables $\{g_{m:n}\}$ (entanglement generation) and $\{f_{m:n}^{m:k}\}$ (entanglement swapping), such that: 1) every swapping between $m{:}k$ and $k{:}n$ at node k consumes an equal number of $m{:}k$- and $k{:}n$-ebits as in Constraint (1a); 2) for non-SD pairs, all acquired ebits will be used for further swapping as in Constraint (1b); 3) for any SD pair $s{:}t$, the input ebits should be no less than the output as in Constraint (1c), with the difference corresponding to *end-to-end* ebits that are kept to be used between $s{:}t$ themselves. Taking a closer look at $I(m{:}n)$, it consists of both ebits generated directly along link $m{:}n$ (the first term), and ebits obtained by swapping at all intermediate nodes k (the second term), both discounted by the corresponding success probabilities p_{mn} and q_k respectively. The output $\Omega(m{:}n)$ consists of all $m{:}n$-ebits contributed to swapping to generate either $m{:}k$ ebits with n as an intermediate node, or $k{:}n$-ebits with m as an intermediate node.

The MRED formulation itself does not consider online decision making as implicitly included in the ESDI problem definition, since there is no input or decision variable with respect to the time T (or any time-related input). It nevertheless constitutes a major building block in our framework design due to its ability to incorporate various optimization objectives and

constraints. For example, define η_{st} as the total *entanglement generation rate (EDR)* of all commodities between SD pair $s{:}t \in U$, and $\eta \triangleq \sum_{s{:}t \in U} \eta_{st}$ the total EDR of all SD pairs. The following theorem generalizes Theorem 1 in [9] (with proof omitted due to page limit) showing optimality of the MRED formulation *w.r.t.* the total EDR of all commodities:

Theorem 1. *The optimal total EDR η^* is upper bounded by $\max_{\mathcal{F}}\{\sum_{s{:}t}(I(s{:}t) - \Omega(s{:}t)) \mid \mathcal{F}$ is feasible to (1)$\}$, and there exists a stationary ESDI protocol with expected total EDR equal to η^*.*

D. A General Framework for ESDI

We design a general ESDI framework based on the MRED formulation and the idea of *prioritization* as shown in Algorithm 1. Based on the set of commodities and their remaining demands, our framework adjusts the prioritization between SD pairs, by dynamically adding or removing objective functions and constraints to the MRED formulation in (1). It further balances between *scheduling* (prioritizing certain SD pairs) and *work conservation* (maximizing network EDR). The solution to the *prioritized* MRED is then executed by a probabilistic protocol to implement the prioritization quickly.

Algorithm 1: ESDI General Framework

---
1   $\mathcal{F} \leftarrow \bot$;
2   **for** *all time slot $T \in \mathbf{T}$* **do**
3   **if** $\mathcal{F} = \bot$ *or priorities may change* **then**
4   Adjust objectives and constraints in Program (1);
5   Solve adjusted (1) to update optimal $\mathcal{F}$;
6   Execute $\mathcal{F}$;

In general, the framework keeps track of the set of active commodities and priorities among them. When initializing, or when priorities among active commodities change, the backbone Program (1) will be adjusted with objective functions and/or constraints reflecting the latest priorities. It then gets solved to derive the optimal solution $\mathcal{F}$, which will be executed over time until priorities change again. The priority changes are decided by a **scheduling algorithm** and could happen in a number of cases such as changes in active commodities and demands. The solution $\mathcal{F}$ is instead executed by a **distribution protocol** consisting of $\{\mathcal{A}_{\mathsf{gen}}, \mathcal{A}_{\mathsf{swap}}, \mathcal{A}_{\mathsf{dis}}\}$ as defined in Definition 1. In the following, we will focus on the design of these elements, including two scheduling algorithms for deadline-agnostic (§V) and deadline-aware (§VI) commodities respectively, and a generic distribution protocol that can be tailored to both types of commodities (§VII).

V. ESDI WITHOUT DEADLINE CONSTRAINTS

In this section, we consider ESDI for commodities without deadlines. An example is QKD between end points [19], [24], whose focus is to accumulate a sufficient number of classical key bits obtained from entanglements as soon as possible. The main goal of scheduling is to minimize the average completion time of all commodities in a quantum network.

A. Motivation: Shortest Job First Scheduling

Shortest job first (SJF) and its variant, *shortest remaining time first (SRTF)*, are optimal scheduling policies for average task completion time in classical real-time task scheduling. SJF allocates the idle processing unit to the task with the shortest completion time among the remaining tasks, while SRTF allows a newly arriving task to *preempt* the currently

processing task. Due to their strong performance and simple implementation, SJF and SRTF are widely used in network traffic scheduling [3], [16]. Nevertheless, extending these policies to ESDI is very challenging. First, scheduling over a network of repeaters is more challenging than scheduling on a single machine and is generally NP-hard [15]. In this case, exclusively scheduling one commodity at a time is clearly inefficient since the network may simultaneously support multiple commodities without resource contention. Second, the probabilistic nature of quantum operations makes it impossible to obtain the exact completion time of each commodity.

Our goal is to design an algorithm that *simulates* the behavior of an SJF scheduler in the quantum network scenario, while balancing between **strict scheduling** (one commodity at a time) and **work conservation** (achieving high utilization of network resources and throughput). The design of our algorithm includes two components: 1) an MRED formulation with strict priorities, and 2) an SJF-based prioritization algorithm.

