Entanglement Distribution in the Quantum Internet: *Knowing when to Stop!*

Angela Sara Cacciapuoti*, *Senior Member, IEEE*, Michele Viscardi, Jessica Illiano,
Marcello Caleffi, *Senior Member, IEEE*

***Abstract*—Entanglement distribution is a key functionality of the Quantum Internet. However, quantum entanglement is very fragile, easily degraded by decoherence, which strictly constraints the time horizon within the distribution has to be completed. This, coupled with the quantum noise irremediably impinging on the channels utilized for entanglement distribution, may imply the need to attempt the distribution process multiple times before the targeted network nodes successfully share the desired entangled state. And there is no guarantee that this is accomplished within the time horizon dictated by the coherence times. As a consequence, in noisy scenarios requiring multiple distribution attempts, it may be convenient to stop the distribution process early. In this paper, we take steps in the direction of *knowing when to stop* the entanglement distribution by developing a theoretical framework, able to capture the quantum noise effects. Specifically, we first prove that the entanglement distribution process can be modeled as a Markov decision process. Then, we prove that the optimal decision policy exhibits attractive features, which we exploit to reduce the computational complexity. The developed framework provides quantum network designers with flexible tools to optimally engineer the design parameters of the entanglement distribution process.**

***Index Terms*—Entanglement Distribution, Quantum Internet, Quantum Communications, Markov Decision Process**

I. Introduction

The Quantum Internet is foreseen to enable several applications with no counterpart in the classical world [2]–[8], such as distributed quantum computing [9] and secure communications [10]. To this aim, the entanglement distribution process plays the *key* role. Indeed, the successful distribution of entangled states among remote network nodes represents a necessary condition for any entanglement-based network [11].

A few theoretical models and designs for entanglement distribution have been recently proposed in literature. In [12], the authors model the distribution of entangled pairs as a discrete time Markov chain. Specifically, they assume infinite coherence time and infinite resources at the central node, with the aim of analyzing the expected capacity of the central node in terms of the number of qubits to be stored to meet the stability condition of the system. In [13], the distribution of entangled pairs is modeled as a continuous time Markov chain. Such a model is based on a Poisson probability distribution for the successful distribution of entangled pairs, and it accounts for some non-idealities, such as decoherence and noisy measurements. In [14], a Markov decision process is used to study the limits of bipartite entanglement distribution via entanglement swapping, by using a chain of quantum repeaters equipped with quantum memories. Finally, in [15] some practical figures of merit for entanglement distribution in quantum repeater networks are provided. In particular, the authors define the average connection time and the average size of the largest distributed entangled state for a fixed scenario.

Despite these research efforts, the fundamental problem of *knowing when to stop* (the entanglement distribution) remains unsolved. And filling this research gap is mandatory for the efficient engineering of any entanglement distribution process.

Specifically, it is well-known that quantum entanglement is a very fragile resource, easily degraded by decoherence [16], [17]. Decoherence severely impacts the time horizon in which freshly-generated entangled states can be successfully distributed and exploited for communication needs. Yet, due to the noise irremediably affecting the quantum communication channels utilized for entanglement distribution, it may be necessary to attempt the distribution process multiple times before that all the selected network nodes successfully share the targeted entangled state.

As a matter of fact, because of the complex and stochastic nature of the physical mechanisms underlying quantum noise, there is no guarantee that all the selected nodes can successfully share the entangled state within the time horizon dictated by the coherence times. As a consequence, in noisy scenarios requiring multiple distribution attempts, it may be convenient to stop the distribution process early, i.e., before entangling all the selected nodes. The rationale for this choice is twofold. On one hand, an early stopping can be required to account for additional delays induced by the network functionalities exploiting the entanglement resource. On the other hand, an early stopping can be convenient whenever "*enough*" nodes – accordingly to a certain figure of merit – already share entanglement, so that the entangled resource can be promptly exploited for the needed communication/computing purpose.

In this paper, we take steps in the direction of *knowing when to stop* by developing a theoretical framework. This framework provides quantum network designers with flexible tools to optimally engineer the design parameters of the entanglement

The authors are with the www.QuantumInternet.it research group, *FLY: Future Communications Laboratory*, University of Naples Federico II, Naples, 80125 Italy. A.S. Cacciapuoti and M. Caleffi are also with the Laboratorio Nazionale di Comunicazioni Multimediali, National Inter-University Consortium for Telecommunications (CNIT), Naples, 80126, Italy.

*Corresponding author.

A preliminary version of this work is currently under review for IEEE QCE23 [1].

Michele Viscardi acknowledges PNRR MUR project CN00000013, Angela Sara Cacciapuoti acknowledges PNRR MUR NQSTI-PE00000023, Marcello Caleffi acknowledges PNRR MUR project RESTART-PE00000001.

distribution. To the best of our knowledge, this is the first work addressing the optimal stopping rule for entanglement distribution.

A. Our contributions

The developed theoretical framework abstracts from the particular state to be distributed and provides a model that can be tweaked to account for the physical characteristics of the process itself. Specifically through the paper:

- we provide a comprehensive characterization of the entanglement distribution problem, by showing that it can be modeled as a Markov decision process with minimal assumptions;
- we provide the optimality conditions of the policy to be adopted, and we prove some key properties of the optimal policy that can be exploited for reducing the computational complexity;
- we analyze the impact of different reward functions on the distribution process through two main figures of merit: the average cluster size and the average distribution time;
- we gain insights on the selection of appropriate reward functions for entanglement distribution process engineering.

In summary, we present an easy-to-use tool for modeling and fine-tuning entanglement distribution systems to meet specific performance requirements. It is important to emphasize that the model we offer in this study is highly adaptable and can be tailored to various scenarios and applications.

The rest of the manuscript is organized as follows. In Sec. II, we introduce the system model along with some preliminaries. In Sec. III, we first formulate the entanglement distribution as a decision process, and then we derive both general (Sec. III-B) and reward-dependent (Sec. III-C) properties of the optimal policy, which we exploit for for reducing the computational complexity of the optimal policy search. In Sec. IV we validate the theoretical analysis through numerical simulations, and we discuss the impact of the reward functions on the performance of the entanglement distribution process. Finally, in Sec. V we conclude the paper, and some proofs are gathered in the Appendix.

II. System Model

Generating and distributing entanglement can be a demanding task due to the delicate nature of quantum states and their susceptibility to environmental disturbances. In many practical scenarios, the generation of entanglement requires sophisticated and resource-intensive setups, often involving complex experimental apparatuses and precise control mechanisms. These technological limitations, coupled with the need for specialized environments that can facilitate quantum communication processes, make it pragmatic to assume a specialized super-node responsible for entanglement generation and distribution [12], [18]–[20].

Remark. We emphasize that, when it comes to the distribution of multipartite entanglement, the assumption of a super-node for the entanglement generation is needed, not only due to the current maturity of the quantum technologies, but also due to the unavoidable requirement of some sort of local interaction among the qubits to be entangled, as discussed in [20].

Accordingly, we consider a scenario where a super-node is in charge of generating and distributing EPR pairs to a set of S quantum nodes – referred to as *clients* in the following – through quantum channels. It is worthwhile to note that the assumption of EPR pair distribution to client nodes is not restrictive, i.e., it does not hold only in EPR-based networks. In fact, when it comes to the distribution of multipartite entanglement, the super-node can, in principle, distribute each entangled qubit (ebit) to each client. However, this approach is not viable for all the classes of multipartite entanglement, which are characterized by different[1] *persistence* properties [2]. Accordingly, in the following we consider the more general case in which multipartite entangled states are distributed – as instance, through teleportation [19], [22] – by exploiting the a-priori distribution of EPR pairs via heralded scheme [23], [24]. As a matter of fact, this strategy is very common in literature and it has been proved to guarantee more resilience to noise and better protection against memory decoherence [19].

In the following we collect some definitions and assumptions that will be used in the paper.

EPR Distribution Model: The distribution attempt of an EPR ebit toward a client node through a noisy quantum channel is modeled with a Bernoulli distribution with parameter p, where p denotes the successful distribution probability.

According to the above, we consider quantum channels modeled as absorbing channels. Such a model constitutes a *worst-case* scenario, since the noise irreversibly corrupts the information carrier without any possibility of ebit recovery [25]–[28]. The channel behavior is captured through the parameter p, i.e., the probability of an ebit propagating through a quantum channel without experiencing absorption. And, $q \overset{\triangle}{=} 1 - p$ denotes the loss probability, i.e., the probability of ebit distribution failure as a consequence of the carrier absorption.

It is worthwhile to highlight that other noisy channel models can be easily incorporated in our analysis. As an example, Pauli channels followed by a purification process can be as well modeled with a Bernoulli distribution with parameter p, where p denotes the success probability of the joint distribution and purification process.

