Quantum Queuing Delay

Wenhan Dai [ID], *Student Member, IEEE*, Tianyi Peng [ID], *Student Member, IEEE*, and Moe Z. Win [ID], *Fellow, IEEE*

Abstract—Queuing delay is an essential topic in the design of quantum networks. This paper introduces a tractable model for analyzing the queuing delay of quantum data, referred to as quantum queuing delay (QQD). The model employs a dynamic programming formalism and accounts for practical aspects such as the finite memory size. Using this model, we develop a cognitive-memory-based policy for memory management and show that this policy can decrease the average queuing delay exponentially with respect to memory size. Such a significant reduction can be traced back to the use of entanglement, a peculiar quantum phenomenon that has no classical counterpart. Numerical results validate the theoretical analysis and demonstrate the near-optimal performance of the developed policy.

Index Terms—Quantum networks, queuing delay, teleportation, memory management.

I. INTRODUCTION

QUANTUM information processing systems have the potential to create the next technological revolution [1]–[3], enabling various applications such as quantum communication [4]–[7], quantum sensing [8]–[10], quantum computing [11]–[15], as well as next-generation positioning, navigation, and timing [16]–[18].

While physical implementation of quantum networks is advancing rapidly [19]–[22], little is known about the design and analysis of operation strategies, beyond the physical layer, that are essential for transmitting quantum information reliably and efficiently. In particular, queuing delay of quantum data, referred to as quantum queuing delay (QQD), is one of the critical issues in transmission of information across quantum networks. Compared to its classical counterpart, queuing delay is even more important for quantum networks: the quantum states interact with the environment and will lose a significant amount of information if not delivered on time [23]–[25]. The difficulties of analyzing QQD are two-fold. First, there is arguably no mathematical model that characterizes the queuing node and the queuing process in quantum networks. Such a model has to tally with the physical realizations (e.g.,

the quantum channels and quantum operations) and practical constraints (e.g., quantum memory size and quantum state lifetime). Second, quantum communication has its own peculiar properties that have no classical counterparts. For example, quantum communication may exploit quantum entanglement [26]–[29], a phenomenon representing non-local interconnection among quantum objects. These peculiar properties make the strategies designed for classical networks ill-suited for quantum networks.

Existing work related to QQD can be divided into two groups: queuing delay in classical networks and operation design in quantum networks. There are myriad studies on queuing delay in classical networks [30]–[32]. Although some of the concepts such as queue length, Little's theorem, blocking probability, and stability [33]–[35] may be borrowed for studying quantum networks, the many methods tailored for the classical queuing theory do not apply directly to QQD. On the other hand, there are only a few studies on the operation designs for quantum networks, proposing ad-hoc protocols and verifying their performance via simulations. In [36], a decentralized entanglement routing protocol is proposed to find the shortest path in a quantum network using local knowledge of quantum nodes. In [37], optimized entanglement routing protocols are developed based on dynamic programming. In [38], link layer protocols are proposed for quantum networks and their performance is evaluated via simulations. These studies either maximize the quantum network throughput instead of queuing delay or provide heuristic protocols without performance guarantee.

The fundamental questions related to QQD are:
- how to develop a tractable quantum queuing model that is consistent with the physical realizations and practical constraints; and
- how to characterize and exploit the properties specific to quantum nature for developing efficient policies that minimize QQD.

The answers to these questions will enable us to minimize QQD, which can further unleash the potential of quantum networks.

The goals of this paper are to build a mathematical model for characterizing QQD and to determine policies for controlling such a delay with performance guarantee. An essential part in this model is teleportation [39], a celebrated technique for sending quantum information from one node to another using entanglement and classical communication. The new technical idea in this paper is the introduction of dynamic programming in the modeling of QQD [40]. Our view is that quantum nature can bring new phenomena in the area of queuing delay. We believe that QQD can be significantly reduced by establishing entanglements and storing them in

the entanglement memory before, instead of waiting until, quantum data arrive. Specifically, entanglements can be seen as resources that are reserved in the memory of a node, and quantum data in the queue can be delivered via teleportation using these resources instead of waiting for the transmission time.

In this paper, we establish a mathematical formalism for characterizing QQD and propose policies to significantly reduce QQD. The key contributions of this paper are as follows:

- we establish a tractable model for characterizing QQD using dynamic programming;
- we determine the average queuing delay for the scenario with one receiver and show that such delay can decrease exponentially with respect to the memory size of a node;
- we develop a cognitive-memory-based policy for minimizing the average queue length in a general scenario and show that it is optimal for the scenario with two receivers; and
- we derive an upper bound for the average queue length corresponding to the proposed cognitive-memory-based policy. This bound implies that the average queuing delay can decrease exponentially with respect to the memory size of a node.

The remainder of the paper is organized as follows. Section II presents the system model and introduces the problem of minimizing queueing delay in quantum networks. Section III presents the analysis for a scenario with one receiver to gain insights into the quantum queuing systems. Section IV presents the cognitive-memory-based policy, shows that it is optimal for the scenario with two receivers, and derives an upper bound of average queue length for the general scenario. The performance of the proposed policies is presented in Section V. Finally, the conclusions are drawn in Section VI.

Notation: Random variables are displayed in sans serif, upright fonts; their realizations in serif, italic fonts. Vectors and matrices are denoted by bold lowercase and uppercase letters, respectively. For example, a random variable and its realization are denoted by x and x; a random vector and its realization are denoted by $\mathbf{x}$ and $\boldsymbol{x}$; a random matrix and its realization are denoted by $\mathbf{X}$ and $\boldsymbol{X}$, respectively. Sets and random sets are denoted by upright sans serif and calligraphic font, respectively. For example, a random set and its realization are denoted by X and $\mathcal{X}$, respectively. The m-by-n matrix of zeros (resp. ones) is denoted by $\mathbf{0}_{m \times n}$ (resp. $\mathbf{1}_{m \times n}$); when $n = 1$, the m-dimensional vector of zeros (resp. ones) is simply denoted by $\mathbf{0}_m$ (resp. $\mathbf{1}_m$). The m-by-m identity matrix is denoted by $\boldsymbol{I}_m$: the subscript is removed when the dimension of the matrix is clear from the context. The set of positive integers is denoted by $\mathbb{N}^*$. The cardinality of a set $\mathcal{S}$ is denoted by $\mathrm{Card}(\mathcal{S})$. The set $\{m, m+1, \ldots, n\}$ is denoted by $\mathcal{K}_{m:n}$.

II. SYSTEM MODEL

This section presents the system model and introduces the problem of minimizing queueing delay in quantum networks.

A. Quantum Node

We consider a quantum system composed of a quantum node and a collection of receivers as illustrated in Fig. 1. A quantum node is composed of an entanglement-generating platform, an entanglement memory, and N_r quantum data queues, where N_r is the number of receivers. Quantum data are quantum states (e.g., the spin of an electron and the polarization of a photon), and they arrive at the quantum node according to a stochastic process.[1] Each quantum datum is associated with a destined receiver and is quantified by a qubit (qubit). Next we describe each component in the quantum node and present the protocol for transmitting quantum data to the corresponding receiver.

Quantum Data Queue: Each quantum data queue is associated with a receiver. Quantum data are stored in the quantum data queue after arrival. These data are assumed to be stored perfectly for infinite time in the quantum data queue.

Entanglement-generating Platform: In practice, there are multiple ways of generating entanglement, i.e., entangled qubit pairs, between two nodes [41]–[43]. We focus on abstract models and leave more detailed physics realizations for later. We consider a platform for making attempts to generate entangled qubit pairs between the node and a receiver. Such attempts may fail due to imperfection in practical operations. Note that the platform does not have the storing capability and qubits are assumed to be collapsed in the next time slot if stored at the platform.

Entanglement Memory: If the attempt of entanglement generation succeeds, an entangled qubit pair will be shared between the quantum node and a receiver, with one qubit at the quantum node and the other at the receiver. The quantum node will use this entangled qubit pair for teleportation immediately or move half of the entangled qubit pair (i.e., the qubit at the quantum node) to the entanglement memory for future use [44]. Such half of the entangled qubit pair will be referred to as entangled qubit in the rest of the paper. The entangled qubit pairs are assumed to be stored perfectly for infinite time in the entanglement memory.

Teleportation Protocol: The quantum node employs the teleportation protocol to transmit quantum data to a receiver [39].[2] In teleportation, the quantum node performs Bell measurements on the qubit to be transmitted, denoted as $|\psi\rangle$, as well as its half of the entangled qubit pair; the measurement result is then sent to the receiver, which will perform operations on its half of the entangled qubit pair accordingly; the resulting qubit at the receiver then becomes $|\psi\rangle$.

The teleportation protocol permits the quantum node to transmit 1 qubit at the cost of sending 2 classical bits (cbits)

---

[1] These quantum data are either generated by the quantum node itself or are sent to the quantum node by a collection of transmitters, which are not illustrated in Fig. 1 explicitly. In the latter case, the quantum data are sent to the node from transmitters via either direct transmission or teleportation; then the quantum node needs to move the data to the corresponding data queues.

[2] The reason for employing teleportation rather direct transmission using quantum channels are two folds. First, it is less challenging to establish entanglement and perform teleportation using near-future quantum technologies compared to direct transmission, which requires entanglement involving many qubits for quantum error correction coding. Second, as will be shown in this paper, using teleportation results in much less QQD compared to direct transmission.
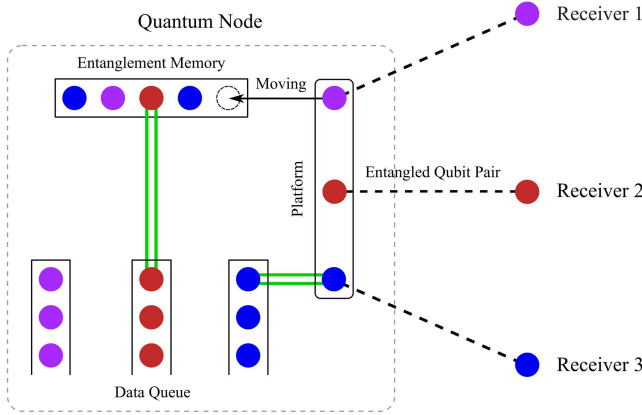
Fig. 1. An illustration of the quantum node and receivers. Quantum nodes are composed of quantum data queues, platforms, and entanglement memories. Purple, red, and blue circles represent qubits corresponding to Receiver 1, 2, 3, respectively. Hollow circles represent vacant spot in the entanglement memory. Double green lines represent teleportation. Dashed lines represent entangled qubit pairs. Black arrows represent the operation of moving qubits from the platform to the entanglement memory.

and consuming 1 entangled qubit pair. In this paper, we consider that classical communication resources are free since in the foreseeable future, the communication capability of classical information will be significantly larger than that of quantum information. Therefore, the delay brought by the classical communication is negligible compared to QQD, and is omitted in this paper. The bottleneck of quantum information transmission then lies in the entanglement generation rate and in the storage capability.

B. Dynamic Programming Formalism

We formulate the quantum data transmission at a quantum node as a dynamic programming problem. Consider a dynamic system with discrete time slots t_n ($n = 0, 1, 2, \dots$). Let $s_i^{(n)}$ denote the number of qubits associated with Receiver i in the system at the beginning of time slot t_n. In particular, $s_i^{(n)} \geqslant 0$ implies there are $s_i^{(n)}$ qubits in the ith data queue; $s_i^{(n)} < 0$ implies there are $-s_i^{(n)}$ entangled qubits associated with Receiver i in the memory.

Each time slot is divided into the following three phases.

1) Data arrival and entanglement generation: $a_i^{(n)}$ qubits of quantum data associated with Receiver i arrive at the node; meanwhile, the node makes an attempt to generate entanglement in the platform with every receiver; $b_i^{(n)}$ entangled qubit pairs associated with Receiver i are successfully generated.

2) Teleportation: the node adopts teleportation to transmit quantum data by consuming entangled qubit pairs. The entangled qubits at the node are in the platform and the entanglement memory. The entangled qubits in the platform are given priority for serving as resources in teleportation.

3) Entanglement storage: the remaining entangled qubits in the platform are moved to the entanglement memory.