B. MRED with Strict Priorities

Consider a *priority list of SD pairs* P_s given as input, which consists of κ SD pairs sorted from high to low priority. Let $\eta_i = \eta_{s_i t_i}$ be the expected EDR between SD pair $s_i{:}t_i \in P_s$ with index i. The MRED with strict priorities (MRED-SP) is formulated by enforcing strict priorities among commodities:

$$\text{(MRED-SP)} \quad \max \eta_1, \ \max \eta_2, \ \ldots, \ \max \eta_\kappa,$$

$$\max \sum_{s{:}t \in U} \eta_{st} \tag{2}$$

$$\text{s.t.} \quad \eta_{st} = I(s{:}t) - \Omega(s{:}t), \quad \forall s{:}t \in U; \tag{2a}$$

$$\text{Constraints (1a)–(1e)}.$$

Explanation: Program (2) is a *multi-objective optimization* problem with up to $\kappa + 1$ objectives. The first κ objectives enforce strict priorities among SD pairs, *i.e.*, the program will first optimize for η_1, then optimize for η_2 while keeping optimality of η_1, then optimize for η_3 while keeping optimality of both η_1 and η_2, so on and so forth. The last objective, which optimizes for the total EDR of all SD pairs, is added to achieve work conservation, *i.e.*, utilizing the remaining resources unused by the κ prioritized SD pairs to maximize the overall throughput of the network. With Constraint (2a) providing a definition of each η_{st}, Program (2) is an optimization version of Program (1) enforcing constraints (1a)–(1e).

Remark: We note that while the list P_s may also be defined upon commodities instead of SD pairs, it is not meaningful in the formulation. Consider two commodities belonging to the same SD pair $s{:}t$. Incorporating both in the priority list results in adding the objective η_{st} twice to Program (2), which has no impact than just adding one for the higher-priority commodity, but will increase solving time of the multi-objective program.

C. ESDI-O: SJF-based Priority Scheduling

The key in utilizing Program (2) for ESDI is to form the priority list P_s of SD pairs. Following the intuition of SJF, P_s should reflect how fast each SD pair will be able to finish its commodity, *i.e.*, the *expected completion time (ECT)* of the commodity, with the lowest demand. This is however challenging because of the difficulty in modeling the exact probability distributions governing the entanglement generation and swapping processes across the network. Nevertheless, as our primary goal is to define priorities instead of obtaining the accurate ECTs, we may use an approximation—a lower bound of the ECT. Specifically,

consider a commodity $z_j^i \in Z_i$ receives an expected EDR of η, and define a random variable $\eta(T)$ as the actual number of ebits generated and distributed to z_j^i in every time slot T. We have $ECT_j^i = \mathbb{E}\left[\min\left\{T' \,\middle|\, \sum_{T=1}^{T'} \eta(T) \geq d_j^i\right\}\right] \approx \mathbb{E}\left[\min\left\{T' \,\middle|\, \mathbb{E}[\eta] \cdot T' \geq d_j^i\right\}\right] = \mathbb{E}[d_j^i/\mathbb{E}[\eta]]$. By Jensen's inequality, we then have $ECT_j^i \geq d_j^i/\mathbb{E}[\eta]$.

To rank SD pairs by priorities, we use $d_j^i/\mathbb{E}[\eta]$ as an approximate ECT for each commodity. Here $\mathbb{E}[\eta]$ is computed by running MRED with only one SD pair in the quantum network, *i.e.*, exclusively allocating all network resources to a single SD pair. SD pairs are then ranked by the approximate ECT of the commodity with the *smallest demand* for each pair. After ranking, a multi-commodity (multi-objective) MRED-SP formulation is solved to obtain the eventual ESDI solution.

Algorithm 2: ESDI-O for Commodities without Deadlines

Input: Network $G = (V, E)$, commodities Z, SD pairs U, scheduling length κ

1 $\mathcal{F} \leftarrow \emptyset$, $Z_T \leftarrow \emptyset$ for $\forall T$, $\eta_{st} \leftarrow \perp$ for $\forall s{:}t \in U$;
2 **for** *SD pair $s{:}t \in U$* **do**
3     $\eta_{st} \leftarrow \max_{\mathcal{F}}\{\eta_{st} \,|\, \text{Program (1)}\}$;
4 **for** $T = 1, 2, \ldots$ **do**
5   Add arriving commodities to Z_T;
6   **if** *active commodity list Z_T has changed* **then**
7        $P_s \leftarrow$ sort SD pairs by $\min_j\{d_j^i\}/\eta_{s_i t_i}$;
8   Keep first κ SD pairs in P_s and drop the rest;
9        $\mathcal{F} \leftarrow$ solve Program (2) with P_s;
10   Execute $\mathcal{F}$ by calling $(\mathcal{A}_{\text{gen}}, \mathcal{A}_{\text{swap}}, \mathcal{A}_{\text{dis}})$;
11     $Z_{T+1} \leftarrow Z_T \setminus \{z_j^i \,|\, d_j^i \text{ is finished}\}$;
12 **return** *when all commodities have finished.*

Based on the above intuition, Algorithm 2 gives the detailed online ESDI-O algorithm to minimize the average completion time of all commodities. Line 1 initializes the solution $\mathcal{F}$, the active commodity list Z_T and the expected EDR η_{st} for every SD pair. In Line 3, the expected EDR for each SD pair is computed *offline*, by assuming it is the only SD pair in the network. In the online process, whenever the active commodity list Z_T changes with either commodity arriving at time T or was completed after $T - 1$, the solution $\mathcal{F}$ will be updated. First, in Line 7, SD pairs will be sorted by *lower bound* of the ECT of each SD pair's commodity with the *lowest demand*, *i.e.*, $\min_j\{d_j^i\}/\eta_{s_i t_i}$. The first κ SD pairs are then entered into Program (2) in the priority list P_s, and the solution $\mathcal{F}$ will be updated after solving Program (2) in Line 9. Then, in every time slot, the up-to-date solution $\mathcal{F}$ is executed, by calling $(\mathcal{A}_{\text{gen}}, \mathcal{A}_{\text{swap}}, \mathcal{A}_{\text{dis}})$ which we detail in §VII-B. Finally, completed commodities will be removed from the active commodity list Z_{T+1} in Line 11.