We observe that, by exploiting heralded schemes, the super-node is able to recognize which client – if any – experienced an absorption over the channel. And, in case of absorption, further distributions can be attempted. Indeed, it may be necessary to attempt the distribution multiple times before having the targeted subset of clients in the network successfully received the ebit. From the above, it follows straightforward to consider, within our model, the *number of possible distribution attempts*

[1] As an example, the direct distribution of GHZ-like states, which are characterized by the lowest persistence, requires all the photons encoding the GHZ state to be successfully distributed to the clients in a single distribution attempt [21].

as the key temporal parameter. Clearly, the maximum number of distribution attempts is determined by the coherence times of the underlying quantum technology, as detailed in the next subsection.

A. Problem Formulation

Definition 1 (**Time horizon**). *We consider the time horizon of the entanglement distribution process constituted by N time slots:*

$$\mathcal{N} = \{1, 2, \dots, N\}. \tag{1}$$

with N implicitly accounting for the minimum guaranteed coherence time.

Remark. Specifically, the value of N in (1) depends on the particulars of the technology adopted for generating and distributing the entangled states, and it is set such that decoherence effects can be considered negligible within the time horizon.

As shown in Fig. 1, the time is organized into N time-slots, where at (the end of) each time-slot the super-node can decide whether another distribution attempt should be performed (or not) in the subsequent time-slot. Clearly, the number of clients having already successfully received an ebit through the noisy channel, referred in the following as "connected" clients, represents a key parameter. We formalize this concept through the following two definitions.

Definition 2 (**Action Set**). *The action set $\mathcal{A}$ denotes the set of actions available at the super-node:*

$$\mathcal{A} = \{C, Q\}, \tag{2}$$

with C denoting the action of attempting another distribution round in the next time slot, and Q denoting the action of not attempting the distribution.

Definition 3 (**State Space**). *The system state space is defined as the pair*

$$(s, n) \in \tilde{\mathcal{S}} \times \mathcal{N}, \tag{3}$$

where $\mathcal{N}$ is given in (1) *and $\tilde{\mathcal{S}}$ is defined as follows:*

$$\tilde{\mathcal{S}} \stackrel{\triangle}{=} \mathcal{S} \cup \{\Delta\}, \tag{4}$$

with $\mathcal{S} \stackrel{\triangle}{=} \{0, 1, 2, \dots, S\}$ denoting the set of possible values for number of connected clients.

Accordingly, the system is in state (s, n) with $s \in \mathcal{S}$ if $s \leq S$ clients have successfully received an ebit from the super-node within the first n distribution attempts. It is worthwhile to note that Δ in (4) represents an auxiliary state, referred to as *absorbing state*, that denotes the state of the system where no further distributions are attempted.

Remark. In the following, we will use the symbols

$$s_n \stackrel{\triangle}{=} (s, n) \tag{5}$$

as a shorthand notation for the system state (s, n), whenever this will not generate confusion.

Definition 4 (**Allowed Action Set**). *The allowed action set A_{s_n} denotes the set of actions available at the super-node when the system state is s_n, and it results:*

$$\mathcal{A}_{s_n} = \begin{cases} \{C, Q\} & s \in \mathcal{S} \setminus \{S\} \wedge n < N \\ \{Q\} & s = S \vee s = \Delta \vee n = N \end{cases} \tag{6}$$

From Def. 4 it results that the only allowed action is Q whenever the system either: i) successfully distributed entanglement to all the clients, or ii) is in the absorbing state $s = \Delta$, or iii) is at the last available time-slot N. Assuming the system being in the state $s_n \in \tilde{\mathcal{S}} \times \mathcal{N}$ and depending on the particular action $a \in \mathcal{A}_{s_n}$ taken, the system will evolve into some state $\tilde{s}_{n+1} \in \tilde{\mathcal{S}} \times \mathcal{N}$ with some probability $p(\tilde{s}_{n+1} | s_n, a)$, which will be derived with Lemma 1 in Section III.

Decision Formulation. During the first time-slot, the super-node simultaneously transmits S ebits to the S clients. In case of absorption, further distributions can be attempted. This requires additional time, thus challenging the decoherence constraints as well as impacting the overall distribution rate. Hence there exists a trade-off between the number of clients that successfully receive an ebit – which we refer to as *distributed cluster size* – and the *distribution time*, i.e., the number of time slots after which the distribution process is either completed or arrested. This trade-off deeply impacts the performance of the overlaying communication functionalities. Thus, its optimization becomes crucial in the design of quantum networks.

To capture this trade-off by abstracting from the particulars of the underlying hardware technology(ies), we model the effects of the action $a \in \mathcal{A}_{s_n}$, taken by the super-node starting from the state s_n, through the notion of an utility function $r(s_n, a)$, referred to as *reward function*. Accordingly, we formalize this concept in the following Definition.

Definition 5 (**Reward function**). *Assuming that action $a \in \mathcal{A}_{s_n}$ is taken when the system is in state $s_n \in \tilde{\mathcal{S}} \times \mathcal{N}$, the overall reward achieved is:*

$$r(s_n, a) = \begin{cases} -f(s_n) & s \in \mathcal{S}, a = C \\ g(s_n) & s \in \mathcal{S}, a = Q \\ 0 & s = \Delta \end{cases} \tag{7}$$

where:

- *$f(s_n)$ denotes the continuation cost function, which models the overall cost of attempting (continuing) the ebits distribution when the system is in s_n;*
- *$g(s_n)$ denotes the pay-off function, which models the gain achievable by stopping the ebits distribution when the system is in the state s_n.*

It is clear that, according to our formulation, once the system reaches the absorption state, no further costs or rewards are obtained since the distribution process has been stopped.

Remark. The notion of reward function allows us to abstract from the particulars of i) the underlying technology for entanglement generation and distribution, and ii) the overlying network functionalities exploiting entanglement as a communication resource. In turn, this enables the following two key

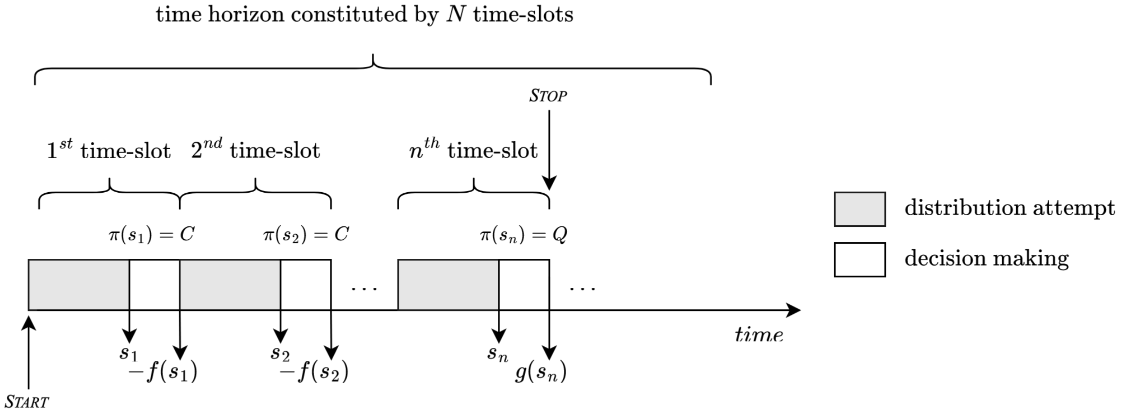


Fig. 1: Pictorial representation of the model for the entanglement distribution process. The overall goal is to decide *when to stop* the entanglement distribution.

features: i) it restricts our attention on the effects of the entanglement distribution process; b) it allows us to measure the performance of an entanglement distribution strategy, and thus it allows us to quantitatively compare different strategies.

In the following we restrict our attention on payoff functions $\{g(s_n)\}$ satisfying the two following properties.

Property 1 (**Monotonicity with s**). *The payoff function $g(s_n)$ is a monotonic non-decreasing function of s:*

$$g(s_n) \leq g(\tilde{s}_n) \quad with \;\; s < \tilde{s}. \tag{8}$$

Property 2 (**Monotonicity with n**). *The payoff function $g(s_n)$ is a monotonic non-increasing function of n:*

$$g(s_n) \geq g(s_m) \quad with \;\; n \leq m. \tag{9}$$

The rationale for these two properties is to model scenarios with meaningful meaning from an entanglement distribution perspective. Specifically, with Property 1 the reward function tunes the system choice towards larger s, i.e., higher number of connected clients. Clearly, this is reasonable since the higher is the number of connected clients, the larger is – as instance – the distributed multipartite entangled state. Conversely, Property 2 tunes the system choice towards shorter distribution times, which is mandatory to account for the fragile, easily degraded nature of entanglement.