Note that in the third phase, if the entanglement memory cannot accommodate all the entangled qubits in the platform, the node has to discard some entangled qubits and move the rest to the memory. The policy for discarding entangled qubits in such a scenario is referred to as *entanglement memory management*. Moreover, we consider that there is a physical limit Q_t for the number of quantum data corresponding to a receiver, meaning that after the teleportation phase, if the number of quantum data in the queue is greater than Q_t, the node has to discard the newly arrived quantum data. This gives $s_i^{(n)} \leqslant Q_t, \forall i, n.$[3]

The number of entangled qubit pairs that are successfully generated, i.e., $b_i^{(n)}$, is known to the node before the control decision $u^{(n)}$ is made. If this assumption does not hold (e.g., if the node and the receiver are far apart), one can create a slightly different model, but the insights obtained in this paper would still be valid.

With the background introduced above, the mathematical model for quantum data transmission at a quantum node has the form

$$\boldsymbol{x}^{(n+1)} = f(\boldsymbol{x}^{(n)}, u^{(n)}, \boldsymbol{w}^{(n)}), \quad n = 0, 1, 2, \dots \quad (1)$$

where $\boldsymbol{x}^{(n)}, u^{(n)}, \boldsymbol{w}^{(n)}$ and f are described below.

- $\boldsymbol{x}^{(n)}$: the state of the system, consisting of the number of qubits at the beginning of time slot t_n, as well as the difference between the number of quantum data arriving at the quantum node and the number of entangled qubit pairs successfully generated during the first phase of the time slot, i.e.,

$$\boldsymbol{x}^{(n)} = \left[\left(\boldsymbol{s}^{(n)}\right)^T \left(\boldsymbol{c}^{(n)}\right)^T \right]^T$$

in which

$$\boldsymbol{s}^{(n)} = \left[s_1^{(n)} \ s_2^{(n)} \ \dots \ s_{N_r}^{(n)} \right]^T$$

$$\boldsymbol{c}^{(n)} = \left[c_1^{(n)} \ c_2^{(n)} \ \dots \ c_{N_r}^{(n)} \right]^T$$

and $c_i^{(n)} = a_i^{(n)} - b_i^{(n)}.$[4]

- $u^{(n)}$: the control policy at time slot n, i.e., the policy for entanglement memory management. In particular, $u^{(n)}$ is a function that maps $\boldsymbol{x}^{(n)}$ into $\boldsymbol{s}^{(n+1)}$ with the constraint that

$$\sum_{i=1}^{N_r} \max\{0, -s_i^{(n+1)}\} \leqslant M$$

where M denotes the capacity of the entanglement memory. Moreover, due to the upper bound for the number of quantum data, we require that

$$s_i^{(n+1)} = Q_t \quad \text{if } s_i^{(n)} + c_i^{(n)} > Q_t. \quad (2)$$

---

[3]If the discarded quantum data are required to be recovered, one can either encode the quantum data with error correction coding techniques, or require the retransmission of the quantum data. In the latter case, the transmitters need to have knowledge about the quantum data so that the no-cloning theorem is not violated.

[4]Note that due to the teleportation in the second phase, $c_i^{(n)}$ is sufficient for entanglement memory management.

- $w^{(n)}$: the instantiation of the random vector that represents all the uncertainty in the system at time slot t_{n+1}, i.e., $w^{(n)} = c^{(n+1)}$. The distribution of $w^{(n)}$ is assumed known *a priori*. In particular, consider that $a_i^{(n)}$ are independent and identically distributed (i.i.d.) Bernoulli random variables with parameter $\mathbb{P}\{a_i^{(n)} = 1\} = p_i$ (independent over i and n) and that $b_i^{(n)}$ are also i.i.d. Bernoulli random variables with parameter $\mathbb{P}\{b_i^{(n)} = 1\} = q_i$.[5] The distribution of $c_i^{(n)}$ is found to be

$$\mathbb{P}\{c_i^{(n)} = k\} = \begin{cases} (1-p_i)q_i & \text{if } k = -1 \\ p_i q_i + (1-p_i)(1-q_i) & \text{if } k = 0 \\ p_i(1-q_i) & \text{if } k = +1 \\ 0 & \text{otherwise.} \end{cases}$$

In the sequel, we assume that $0 < p_i, q_i < 1$, $i \in \mathcal{K}_{1:N_r}$.

- f: a function that describes the system. Note that $u^{(n)}(x^{(n)}) = s^{(n+1)}$ and $w^{(n)} = c^{(n+1)}$; therefore, f can be expressed as

$$f(x^{(n)}, u^{(n)}, w^{(n)}) = \left[\left(u^{(n)}(x^{(n)}) \right)^{\mathrm{T}} \left(w^{(n)} \right)^{\mathrm{T}} \right]^{\mathrm{T}}.$$

This model is illustrated in Fig. 2. When $n = 0$, $s_i^{(0)} = 0$, $i \in \mathcal{K}_{1:N_r}$, $a_1^{(0)} = 0$, $a_2^{(0)} = 1$, $a_3^{(0)} = 1$, $b_1^{(0)} = 1$, $b_2^{(0)} = 0$, $b_3^{(0)} = 1$. Consequently, $c_1^{(0)} = -1$, $c_2^{(0)} = 1$, $c_3^{(0)} = 0$. The entangled qubit corresponding to Receiver 1 is moved to the memory, which gives $s_1^{(1)} = -1$. When $n = 1$, $s_1^{(1)} = -1$, $s_2^{(1)} = 1$, $s_3^{(1)} = 0$, $a_1^{(1)} = 0$, $a_2^{(1)} = 1$, $a_3^{(1)} = 0$, $b_1^{(1)} = 1$, $b_2^{(1)} = 0$, $b_3^{(1)} = 1$. Consequently, $c_1^{(1)} = -1$, $c_2^{(1)} = 1$, $c_3^{(1)} = -1$. Note that in this case, there are three entangled qubits in the memory and the platform after the teleportation phase, and this is beyond the capacity of the memory. The control policy in this case chooses to discard the entangled qubit in the platform corresponding to Receiver 1, which gives $s_1^{(2)} = -1$ and $s_3^{(2)} = -1$. When $n = 2$, $s_1^{(2)} = -1$, $s_2^{(2)} = 2$, $s_3^{(2)} = -1$, $a_i^{(2)} = 1$, $b_i^{(2)} = 0$, $i \in \mathcal{K}_{1:N_r}$. Consequently, $c_i^{(2)} = 1$, $i \in \mathcal{K}_{1:N_r}$. After the teleportation phase, there are three qubits in the second data queue, which is beyond the capacity of the data queue. The node has to discard one qubit, which gives $s_2^{(3)} = 2$.

The goal is to minimize the average delay of the quantum queuing system. Due to the relationship between average delay and average queue length shown by Little's law [45], we aim at designing the control policy $u^{(n)}$ for minimizing the expected average queue length.[6] In particular, let $J_{\mathrm{L}}(x^{(1)})$ denote the

expected average queue length starting in the state $x^{(1)}$, i.e.,

$$J_{\mathrm{L}}(x^{(1)}) := \limsup_{N \to \infty} \mathbb{E}\left\{ \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{N_r} \max\left\{0, s_i^{(n)}\right\} \Big| x^{(1)} \right\}. \tag{3}$$

C. Practical Implementation

We now consider one of the practical methods of entanglement generation and justify the Bernoulli distribution assumption of $b_i^{(n)}$ in the previous subsection. As mentioned in [38], [46], Nitrogen-Vacancy (NV) platforms are available for entanglement generation. There are three practical factors that need to be considered as described below.

- Time for one entanglement attempt: consider that the distance between the quantum node and a receiver is about 25 kilometers (a typical distance between two neighboring European cities) [38]. The time for an attempt to generate an entangled qubit pair between the node and the receiver is about 145 μs, including photon emission, electron readout, and the communication delay.
- Success probability for one entanglement attempt: according to [38], the success probability for one entanglement attempt is about $\alpha \times 10^{-3}$, where α is a parameter such that the fidelity of the entanglement is $1 - \alpha$.[7] For example, if the fidelity of the entanglement is required to be 0.9, the success probability is about 10^{-4}.
- Times for teleportation and entanglement storage: the operation for teleportation in the transmitter's side is essentially a Bell-state measurement, which takes about 100 μs using an NV platform [46]; the time for moving qubits to the memory or the queue is about 1040 μs.

Note that the times for teleportation and entanglement storage are akin to the overhead in the classical communication networks, whereas the time for the entanglement attempt is akin to communication time. Moreover, the total time for teleportation and entanglement storage is much larger than that for one entanglement attempt. To improve entanglement generation efficiency, multiple attempts of entanglement generation can be made in the first phase of a time slot. Suppose there are 500 attempts in the first phase. Then the duration of the phase is 7.25×10^4 μs, which is much larger than the time of the second and third phase (about 1.1×10^3 μs). With the mild assumption that the results of these attempts are i.i.d., the number of generated entanglements follows a binomial distribution. If the success probability for one attempt is set to be 10^{-4}, corresponding to a fidelity value of 0.9 as shown above, then

$$\mathbb{P}\{\text{number of generated entanglements} = k\}$$
$$\begin{cases} = 0.9512 & \text{if } k = 0 \\ = 0.0476 & \text{if } k = 1 \\ \approx 0 & \text{if } k > 1 \end{cases}$$

---

[5]As no large-scale quantum networks have been implemented, it is not clear how to model the distribution of the quantum data arrival. In this paper, motivated by classical networks, we consider that $a_i^{(n)}$ follows the Bernoulli distribution. The justification for $b_i^{(n)}$ following a Bernoulli distribution will be shown in the next subsection.

[6]Little's law shows that the average queue length is the multiplication of the average queuing delay and the effective arrival rate of quantum data. Later we will see that the blocking probability is almost zero for the developed policies and hence the effective arrival rate is almost a constant.

[7]Fidelity is a real number in $[0,1]$ that characterizes the quality of entanglement. Higher fidelity implies higher quality.
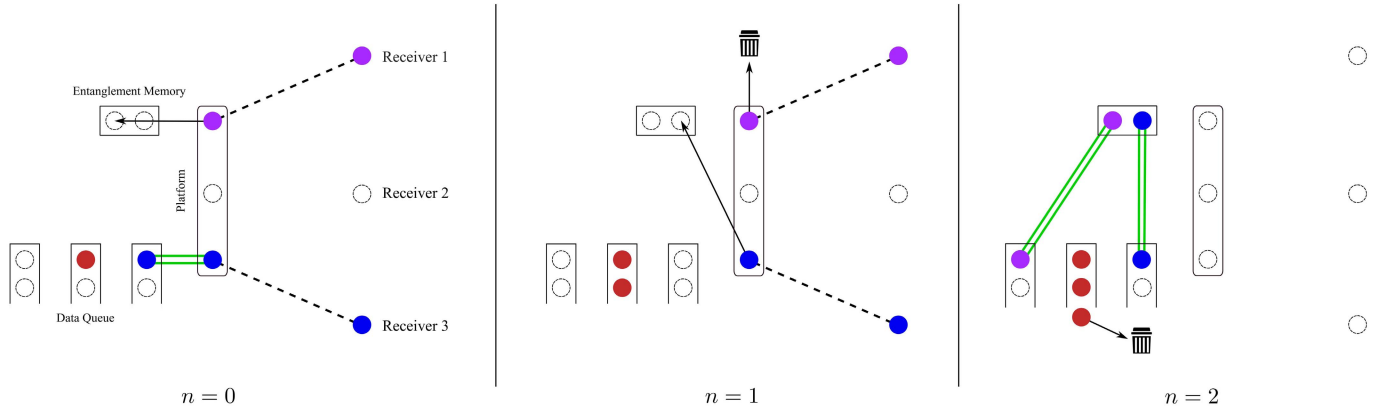
Fig. 2. An illustration of dynamic programming formalism for $N_r = 3$, $M = 2$, $Q_t = 2$. Purple, red, and blue circles represent qubits corresponding to Receiver 1, 2, and 3, respectively. Hollow circles represent vacant spot in the receivers, platforms, data queues, and the entanglement memory. Double green lines represent teleportation. Dashed lines represent entangled qubit pairs. Black arrows represent the operation of moving qubits from the platform to the entanglement memory and the operation of discarding qubits.

which can be approximated as a Bernoulli distribution as the probability of generating more than one entangled qubit pair is lower than 1.2×10^{-3}. If the required fidelity value is higher than 0.9, $b_i^{(n)}$ can be approximated only better as a Bernoulli random variable. This justifies the assumption that $b_i^{(n)}$ follows a Bernoulli distribution.