Remark: In addition to following the general intuition of SJF that we outlined before, Algorithm 2 also contains several practical design elements to improve its performance in practice. First, the solution $\mathcal{F}$ is only updated when the commodity list changes. Second, when scheduling with different sets of commodities, we seek to *preserve the relative priorities between commodities*, by sorting only based on the original demand d_j^i instead of the remaining demand of each commodity. While these may seem counter-intuitive from the scheduling perspective, we find that they actually help improve the performance by achieving better *work conservation*. Specifically, because the network generates ebits probabilistically between

each pair of swapping node pairs $m{:}k$ and $k{:}n$, almost in any time slot T there is one side with more generated ebits than the other side, leading to intermediate ebits not being utilized in time T. These intermediate ebits would be wasted if the underlying swapping decisions have changed, wasting resources and degrading throughput. By updating the solution $\mathcal{F}$ only when active commodities change, and preserving relative priorities among commodities, the algorithm can minimize the number of times that the underlying swapping decisions change, hence reducing wastage and accelerating the completion of commodities.

VI. ESDI WITH DEADLINE CONSTRAINTS

In this section, we consider ESDI for commodities with deadlines. In some quantum applications like DQC, decoherence is a critical challenge, where the quantum information stored in qubits gradually decoheres over time (even with quantum error correction) [5]. It is thus crucial to finish transmitting the information before irreversible errores happen.

A. Motivation: Earliest Deadline First Scheduling
When scheduling tasks with deadlines, *Earliest Deadline First (EDF)* is a provably optimal *preemptive* scheduling policy in classical real-time scheduling [16]. But similar to SJF, it cannot be directly extended due to the quantum network characteristics, more specifically, difficulty in accurately estimating the ECT and the multi-resource contention among multiple commodities. Further, the previous formulation in (2) is no longer suitable because of deadlines, since it would prioritize one commodity strictly over another, neglecting cases where a set of deadlines may mostly or all be satisfied when the commodities jointly share the network resources.

This section develops an EDF-inspired algorithm following the same structure as in the last section, including: 1) an MRED formulation enforcing deadline-aware priorities, and 2) an EDF-based prioritization algorithm.

B. MRED with Deadline Constraints
Here, instead of giving a list of SD pairs with strict priorities among them, we are given a *priority list of commodities P_c*, such that we seek to make sure *all* commodities in P_c can be completed by their deadlines *on expectation*. Each commodity $z_j^i \in P_c$ has a *remaining demand* Θ_j^i denoting the number of end-to-end ebits yet to be distributed, and Δ_j^i remaining time slots until the deadline of the commodity. Let U_c be the set of SD pairs that commodities in P_c belong to, and let $P_c^i \subseteq P_c$ be the set of commodities belonging to $i \in U_c$. The commodities in each P_c^i are sorted in **non-decreasing order of their remaining time slots** Δ_j^i. We further facilitate notation by defining $P_c^i[l]$ as the *first l commodities in list P_c^i*.

Note that SD pairs in U_c are not prioritized over each other, *i.e.*, all SD pairs are treated the same; but priorities are defined among commodities belonging to each SD pair in the priority list P_c^i. Let $\eta_{s_i t_i}$ be the expected EDR between SD pair $s_i{:}t_i \in U$ as previously defined. The MRED with deadline constraints (MRED-DC) is formulated as followed:

$$\text{(MRED-DC)} \quad \max \sum_{s{:}t \in U} \eta_{st} \tag{3}$$

$$\text{s.t.} \quad \eta_{s_i t_i}\Delta_j^i \geq \sum_{z_j^i \in P_c^i[l]} \Theta_j^i, \tag{3a}$$

$$\forall s_i{:}t_i \in U_c, l = 1, 2, \ldots, |P_c^i|;$$

$$\text{Constraints (1a)–(1e) and (2a).}$$

Explanation: Different from Program (2) in which the objectives enforce strict priorities, the objective function in Program (3) is merely to achieve *work conservation*, *i.e.*, maximizing overall network throughput. Instead, the prioritization is fully enforced through Constraint (3a), which specifies **requirements** on the expected EDR of each SD pair in the priority set U_c. Consider an SD pair i with only one commodity $z_j^i \in P_c^i$. Constraint (3a) basically specifies that the expected number of ebits distributed within the remaining time slots Δ_j^i, must be able to satisfy all the remaining demand Θ_j^i of the commodity, that is, $\eta_{s_i t_i} \Delta_j^i \geq \Theta_j^i$.

The case gets trickier when an SD pair has multiple commodities in P_c^i. In this case, having $\eta_{s_i t_i} \Delta_j^i \geq \Theta_j^i$ for each individual $z_j^i \in P_c^i$ is no longer sufficient, since the distributed end-to-end ebits must be shared among commodities. Nevertheless, a close inspection reveals that with respect to a single SD pair, the contention among commodities precisely replicates the classical single-machine task scheduling problem with deadlines, in which case EDF is proved to be optimal.