Remark. It is worthwhile to note that the theoretical framework developed in Sec. III-A continues to hold regardless of whether the reward exhibits any monotonicity. Conversely, we will exploit these two properties in Sec. III-B for reducing the computational complexity of the optimal decision strategy.

According to the theoretical framework developed so far, the entanglement distribution process is modeled through the quintuple:

$$\{\tilde{\mathcal{S}}, \mathcal{N}, \mathcal{A}_{s_n}, p(\tilde{s}_{n+1}|s_n, a), r(s_n, a)\}. \tag{10}$$

III. Knowing when to Stop

Here, we develop the theoretical framework for modeling the entanglement distribution process. Specifically, in Sec. III-A, we prove that – with the minimal set of assumptions about the quantum technologies underlying entanglement generation and distribution – the entanglement distribution process can be modeled as a Markov decision processes. Then in Sec. III-B we prove some key properties that we will exploit to reduce the computational complexity of the problem.

A. Optimal Decision Model

In Theorem 1 we prove that the entanglement distribution process can be modeled as a Markov Decision Process. To this aim, the preliminary result in Lemma 1 is needed.

Lemma 1. *Assuming action $a \in \mathcal{A}_{s_n}$ is taken when the system is in state $s_n \in \tilde{\mathcal{S}} \times \mathcal{N}$, the probability $p(\tilde{s}_{n+1}|s_n, a)$ of the system evolving into state $\tilde{s}_{n+1} \in \tilde{\mathcal{S}} \times \mathcal{N}$ depends only on current state and action, and it is given by:*

$$p(\tilde{s}_{n+1}|s_n, a) = \begin{cases} p(\tilde{s}|s), & \textit{if } a = C \wedge s, \tilde{s} \in \mathcal{S} : \tilde{s} \geq s \\ 1 & \textit{if } a = Q \wedge \tilde{s} = \Delta \\ 0 & \textit{otherwise} \end{cases}, \tag{11}$$

with

$$p(\tilde{s}|s) = \binom{S-s}{\tilde{s}-s} q^{S-\tilde{s}} p^{\tilde{s}-s}. \tag{12}$$

Proof: See Appendix A ■

Remark. The available actions defined in (6) establish two disjoint functioning regimes for the system, namely, the regime of action C and the regime of action Q, as shown in Fig. 2 with reference to a system with $S = 3$ clients. Specifically, Fig. 2a represents the regime of action C. Here, the system evolves according to the transition probabilities $p(\tilde{s}|s)$ in (11). It is worth noting that there exist no transition towards the absorbing state through action C. Differently, Fig. 2b represents the region of action Q. Specifically, by accounting for (11), once the super-node decides to perform action Q, the system will only evolve towards (or remain in) the absorbing state Δ, where no further ebit transmissions are attempted.

Theorem 1. *The entanglement distribution process can be modeled as a Markov Decision Process.*

Proof: The proof follows from Lemma 1 by accounting for the Markov property of the transition probabilities [29]. ■

In the following, stemming from the result stated in Theorem 1, we will embrace the powerful framework of the Markov Decision Process to (optimal) "*know when to stop*" the entanglement distribution process. To this aim, the following definition is needed.

Definition 6 (**Policy**). *A policy $\pi(\cdot)$ is a rule determining the action to be taken in any possible state of the considered system. Hence, it is a function that maps the set of system states over the set of the allowed actions:*

$$\forall s_n \in \tilde{\mathcal{S}} \times \mathcal{N} : \pi(s_n) \in \mathcal{A}_{s_n} \tag{13}$$

In the following, Π denotes the set of all possible policies.

We note that, in (13), we exploited the Markovianity by considering policies $\pi(\cdot)$ depending on the current system state only, rather than on the entire history of the system state evolution [29]. Furthermore, we note that the overall reward achieved by adopting any policy $\pi(\cdot) \in \Pi$ is inherently stochastic, due to the noise affecting entanglement distribution. Thus, to assess and to compare the decision maker's preference toward different policies, we need a criterion to measure the performance of the selected policy. One widely adopted criterion in literature is the *expected total reward*, which we introduce in the following.

Expected Rewards. Given that the strategy $\pi(\cdot)$ is adopted, the **total expected reward** $v_\pi(s_1)$, obtained when the system state starts in state s_1, is recursively defined as:

$$v_\pi(s_1) = r\big(s_1, \pi(s_1)\big) + \sum_{\tilde{s} \in \tilde{\mathcal{S}} : \tilde{s}_2 = (\tilde{s}, 2)} p\big(\tilde{s}_2 | s_1, \pi(s_1)\big) v_\pi(\tilde{s}_2), \tag{14}$$

where $v_\pi(\tilde{s}_n)$ denotes the **expected remaining reward** at time slot n, and it is given by (15) shown at the top of the next page. Specifically, the boundary condition at time slot N in (15) prevents from infinite loops in the absorbing state.

We note that, for deriving the expression in (14), we exploited Theorem 4.2.1 in [29]. Accordingly, it is possible to restrict our attention on deterministic policies $\pi(\cdot) \in \Pi$ with no loss of optimality. Furthermore, we note that the expected total reward $v_\pi(s_1)$ has been defined as a recursive function, where the recursive step $v_\pi(s_n)$ at time slot n is function of three key parameters. That are the number of connected clients s, the policy $\pi(\cdot)$ through action $\pi(s_n)$, and the reward at time slot $n+1$ via the transition probabilities $p\big(\cdot | s_n, \pi(s_n)\big)$.

Stemming from the above, we are ready now to formally define the problem of (optimal) *knowing when to stop* the entanglement distribution.

(Optimally) Knowing When to Stop. By accounting for (14), the overall objective is to find the strategy $\pi^* \in \Pi$ that maximizes the expected total reward when the system is in state s_1:

$$v_{\pi^*}(s_1) = \max_{\pi \in \Pi} \big\{ v_\pi(s_1) \big\} \tag{16}$$

As a matter of fact, being the considered sets $\tilde{\mathcal{S}}$ and $\mathcal{N}$ finite, there always exists a deterministic strategy achieving the maximum in (16) [29]. Furthermore, we have implicitly assumed as overall goal to maximize the reward for some specific initial state s_1. Alternatively, the goal might be to find the optimal policy π^* prior to know the initial state s_1. In such a case, by accounting for (14), the total expected reward v_π is given by:

$$v_\pi = \sum_{s \in \mathcal{S} : s_1 = (s, 1)} p(s) v_\pi(s_1) \tag{17}$$

with $p(s)$, namely, the probability of successfully distributing ebits to s clients during the first distribution attempt, given by:

$$p(s) = p^s q^{S-s} \tag{18}$$

However, the reward in (17) is maximized by maximizing the reward in (14) for each s_1 in $\mathcal{S}$ [29]. Hence in the following we will focus on the problem formulation in (16) without any loss in generality.

B. Optimal Decision Strategy: Properties

In this subsection, we prove that the optimal policy $\pi^*(\cdot)$ exhibits specific properties with respect to the reward function. Then, we will engineer these properties to derive effective, practical strategies for reducing the computational complexity of the decision problem. To this aim, some preliminaries are needed.

First, we explicit the expression of the expected remaining reward in (15). Specifically, let us denote with $v^*(s_1)$ the maximum expected total reward, which is equivalent to the expected total reward achieved by the optimal policy π^* given in (16):

$$v^*(s_1) \overset{\triangle}{=} v_{\pi^*}(s_1) \tag{19}$$

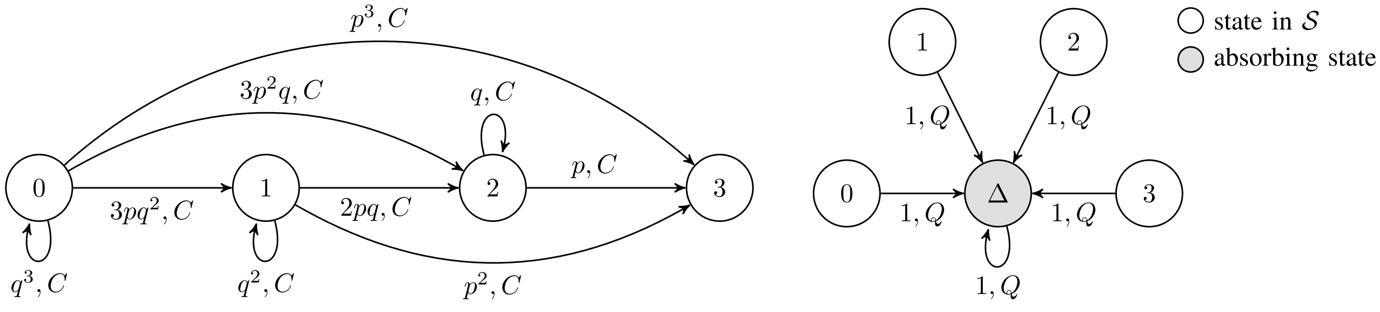
By accounting for the allowed action set $\mathcal{A}_{s_n}$ given in (6) and for the reward function defined in Def. 5, the maximum expected total reward $v^*(s_1)$ is given in (20) shown at the top of the next page, with the maximum expected remaining reward at the n-th recursive step given by:

$$v^*(s_n) = \begin{cases} \max\big\{ v^*_Q(s_n), v^*_C(s_n) \big\} & \text{if } n < N \\ r(s_N, Q) & \text{otherwise} \end{cases} \tag{21}$$

In (20), $v^*_Q(s_1)$ and $v^*_C(s_1)$ denote the maximum expected reward achievable when action Q or C is taken, respectively, starting from state s_n.