III. ONE-RECEIVER SCENARIO

We consider a simple scenario, where there is only one receiver, i.e., $N_r = 1$, to gain insights into the quantum queuing system.

Recall that the number of quantum data $a_i^{(n)}$ and the number of entangled qubits $b_i^{(n)}$ are i.i.d. Bernoulli random variables. In the presence of only one receiver, the subscript is dropped in this section, e.g.,

$$\mathbb{P}\{a^{(n)} = 1\} = p \qquad \mathbb{P}\{b^{(n)} = 1\} = q. \qquad (4)$$

In the scenario with $N_r = 1$, the control $u^{(n)}$ is trivial: in the third phase, the quantum node simply moves the remaining entangled qubits in the platform to the entanglement memory as long as the memory does not reach its full capacity, i.e.,

$$u^{(n)}\left(\begin{bmatrix} s^{(n)} & c^{(n)} \end{bmatrix}^T\right) = \begin{cases} Q_t & \text{if } s^{(n)} + c^{(n)} > Q_t \\ -M & \text{if } s^{(n)} + c^{(n)} < -M \\ s^{(n)} + c^{(n)} & \text{otherwise.} \end{cases}$$

Note that if there are multiple receivers, then determining $u^{(n)}$ becomes challenging as it involves the allocation of memory among different receivers. We next evaluate the expected average queue length in the following proposition.

Proposition 1: For $N_r = 1$, if $p, q \in (0, 1)$, the expected average queue length is

$$J_L(\boldsymbol{x}^{(1)}) = \begin{cases} \dfrac{\alpha^M[\alpha - \alpha^{Q_t+1} - (1-\alpha)Q_t\alpha^{Q_t+1}]}{(1-\alpha)(1-\alpha^{Q_t+M+1})} & \text{if } p \neq q \\[4mm] \dfrac{Q_t(Q_t+1)}{2(Q_t+M+1)} & \text{if } p = q \end{cases}$$

where

$$\alpha = \frac{p(1-q)}{q(1-p)}.$$

Proof: Consider the evolution of $s^{(n)}$. The transition probability from the state $s^{(n)}$ to the state $s^{(n+1)}$ is

$$\mathbb{P}\left\{s^{(n+1)} = s^{(n+1)} \big| s^{(n)} = s^{(n)}\right\}$$

$$= \begin{cases} p(1-q) & \text{if } s^{(n+1)} = s^{(n)} + 1 \text{ and} \\ & s^{(n+1)} \leqslant Q_t \\ pq + (1-q)(1-p) & \text{if } s^{(n+1)} = s^{(n)} \text{ and} \\ & s^{(n)} \in \mathcal{K}_{-M+1:Q_t-1} \\ q(1-p) & \text{if } s^{(n+1)} = s^{(n)} - 1 \text{ and} \\ & s^{(n+1)} \geqslant -M \\ 1 - p + pq & \text{if } s^{(n+1)} = s^{(n)} = -M \\ 1 - q + pq & \text{if } s^{(n+1)} = s^{(n)} = Q_t. \end{cases}$$

Let π denote the stationary distribution of $s^{(n)}$, i.e.,

$$\pi_i = \lim_{n \to \infty} \mathbb{P}\{s^{(n)} = i | s^{(1)} = s^{(1)}\}, \quad i \in \mathcal{K}_{-M:Q_t}.$$

The stationary distribution π satisfies the following properties:

$$\pi_{-M} = (1 - p + pq)\pi_{-M} + q(1-p)\pi_{-M+1}$$
$$\pi_i = (1-q)p\pi_{i-1} + [pq + (1-p)(1-q)]\pi_i$$
$$\qquad + q(1-p)\pi_{i+1}, \qquad i \in \mathcal{K}_{-M+1:Q_t-1}$$
$$\pi_{Q_t} = (1-q)p\pi_{Q_t-1} + (1 - q + pq)\pi_{Q_t}.$$

Solving these equations gives

$$\pi_i = \begin{cases} \alpha^i \dfrac{\alpha^M(1-\alpha)}{1-\alpha^{Q_t+M+1}} & \text{if } p \neq q \\[4mm] \dfrac{1}{Q_t + M + 1} & \text{if } p = q \end{cases}$$

where $i \in \mathcal{K}_{-M:Q_t}$. The expected average queue length is

$$J_L(\boldsymbol{x}^{(1)}) = \sum_{i=1}^{Q_t} i\,\pi_i$$

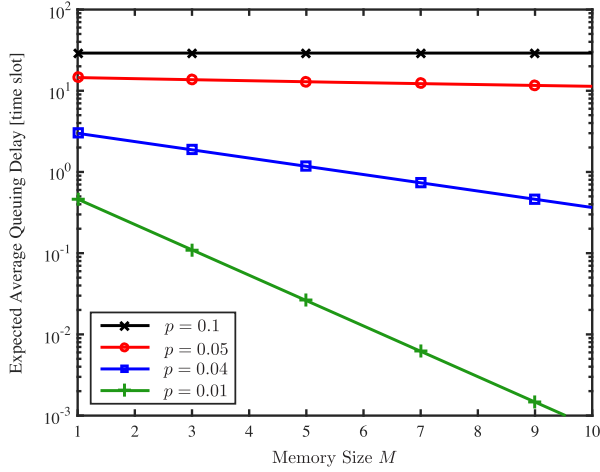which is the desired result after some calculation. $\qquad \square$

Fig. 3.   Expected average queue length as a function of the memory size M with $N_r = 1$ and $Q_t = 30$. The parameter q is fixed as 0.05, whereas different values of p are considered.

Fig. 3 shows the expected average queue length as a function of M with $N_r = 1$ and $Q_t = 30$. The parameter q is set as 0.05 for consistency with the analysis in Section II-C. The observations obtained from Fig. 3 are as follows. These observations will also hold for other values of q.

- If $p < q$, i.e., $\alpha < 1$, the expected average queue length decreases exponentially as a function of M. This is because for $\alpha < 1$,

$$C_1 \alpha^M \leqslant J_L(\boldsymbol{x}^{(1)}) \leqslant C_2 \alpha^M$$

where $C_1 = \left[\alpha - \alpha^{Q_t+1} - (1-\alpha)\,Q_t\,\alpha^{Q_t+1}\right]/(1-\alpha)$ and $C_2 = C_1/(1 - \alpha^{Q_t+1})$, and they do not rely on M. This observation manifests the peculiar property of quantum information transmission via teleportation: entanglement can be built before the quantum data arrive so that the delay can be significantly reduced. Note that the scenario with $\alpha < 1$ is common in practice since it is akin to the case that the arrival rate is lower than the processing rate in classical queuing problems.

- If $p = q$, the expected average queue length is approximately inversely proportional as a function of M. If $p > q$, the expected average queue length converges to Q_t (for sufficiently large Q_t). These two scenarios correspond to the uncommon case for which the arrival rate is no less than the processing rate.

One can also verify that the "blocking probability," i.e., the probability that the quantum data are dropped because the queue is full, is

$$
\begin{aligned}
&\mathbb{P}\{\text{blocking of quantum data}\} \\
&\quad = (1-q) \lim_{n\to\infty} \mathbb{P}\{\mathsf{s}^{(n)} = Q_t | \mathsf{s}^{(1)} = s^{(1)}\} \\
&\quad = \begin{cases} \dfrac{(1-q)\alpha^{Q_t+M}(1-\alpha)}{1 - \alpha^{Q_t+M+1}} & \text{if } p \neq q \\[2ex] \dfrac{1-q}{Q_t+M+1} & \text{if } p = q. \end{cases}
\end{aligned}
$$

It is straightforward to see that this probability is a decreasing function of Q_t, whereas the expected average queue length is

an increasing function of Q_t. One may optimize over Q_t to achieve a desirable tradeoff between the blocking probability and the expected average queue length.

Before finishing the analysis of single-receiver scenario, we would like to discuss the scenario where the discarded qubits are required for retransmission, referred to as *retransmission model*. In this model, the assumption that $\mathsf{a}^{(n)}$ are i.i.d. may not hold. However, one can show that the expected average queue length under the retransmission model is lower-bounded by $J_L(\boldsymbol{x}^{(1)})$ shown in Proposition 1 and upper-bounded by

$$
\overline{J}_{\mathrm{L,retra}}(\boldsymbol{x}^{(1)}) = \begin{cases} \dfrac{\alpha^{M+1}}{1-\alpha} & \text{if } p \neq q \\[2ex] \infty & \text{if } p = q \end{cases}
$$

where $\overline{J}_{\mathrm{L,retra}}(\boldsymbol{x}^{(1)})$ is obtained by letting Q_t go to infinity in $J_L(\boldsymbol{x}^{(1)})$. Note that if $p < q$, both $J_L(\boldsymbol{x}^{(1)})$ and $\overline{J}_{\mathrm{L,retra}}(\boldsymbol{x}^{(1)})$ decrease to zero exponentially with respect to the memory size M; hence, the expected average queue length under the retransmission model also decreases to zero exponentially with M.

IV. MULTIPLE-RECEIVER SCENARIO

In this section, we design the control policy $u^{(n)}$ for $N_r \geqslant 2$. Recall that $u^{(n)}$ is a function that maps $\boldsymbol{x}^{(n)}$ into $\boldsymbol{s}^{(n+1)}$. We first consider the scenario where $N_r = 2$, and extend the analysis to the scenario where $N_r > 2$.

A. The Two-Receiver Scenario

In this subsection, $N_r = 2$. Recall that $\boldsymbol{s}^{(n)}$ represents the numbers of qubits in the system at the beginning of the time slot t_n, and $\boldsymbol{c}^{(n)}$ represents the difference between the numbers of qubits arriving at the node and the numbers of entangled qubit pairs generated in the first phase of the time slot t_n. For certain values of $\boldsymbol{s}^{(n)}$ and $\boldsymbol{c}^{(n)}$, the optimal control $u^{(n)}$ is trivial: in the third phase, the quantum node simply moves the remaining entangled qubits in the platform to the entanglement memory as long as the memory does not reach its full capacity. Specifically, recall that $u^{(n)}\left(\left[\,\boldsymbol{s}^{(n)}\ \boldsymbol{c}^{(n)}\,\right]^\mathrm{T}\right) = \boldsymbol{s}^{(n+1)}$ and

$$s_i^{(n+1)} = \min\{Q_t, s_i^{(n)} + c_i^{(n)}\} \tag{5}$$

provided that

$$\sum_{i=1}^{2} \max\left\{0, -\left(s_i^{(n)} + c_i^{(n)}\right)\right\} \leqslant M.$$

In addition, one can verify that (5) also holds if $s_j^{(n)} + c_j^{(n)} \geqslant 0$, $j = 1$ or 2.

We now consider the scenarios where

$$\sum_{i=1}^{2} \max\left\{0, -\left(s_i^{(n)} + c_i^{(n)}\right)\right\} > M \tag{6}$$

$$s_i^{(n)} + c_i^{(n)} < 0, \quad i = 1, 2. \tag{7}$$

Note that in these scenarios, $c_i^{(n)} \leqslant 0$ $(i = 1, 2)$ since $\sum_{i=1}^{2} \max\{0, -s_i^{(n)}\} \leqslant M$. The next theorem shows that the optimal control is a threshold-based policy for these scenarios.