Indeed, Constraint (3a) fully simulates EDF with respect to each SD pair, by always requiring the commodity with the smallest remaining time slot Δ_j^i to be completed first. The first commodity $z_{j_1}^i \in P_c^i[1]$ thus satisfies the same condition $\eta_{s_i t_i} \Delta_{j_1}^i \geq \Theta_{j_1}^i$. For the second commodity $z_{j_2}^i \in P_c^i[2] \setminus P_c^i[1]$, it can only start after $z_{j_1}^i$ has finished, and hence the condition for it to finish becomes $\eta_{s_i t_i} \Delta_{j_2}^i \geq \Theta_{j_1}^i + \Theta_{j_2}^i$. The third commodity similarly requires $\eta_{s_i t_i} \Delta_{j_3}^i \geq \Theta_{j_1}^i + \Theta_{j_2}^i + \Theta_{j_3}^i$, so on and so forth. Constraint (3a) gives the general form.

Remark: MRED-DC prioritizes commodities via constraints, which is generally stronger than prioritization via objectives. This is to enforce the deadline requirement of commodities. Meanwhile, MRED-DC can be more *work conserving* than MRED-SP, in the sense that each prioritized commodity or SD pair does not try to take all the network resources, but instead would only take what is needed to complete before the deadline, leaving more room for network-wide throughput optimization. Note that despite MRED-DC requiring completion *on expectation*, a prioritized commodity may not be able to finish after all due to statistical inevitability. But we find such cases relatively rare and do not impact MRED-DC's practical performance with the prioritization algorithm in §VI-C.

On the other hand, one implication of prioritization via constraint is the possibility of an **infeasible** program, in which it may not be possible for all input commodities to complete on expectation (due to too short remaining deadlines or too much contention). We address this by the following algorithm.

C. ESDI-E: EDF-based Priority Scheduling

The essence of utilizing Program (3) is to adjust the commodity priority list based on the active commodities dynamically. The priority list P_c should be able to reflect how many time slots commodities have for completing their demands. At time T, the remaining time slots for commodity z_j^i is $\Delta_j^i = \delta_j^i - T + 1$.

Algorithm 3 provides the detailed online ESDI-E algorithm to maximize the number of commodities that could be finished before their deadlines. Line 1 initializes the solution $\mathcal{F}$, the active commodity list Z_T, and the expected EDR η_{st} for every SD pair. For the online process, any commodity not finished by its deadline will be dropped in Line 3. The solution $\mathcal{F}$ and commodities in the priority list P_c will be updated once new commodities arrive or commodities leave at time slot T,

---

Algorithm 3: ESDI-E for Commodities with Deadlines

> **Input:** Network $G = (V, E)$, commodities Z, SD pairs U_T, scheduling length κ
> **1** $\mathcal{F} \leftarrow \emptyset$, $Z_T \leftarrow \emptyset$ for $\forall T$, $\eta_{st} \leftarrow \bot$ for $\forall s{:}t \in U$;
> **2** **for** $T = 1, 2, \dots$ **do**
> **3** $Z_T \leftarrow Z_T \setminus \{z_j^i \,|\, \delta_j^i < T\}$;
> **4**   Add arriving commodities to Z_T;
> **5** $\Delta_j^i \leftarrow \delta_j^i - T + 1$ for $\forall z_j^i \in Z_T$;
> **6**   **if** *active commodity list Z_T has changed* **then**
> **7** $P_c \leftarrow \emptyset$;
> **8** $\mathcal{F} \leftarrow$ solve Program (3) with P_c;
> **9**   **for** $z_j^i \in Z_T$ *in increasing order of Δ_j^i* **do**
> **10**   **if** $|P_c| = \kappa$ **then break**;
> **11** $P_c \leftarrow P_c \cup \{z_j^i\}$;
> **12** $\mathcal{F} \leftarrow$ solve Program (3) with P_c;
> **13**   **if** $\mathcal{F}$ *is infeasible* **then** $P_c \leftarrow P_c \setminus \{z_j^i\}$;
> **14**   Execute last feasible $\mathcal{F}$ via $(\mathcal{A}_{\mathsf{gen}}, \mathcal{A}_{\mathsf{swap}}, \mathcal{A}_{\mathsf{dis}})$;
> **15**   Update remaining demands Θ_j^i for $z_j^i \in Z_T$;
> **16** $Z_{T+1} \leftarrow Z_T \setminus \{z_j^i \,|\, d_j^i \text{ is finished}\}$;
> **17** **return** *when all commodities have finished.*

---

starting from Line 6. Active commodities will be sorted by the remaining deadlines Δ_j^i to ensure commodities with more urgent deadlines could be served as soon as possible in Line 9.

The priority list P_c is built *incrementally* via sequentially solving Program (3), with the goal of forming a *maximally feasible* priority list with up to κ prioritized commodities. In each iteration at Line 9, one commodity is added to P_c, and Program (3) is solved to decide if all commodities in P_c can finish by their deadlines on expectation. If adding a commodity makes Program (3) infeasible, it will be dropped from P_c. The eventual solution $\mathcal{F}$ at time T is the last feasible solution of Program (3). The rest for executing the solution $\mathcal{F}$ and updating the active commodity list are the same as in Algorithm 2.