Furthermore, let us denote with $p(\check{s}_{n+k} | s_n, C)$ the probability to evolve into state $\check{s}_{n+k} = (\check{s}, n+k)$ at time slot $n+k$, starting from state $s_n = (s, n)$ with $s \neq \Delta$, by having chosen always action C at the end of each of time-slot[2] between n and $n+k-1$. By exploiting the Markovianity in Lemma 1,

[2] Namely, by choosing action C regardless whether the number of connected clients s is either $s < S$ or $s = S$.



(a) (b)

Fig. 2: Representation of the two functioning regimes for a network with $S = 3$ clients: (a): regime of the action C. (b): regime of the action Q.

$$v_\pi(s_n) = \begin{cases} r\big(s_n, \pi(s_n)\big) + \sum_{\tilde{s}\in\tilde{\mathcal{S}}} p\big(\tilde{s}_{n+1}|s_n, \pi(s_n)\big)v_\pi(\tilde{s}_{n+1}) & \text{if } n < N \\ r\big(s_N, Q\big) & \text{otherwise} \end{cases} \tag{15}$$

$$v^*(s_1) = \max\left\{ \overbrace{r\big(s_1, Q\big)}^{\triangleq v_Q^*(s_1)}, \overbrace{r\big(s_1, C\big) + \sum_{\tilde{s}\in\tilde{\mathcal{S}}} p\big(\tilde{s}_2|s_1, C)\big)v^*(\tilde{s}_2)}^{\triangleq v_C^*(s_1)} \right\} = \max\left\{ g(s_1)\,,\, -f(s_1) + \sum_{\tilde{s}\in\tilde{\mathcal{S}}} p\big(\tilde{s}_2|s_1, C)\big)v^*(\tilde{s}_2) \right\} \tag{20}$$

this probability, referred to us *extended transition probability*, can be recursively written as follows:

$$p(\breve{s}_{n+k}|s_n, C) = \sum_{\tilde{s}=s}^{\breve{s}} p\left(\breve{s}_{n+k}|\tilde{s}_{n+1}, C\right) p\left(\tilde{s}_{n+1}|s_n, C\right), \tag{22}$$

with the expression of $p\left(\tilde{s}_{n+1}|s_n, C\right)$ given in Lemma 1.

Stemming from the extended transition probabilities given in (22), we are ready to define now two rewards functions, that will be exploited in the following for efficiently deriving the optimal policy.

Reward Majorant and Minorant. Given that the system is in state $s_n = (s, n)$, with $s \neq \Delta$ and $n < N$, we introduce the quantities $v^+(s_n)$ and $v^-(s_n)$, referred to as the reward majorant and the reward minorant, respectively:

$$\begin{aligned} v^+(s_n) &= r\big(s_n, C\big) + \sum_{\breve{s}\in\tilde{\mathcal{S}}} p\big(\breve{s}_N|s_n, C\big)v_Q^*\big(\breve{s}_{n+1}\big) \qquad (23) \\ &= -f(s_n) + \sum_{\breve{s}\in\tilde{\mathcal{S}}} p\big(\breve{s}_N|s_n, C\big)g(\breve{s}_{n+1}) \\ v^-(s_n) &= r\big(s_n, C\big) + \sum_{\tilde{s}\in\tilde{\mathcal{S}}} p\big(\tilde{s}_{n+1}|s_n, C\big)v_Q^*\big(\tilde{s}_{n+1}\big) = \qquad (24) \\ &= -f(s_n) + \sum_{\tilde{s}\in\tilde{\mathcal{S}}} p\big(\tilde{s}_{n+1}|s_n, C\big)g(\tilde{s}_{n+1}), \end{aligned}$$

with $\tilde{s}_{n+1} = (\tilde{s}, n+1)$ and $\breve{s}_{n+1} = (\breve{s}, n+1)$.

Both the majorant and the minorant model the reward achievable by deciding first to continue the entanglement distribution at time slot n and, then, to stop the distribution at the subsequent time slot $n+1$. Yet, they significantly differ each other:

- The reward minorant $v^-(s_n)$ is obtained by assuming the system evolving from state s_n to state $\tilde{s}_{n+1}$ in agreement with the transition probabilities given in (13).
- Conversely, the reward majorant $v^+(s_n)$ is obtained by assuming the system able to evolve freely from state s_n to state $\breve{s}_N$ – with $\breve{s}_N = (\breve{s}, N)$ representing the state that would have been reached by performing $N - n$ subsequent distributions attempts by choosing only action C and never action Q – yet in a single time slot. In other words, the majorant models the expected reward achieved when the system performs $N-n$ subsequent distributions attempts, yet i) by paying only a single continuation cost $-f(s_n)$, and ii) by obtaining a pay-off $g(\breve{s}_{n+1})$ as if $\breve{s}$ would have been reached in a single time slot.

The proof of the main result, namely, Theorem 2 requires the following preliminary lemma.

Lemma 2. *Given that the system state is s_n with $s \in \mathcal{S}$ and $n < N$, it results:*

$$v^-(s_n) \leq v_C^*(s_n) \leq v^+(s_n) \tag{25}$$

Proof: See Appendix A. ■

Theorem 2. *Given that the system state is s_n with $s \in \mathcal{S}$ and $n < N$, it results:*

$$\pi^*(s_n) = \begin{cases} Q & \textit{if } g(s_n) \geq v^+(s_n) \\ C & \textit{if } g(s_n) \leq v^-(s_n) \end{cases} \tag{26}$$

Proof: The proof follows directly from Lemma 2, by accounting for the definition of $v^*_C(s_n)$ and $v^*_Q(s_n)$ given in (20). ■

Markov decision problems as the one we considered in (16) are generally solved with backward induction [29]. Specifically, stemming from the expression of the maximum expected remaining reward given in (21), backward induction works as follows: starting from $n = N$ and going backward in time, the optimal action maximizing the expected total reward is obtained for each state s_n by exploiting the already-derived optimal actions for states $\tilde{s}_{n+1}$, with $\tilde{s} > s$.

Remark. When the system state is s_n, backward induction requires to preliminarly evaluate $\big(S - s + 1\big)^{N-n}$ optimal actions – i.e., to compute the optimal action for each possible future state – before determining the optimal action $\pi^*(s_n)$ for the current state. Luckily, with Theorem 2 we have derived an efficient strategy for finding the optimal action without the need of evaluating the future evolution of the system. Specifically, whenever $g(s_n)$ satisfies one of the conditions in (26), the optimal action can be decided regardless of any further evolution of the system. We validate this result with the first experiment in Sec. IV.

Finally, it is important to discuss the assumptions underlying Theorem 2. As regards to the continuation cost $f(\cdot)$, Theorem 2 does not require any assumption or constraint, except $f(\cdot)$ being reasonable non negative[3]. As regards to the pay-off function $g(\cdot)$, Theorem 2 requires Properties 1-2 being satisfied. Yet these properties are not restrictive, since they reasonably drive the entanglement distribution toward entangling the *larger* number of client nodes in the *shorter* possible time-frame.

In the next subsection, we will introduce and discuss some (reasonable) assumptions on the pay-off function which allows us to further simplify the search of the optimal policy.

C. One-Step Look Ahead

Here we depart from the general discussion of Sec. III-B, by further extending the result of Theorem 2 for deriving the optimal policy, albeit imposing additional constraints on the rewards. To this aim, the following preliminaries are need.

Given that there exists only two actions in (2) – namely, continue or stop – the entanglement distribution problem belongs to the framework of optimal *stopping* problems, for which there exists a very simple (hence, computational efficient) rule – namely, one-step look ahead (OLA) rule – for deciding the action to be taken.

Definition 7 (**OLA Set**). *At time-step n, the one-step look ahead (OLA) set $\mathcal{S}^Q_n \subseteq \tilde{\mathcal{S}}$ is the set of system states where the instantaneous reward achievable by stopping is not lower than the expected reward achievable by attempting a further distribution attempt and then deciding to stop the distribution.*

$$\mathcal{S}^Q_n = \big\{s \in \mathcal{S} : g(s_n) \geq v^-(s_n)\big\} \tag{27}$$

with $v^-(s_n)$ given in (24).