Algorithm 1 Control for the Two-Receiver Scenario

Input: the current state $(s_1^{(n)}, s_2^{(n)}, c_1^{(n)}, c_2^{(n)})$, the threshold T, the memory size M, the upper bound of the queue length Q_t

Output: the qubit number in the system $s_1^{(n+1)}, s_2^{(n+1)}$ at the next time slot

1: **if** $s_i^{(n)} + c_i^{(n)} \geqslant 0$, $i \in \{1, 2\}$ or $\sum_{i=1}^{2} \max \left\{ 0, -\left(s_i^{(n)} + c_i^{(n)}\right) \right\} \leqslant M$ **then**

2:     // Control is trivial in these scenarios

3:     **for** $i = 1 : 2$ **do**

4:         $s_i^{(n+1)} = \min\{Q_t, s_i^{(n)} + c_i^{(n)}\}$;

5:     **end for**

6: **else**

7:     // Entanglement memory management

$$s_1^{(n+1)} = \begin{cases} s_1^{(n)} + c_1^{(n)} & \text{if } s_1^{(n)} + c_1^{(n)} \geqslant -T \\ -M - \left(s_2^{(n)} + c_2^{(n)}\right) & \text{otherwise} \end{cases}$$

and

$$s_2^{(n+1)} = \begin{cases} -M - \left(s_1^{(n)} + c_1^{(n)}\right) & \text{if } s_1^{(n)} + c_1^{(n)} \geqslant -T \\ s_2^{(n)} + c_2^{(n)} & \text{otherwise.} \end{cases}$$

8: **end if**

Theorem 1: For $N_r = 2$, if (6) and (7) hold, there exists $T^* \in \{0, -1, -2, \ldots, -M\}$ such that the optimal control u^* for minimizing the expected average queue length gives

$$s_1^{(n+1)} = \begin{cases} s_1^{(n)} + c_1^{(n)} & \text{for } s_1^{(n)} + c_1^{(n)} \geqslant T^* \\ -M - \left(s_2^{(n)} + c_2^{(n)}\right) & \text{otherwise} \end{cases}$$

and

$$s_2^{(n+1)} = \begin{cases} -M - \left(s_1^{(n)} + c_1^{(n)}\right) & \text{for } s_1^{(n)} + c_1^{(n)} \geqslant T^* \\ s_2^{(n)} + c_2^{(n)} & \text{otherwise.} \end{cases}$$

Proof: See Appendix A and therein Lemmas 1-4. □

Remark 1: Given a threshold T, we develop the corresponding control policy for $N_r = 2$, summarized in Algorithm 1. One way to interpret the entanglement memory management in Algorithm 1 is as follows. We assign $M_1 = T$ and $M_2 = M - T$ slots of memory budget to the entangled qubit pairs corresponding to the Receivers 1 and 2, respectively. When the budget of Receiver i is not used up, the entangled qubits corresponding Receiver i have a higher priority of using the memory: the entangled qubits are moved to the memory, even at the cost of discarding the entangled qubits corresponding to the other receiver if the memory is full.

This structure of memory use is akin to the spectrum use in cognitive radio. Consider two virtual sets of entanglement memory slots, denoted as $\mathcal{R}_1$ and $\mathcal{R}_2$ with $\mathrm{Card}(\mathcal{R}_1) = M_1$ and $\mathrm{Card}(\mathcal{R}_2) = M_2$. The entangled qubits associated with Receiver i have a higher priority on the usage of the memory set $\mathcal{R}_i$, and we can exploit the memory set of the other Receiver j as long as $\mathcal{R}_j$ is not full. In other words, the entangled qubits associated with Receiver i are akin to the primary user regarding the usage of $\mathcal{R}_i$; and they are akin to

the secondary user regarding the usage of $\mathcal{R}_j$. This idea is referred to as the cognitive memory and can be used for the policy design in general scenarios where $N_r > 2$.

Remark 2: Algorithm 1 requires the value of the threshold T as input. It remains unclear how to determine the optimal value threshold T^*, but in the next subsection we will provide an upper bound for the expected average queue length of Algorithm 1 as a function of T, which can guide the choice of T.

B. Harnessing Cognitive Memory

For the design of the control policy in the scenario with $N_r > 2$, one can use the value iteration method or the policy iteration method, which is commonly used for solving the dynamic programming problem [47]. However, these methods require computing the values of certain functions corresponding to every state in the system in which the number of states increases in the order of $O(M^{N_r})$. Therefore, these methods become computationally unfavorable for large N_r.

Alternatively, inspired by the idea of cognitive memory, we assign M_i slots of memory budget to the entangled qubits corresponding to Receiver i, $i \in \mathcal{K}_{1:N_r}$, where

$$M_i \in \mathbb{N}^*, \quad i \in N_r \tag{8}$$

$$\sum_{i=1}^{N_r} M_i \leqslant M. \tag{9}$$

Similarly to Algorithm 1, we design control policies for $N_r > 2$, summarized in Algorithm 2. The control policy based on the output of Algorithm 2 is referred to as the cognitive-memory-based policy. In particular, the threshold-based policy for $N_r = 2$ can be viewed as a special case of the cognitive-memory-based policy.

A subroutine in the line 12 of Algorithm 2 determines how to discard the entangled qubits that use up their memory budgets. Note that this subroutine can be designed heuristically. Later in Section V, we will see that even a simple design of the subroutine can achieve near-optimal performance. The performance of Algorithm 2 relies on the subroutine. However, we can provide an upper bound for the expected average queue length regardless of the design of the subroutine.

Proposition 2: The expected average queue length achieved by Algorithm 2 is upper bounded by

$$
\begin{aligned}
&J(M_1, M_2, \ldots, M_{N_r}) \\
&= \sum_{i=1}^{N_r} \left[(1 - \delta_{p_i, q_i}) \frac{\alpha_i^{M_i} \left[\alpha_i - \alpha_i^{Q_t+1} - (1 - \alpha_i) Q_t \alpha_i^{Q_t+1} \right]}{(1 - \alpha_i)(1 - \alpha_i^{Q_t + M_i + 1})} \right. \\
&\qquad \left. + \delta_{p_i, q_i} \frac{Q_t(Q_t + 1)}{2(Q_t + M_i)} \right]
\end{aligned} \tag{10}
$$

where $\delta_{x,y}$ is the Kronecker delta function and

$$\alpha_i = \frac{p_i(1 - q_i)}{q_i(1 - p_i)}.$$

Proof: The set $\mathcal{K}_{1:N_r} \backslash \mathcal{D}$ represents the set of receivers whose corresponding entangled qubits use up the memory budget. The subroutine in the line 12 aims to determine $s_i^{(n+1)}$, $i \in \mathcal{K}_{1:N_r} \backslash \mathcal{D}$. From the way that $\mathcal{D}$ is generated, we have that

---

Algorithm 2 Control for the N_r-Receiver Scenario

Input: the current state $s_i^{(n)}$, $c_i^{(n)}$, the budget M_i, the probabilities p_i and q_i, $i \in \mathcal{K}_{1:N_\mathrm{r}}$, the memory size M, and the upper bound of the queue length Q_t

Output: the qubit number in the system $s_i^{(n+1)}$, $i \in \mathcal{K}_{1:N_\mathrm{r}}$ at the next time slot

1: Let $\mathcal{D} = \varnothing$;
2: **if** $\sum_{i=1}^{N_\mathrm{r}} \max\left\{0, -\left(s_i^{(n)} + c_i^{(n)}\right)\right\} \leqslant M$ **then**
3:    $s_i^{(n+1)} = \min\{Q_\mathrm{t}, s_i^{(n)} + c_i^{(n)}\}$;
4: **else**
5:    **for** $i = 1 : N_\mathrm{r}$ **do**
6:       **if** $s_i^{(n)} + c_i^{(n)} \geqslant -M_i$ **then**
7:          $s_i^{(n+1)} = \min\{Q_\mathrm{t}, s_i^{(n)} + c_i^{(n)}\}$;
8:          $\mathcal{D} \to \mathcal{D} \cup \{i\}$;
9:       **end if**
10:   **end for**
11:   **if** $\mathrm{Card}(\mathcal{D}) \neq N_\mathrm{r}$ **then**
12:      Call a subroutine to determine $s_j^{(n+1)}$, $j \in \mathcal{K}_{1:N_\mathrm{r}} \backslash \mathcal{D}$;
13:   **end if**
14: **end if**

$s_i^{(n)} = -M_i$ and $c_i^{(n)} = -1$, $i \in \mathcal{K}_{1:N_\mathrm{r}} \backslash \mathcal{D}$. Consider a naive algorithm that discards the entangled qubits corresponding to Receiver $i \in \mathcal{K}_{1:N_\mathrm{r}} \backslash \mathcal{D}$ in the platform, i.e.,

$$s_i^{(n+1)} = -M_i, \quad i \in \mathcal{K}_{1:N_\mathrm{r}} \backslash \mathcal{D}.$$

This algorithm has the worst performance among all possible designs of subroutines. Using this algorithm, the queuing system is decomposed into N_r independent subsystems. Each subsystem i has a memory size M_i and the corresponding expected average queue length is derived in Proposition 1. The total expected average queue length is then given by Proposition 2. □

Next, consider the choice of M_i for minimizing the expected average queue length. Since the expected average queue length depends on the design of subroutine, we use the upper bound $J(M_1, M_2, \ldots, M_{N_\mathrm{r}})$ as the objective and formulate an optimization problem as follows:

$$\mathscr{P} : \minimize_{\{M_i, i \in \mathcal{K}_{1:N_\mathrm{r}}\}} J(M_1, M_2, \ldots, M_{N_\mathrm{r}})$$
$$\text{subject to} \quad (8) \text{ and } (9).$$

Note that $\mathscr{P}$ is an integer programming problem and solving $\mathscr{P}$ may be computationally cumbersome.[8] Therefore, we relax the constraints (8) and (9) to $M_i \geqslant 0$, $i \in N_\mathrm{r}$, and $\sum_{i=1}^{N_\mathrm{r}} M_i \leqslant M$, respectively. The effect brought by such relaxation becomes negligible for large M. Moreover, we consider the scenarios where the following conditions hold:

$$p_i < q_i, \quad i \in N_\mathrm{r} \tag{11}$$
$$\alpha^{Q_\mathrm{t} + M_i} \ll 1, \quad i \in N_\mathrm{r}. \tag{12}$$

The condition (11) is consistent with the discussion in Section III and the condition (12) is reasonable for achieving

---

---

the low blocking probability shown in Section III. With these two conditions, the objective function of $\mathscr{P}$ can be approximated by

$$\sum_{i=1}^{N_\mathrm{r}} \frac{\alpha_i^{M_i}\left[\alpha_i - \alpha_i^{Q_\mathrm{t}+1} - (1-\alpha_i)Q_\mathrm{t}\alpha_i^{Q_\mathrm{t}+1}\right]}{1 - \alpha_i}.$$

Using this approximated term as the objective function and the relaxation of the constraints on M_i from integers to real numbers, we obtain the following relaxed optimization problem:

$$\mathscr{P}_\mathrm{R} : \minimize_{\{M_i, i \in \mathcal{K}_{1:N_\mathrm{r}}\}} \quad \sum_{i=1}^{N_\mathrm{r}} \lambda_i \alpha_i^{M_i}$$
$$\text{subject to} \quad M_i \geqslant 0, \quad i \in N_\mathrm{r}$$
$$\sum_{i=1}^{N_\mathrm{r}} M_i \leqslant M$$

where

$$\lambda_i = \frac{\alpha_i - \alpha_i^{Q_\mathrm{t}+1} - (1-\alpha_i)Q_\mathrm{t}\alpha_i^{Q_\mathrm{t}+1}}{1 - \alpha_i}.$$

One can easily verify that $\lambda_i \geqslant 0$; moreover, the objective function is convex and the constraints are linear. Therefore, $\mathscr{P}_\mathrm{R}$ can be solved by standard convex programming. We next show that, by checking the Karush-Kuhn-Tucker conditions, we can obtain a closed-form solution for the problem $\mathscr{P}_\mathrm{R}$ under some mild assumptions.