Remark: For the same reason in ESDI-O (minimizing wasted intermediate ebits), ESDI-E also updates the solution only when the commodity list changes. Sorting commodities by remaining deadlines naturally preserves the relative priorities among commodities, similar to using the original demands for sorting in ESDI-O. Meanwhile, ESDI-E may be significantly slower than ESDI-O with a large κ, due to the need of repeatedly solving Program (3) in each update iteration to ensure feasibility. For this reason, κ should be kept as a small value, which additional benefits work conservation for maximizing network throughput.

VII. ESDI: ENTANGLEMENT DISTRIBUTION DESIGN

With the solution $\mathcal{F}$ output by either ESDI-O or ESDI-E, we next design algorithms $\{\mathcal{A}_{\mathsf{gen}}, \mathcal{A}_{\mathsf{swap}}, \mathcal{A}_{\mathsf{dis}}\}$ to implement the solution, targeting at a buffered quantum network scenario.

A. Buffered Quantum Network

In existing work on entanglement routing or distribution, it is commonly assumed that quantum memories are unavailable or have extremely short coherence time [31], [36]–[38]. Hence ebits generated for swapping must be either consumed or discarded in one time slot. This severely limits the entanglement generation rate of the quantum network. Nevertheless, recent advances in quantum memories have demonstrated relatively long storage times of coherent qubits from seconds [11] to over 1 hour [20]. This has motivated recent designs of
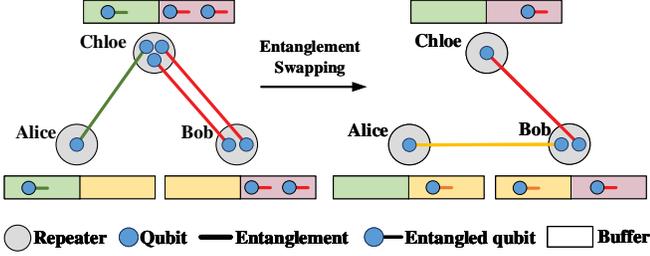
Fig. 3: Buffered quantum network. Different colors of buffers and entanglements represent half part of an ebit being stored at the corresponding repeater's buffer and entanglement between the corresponding repeaters, respectively.

buffered quantum networks, where intermediate repeaters can store ebits for an extended period before entanglement swapping [8], [9], [14]. Below, we first describe a general model for a buffered quantum network and then present protocols implementing our optimized entanglement scheduling and distribution solutions.

As is shown in Fig. 3, We consider establishing ebits between one SD pair Alice and Bob, with the help of repeater Chloe. Assume after the entanglement generation, there are one entanglement between Alice and Chloe and two entanglements between Chloe and Bob. Each ebit would be stored in buffers at both ends, respectively. Now Chloe performs entanglement swapping for one ebit pair from each buffer. If swapping succeeds, an ebit is generated between Alice and Bob while the two qubits in Chloe's buffer are both consumed. After getting the swapping result, Alice would move her local qubit from her buffer with Chloe to her buffer with Bob, and Bob would move his local qubit from his buffer with Chloe to his buffer with Alice, which can be done by simply relabeling these qubits. The remaining unused ebits between Chloe and Bob would be stored in Chloe's and Bob's buffers to wait for another Alice-Chloe pair for swapping.

B. Entanglement Distribution Design

Given a solution $\mathcal{F}$ output by a central quantum network controller, we design a *multi-commodity* extension of the protocol in [8] to achieve the expected entanglement distribution rate.

For the generation process ($\mathcal{A}_{\mathsf{gen}}$), each link $m{:}n$ will continuously attempt to generate elementary ebits at rate $c_{mn} \cdot g_{m:n}$. For any node pair $m{:}n$, we consider set $\mathcal{M}_{m:n}$ to store established ebits between node m and node n, and set $\mathcal{D}_{m:k}^{m:n}$ to store $m{:}n$-ebits that will be used to swap into $m{:}k$-ebits. For an SD pair $s{:}t \in U$, an additional set $\mathcal{R}_{s:t}$ is kept to store all end-to-end ebits distributed between source s and destination t. All sets are physically implemented by *quantum buffers* with classical labels at both m and n, with $\mathcal{M}_{m:n}$ denoting the **input buffer** of the node pair, $\mathcal{D}_{m:k}^{m:n}$ denoting the **output buffer** for "next-hop" node pair $m{:}k$, and $\mathcal{R}_{m:n}$ denoting the **receiving buffer** for storing "completed" end-to-end ebits to be utilized by upper-layer quantum communication protocols or applications.

The swapping process ($\mathcal{A}_{\mathsf{swap}}$) is a probabilistic process with two steps: 1) moving generated/established ebits from the input buffer to output buffers *probabilistically* (switching), and 2) swapping when corresponding buffers are non-empty (swapping). Whenever an ebit is added to $\mathcal{M}_{m:n}$, the two endpoints will implement **opportunistic switching**, by jointly tossing a random coin and moving the ebit from $\mathcal{M}_{m:n}$ to

$\mathcal{D}_{m:k}^{m:n}$ or $\mathcal{R}_{m:n}$ with the following probabilities:

$$\Pr[\text{move to } \mathcal{D}_{m:k}^{m:n}] = \frac{f_{m:k}^{m:n}}{\sum_{k'} f_{m:k'}^{m:n} + \eta_{mn}},$$

$$\Pr[\text{move to } \mathcal{R}_{m:n}] = \frac{\eta_{mn}}{\sum_{k'} f_{m:k'}^{m:n} + \eta_{mn}}.$$

where $\eta_{mn} = 0$ for a non-SD pair $m{:}n \notin U$. Finally, each node k will check if for any $m{:}n$, there exists
1) $f_{m:n}^{m:k} = f_{m:n}^{k:n} > 0$;
2) $\mathcal{D}_{m:n}^{m:k} \neq \emptyset$, and $\mathcal{D}_{m:n}^{k:n} \neq \emptyset$.