Definition 8 (**OLA Rule**).

$$\pi(s_n) = \begin{cases} Q & \textit{if } s_n \in \mathcal{S}^Q_n \iff g(s_n) \geq v^-(s_n) \\ C & \textit{otherwise} \end{cases} \tag{28}$$

The naming for the OLA rule follows by noting that the reward minorant $v^-(s_n)$ represents the *expected reward* when the policy is *to continue for one-step and then to stop*, namely:

$$v^-(s_n) = -f(s_n) + E[g(\mathfrak{S}_{n+1})] \tag{29}$$

with $\mathfrak{S}_{n+1}$ denoting the random variable describing the system state at step $n + 1$.

The OLA rule is optimal whenever the OLA set is closed [30], [31], namely, whenever the system state remains confined within the OLA set, once entering. Unfortunately, the optimality of the OLA rule strictly depends on the particulars of the cost $f(\cdot)$ and pay-off $g(\cdot)$ functions, and no general conclusions can be taken independently.

Yet, we can consider different settings for the cost/pay-off functions – which allows us to model a wide range of possible communication scenarios – and discuss the optimality of the OLA rule with respect to this setting. More into details, we consider the following three base-cases:

$$g(s_n) = \frac{s}{n} \tag{30}$$

$$g(s_n) = \lambda^n s, \text{ with } \lambda \in (0, 1] \tag{31}$$

$$g(s_n) = \frac{s}{S} - \frac{n}{N} \tag{32}$$

with $f(s_n) = 0$ since we already incorporated the cost arising with additional distribution attempts into the reward.

Remark. As an example, with the first base-case given in (30) we model a scenario where the reward, represented by the number s of entangled clients, is discounted by a factor equal to the number of time-slots used for entangling such clients. The rationale for this scenario is to model the reward as a sort of *entanglement throughput* – namely, as an *average entanglement per unit of time* – similarly to the bit throughput that represents one of the key metric for classical networks. As regards to the second base-case given in (31), it introduces a discount factor λ which exponentially weights the reward s as time passes. As a matter of fact, multiplicative decreasing the rate of some process such as in (31) is widely adopted in classical networks, with TCP exponential back-off constituting the most famous case. Finally, with (32) we meant to introduce another base-case for conferring generality to the discussion.

By considering the settings of the base-cases, we have the following result.

Proposition 1. *When the rewards are modeled as in* (31) *or* (31)*, the OLA rule is optimal and it results:*

$$\pi^*(s_N) = Q \iff \begin{cases} s \geq \dfrac{\lambda S p}{1 - \lambda + \lambda p} & \textit{if } g(s_n) = \lambda^n s \\ s \geq S - \frac{S}{Np} & \textit{if } g(s_n) = \frac{s}{S} - \frac{n}{N} \end{cases} \tag{33}$$

whereas when the rewards are modeled as in (30)*, the OLA rule is not optimal.*

Proof: See Appendix C. ■

[3] Otherwise it would represent a pay-off rather than a cost.

From an engineering perspective, it is evident that having an efficient (i.e., low-computational-complexity) optimal rule, such as the OLA rule, for deriving the optimal policy – namely, for deciding when to stop distributing entanglement within a quantum network – is highly advantageous. Hence, whenever possible, the opportunity of choosing rewards satisfying the optimality condition of the OLA rule should be preferred.

Nevertheless, whenever this should not be possible, we can still exploit the main result – namely, Theorem 2 – for designing an efficient rule, as long as we tolerate finding a sub-optimal policy rather than an optimal one.

Definition 9 (**Sub-Optimal Rule**).

$$\pi(s_n) = \begin{cases} Q & \textit{if } s_n \geq \frac{v^+(s_n)+v^-(s_n)}{2} \\ C & \textit{otherwise} \end{cases} \tag{34}$$

Clearly, the "amount" of sub-optimality – hence, the loss in reward – introduced by such a rule strictly depends on the particular settings of the rewards. In the next subsection, we will evaluate such a sub-optimality for the three base-cases introduced above.

IV. Performance evaluation

In this section, we first validate the theoretical results derived in Secs. III-B and III-C.

Then, we discuss the impact of the reward functions on the performance of the entanglement distribution process. To this aim, we focus on two key metrics:

- average distribution time, namely, the average number of time-slots before the distribution is arrested;
- average cluster size, namely, the average number of client nodes successfully entangled;

More into details, we investigate how the choice of the reward setting influences these two key metrics. This allows us to to draft some guidelines for selecting a reward function able to drive the system to fulfill some specific performance requirements.

With the first experiment, we evaluate in Fig. 3 the expected total reward v_π given in (17) as a function of the ebit propagation probability p. The adopted simulation set is as follows: the number of clients is $S = 100$, the time-horizon is constituted by $N = 100$ time-slots, the rewards are modeled as in (30) with $g(s_n) = \frac{s}{n}$, and p varies with step equal to 0.025. Within the experiment, we consider four different rewards.

First, we consider the reward v_{π^*} achieved with the optimal policy π^*, with π^* obtained via exhaustive search through backward induction. Clearly, this is the maximum expected reward that can be achieved, and it represents the performance baseline for any sub-optimal policy. We note that, the higher is p, the higher is the reward v_{π^*}. This result is reasonable, since higher distribution probabilities allow the system to evolve toward states characterized by higher cluster sizes s and lower distribution times n.

Additionally, we consider the reward v_{π^*} achieved with the policy π^* computed via Theorem 2. More into detail, $\pi^*(s_n)$ is obtained with Theorem 2 whenever either of the two constrains in (26) holds, and via backward induction otherwise. Clearly, by comparing this reward with the optimal reward v_{π^*}, we can observe a perfect agreement between the two rewards. This constitutes an experimental validation of the analytical results derived in Theorem 2.

Furthermore, we consider the reward v_π achieved when the policy π is obtained with the OLA rule given in Definition 8. Indeed, it must be noted that – although barely noticeable even in the zoomed-in inset of Fig. 3 – the reward achievable with the OLA rule is lower than the reward v_{π^*} achievable with the optimal policy for any value of p. This validates the theoretical results derived in Prop 1, and, specifically, the sub-optimality of the OLA rule for $g(s_n) = \frac{s}{n}$. Yet, the performance degradation of the OLA rule is practically negligible.

Finally, we consider the reward v_π achieved with the policy π obtained via the sub-optimal rule given in Definition 9. From Fig. 3, one might question the rationale for this sub-optimal rule and, specifically, one might incorrectly believes that – given that the OLA rule significantly outperforms the sub-optimal rule given in Definition 9 – the last rule is useless. Yet it must be noted that the performance of the OLA rule strictly depends on some specific assumptions on the cost $f(\cdot)$ and pay-off $g(\cdot)$ functions, assumptions which are not required by the rule given in Definition 9.

Remark. From the above discussion, it becomes clear that there exists a trade-off between optimality and computational-efficiency, that must be properly engineered by the quantum network designers. Specifically, designers can decide to adopt generalist heuristic policies – such as the one in Def. 9 – which does not impose limitations on the choice of the reward functions albeit at the price of sub-optimal decisions. Or they can leverage optimal, efficient policies – such as the OLA one – as long as they can tolerate additional constraints in the reward function definition.

With the second experiment, we aim at assessing the importance of an optimal policy for achieving the highest total reward. For this, in Fig. 4 we plot the average total reward v_π given in (17) as a function of the ebit propagation probability p for 10^6 Montecarlo distribution process trials, for the same simulation set adopted in Fig. 3. We note that lines denote the mean of the total rewards over the different trials, whereas shading areas denote the standard deviation of the different trials[4].

We extend the set of policies by considering – along with the optimal and the two sub-optimal policies already considered in the previous experiment – 20 random policies. We observe that the higher is the ebit distribution probability p , the higher is the performance gap between the expected total reward achieved by the optimal strategy and the reward achieved by a random strategy. As a matter of fact, the performance gap remains evident even if we consider the distribution of the optimal reward via standard deviation. This result shows the importance of the considered problem for scenarios of

[4] With the shading areas of optimal and OLA rewards practically overlapping.