Proposition 3: For

$$\mu := -\exp\left\{\frac{1}{\sum_{i=1}^{N_\mathrm{r}}(1/\log\alpha_i)}\left[M + \sum_{i=1}^{N_\mathrm{r}} \frac{\log(-\lambda_i \log\alpha_i)}{\log\alpha_i}\right]\right\}$$
$$\geqslant \lambda_i \log\alpha_i, \quad i \in \mathcal{K}_{1:N_\mathrm{r}}$$

the optimal solution for $\mathscr{P}_\mathrm{R}$ is

$$M_i^* = \frac{1}{\log\alpha_i} \log\left(\frac{\mu}{\lambda_i \log\alpha_i}\right).$$

Remark 3: The condition in Proposition 3 holds for a large M. In such a scenario, the minimum of the objective function is

$$\mu \sum_{i=1}^{N_\mathrm{r}} \frac{1}{\log\alpha_i}$$

which decreases to zero exponentially as a function of M and the exponent is $1/(\sum_{i=1}^{N_\mathrm{r}} 1/\log\alpha_i)$. This shows that for $N_\mathrm{r} > 1$, the delay can be significantly decreased with entanglement built before the quantum data arrive. Such an observation is consistent with the one shown in Section III, where $N_\mathrm{r} = 1$.

V. NUMERICAL RESULTS

This section illustrates the performance of the proposed policies through numerical results. The probabilities are set as $q_i = 0.05$, $i \in \mathcal{K}_{1:N_\mathrm{r}}$. Here q_i is chosen according to the analysis of the practical implementation in Section II-C.

Fig. 4. Expected average queue length as a function of the memory size M with $N_r = 2$.
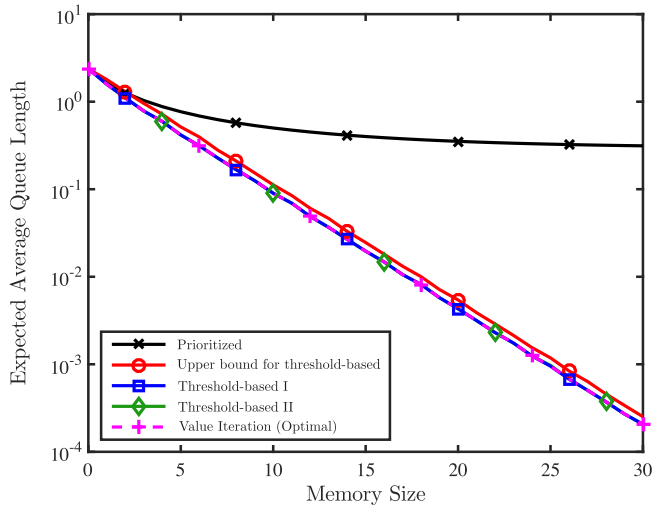

Fig. 5. Expected average queue length as a function of the memory size M with $N_r = 3$.

A. Two-Receiver Scenario

We first evaluate the expected average queue length in the scenario with two receivers. As an example, consider $p_1 = 0.5 \, q_1$ and $p_2 = 0.6 \, q_2$. We compare the following four policies.

- Value Iteration (Optimal): the policy obtained by value-iteration [40], which is also the optimal policy.
- Threshold-based I: the policy shown in Algorithm 1, with the optimal choice of the threshold T. The optimal choice is obtained by evaluating the performance of all choices of the threshold.
- Threshold-based II: the policy shown in Algorithm 1, with the choice of threshold T by solving the problem $\mathscr{P}$.
- Prioritized: the policy that gives priority to discarding the entangled qubits of the receiver with a lower arrival rate when the memory is full.

Note that the prioritized policy serves as a baseline. The upper bound (10) for the threshold-based policy will also be shown for evaluating the performance of the policies.

Fig. 4 shows the expected average queue length of the Value Iteration policy, the Threshold-based policies, and the Prioritized policy, as well as the upper bound for the performance of the threshold-based policy as a function of the memory size. First, the curves associated with the Value Iteration policy, the Threshold-based policies, and the upper bound demonstrate exponential decrease as a function of the memory size M. This observation is consistent with the analysis in Section IV-B and further validates the insight into QQD: by building and storing entanglement before the quantum data arrive, the queuing delay can be significantly reduced and such reduction largely relies on the memory size. Second, Threshold-based Policy I achieves the same performance as the optimal policy. This observation is consistent with Theorem 1, which shows that the Value Iteration policy is threshold-based. Third, the benefit of the Value Iteration and the Threshold-based policy is evident. For example, the expected average queue length is 0.4143 for the Prioritized policy when $M = 14$, whereas it is 0.0268 and 0.0268 for Threshold-based Policy I and Threshold-based
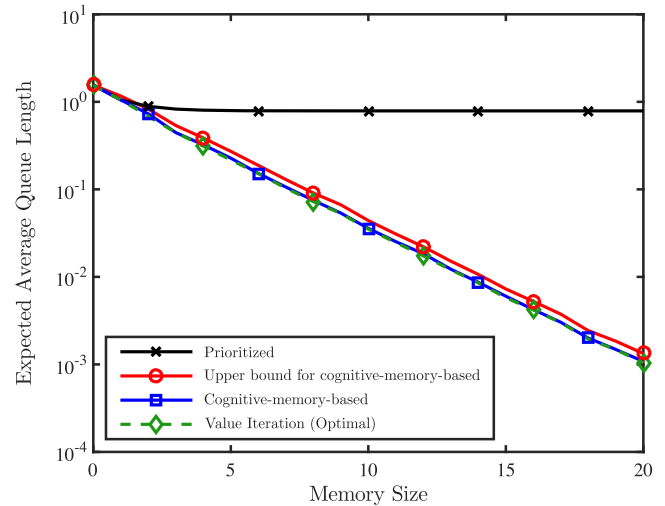
Policy II, respectively. This corresponds to an average queue length reduction of 93.5%. Fourth, the curve associated with the upper bound is close to the one associated with the Value Iteration policy, showing that the upper bound is tight.

B. Multi-Receiver Scenario

We next evaluate the expected average queue length in the scenario with multiple receivers. In particular, we consider that $N_r = 3$ and $p_1 = 0.3 \, q_1$, $p_2 = 0.35 \, q_2$, $p_3 = 0.4 \, q_3$.[9] We compare the following three policies.

- Value Iteration: the policy obtained by value-iteration [40], which is also the optimal policy.
- Cognitive-memory-based: the policy shown in Algorithm 2, with the choice of M_i by solving the problem $\mathscr{P}_R$. The subroutine in Algorithm 2 gives priority to discarding the entangled qubits of the receiver with a lower arrival rate.
- Prioritized: the policy that gives priority to discarding the entangled qubits of the receiver with a lower arrival rate when the memory is full.

Note that the prioritized policy serves as a baseline. The upper bound (10) will also be shown for evaluating the performance of the policies.

Fig. 5 shows the expected average queue length of the Value Iteration policy, the Cognitive-memory-based policy, and the Prioritized policy, as well as the upper bound for the performance of the Cognitive-memory-based policy as a function of the memory size. First, the curves associated with the Value Iteration policy, the Cognitive-memory-based policy, and the upper bound demonstrate exponential decrease as a function of the memory size M. Again, this observation shows that the queuing delay can be significantly reduced and such reduction largely relies on the memory size. Second, the benefit of the Value Iteration and the Cognitive-memory-based policy is evident. For example, the expected

[9]For large N_r, the optimal policy is computationally cumbersome as mentioned in Section IV-B. Therefore, we choose a moderate value of N_r in this section.
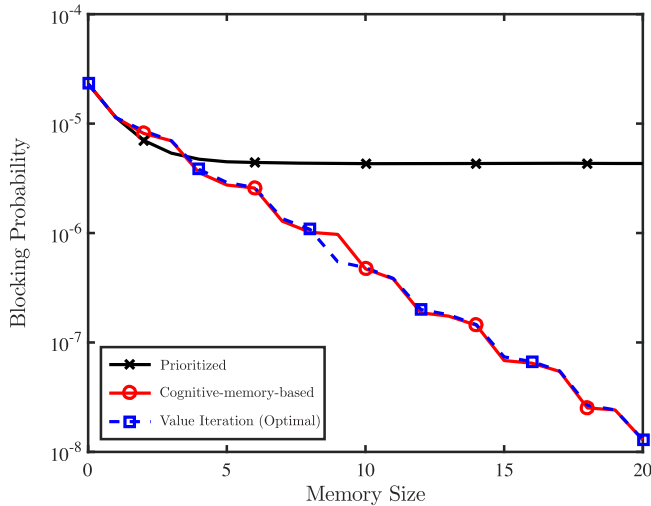
Fig. 6. Blocking probability as a function of the memory size M with $N_r = 3$.

average queue length is 0.7644 for the Prioritized policy when $M = 10$, whereas it is 0.0738 and 0.0815 for the Value Iteration policy and the Cognitive-memory-based policy, respectively. This corresponds to expected average queue length reductions of 90.4% and 89.3%, respectively. This observation is consistent with the two-receiver scenario and Proposition 2. Third, the curve associated with the upper bound is close to the one associated with the Value Iteration policy, showing that the upper bound is tight.

Next, we evaluate the blocking probability in the scenario with multiple receivers. Recall that the blocking probability is the probability that the quantum data are dropped because the queue is full. Fig. 6 shows the blocking probability of the Value Iteration policy, Cognitive-memory-based policy, and the Prioritized policy as a function of the memory size. The blocking probability converges to zero quickly a function of the memory size M for all the policies. This is consistent with the analysis in Section III and shows that almost none of the quantum data are dropped in the considered scenario. Together with Little's law, this shows that the expected average queuing delay is proportional to the expected average queue length, which justifies the use of expected average queue length as the objective function.

VI. CONCLUSION

In this paper, we introduce a formalism to analyze QQD. The main methodologies used in this paper are dynamic programming and stochastic processes. We develop a cognitive-memory-based policy and show that this policy is optimal for the scenario with two receivers. For a general scenario, we derive an upper bound for the expected average queue length achieved by the cognitive-memory-based policy. With this upper bound, we show that the expected average queuing delay can decrease exponentially with respect to the memory size. The performance of the proposed policies are evaluated in practical scenarios. Numerical results verify the optimality/near-optimality of the policies and show that the developed upper bound is tight.

A key insight is that, unlike classical queuing delay, QQD can be significantly reduced by establishing entanglements before the quantum data arrive. The exponential decrease of QQD with respect to the memory size shows that a moderate memory size suffices to achieve a near-zero QQD. The results of this paper enable a deeper understanding of quantum information science and pave the way of operation design in quantum networks.

APPENDIX A
A PROOF OF THEOREM 1

The proof for Theorem 1 is organized in five steps.

Step 1 Connection to the Blackwell Optimal Policy:

We first introduce the $\beta-$discounted dynamic programming problem in the context of quantum queuing system: Given an initial state $x^{(1)}$ and $\beta \in (0,1)$, the aim is to find a policy $\pi = \{u^{(0)}, u^{(1)}, \ldots\}$, that minimizes the cost function

$$J_\beta(x^{(1)}) = \limsup_{N \to \infty} \mathbb{E}\left\{\sum_{n=0}^{N-1}\sum_{i=1}^{N_r} \beta^n \max\{0, s_i^{(n)}\} \Big| x^{(1)}\right\}$$

subject to the system equation constraint (1).

The definition of Blackwell optimal policy is described below.

Definition 1 ([47], Chapter 4): A stationary policy μ is said to be Blackwell optimal if it is simultaneously optimal for all the $\beta-$discounted problems with β in an interval $(\beta^*, 1)$, where β^* is some scalar with $0 < \beta^* < 1$.

Lemma 1 ([47], Proposition 4.1.3): There exists a Blackwell optimal policy.

Lemma 2 ([47], Proposition 4.1.7): A Blackwell policy is optimal over all policies in the average cost problem.

Based on Lemma 1, we can consider a Blackwell optimal policy $u^{*(n)}$. Noting that the Blackwell optimal policy is stationary, i.e., independent on the time index n, we can drop the superscript (n) and write the optimal control as u^*. By Definition 1 and Lemma 1, there exists $\beta^* \in (0,1)$ such that u^* is optimal for $\beta-$discounted problems with $\beta \in (\beta^*, 1)$. We next show that either u^* is threshold-based control as stated in Theorem 1, or we can create threshold-based control that is also a Blackwell optimal policy.