For each such case, node k performs swapping between each pair of ebits in $\mathcal{D}_{m:n}^{m:k}$ and $\mathcal{D}_{m:n}^{k:n}$ respectively. Upon success, the ebit will then be added to $\mathcal{M}_{m:n}$ by m and n. Compared to the single-commodity protocol in [8], the additional receiving buffer $\mathcal{R}_{s:t}$ is required for implementing MRED, since each SD pair may also be assigned to contribute established ebits for other SD pairs, instead of keeping all ebits to their own use. All the above can be parallel and asynchronous.

Once end-to-end ebits are established and stored in $\mathcal{R}_{s:t}$, it further needs to be distributed to commodities of the same SD pair. This distribution process ($\mathcal{A}_{\mathsf{dis}}$) can be deterministic. Following our design principles in §V and §VI, we apply the following policies:
1) For commodities without deadlines, the commodity with the least remaining demand will be served first, followed by the second, so on and so forth.
2) For commodities with deadlines, the commodity with the earliest deadline would be served, followed by the second, so on and so forth.

Both are due to that commodity scheduling for single SD pair reduces to (deterministic) single-machine classical scheduling, where SJF and EDF are optimal in corresponding scenarios.

VIII. PERFORMANCE EVALUATION

A. Evaluation Methodology

To evaluate our solution, we developed a discrete-time quantum network simulator. We used random Waxman graphs [33] with model parameters $\alpha = \beta = 0.8$. The success probability of entanglement generation and swapping were 0.9. Each link had a capacity uniformly sampled from $[3, 10]$. We generated graphs with 20 nodes and picked 1000 SD pairs randomly.

Our simulator was based on a time-slotted model to be compatible with existing algorithms. We implemented our ESDI protocol and two state-of-the-arts. Linear programs were solved using the Gurobi solver [1].

The following algorithms were compared:
- **ESDI-B:** basic ESDI without scheduling as in (1);
- **ESDI-O:** ESDI without deadlines in Algorithm 2;
- **ESDI-E:** ESDI with deadlines in Algorithm 3;
- **E2E-F:** fidelity-aware protocol in [38] maximizing end-to-end EDR. We set fidelity as 1 since it is not considered.
- **QPASS:** QPASS protocol in [31] trying to maximize end-to-end EDR for multiple SD pairs.

Commodities arrived following a Poisson distribution with arrival rate $\lambda = 1$ by default. Demands of each commodity followed an exponential distribution with mean of 600 ebits, and a minimum demand of 100 ebits per commodity. For a commodity with a deadline, we considered the difference between its deadline and arrival time to be a random function with expectation proportional to its demand. Define $\mu = 0.4$. A unit deadline $\bar{\delta}_j^i$ was drawn from a uniform distribution in the range $[\mu - 0.1, \mu + 0.1]$ for each commodity, and the deadline of
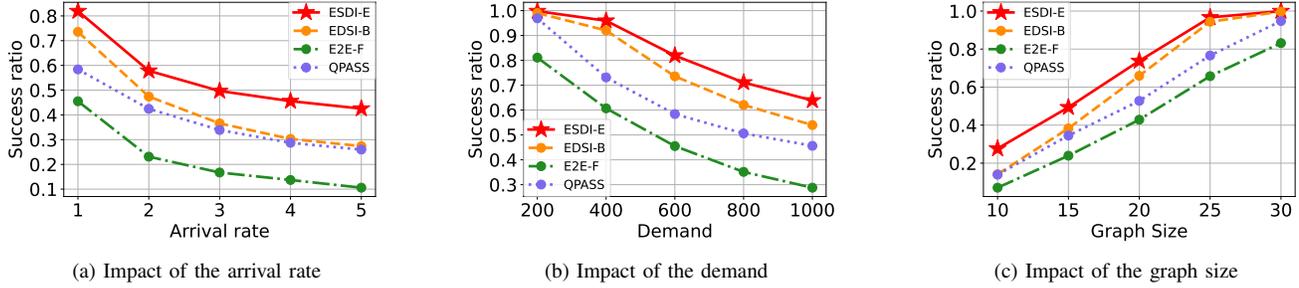
(a) Impact of the arrival rate
(b) Impact of the demand
(c) Impact of the graph size

Fig. 4: Success ratio between ESDI and state-of-the-art algorithms

(a) Impact of the arrival rate
(b) Impact of the demand
(c) Impact of the graph size

Fig. 5: Average completion time between ESDI and state-of-the-art algorithms

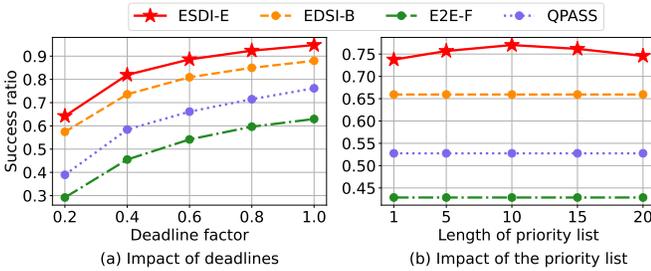
(a) Impact of deadlines
(b) Impact of the priority list

Fig. 6: Impact of deadlines and the length of priority list

the commodity with demand d_j^i was set as $\delta_j^i = a_j^i + \bar{\delta}_j^i \cdot d_j^i$. We set the default scheduling length $\kappa = 1$. Both QPASS and E2E-F use Yen's algorithm where we set the number of paths to 15. Since they are *entanglement routing* algorithms for bufferless quantum networks, we simulated their bufferless behavior by dropping all ebits after one time slot. Each simulation was run with 5 seeds to reduce random noise.