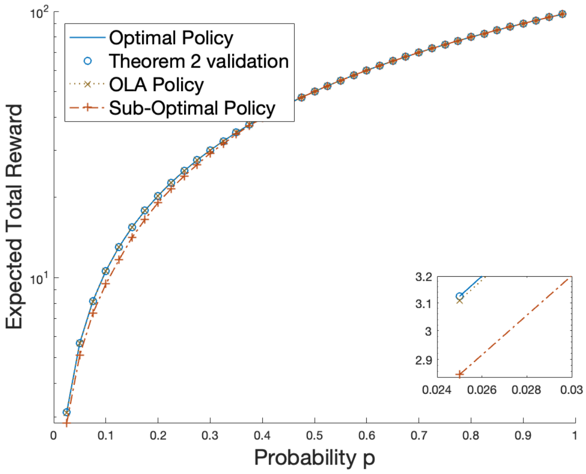


Fig. 3: Expected total reward v_π as a function of the ebit propagation probability p for $S = 100$, $N = 100$ and $g(s_n) = \frac{s}{n}$. Logarithmic scale for axis y.

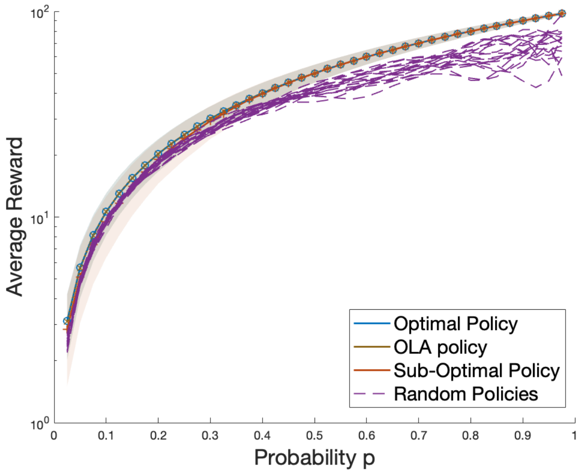


Fig. 4: Average total reward v_π as a function of the ebit propagation probability p for the same setting of Fig. 3. Lines denote the mean, whereas shading areas denote the standard deviation. Logarithmic scale for axis y.

practical interest, namely, for scenarios where entanglement can be fairly distributed.

In Fig. 5, we present the average cluster size s as a function of the ebit propagation probability p, computed with the same 10^6 Montecarlo distribution process trials of Fig. 4. As before, lines denote the mean over the different trials, whereas shading areas denote the standard deviation of the different trials. First, we note that the random policies might achieve larger cluster sets with respect to the optimal policy. The rationale for this behaviour is that the optimal policy aims at: i) maximizing the cardinality of the cluster set, while simultaneously ii) minimizing the distribution time. Hence, depending on $g(\cdot)$ and p, the optimal policy might prefer an earlier stop of the distribution process. And this was, indeed, the overall objective of our modeling.

These considerations are confirmed by Fig. 6, which presents the average distribution time n as a function of ebit propagation probability p, computed with the same 10^6 Montecarlo distribution process trials of Fig. 4. Indeed, it is possible to note that the values of p in Fig. 6 – for which the random policies achieve larger cluster sizes with respect to the optimal policy – are characterized by longer distribution times.

Finally, with the latest experiment, we aim at discussing the impact of the rewards settings – and, specifically, of the three base-cases introduced in (30)-(32) – on the overall entanglement distribution process.

For this, we preliminary compare the optimal policy π^* for the different settings of the pay-off function via the action matrices represented in Fig 7. Formally, the action matrix $A^* : S \times N \longrightarrow p \in [0, 1]$ is defined as follows:

$$a^*_{s,n} \in A^* = \tilde{p} \iff \pi^*(s_n) = \begin{cases} Q & \forall p \leq \tilde{p} \\ C & \forall p > \tilde{p} \end{cases} \tag{35}$$

As an example, by considering the action map for the pay-off function $g(s_n) = \frac{s}{n}$ represented in Fig. 7a, we note that, for an arbitrary time-slot n, $a^*_{s,n}$ increases as the cluster size s increases. This means that, as the cluster size s increases, higher values of p are needed for having action C being the optimal action. Clearly, for a given n, for the lowest values of s, action C is optimal for almost all the values of p. This is very reasonable: when the current cluster size s is very small, so is the pay-off reward. Hence, it is likely more convenient to attempt another entanglement distribution rather than to stop here. And, vice-versa, for the highest values of s, action Q is optimal for almost all the values of p.

Furthermore, we observe that the values of the action matrices in Fig. 7 strongly depend on the particular pay-off function.

As instance, the action map for the pay-off function $g(s_n) = \lambda^n$ represented in Fig. 7b strongly depends on the cluster size, whereas it is largely independent from the time-slot. As a result, the pay-off function $g(s_n) = \lambda^n$ drives the entanglement distribution process towards larger cluster sizes at the price of significantly longer distribution times.

These considerations are are clearly confirmed by Fig. 8, which presents the average cluster size s as a function of the ebit propagation probability p – for the same 10^6 trials of Fig. 4 – for the different settings of the pay-off function given in (30)-(32). As before, lines denote the mean over the different trials, whereas shading areas denote the standard deviation of the different trials.

First, we note that the larger is the parameter λ in (31), the larger is the average cluster size s and the steeper is the slope of the related curve. As a matter of fact, the largest values of the average cluster size are achieved when the pay-off function is $g(s_n) = \frac{s}{S} - \frac{n}{N}$ as in (32). This agrees with the action matrix in Fig 7c, where action Q becomes optimal only for the largest values of s.

Interestingly, the pay-off functions significantly impact the performances for lower values of p. Indeed, both in Fig. 8 and Fig. 9, as p increases, the distance between the curves in the

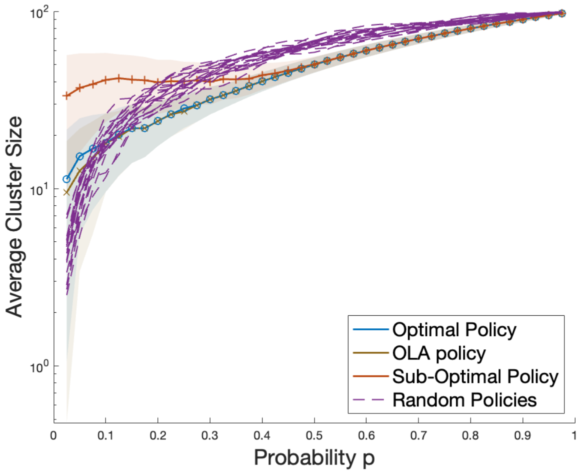


Fig. 5: Average cluster size s as a function of the ebit propagation probability p for the same setting of Fig. 3. Lines denote the mean, whereas shading areas denote the standard deviation. Logarithmic scale for axis y.

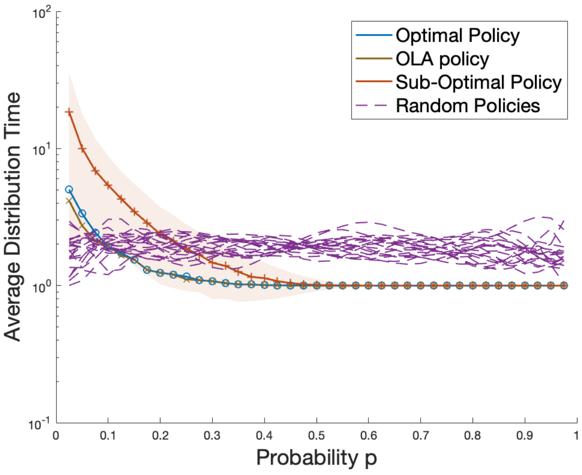
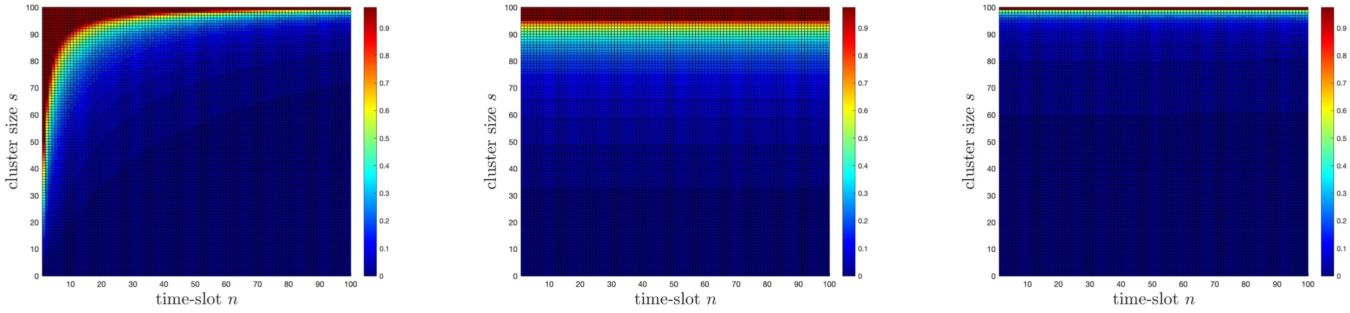


Fig. 6: Average distribution time n as a function of the ebit propagation probability p for the same setting of Fig. 3. Lines denote the mean, whereas shading areas denote the standard deviation. Logarithmic scale for axis y.