Step 2 Introduction of Some Notations:

Since $N_r = 2$, $x^{(1)} \in \mathbb{R}^4$. With a little abuse of notation, we write J_β as a function of four variables, i.e.,

$$J_\beta(x) = J_\beta(s_1, s_2, c_1, c_2) \tag{13}$$

where $x = [s_1, s_2, c_1, c_2]^T$. We also introduce a function that represents the future cost

$$\widetilde{J}_\beta(s_1, s_2)$$
$$:= \mathbb{E}\{J_\beta(s_1, s_2, \mathsf{c}_1, \mathsf{c}_2)\}$$
$$= \sum_{c_1, c_2 \in \{-1, 0, 1\}} \mathbb{P}\{\mathsf{c}_1^{(1)} = c_1, \mathsf{c}_2^{(1)} = c_2\}\ J_\beta(s_1, s_2, c_1, c_2).$$

One can verify that

$$J_\beta(s_1, s_2, c_1, c_2) = \max\{0, s_1\} + \max\{0, s_2\}$$
$$+ \beta \widetilde{J}_\beta(u_\beta^*(s_1, s_2, c_1, c_2))$$

where u_β^* denotes the optimal control policy for the β-discounted problem.[10]

Step 3 Connection to the "convexity" of $\widetilde{J}_\beta$:

Note that for the $\beta-$discounted problem, for a state $[\,s_1, s_2, c_1, c_2\,]^{\mathrm{T}}$, if

$$\sum_{i=1}^{N_{\mathrm{r}}} \max\left\{0, -(s_i + c_i)\right\} > M$$

$$s_i + c_i < 0, \quad i = 1, 2$$

the optimal control is

$$[\tilde{s}_1, \tilde{s}_2]^{\mathrm{T}} = \underset{[\tilde{s}_1, \tilde{s}_2]^{\mathrm{T}} \in \mathcal{F}(s_1, s_2, c_1, c_2)}{\arg\min} \beta \mathbb{E}\{J_\beta(\tilde{s}_1, \tilde{s}_2, \mathsf{c}_1, \mathsf{c}_2)\}$$

$$= \underset{[\tilde{s}_1, \tilde{s}_2]^{\mathrm{T}} \in \mathcal{F}(s_1, s_2, c_1, c_2)}{\arg\min} \widetilde{J}_\beta(\tilde{s}_1, \tilde{s}_2)$$

where $\mathcal{F}(s_1, s_2, c_1, c_2)$ is the feasible set, given by

$$\mathcal{F}(s_1, s_2, c_1, c_2) := \left\{[x, y]^{\mathrm{T}} \in \mathbb{Z}^2 : x \geqslant s_1 + c_1, y \geqslant s_2 + c_2\right.$$
$$\left. x + y \geqslant -M\right\}.$$

Note that the feasible set has a physical meaning: a quantum node can discard established entangled qubits, but cannot store entangled qubits beyond the size of the entanglement memory.

In the remaining parts of the proof, we will show that $\widetilde{J}_\beta$ has the following property

$$2\widetilde{J}_\beta(s_1, s_2) \leqslant \widetilde{J}_\beta(s_1 + 1, s_2 - 1) + \widetilde{J}_\beta(s_1 - 1, s_2 + 1) \quad (14)$$

provided that $s_1 \leqslant -1$, $s_2 \leqslant -1$, and $s_1 + s_2 = -M$. The property (14) implies that $\widetilde{J}_\beta(s_1, -M - s_1)$ is discretely convex as a function of s_1 for $s_1 \in \mathcal{K}_{-M:0}$. One can easily verify that if (14) holds, then either u_β^* is threshold-based control or there exists threshold-based control that achieves the same performance as u_β^* (the latter occurs only when $\widetilde{J}_\beta(s_1, -M - s_1)$ first decreases, stays constant, and then increases as a function of s_1 when $s_1 \in \mathcal{K}_{-M:0}$).

We will prove a stronger claim than (14) as follows:

Lemma 3: There exists $\beta^* \in (0, 1)$ such that for $\beta \in (\beta^*, 1)$,

$$\widetilde{J}_\beta(s_1, s_2 - 1) - \widetilde{J}_\beta(s_1 - 1, s_2)$$
$$\leqslant \widetilde{J}_\beta(s_1, s_2 - 2) - \widetilde{J}_\beta(s_1 - 1, s_2 - 1) \quad (15)$$
$$\widetilde{J}_\beta(s_1 - 1, s_2) - \widetilde{J}_\beta(s_1, s_2 - 1)$$
$$\leqslant \widetilde{J}_\beta(s_1 - 2, s_2) - \widetilde{J}_\beta(s_1 - 1, s_2 - 1) \quad (16)$$

provided that $s_1 \leqslant 1$, $s_2 \leqslant 1$, and s_1 and s_2 are valid numbers.[11]

If (15) and (16) hold, then adding these two equations gives

$$2\widetilde{J}_\beta(s_1 - 1, s_2 - 1) \leqslant \widetilde{J}_\beta(s_1, s_2 - 2) + \widetilde{J}_\beta(s_1 - 2, s_2)$$

for $s_1 \leqslant 1$ and $s_2 \leqslant 1$, which is stronger than (14).

Step 4 Value Iteration of J_β and $\widetilde{J}_\beta$:

---

[10]Note that the optimal control policy for the β-discounted problem is stationary and does not rely on the time index n. Similarly to (13), we write u_β^* as a function of four variables with an output of two variables.

[11]In this paper, "valid numbers" are defined as numbers that lead to valid state inputs for the function $\widetilde{J}_\beta$.

We consider the value iteration of J_β and $\widetilde{J}_\beta$. In particular, consider

$$J_\beta^{(0)}(s_1, s_2, c_1, c_2) = \max\{0, s_1\} + \max\{0, s_2\}$$

and

$$J_\beta^{(k+1)}(s_1, s_2, c_1, c_2)$$
$$= \max\{0, s_1\} + \max\{0, s_2\}$$
$$+ \beta \min_{u(s_1, s_2, c_1, c_2) \in \mathcal{F}(s_1, s_2, c_1, c_2)} \mathbb{E}\{J_\beta^{(k)}(u(s_1, s_2, c_1, c_2), \mathsf{c}_1, \mathsf{c}_2)\}$$
$$= \max\{0, s_1\} + \max\{0, s_2\}$$
$$+ \beta \mathbb{E}\{J_\beta^{(k)}(u_k(s_1, s_2, c_1, c_2), \mathsf{c}_1, \mathsf{c}_2)\}$$

where u_k denotes the policy that achieves the minimum in the first equality. Correspondingly, consider

$$\widetilde{J}_\beta^{(0)}(s_1, s_2) = \mathbb{E}\{J_\beta^{(0)}(s_1, s_2, \mathsf{c}_1, \mathsf{c}_2)\}$$
$$= \max\{0, s_1\} + \max\{0, s_2\} \quad (17)$$

and

$$\widetilde{J}_\beta^{(k)}(s_1, s_2) = \mathbb{E}\{J_\beta^{(k)}(s_1, s_2, \mathsf{c}_1, \mathsf{c}_2)\}.$$

One can verify that the function $\widetilde{J}_\beta^{(k)}$ has the following iteration:

$$\widetilde{J}_\beta^{(k+1)}(s_1, s_2) = \max\{0, s_1\} + \max\{0, s_2\}$$
$$+ \beta \mathbb{E}\left\{\widetilde{J}_\beta^{(k)}\left(u_k(s_1, s_2, \mathsf{c}_1, \mathsf{c}_2)\right)\right\}.$$

Value iteration shows that $\lim_{k\to\infty} J_\beta^{(k)} = J_\beta$ [47]; consequently, $\lim_{k\to\infty} \widetilde{J}_\beta^{(k)} = \widetilde{J}_\beta$. Therefore, for proving Lemma 3, it suffices to prove that for all k,

$$F^{(k)}(s_1, s_2) \leqslant 0, \quad s_1 \leqslant Q_{\mathrm{t}}, \ s_2 \leqslant 0$$
$$G^{(k)}(s_1, s_2) \leqslant 0, \quad s_1 \leqslant 0, \quad s_2 \leqslant Q_{\mathrm{t}}$$

where

$$F^{(k)}(s_1, s_2) := \left[\widetilde{J}_\beta^{(k)}(s_1, s_2 - 1) - \widetilde{J}_\beta^{(k)}(s_1 - 1, s_2)\right]$$
$$- \left[\widetilde{J}_\beta^{(k)}(s_1, s_2 - 2) - \widetilde{J}_\beta^{(k)}(s_1 - 1, s_2 - 1)\right]$$
$$G^{(k)}(s_1, s_2) := \left[\widetilde{J}_\beta^{(k)}(s_1 - 1, s_2) - \widetilde{J}_\beta^{(k)}(s_1, s_2 - 1)\right]$$
$$- \left[\widetilde{J}_\beta^{(k)}(s_1 - 2, s_2) - \widetilde{J}_\beta^{(k)}(s_1 - 1, s_2 - 1)\right].$$

The proof will be shown in the next step.

Step 5 Induction Method to Prove Lemma 3:

We use the induction method to prove the following claims simultaneously:

Lemma 4: For all k, the following inequalities hold:

$$F^{(k)}(s_1, s_2) \leqslant 0, \qquad s_1 \leqslant Q_{\mathrm{t}}, \qquad s_2 \leqslant 0 \quad (18)$$
$$H^{(k)}(s_1, s_2) \leqslant 0, \qquad s_1 \leqslant Q_{\mathrm{t}}, \qquad s_2 \leqslant 0 \quad (19)$$
$$G^{(k)}(s_1, s_2) \leqslant 0, \qquad s_1 \leqslant 0, \qquad s_2 \leqslant Q_{\mathrm{t}} \quad (20)$$
$$I^{(k)}(s_1, s_2) \leqslant 0, \qquad s_1 \leqslant 0, \qquad s_2 \leqslant Q_{\mathrm{t}} \quad (21)$$
$$F^{(k)}(s_1, s_2) \leqslant \eta_{s_2}^{(k)}, \qquad s_1 \leqslant Q_{\mathrm{t}}, \qquad s_2 \geqslant 1 \quad (22)$$
$$H^{(k)}(s_1, s_2) \leqslant \eta_{s_2}^{(k)}, \qquad s_1 \leqslant Q_{\mathrm{t}}, \qquad s_2 \geqslant 1 \quad (23)$$
$$G^{(k)}(s_1, s_2) \leqslant \zeta_{s_1}^{(k)}, \qquad s_1 \geqslant 1, \qquad s_2 \leqslant Q_{\mathrm{t}} \quad (24)$$
$$I^{(k)}(s_1, s_2) \leqslant \zeta_{s_1}^{(k)}, \qquad s_1 \geqslant 1, \qquad s_2 \leqslant Q_{\mathrm{t}} \quad (25)$$

$$\widetilde{J}_\beta^{(k)}(s_1, s_2) - \widetilde{J}_\beta^{(k)}(s_1, s_2 - 1) \leqslant \nu_{s_2}^{(k)}, \quad s_1 \leqslant Q_\text{t}, \ s_2 \geqslant 1$$
$$(26)$$

$$\widetilde{J}_\beta^{(k)}(s_1, s_2) - \widetilde{J}_\beta^{(k)}(s_1 - 1, s_2) \leqslant \upsilon_{s_1}^{(k)}, \quad s_1 \geqslant 1, \ s_2 \leqslant Q_\text{t}$$
$$(27)$$

where

$$H^{(k)}(s_1, s_2) := \big[\widetilde{J}_\beta^{(k)}(s_1, s_2 - 1) - \widetilde{J}_\beta^{(k)}(s_1, s_2)\big]$$
$$- \big[\widetilde{J}_\beta^{(k)}(s_1, s_2 - 2) - \widetilde{J}_\beta^{(k)}(s_1, s_2 - 1)\big]$$

$$I^{(k)}(s_1, s_2) := \big[\widetilde{J}_\beta^{(k)}(s_1 - 1, s_2) - \widetilde{J}_\beta^{(k)}(s_1, s_2)\big]$$
$$- \big[\widetilde{J}_\beta^{(k)}(s_1 - 2, s_2) - \widetilde{J}_\beta^{(k)}(s_1 - 1, s_2)\big].$$