The following metrics were used for evaluation. ***Success ratio*** measures the ratio of the number of commodities finished before their deadlines. ***Average completion time*** measures the average time between each commodity's arrival and completion when there is no deadline.

B. Evaluation Results

1) Success ratio for commodities with deadline: Fig. 4 shows the success ratio of ESDI-E and other algorithms with varying parameters, while all commodities have deadlines. The success ratio decreased with increasing arrival rate in Fig. 4(a) and increasing per-commodity demand in Fig. 4(b) due to more severe resource contention, and increased with increasing graph size in Fig. 4(c) due to more abundant resources in the network. From all figures, ESDI-E and ESDI-B achieved the highest success ratio compared to other algorithms, and ESDI-E further improved success ratio by up to 55% over ESDI-B. This leads to two critical observations: 1) ***MRED*** with optimal EDR in a buffered quantum network can significantly ***improve network-wide throughput*** over existing heuristic-based bufferless algorithms, and 2) ***scheduling via prioritization*** (ESDI-E)

can additionally ***finish more commodities before deadlines***.

2) Average completion time for commodities without deadline: Fig. 5 shows the average completion time for ESDI-O and other algorithms with varying parameters. With higher arrival rate in Figs. 5(a) or higher demand in Fig. 5(b), the average completion time of all algorithms increased, again due to more severe contention of resources. Larger graph size decreased average completion time in Fig. 5(c) by alleviating the contention. The key observation across figures is again that ESDI-O and ESDI-B outperformed the other algorithms by 1) ***maximizing network-wide throughput*** with MRED in ESDI-B, and 2) ***scheduling*** via prioritization in ESDI-O.

3) Impact of deadlines: Fig. 6(a) shows the success ratio for ESDI-E and other algorithms with varying deadline factors. For each deadline factor value (such as 0.8), we multiplied the deadline of every commodity with the factor. From the figure, we observe increasing success ratios with increasing deadline factors, due to more commodities being able to finish before their deadlines. The advantage of ESDI-E (and ESDI-B compared to other heuristics) was consistent across different deadlines, showing the constant advantage of our algorithms.

4) Impact of scheduling length κ: Fig. 6(b) shows the success ratio for ESDI-E and other algorithms with varying lengths of the priority list for scheduling. Because there was no priority list in ESDI-B, E2E-F, and QPASS, their success ratios kept the same. For ESDI-E, we observed a trade-off between *scheduling* and *work conservation*, the two primary goals we defined in §IV-D. Specifically, increasing the scheduling length κ from the minimum value 1 to an intermediate value 10 slightly increased the success ratio of ESDI-E, while further increasing κ led to decrease in the success ratio. The initial increase was due to our algorithm being able to *prioritize* commodities without significant impact on the throughput of other commodities in the network. As prioritization increased further, throughput started to degrade since some prioritized commodities (those with earlier deadlines) might span across longer paths, thus having lower end-to-end EDR than other commodities with later deadlines but shorter paths. In practice, the exact optimal length κ depends on the network topology

and the set of commodities, and finding this optimal scheduling length is an important future work. Fortunately, as we can observe, even setting $\kappa = 1$ (scheduling only one commodity at a time) still led to a substantial advantage over any non-scheduling baseline, thus motivating us to set $\kappa = 1$ as the default value for evaluation.

IX. CONCLUSIONS

In this paper, we designed a general optimization framework **ESDI** for entanglement scheduling and distribution in a general quantum network. Motivated by scenarios where different quantum communication applications have different demands and time requirements (such as deadlines), we first developed a multi-commodity entanglement distribution formulation, and then designed two scheduling and distribution algorithms based on the idea of *scheduling via prioritization*. We further designed a practical probabilistic protocol to implement the optimized ESDI decisions in a buffered quantum network. We developed a discrete-time quantum network simulator for evaluation. Extensive simulation results showed our solutions could significantly reduce the average completion time of quantum communication demands and increase the success ratio of commodities when compared to existing entanglement routing and distribution algorithms.

REFERENCES

[1] "Gurobi Optimizer," accessed 2022-07-25. URL: http://www.gurobi.com/products/gurobi-optimizer

[2] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center tcp (dctcp)," in *ACM SIGCOMM*, 2010, pp. 63–74.

[3] W. Bai, L. Chen, K. Chen, D. Han, C. Tian, and H. Wang, "Information-agnostic flow scheduling for commodity data centers," in *USENIX NSDI*, 2015, pp. 455–468.

[4] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theoretical Computer Science*, vol. 560, pp. 7–11, 2014.

[5] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, "Quantum Internet: Networking challenges in distributed quantum computing," *IEEE Network*, vol. 34, no. 1, pp. 137–143, 2019.

[6] C. Cicconetti, M. Conti, and A. Passarella, "Resource allocation in quantum networks for distributed quantum computing," *arXiv preprint arXiv:2203.05844*, 2022.

[7] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpedek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho *et al.*, "A link layer protocol for quantum networks," in *ACM SIGCOMM*, 2019, pp. 159–173.

[8] W. Dai, T. Peng, and M. Z. Win, "Optimal protocols for remote entanglement distribution," in *IEEE ICNC*, 2020, pp. 1014–1019.

[9] ——, "Optimal remote entanglement distribution," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 3, pp. 540–556, 2020.