(a) Action matrix for pay-off function $g(s_n) = \frac{s}{n}$.

(b) Action matrix for pay-off function $g(s_n) = \lambda^n s$, with $\lambda = 0.95$.

(c) Action matrix for pay-off function $g(s_n) = \frac{s}{S} - \frac{n}{N}$.

Fig. 7: Action matrices: compact representation of the optimal policy $\pi^*(s_n)$ as a function of the system state $s_n = (s, n)$ and ebit distribution probability p. Setting: $S = 100$ and $N = 100$.

graph tends to reduce. The rationale is that, as p increases, the target system state – namely, the system state maximizing the reward – can be quickly achieved in shorter distribution times. Thus, different reward functions result in vastly different ebit distribution performances under bad transmission conditions.

Remark. From the above, it becomes evident that, whenever there exist requirements in terms of average cluster size or average distribution time, our modeling allows to meet the performance requirements by choosing a suitable reward function, as instance by tuning the value of λ in $g(s_n) = \lambda^n s$. Thus, our formulation of the entanglement distribution process as an optimal decision problem constitutes an effective, handy tool for quantum network designers aiming at engineering the entanglement distribution process.

V. Conclusion

In this work, we provided a formulation of the entanglement distribution process as a Markov Decision Process. Our theoretical model jointly accounts for the constraints arising from the underlying technologies as well as for the overlaying communication protocol requirements. We exploited this formulation for discussing the trade-off arising between the two key performance metrics – i.e., the average cluster size and the average distribution time – and for discussing the impact of the reward function and the decision-making policy on the entanglement distribution performance. Our formulation provides quantum network designers with an effective, handy tool for tuning and engineering the entanglement distribution process, so that it can meet the performance requirements through proper reward functions.

Appendix A
Proof of Lemma 1

According to the model developed in Sec. II, a distribution attempt takes place only if action $a = C$ is taken. And in this case, at time slot $n + 1$, the system – as a result of the

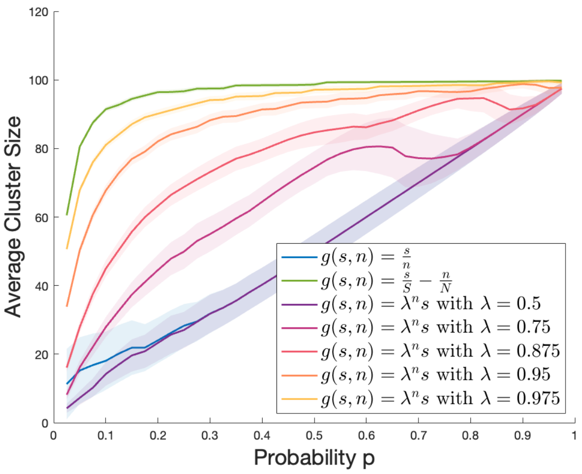


Fig. 8: Average cluster size as a function of the ebit propagation probability p for $S=100$, $N=100$ and different settings of the pay-off function $g(\cdot)$. Lines denote the mean, whereas shading areas denote the standard deviation.

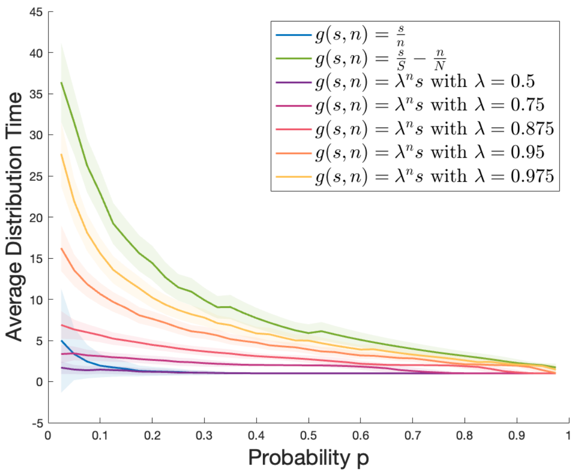


Fig. 9: Average distribution time as a function of the ebit propagation probability p for $S=100$, $N=100$ and different settings of the pay-off function $g(\cdot)$. Lines denote the mean, whereas shading areas denote the standard deviation.

distribution attempts – evolves into another state characterized by a number $\tilde{s}$ of “connected nodes”, which cannot be smaller than the number s of “connected nodes” in the time slot n. The reason for which $\tilde{s} \geq s$ is twofold: i) the heralded scheme allows the super-node to recognize which node – if any – experienced an ebit loss in a given time-slot. Hence, in the successive time slot, the super-node distributes entanglement only to the missing nodes; ii) by restricting the distribution attempts within a time interval N where the decoherence effects are negligible, the system state evolution is restricted from “backward” transitions towards smaller connected sets with $\tilde{s} < s$. Stemming from this and in according to the EPR distribution model given in Sec. II, each ebit distribution attempt follows a Bernoulli distribution with parameter p. Accordingly, it follows that when $a=C$ and $s,\tilde{s}\in\mathcal{S}:\tilde{s}\geq s$, the transition probability $p(\tilde{s}_{n+1}|s_n,C)$ is given by:

$$p(\tilde{s}_{n+1}|s_n,C)=p(\tilde{s}|s)=\binom{S-s}{\tilde{s}-s}q^{S-\tilde{s}}p^{\tilde{s}-s}. \tag{36}$$

Conversely, when action $a=Q$ is taken, the system can only evolve in the absorption state Δ, which is a fictitious state modeling the state where no further distribution attempts are performed. It is worthwhile to observe that once in the absorption state $s=\Delta$, the system remains in such a state, i.e., no evolution towards $\tilde{s}\neq\Delta$ is allowed. As a consequence the proof follows.

Appendix B
Proof of Lemma 2

We have two statements to prove within the inequality given in (25).

Proof: First Inequality.

We start by proving the first part of the inequality in (25), namely:

$$v^-(s_n)\leq v_C^*(s_n)\;\forall\, s\in\mathcal{S}\wedge n<N \tag{37}$$

By exploiting the expression of $v_C^*(s_n)$ in (20), one can recognize that:

$$v_C^*(s_n)=-f(s_n)+\sum_{\tilde{s}\in\tilde{\mathcal{S}}}p(\tilde{s}_{n+1}|s_n,C)v^*\big(\tilde{s}_{n+1}\big) \tag{38}$$

According to (21), $v^*\big(\tilde{s}_{n+1}\big)\geq v_Q^*(\tilde{s}_n+1)$ and by accounting for the expression of $v^-(s_n)$ in (24), the proof follows. ■

Proof: Second Inequality.

We now prove the second part of the inequality in (25), i.e.:

$$v^+(s_n)\geq v_C^*(s_n)\;\forall\, s\in\mathcal{S}\wedge n<N \tag{39}$$

By exploiting the expressions of $v^+(s_n)$ in (23) and $v_C^*(s_n)$ reported in (38), one recognizes that proving (39) is equivalent to prove that:

$$\sum_{\check{s}\in\tilde{\mathcal{S}}}p(\check{s}_N|s_n,C)v_Q^*\big(\check{s}_{n+1}\big)\geq\sum_{\tilde{s}\in\tilde{\mathcal{S}}}p(\tilde{s}_{n+1}|s_n,C)v^*\big(\tilde{s}_{n+1}\big), \tag{40}$$

where, by definition

$$v^*\big(\tilde{s}_{n+1}\big)=\max\big\{v_Q^*(\tilde{s}_{n+1}),v_C^*(\tilde{s}_{n+1})\big\} \tag{41}$$

To prove (40), we can consider the two elements in (41) separately. To this aim, let us consider the more general case, namely, the case where $n+1<N$[5].