Moreover, $\eta_i^{(k)}$ and $\nu_i^{(k)}$ are auxiliary sequences given by

$$\eta_i^{(0)} = \eta_\beta^*(i), \quad i \in \mathcal{K}_{1:Q_\text{t}}$$
$$\eta_1^{(k)} = -1 + \beta\big[\tilde{p}_2\, \eta_2^{(k-1)} + (1 - \tilde{p}_2 - \tilde{q}_2)\, \eta_1^{(k-1)}\big]$$
$$\eta_i^{(k)} = \beta\big[\tilde{p}_2\, \eta_{i+1}^{(k-1)} + (1 - \tilde{p}_2 - \tilde{q}_2)\, \eta_i^{(k-1)} + \tilde{q}_2\, \eta_{i-1}^{(k-1)}\big],$$
$$i \in \mathcal{K}_{2:Q_\text{t}-1}$$
$$\eta_{Q_\text{t}}^{(k)} = \beta\big[\tilde{p}_2\, \nu_{Q_\text{t}}^{(k-1)} + (1 - \tilde{p}_2 - \tilde{q}_2)\, \eta_{Q_\text{t}}^{(k-1)} + \tilde{q}_2\, \eta_{Q_\text{t}-1}^{(k-1)}\big]$$

and

$$\nu_i^{(0)} = \nu_\beta^*(i), \quad i \in \mathcal{K}_{1:Q_\text{t}}$$
$$\nu_1^{(k)} = 1 + \beta\big[\tilde{p}_2\, \nu_2^{(k-1)} + (1 - \tilde{p}_2)\, \nu_1^{(k-1)}\big]$$
$$\nu_i^{(k)} = 1 + \beta\big[\tilde{p}_2\, \nu_{i+1}^{(k-1)} + (1 - \tilde{p}_2 - \tilde{q}_2)\, \nu_i^{(k-1)} + \tilde{q}_2\, \nu_{i-1}^{(k-1)}\big],$$
$$i \in \mathcal{K}_{2:Q_\text{t}}$$

where $\tilde{p}_2 = \mathbb{P}\{\mathsf{c}_2^{(1)} = 1\}$, $\tilde{q}_2 = \mathbb{P}\{\mathsf{c}_2^{(1)} = -1\}$; $\eta_\beta^*(i)$ and $\nu_\beta^*(i)$ are steady states of the system above, i.e., $\eta_\beta^*(i) = \lim_{k\to\infty} \eta_i^{(k)}$ and $\nu_\beta^*(i) = \lim_{k\to\infty} \nu_i^{(k)}$;[12] and $\zeta_i^{(k)}$ and $\upsilon_i^{(k)}$ are obtained by replacing η, ν, $\tilde{p}_2$, $\tilde{q}_2$ with ζ, υ, $\tilde{p}_1$, $\tilde{q}_1$, respectively.

Proof: See Appendix B. $\qquad\square$

The completion of proving Lemma 4 finishes the proof of Theorem 1.

APPENDIX B
A PROOF OF LEMMA 4

We first prove a lemma that will be used in the later proof.
Lemma 5: There exists β^* such that for any $\beta \in (\beta^*, 1)$

$$\eta_\beta^*(1) \in (-1, 0)$$
$$\eta_\beta^*(i) > 0, \quad i \neq 0$$
$$\nu_\beta^*(i) > 0.$$

Proof: Denote $Q_\text{t} \times Q_\text{t}$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ as follows:

$$\boldsymbol{A} = \begin{bmatrix} 1 - \tilde{p}_2 & \tilde{p}_2 & & & & \\ \tilde{q}_2 & 1 - \tilde{p}_2 - \tilde{q}_2 & \tilde{p}_2 & & & \\ & \tilde{q}_2 & 1 - \tilde{p}_2 - \tilde{q}_2 & \tilde{p}_2 & & \\ & & & \ddots & & \\ & & & & \tilde{q}_2 & 1 - \tilde{p}_2 - \tilde{q}_2 \end{bmatrix}$$

[12]The expressions of $\eta_\beta^*(i)$ and $\nu_\beta^*(i)$ will be explicitly given in Appendix B.

and

$$\boldsymbol{B} = \boldsymbol{A} - \tilde{q}_2 \boldsymbol{E}_{1,1}.$$

Consider the vectors $\boldsymbol{\nu}_\beta^*$ and $\boldsymbol{\eta}_\beta^*$ that represent steady states, i.e., $[\boldsymbol{\nu}_\beta^*]_i = \nu_\beta^*(i)$ and $[\boldsymbol{\eta}_\beta^*]_i = \eta_\beta^*(i)$.

The steady states of $\nu_k^{(i)}$ and $\nu_k^{(i)}$ are given by

$$\boldsymbol{\nu}_\beta^* = (\boldsymbol{I} - \beta\boldsymbol{A})^{-1}\boldsymbol{1}$$
$$\boldsymbol{\eta}_\beta^* = (\boldsymbol{I} - \beta\boldsymbol{B})^{-1}\big(-\boldsymbol{e}_1 + \beta\tilde{p}_2\boldsymbol{E}_{Q_\text{t},Q_\text{t}}\boldsymbol{\nu}_\beta^*\big).$$

One can easily verify that $\boldsymbol{I} - \beta\boldsymbol{A}$ and $\boldsymbol{I} - \beta\boldsymbol{B}$ are invertible for $\beta \in (0, 1]$.

Note that for $\beta = 1$,

$$[\boldsymbol{\nu}_\beta^*]_i = \sum_{m=i}^{Q_\text{t}} \sum_{n=0}^{m-1} \frac{\tilde{p}_2^n \tilde{q}_2^{m-1-n}}{\tilde{p}_2^m} > 0. \qquad (28)$$

Moreover, for $\beta = 1$,

$$[\boldsymbol{\eta}_\beta^*]_i = \frac{1}{\sum_{m=1}^{Q_\text{t}} \tilde{p}_2^m \tilde{q}_2^{Q_\text{t}-m}}\Big(-\sum_{m=i-1}^{Q_\text{t}-1} \tilde{p}_2^{Q_\text{t}-1-m} \tilde{q}_2^m$$
$$+ \frac{\sum_{m=0}^{Q_\text{t}-1} \tilde{p}_2^m \tilde{q}_2^{Q_\text{t}-1-m}}{\tilde{p}_2^{Q_\text{t}-1}} \sum_{m=0}^{i-1} \tilde{p}_2^{Q_\text{t}-1-m} \tilde{q}_2^m\Big).$$

It is straightforward to verify that

$$[\boldsymbol{\eta}_\beta^*]_1 = 0 \quad \text{and} \quad [\boldsymbol{\eta}_\beta^*]_i > 0, \quad i > 1. \qquad (29)$$

Moreover,

$$\frac{\partial\boldsymbol{\eta}^*}{\partial\beta}\Big|_{\beta=1} = (\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{B}\boldsymbol{\eta}_\beta^* + p_2(\boldsymbol{I} - \boldsymbol{B})^{-1}\boldsymbol{E}_{Q_\text{t},Q_\text{t}}\boldsymbol{\nu}_\beta^*\Big|_{\beta=1}.$$

Noticing that

$$(\boldsymbol{I} - \boldsymbol{B})^{-1} = \sum_{j=0}^\infty \boldsymbol{B}^j$$

one can see that each element $(\boldsymbol{I} - \boldsymbol{B})^{-1}$ is positive. As a consequence, one can easily derive that

$$\frac{\partial[\boldsymbol{\eta}^*]_i}{\partial\beta}\Big|_{\beta=1} > 0. \qquad (30)$$

Combining (28), (29), and (30), together with the continuity of $\boldsymbol{\eta}_\beta^*$ and $\boldsymbol{\nu}_\beta^*$ as functions of β, we arrive at the desired results.
$\qquad\square$

Lemma 6: The functions $\widetilde{J}_\beta^{(k)}(s_1, \cdot)$ and $\widetilde{J}_\beta^{(k)}(\cdot, s_2)$ are increasing functions for all valid s_1 and s_2.

The proof of such monotonicity is omitted due to space constraints.

We now prove Lemma 4 by induction. The base case can be easily verified with the initial values (17). We next consider the induction step. Suppose (19) to (27) hold for the case k, we next consider the case $k+1$. Note that adding (18) and (20) as well as adding (22) and (24) give

$$2\widetilde{J}_\beta^{(k)}(s_1 - 1, s_2 - 1)$$
$$\leqslant \widetilde{J}_\beta^{(k)}(s_1, s_2 - 2) + \widetilde{J}_\beta^{(k)}(s_1 - 2, s_2) \quad (31)$$

for $s_1 \leqslant 1$ and $s_2 \leqslant 1$.

Due to the space constraint, we only show the proof of (19) for the case $k + 1$. The other cases can be proved similarly.

For (19), we first show that for $c_1, c_2 \in \{-1, 0, 1\}$ and $s_1 \leqslant Q_{\mathrm{t}}, s_2 \leqslant 0$,

$$
2\widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 1, c_1, c_2)\big)
$$
$$
\leqslant \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 2, c_1, c_2)\big) + \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2, c_1, c_2)\big). \tag{32}
$$

Note that $\max\{0, -s_1\} + \max\{0, -(s_2 - 2)\} \leqslant M$, we have $\max\{0, -(s_1 + c_1)\} + \max\{0, -(s_2 - 1 + c_2)\} \leqslant M + 1$.

If $\max\{0, -(s_1 + c_1)\} + \max\{0, -(s_2 - 1 + c_2)\} \leqslant M$, then

$$
\widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 1, c_1, c_2)\big) - \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2, c_1, c_2)\big)
$$
$$
= \widetilde{J}_\beta^{(k)}(Q_{\mathrm{m}}, s_2 - 1 + c_2) - \widetilde{J}_\beta^{(k)}(Q_{\mathrm{m}}, s_2 + c_2)
$$
$$
\leqslant \min\Big\{ \widetilde{J}_\beta^{(k)}(Q_{\mathrm{m}}, \max\{s_2 - 2 + c_2, -M\}),
$$
$$
\widetilde{J}_\beta^{(k)}(s_1 + c_1 + 1, s_2 + c_2 - 2),
$$
$$
\widetilde{J}_\beta^{(k)}(s_1 + c_1, s_2 + c_2 - 1)\Big\}
$$
$$
- \widetilde{J}_\beta^{(k)}(Q_{\mathrm{m}}, s_2 - 1 + c_2) \tag{33}
$$
$$
= \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 2, c_1, c_2)\big) - \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 1, c_1, c_2)\big)
$$

where $Q_{\mathrm{m}} := \min\{Q_{\mathrm{t}}, s_1 + c_1\}$. The equalities are due to the definition of u_k and the assumption that $\max\{0, -(s_1 + c_1)\} + \max\{0, -(s_2 - 1 + c_2)\} \leqslant M$. The inequality can be verified using (18) and (19) for the case k and Lemma 6. For example, if $\widetilde{J}_\beta^{(k)}(s_1 + c_1 + 1, s_2 + c_2 - 2)$ achieves the minimum in (33), then $s_1 + c_1 + 1 \leqslant 0$ and

$$
\widetilde{J}_\beta^{(k)}(Q_{\mathrm{m}}, s_2 - 1 + c_2) - \widetilde{J}_\beta^{(k)}(Q_{\mathrm{m}}, s_2 + c_2)
$$
$$
= \widetilde{J}_\beta^{(k)}(s_1 + c_1, s_2 - 1 + c_2) - \widetilde{J}_\beta^{(k)}(s_1 + c_1, s_2 + c_2)
$$
$$
\leqslant \widetilde{J}_\beta^{(k)}(s_1 + c_1 + 1, s_2 - 1 + c_2) - \widetilde{J}_\beta^{(k)}(s_1 + c_1, s_2 + c_2)
$$
$$
\leqslant \widetilde{J}_\beta^{(k)}(s_1 + c_1 + 1, s_2 + c_2 - 2) - \widetilde{J}_\beta^{(k)}(s_1 + c_1, s_2 - 1 + c_2)
$$
$$
= \widetilde{J}_\beta^{(k)}(s_1 + c_1 + 1, s_2 + c_2 - 2) - \widetilde{J}_\beta^{(k)}(Q_{\mathrm{m}}, s_2 - 1 + c_2)
$$

where the first inequality is because of Lemma 6 and the second inequality is because of (18). If the minimum in (33) is achieved by other terms, the discussions are similar.