[10] ——, "Quantum queuing delay," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 3, pp. 605–618, 2020.

[11] Y. Dudin, L. Li, and A. Kuzmich, "Light storage on the time scale of a minute," *Physical Review A*, vol. 87, no. 3, p. 031801, 2013.

[12] C. Elliott, "Building the quantum network," *New Journal of Physics*, vol. 4, no. 1, p. 46, 2002.

[13] A. Farahbakhsh and C. Feng, "Opportunistic routing in quantum networks," in *IEEE INFOCOM*, 2022.

[14] H. Gu, Z. Li, R. Yu, X. Wang, F. Zhou, and J. Liu, "Fendi: High-fidelity entanglement distribution in the quantum internet," *arXiv preprint arXiv:2301.08269*, 2023.

[15] K. Han, Z. Hu, J. Luo, and L. Xiang, "Rush: Routing and scheduling for hybrid data center networks," in *IEEE INFOCOM*, 2015, pp. 415–423.

[16] C.-Y. Hong, M. Caesar, and P. B. Godfrey, "Finishing flows quickly with preemptive scheduling," in *ACM SIGCOMM*, 2012.

[17] M. Koashi and N. Imoto, "No-cloning theorem of entangled states," *Physical Review Letters*, vol. 81, no. 19, p. 4264, 1998.

[18] C. Li, T. Li, Y.-X. Liu, and P. Cappellaro, "Effective routing design for remote entanglement generation on quantum networks," *npj Quantum Information*, vol. 7, no. 1, p. 10, 2021.

[19] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, Y. Li, J.-G. Ren, J. Yin, Q. Shen, Y. Cao, Z.-P. Li *et al.*, "Satellite-to-ground quantum key distribution," *Nature*, vol. 549, no. 7670, pp. 43–47, 2017.

[20] Y. Ma, Y.-Z. Ma, Z.-Q. Zhou, C.-F. Li, and G.-C. Guo, "One-hour coherent optical storage in an atomic frequency comb memory," *Nature Communications*, vol. 12, no. 1, pp. 1–6, 2021.

[21] S. P. Neumann, A. Buchner, L. Bulla, M. Bohmann, and R. Ursin, "Continuous entanglement distribution over a transnational 248 km fiber link," *Nature Communications*, vol. 13, no. 1, p. 6134, 2022.

[22] N. K. Panigrahy, T. Vasantam, D. Towsley, and L. Tassiulas, "On the capacity region of a quantum switch with entanglement purification," in *IEEE INFOCOM*, 2023.

[23] M. Pant, H. Krovi, D. Towsley, L. Tassiulas, L. Jiang, P. Basu, D. Englund, and S. Guha, "Routing entanglement in the quantum internet," *npj Quantum Information*, vol. 5, no. 1, pp. 1–9, 2019.

[24] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. Dynes *et al.*, "The secoqc quantum key distribution network in vienna," *New Journal of Physics*, vol. 11, no. 7, p. 075001, 2009.

[25] S. Pirandola, R. García-Patrón, S. L. Braunstein, and S. Lloyd, "Direct and reverse secret-key capacities of a quantum channel," *Physical Review Letters*, vol. 102, no. 5, p. 050503, 2009.

[26] S. Pouryousef, N. K. Panigrahy, and D. Towsley, "A quantum overlay network for efficient entanglement distribution," in *IEEE INFOCOM*, 2023.

[27] S. I. Salim, A. Quaium, S. Chellappan, and A. B. M. A. Al Islam, "Enhancing fidelity of quantum cryptography using maximally entangled qubits," in *IEEE GLOBECOM*, 2020, pp. 1–6.

[28] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka *et al.*, "Field test of quantum key distribution in the Tokyo QKD Network," *Optics Express*, vol. 19, no. 11, pp. 10 387–10 409, 2011.

[29] E. Schoute, L. Mancinska, T. Islam, I. Kerenidis, and S. Wehner, "Shortcuts to quantum network routing," *arXiv preprint arXiv:1610.05238*, 2016.

[30] D. Shah and J. Shin, "Randomized scheduling algorithm for queueing networks," *The Annals of Applied Probability*, vol. 22, no. 1, pp. 128–171, 2012.

[31] S. Shi and C. Qian, "Concurrent entanglement routing for quantum networks: Model and designs," in *ACM SIGCOMM*, 2020, pp. 62–75.

[32] A. Singh, K. Dev, H. Siljak, H. D. Joshi, and M. Magarini, "Quantum internet—applications, functionalities, enabling technologies, challenges, and research directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2218–2247, 2021.

[33] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[34] J. Yin, Y. Cao, Y.-H. Li, S.-K. Liao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, B. Li, H. Dai *et al.*, "Satellite-based entanglement distribution over 1200 kilometers," *Science*, vol. 356, no. 6343, pp. 1140–1144, 2017.

[35] R. Yu, G. Xue, X. Zhang, and J. Tang, "Non-preemptive coflow scheduling and routing," in *IEEE GLOBECOM*, 2016, pp. 1–6.

[36] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Multi-entanglement routing design over quantum networks," in *IEEE INFOCOM*, 2022.

[37] Y. Zhao and C. Qiao, "Redundant entanglement provisioning and selection for throughput maximization in quantum networks," in *IEEE INFOCOM*, 2021, pp. 1–10.

[38] Y. Zhao, G. Zhao, and C. Qiao, "E2E fidelity aware routing and purification for throughput maximization in quantum networks," in *IEEE INFOCOM*, 2022.