If $\max\{0, -(s_1 + c_1)\} + \max\{0, -(s_2 - 1 + c_2)\} = M + 1$, then $c_1 = -1$ and $s_1 \leqslant 0$. One can verify that $-s_1 > M$. As a consequence, then $c_2 = -1$, $s_2 + c_2 - 1 < 0$, and

$$
\widetilde{J}_\beta^{(k)}(u_k(s_1, s_2 - 1, c_1, c_2)) - \widetilde{J}_\beta^{(k)}(u_k(s_1, s_2, c_1, c_2))
$$
$$
\leqslant \min\Big\{ \widetilde{J}_\beta^{(k)}(s_1 - 1, s_2 - 1),
$$
$$
\widetilde{J}_\beta^{(k)}(s_1, s_2 - 2),
$$
$$
\widetilde{J}_\beta^{(k)}(s_1 + 1, s_2 - 3)\Big\}
$$
$$
- \min\Big\{ \widetilde{J}_\beta^{(k)}(s_1, s_2 - 2), \widetilde{J}_\beta^{(k)}(s_1 - 1, s_2 - 1)\Big\}
$$
$$
= \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 2, c_1, c_2)\big) - \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 1, c_1, c_2)\big)
$$

where the inequality can be verified by discussing all the cases and using (31), (32), and (19) for the case k, as well as Lemma 6. This proves (32).

We now prove (19).

$$
H^{(k+1)}(s_1, s_2)
$$
$$
= \Big[\widetilde{J}_\beta^{(k+1)}(s_1, s_2 - 1) - \widetilde{J}_\beta^{(k+1)}(s_1, s_2) \Big]
$$
$$
\quad - \Big[\widetilde{J}_\beta^{(k+1)}(s_1, s_2 - 2) - \widetilde{J}_\beta^{(k+1)}(s_1, s_2 - 1) \Big]
$$
$$
= 2\max\{0, s_2 - 1\} - \max\{0, s_2\} - \max\{0, s_2 - 2\}
$$
$$
\quad + 2\mathbb{E}\Big\{ \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 1, \mathsf{c}_1, \mathsf{c}_2)\big)\Big\}
$$
$$
\quad - \mathbb{E}\Big\{ \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2 - 2, \mathsf{c}_1, \mathsf{c}_2)\big)\Big\}
$$
$$
\quad - \mathbb{E}\Big\{ \widetilde{J}_\beta^{(k)}\big(u_k(s_1, s_2, \mathsf{c}_1, \mathsf{c}_2)\big)\Big\}
$$
$$
\leqslant \beta \sum_{c_1, c_2 \in \{-1, 0, +1\}} \mathbb{P}\{\mathsf{c}_1^{(1)} = c_1, \mathsf{c}_2^{(1)} = c_2\}
$$
$$
\quad \times \Big[J_\beta^{(k)}\big(u_k(s_1, s_2 - 1, c_1, c_2)\big) - J_\beta^{(k)}\big(u_k(s_1, s_2, c_1, c_2)\big)
$$
$$
\quad - J_\beta^{(k)}\big(u_k(s_1, s_2 - 2, c_1, c_2)\big) + J_\beta^{(k)}\big(u_k(s_1, s_2 - 1, c_1, c_2)\big)\Big]
$$
$$
\leqslant 0
$$

where the inequality is due to $2\max\{0, s_2 - 1\} - \max\{0, s_2\} - \max\{0, s_2 - 2\} \leqslant 0$ and (32). This proves (19) for the case $k + 1$.

REFERENCES

[1] J. P. Dowling and G. J. Milburn, "Quantum technology: The second quantum revolution," *Philos. Trans. Royal Soc. A, Math., Phys. Eng. Sci.*, vol. 361, no. 1809, pp. 1655–1674, Jun. 2003.

[2] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[3] *National Strategic Overview for Quantum Information Science*, Nat. Sci. Technol. Council, Washington, DC, USA, Sep. 2018.

[4] A. S. Fletcher, P. W. Shor, and M. Z. Win, "Channel-adapted quantum error correction for the amplitude damping channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5705–5718, Dec. 2008.

[5] N. Hosseinidehaj, Z. Babar, R. Malaney, S. X. Ng, and L. Hanzo, "Satellite-based continuous-variable quantum communications: State-of-the-art and a predictive outlook," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 881–919, 1st Quart., 2019.

[6] S. Guerrini, M. Chiani, M. Z. Win, and A. Conti, "Quantum pulse position modulation with photon-added coherent states," in *Proc. IEEE Workshop Quantum Commun. Inf. Technol. (QCIT), Global Telecomm. Conf.*, Waikoloa, HI, USA, Dec. 2019, pp. 1–5.

[7] S. Guerrini, M. Chiani, and A. Conti, "Secure key throughput of intermittent trusted-relay QKD protocols," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–5.

[8] G. M. D'Ariano, P. L. Presti, and M. G. A. Paris, "Using entanglement improves the precision of quantum measurements," *Phys. Rev. Lett.*, vol. 87, no. 27, Dec. 2001, Art. no. 270404.

[9] Z. Huang, C. Macchiavello, and L. Maccone, "Usefulness of entanglement-assisted quantum metrology," *Phys. Rev. A*, vol. 94, no. 1, Jul. 2016, Art. no. 012101.

[10] R. Demkowicz-Dobrzański and L. Maccone, "Using entanglement against noise in quantum metrology," *Phys. Rev. Lett.*, vol. 113, no. 25, Dec. 2014, Art. no. 250801.

[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, Los Alamitos, CA, USA: IEEE Press, 1994.

[12] A. Y. Kitaev, "Quantum computations: Algorithms and error correction," *Russ. Math. Surv.*, vol. 52, no. 6, pp. 1191–1249, Dec. 1997.

[13] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, no. 15, 2009, Art. no. 150502.

[14] D. Bruß and C. Macchiavello, "Multipartite entanglement in quantum algorithms," *Phys. Rev. A*, vol. 83, no. 5, May 2011, Art. no. 052313.

[15] T. Peng, A. Harrow, M. Ozols, and X. Wu, "Simulating large quantum circuits on a small quantum computer," 2019, *arXiv:1904.00102*. [Online]. Available: https://arxiv.org/abs/1904.00102

[16] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum-enhanced positioning and clock synchronization," *Nature*, vol. 412, no. 6845, pp. 417–419, Jul. 2001.

[17] Y. Aharonov, J. Oppenheim, S. Popescu, B. Reznik, and W. G. Unruh, "Measurement of time of arrival in quantum mechanics," *Phys. Rev. A*, vol. 57, no. 6, pp. 4130–4139, Jul. 2002.

[18] P. Komar *et al.*, "A quantum network of clocks," *Nature Phys.*, vol. 10, no. 8, pp. 582–587, Aug. 2014.

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

[20] S. Lloyd, J. H. Shapiro, F. N. C. Wong, P. Kumar, S. M. Shahriar, and H. P. Yuen, "Infrastructure for the quantum Internet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 5, pp. 9–20, Oct. 2004.

[21] A. Reiserer and G. Rempe, "Cavity-based quantum networks with single atoms and optical photons," *Rev. Mod. Phys.*, vol. 87, no. 4, pp. 1379–1418, Dec. 2015.

[22] A. S. Cacciapuoti and M. Caleffi, "Toward the quantum Internet: A directional-dependent noise model for quantum signal processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brighton, U.K., May 2019, pp. 7978–7982.

[23] M. Schlosshauer, "Decoherence, the measurement problem, and interpretations of quantum mechanics," *Rev. Mod. Phys.*, vol. 76, no. 4, pp. 1267–1305, Feb. 2005.

[24] K. Jagannathan, A. Chatterjee, and P. Mandayam, "Qubits through queues: The capacity of channels with waiting time dependent errors," in *Proc. Nat. Conf. Commun. (NCC)*, Bengaluru, India, Feb. 2019, pp. 1–6.

[25] P. Mandayam, K. Jagannathan, and A. Chatterjee, "The classical capacity of a quantum erasure queue-channel," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Cannes, France, Jul. 2019, pp. 1–6.

[26] F.-G. Deng, G. L. Long, and X.-S. Liu, "Two-step quantum direct communication protocol using the Einstein-Podolsky-Rosen pair block," *Phys. Rev. A*, vol. 68, no. 4, Apr. 2003, Art. no. 042317.

[27] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Rev. Modern Phys.*, vol. 81, no. 2, pp. 865–942, Jun. 2009.

[28] M. Epping, H. Kampermann, C. Macchiavello, and D. Bruß, "Multi-partite entanglement can speed up quantum key distribution in networks," *New J. Phys.*, vol. 19, no. 9, Sep. 2017, Art. no. 093012.

[29] Y. Xie, J. Yuan, and Q. T. Sun, "Design of quantum LDPC codes from quadratic residue sets," *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 3721–3735, Sep. 2018.

[30] R. B. Cooper, *Introduction to Queueing Theory*, 2nd ed. New York, NY, USA: Elsevier, 1981.

[31] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1992.

[32] L. Kleinrock, *Queueing Systems*, vol. 1, 1st ed. New York, NY, USA: Wiley, 1975.

[33] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE Trans. Autom. Control.*, vol. AC-25, no. 4, pp. 690–693, Aug. 1980.

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

[35] J. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. COM-29, no. 10, pp. 1474–1481, Oct. 1981.

[36] L. Gyongyosi and S. Imre, "Decentralized base-graph routing for the quantum Internet," *Phys. Rev. A*, vol. 98, no. 2, Aug. 2018, Art. no. 022310.

[37] L. Jiang, J. M. Taylor, N. Khaneja, and M. D. Lukin, "Optimal approach to quantum communication using dynamic programming," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 44, pp. 17291–17296, Oct. 2007.

[38] A. Dahlberg *et al.*, "A link layer protocol for quantum networks," in *Proc. ACM Special Interest Group Data Commun.*, Beijing, China, Aug. 2019, pp. 159–173.

[39] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels," *Phys. Rev. Lett.*, vol. 70, no. 13, pp. 1895–1899, Jul. 2002.

[40] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, 4th ed. Belmont, MA, USA: Athena Scientific, 2017.

[41] N. Kalb *et al.*, "Entanglement distillation between solid-state quantum network nodes," *Science*, vol. 356, no. 6341, pp. 928–932, Jun. 2017.

[42] R. Reichle *et al.*, "Experimental purification of two-atom entanglement," *Nature*, vol. 443, no. 7113, pp. 838–841, Oct. 2006.

[43] L. Ruan, W. Dai, and M. Z. Win, "Adaptive recurrence quantum entanglement distillation for two-Kraus-operator channels," *Phys. Rev. A*, vol. 97, no. 5, May 2018, Art. no. 052332.

[44] S. L. Mouradian *et al.*, "Scalable integration of long-lived quantum memories into a photonic circuit," *Phys. Rev. X*, vol. 5, no. 3, Jul. 2015, Art. no. 031009.

[45] J. D. C. Little, "A proof for the queuing formula: $L = \lambda W$," *Oper. Res.*, vol. 9, no. 3, pp. 296–435, May 1961.

[46] W. Pfaff *et al.*, "Unconditional quantum teleportation between distant solid-state quantum bits," *Science*, vol. 345, no. 6196, pp. 532–535, Aug. 2014.

[47] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 2. Belmont, MA, USA: Athena Scientific, 2011.

[48] A. S. Fletcher, P. W. Shor, and M. Z. Win, "Structured near-optimal channel-adapted quantum error correction," *Phys. Rev. A*, vol. 77, no. 1, Jan. 2008, Art. no. 012320.

[49] S. Croke, S. M. Barnett, and G. Weir, "Optimal sequential measurements for bipartite state discrimination," *Phys. Rev. A*, vol. 95, no. 5, May 2017, Art. no. 052308.

[50] A. S. Fletcher, P. W. Shor, and M. Z. Win, "Optimum quantum error recovery using semidefinite programming," *Phys. Rev. A*, vol. 75, no. 1, Jan. 2007, Art. no. 012338.

[51] W. Dai, T. Peng, and M. Z. Win, "Optimal remote entanglement distribution," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 3, pp. 540–556, Mar. 2020.