Case 1: $v^*\big(\tilde{s}_{n+1}\big)=v_Q^*(\tilde{s}_{n+1})$**.**

Let us conduct a proof with a *reductio ad absurdum*, i.e., let us suppose that:

$$\sum_{\check{s}\in\tilde{\mathcal{S}}}p(\check{s}_N|s_n,C)v_Q^*\big(\check{s}_{n+1}\big)<\sum_{\tilde{s}\in\tilde{\mathcal{S}}}p(\tilde{s}_{n+1}|s_n,C)v_Q^*\big(\tilde{s}_{n+1})\big). \tag{42}$$

By accounting for the extended transition probabilities given in (22), we obtain equation (43) given at the top of the next page. We note that (43) is satisfied only if there exists at least

[5] Indeed, when $n+1=N$, no decision has to be made since the distribution is interrupted and the system goes in the absorption state

$$\sum_{\breve{s}\geq s} p\big(\breve{s}_N|s_n,C\big)g(\breve{s}_{n+1}) = \sum_{\breve{s}\geq\tilde{s}} \sum_{\tilde{s}\geq s}^{\breve{s}} p\big(\breve{s}_N|\tilde{s}_{n+1},C\big)p\big(\tilde{s}_{n+1}|s_n,C\big)g(\breve{s}_{n+1}) < \sum_{\tilde{s}\geq s} p\big(\tilde{s}_{n+1}|s_n,C\big)g(\tilde{s}_{n+1}) \tag{43}$$

one $\tilde{s} \in \mathcal{S} : \tilde{s} \geq s$ so that:

$$\begin{aligned}\sum_{\breve{s}\geq\tilde{s}} p\big(\breve{s}_N|\tilde{s}_{n+1},C\big)p\big(\tilde{s}_{n+1}|s_n,C\big)g(\breve{s}_{n+1}) <\\ < p\big(\tilde{s}_{n+1}|s_n,C\big)g(\tilde{s}_{n+1}) \Longleftrightarrow\\ \Longleftrightarrow \sum_{\breve{s}\geq\tilde{s}} p\big(\breve{s}_N|\tilde{s}_{n+1},C\big)g(\breve{s}_{n+1}) < g(\tilde{s}_{n+1}).\end{aligned} \tag{44}$$

By accounting for Property 1 and by recognizing that $\sum_{\breve{s}\geq\tilde{s}} p\big(\breve{s}_N|\tilde{s}_{n+1},C\big) = 1$, (44) constitutes a *reductio ab absurdum* and so does (42).

Case 2: $v^*\big(\tilde{s}_{n+1}\big) = v_C^*(\tilde{s}_{n+1})$**.**
Let us conduct the proof again with a *reductio ad absurdum* by supposing that:

$$\sum_{\breve{s}\in\tilde{\mathcal{S}}} p\big(\breve{s}_N|s_n,C\big)v_Q^*\big(\breve{s}_{n+1}\big) < \sum_{\tilde{s}\in\tilde{\mathcal{S}}} p\big(\tilde{s}_{n+1}|s_n,C\big)v_C^*(\tilde{s}_{n+1})\big) \tag{45}$$

By accounting for the extended probabilities given in (22), we obtain equation (46) given at the top of the next page. For the sake of notation simplicity and with no loss in generality – as discussed at the end of this proof – let us assume $N = n+2$. Accordingly, $v^*(\breve{s}_{n+2}) = g(\breve{s}_N)$ and (46) holds only if there exists at least one $\tilde{s} \in \mathcal{S} : \tilde{s} \geq s$ so that:

$$\begin{aligned}\sum_{\breve{s}\geq\tilde{s}} p\big(\breve{s}_N|\tilde{s}_{n+1},C\big)g(\breve{s}_{n+1}) <\\ < -f(\tilde{s}_{n+1}) + \sum_{\breve{s}\geq\tilde{s}} p\big(\breve{s}_N|\tilde{s}_{n+1},C\big)g(\breve{s}_N)\end{aligned} \tag{47}$$

Hence, by accounting for Property 2, (47) constitutes a *reductio ab absurdum* and so does (45). We finally note that, whether N should be greater than $n+2$ – say $N = n+3$ as instance – we have that $v^*(\breve{s}_{n+2})$ is equal to $\max\big\{v_Q^*(\tilde{s}_{N-1}), v_C^*(\tilde{s}_{N-1})\big\}$, and the proof follows recursively by adopting the same reasoning adopted for the two elements in (41). ■

Appendix C
Proof of Proposition 2

A. Case I: rewards modeled as in (30).

Here we prove that the OLA rule is not optimal when the rewards are modeled as in (30), namely, when:

$$g(s_n) = \frac{s}{n} \tag{48}$$

Let us assume the system state being $s_n \in \mathcal{S}$. Whether action C is chosen, the expected state $E[\mathfrak{S}_{n+1}]$ is given by:

$$E[\mathfrak{S}_{n+1}] = \sum_{\tilde{s}\in\mathcal{S}} \binom{S-s}{\tilde{s}-s} q^{S-\tilde{s}} p^{\tilde{s}-s} = s + p(S-s) \tag{49}$$

Accordingly, stemming from the definition of OLA set in (27) and by accounting for (29), we have that S_n^Q and S_{n+1}^Q are given by:

$$S_n^Q = \left\{x \in \mathcal{S} : \frac{x}{n} \geq \frac{x+p(S-x)}{n+1}\right\} \tag{50}$$

$$S_{n+1}^Q = \left\{x \in \mathcal{S} : \frac{x}{n+1} \geq \frac{x+p(S-x)}{n+2}\right\} \tag{51}$$

Hence, after simple algebraic manipulations, it results:

$$s \in S_n^Q \Longrightarrow s \geq \frac{np}{1+np}S \tag{52}$$

$$\tilde{s} \in S_{n+1}^Q \Longrightarrow \tilde{s} \geq \frac{(n+1)p}{1+(n+1)p}S \tag{53}$$

Let us conduct the proof with a *reductio ab absurdum* by assuming that, starting from state $s_n : s \in S_n^Q$ and evolving into state $\tilde{s}_{n+1}$, it must result $\tilde{s} \in S_{n+1}^Q$ for any $\tilde{s}$. Without any loss of generality, we assume:

$$s = \frac{np}{1+np}S \quad \wedge \quad \tilde{s} = s \tag{54}$$

and, by jointly accounting for (53) and (54), it results:

$$\tilde{s} = s = \frac{np}{1+np}S > \frac{(n+1)p}{1+(n+1)p}S \Longrightarrow p < 0 \tag{55}$$

which clearly constitutes a *reductio ab absurdum*.

B. Case II: rewards modeled as in (31).

Here we prove that the OLA rule is optimal when the rewards are modeled as in (31), namely, when:

$$g(s_n) = \lambda^n s \tag{56}$$

To this aim, let us assume $s_n \in \mathcal{S}_n^Q$ and let us conduct the proof with a *reductio ab absurdum* by assuming that the system can evolve into a $\tilde{s}_{n+1} \notin \mathcal{S}_{n+1}^Q$. From (49), we have that S_n^Q and S_{n+1}^Q are given by:

$$S_n^Q = \big\{x \in \mathcal{S} : \lambda^n x \geq \lambda^{n+1}x + p(S-x)\big\} \tag{57}$$

$$S_{n+1}^Q = \big\{x \in \mathcal{S} : \lambda^{n+1} x \geq \lambda^{n+2}x + p(S-x)\big\} \tag{58}$$

Hence, after simple algebraic manipulations, it results:

$$s \in S_n^Q \Longrightarrow s \geq \frac{\lambda pS}{1-\lambda-\lambda p} \tag{59}$$

$$\tilde{s} \notin S_{n+1}^Q \Longrightarrow \tilde{s} < \frac{\lambda pS}{1-\lambda-\lambda p} \tag{60}$$

which constitutes a *reductio ab absurdum*, given that the system cannot evolve from s_n to $\tilde{s}_{n+1}$ with $\tilde{s} < s$.

$$\sum_{\breve{s}\geq\tilde{s}}\sum_{\tilde{s}\geq s}^{\breve{s}} p\big(\breve{s}_N|\tilde{s}_{n+1},C\big)p\big(\tilde{s}_{n+1}|s_n,C\big)g(\breve{s}_{n+1}) < \sum_{\tilde{s}\geq s} p\big(\tilde{s}_{n+1}|s_n,C\big)\left(-f(\tilde{s}_{n+1}) + \sum_{\breve{s}\geq\tilde{s}} p\big(\breve{s}_{n+2}|\tilde{s}_{n+1},C\big)v^*(\breve{s}_{n+2})\right) \tag{46}$$

C. Case III: rewards modeled as in (32).

Here we prove that the OLA rule is optimal when the rewards are modeled as in (32), namely, when:

$$g(s_n) = \frac{s}{S} - \frac{n}{N} \tag{61}$$

To this aim, let us assume $s_n \in \mathcal{S}_n^Q$ and let us conduct the proof with a *reductio ab absurdum* by assuming that the system can evolve into a $\tilde{s}_{n+1} \notin \mathcal{S}_{n+1}^Q$. From (49), we have that S_n^Q and S_{n+1}^Q are given by:

$$S_n^Q = \left\{ x \in \mathcal{S} : \frac{x}{S} - \frac{n}{N} \geq \frac{x+px}{S} + p + \frac{n+1}{N} \right\} \tag{62}$$

$$S_{n+1}^Q = \left\{ x \in \mathcal{S} : \frac{x}{S} - \frac{n+1}{N} \geq \frac{x+px}{S} + p + \frac{n+2}{N} \right\} \tag{63}$$

Hence, after simple algebraic manipulations, it results:

$$s \in S_n^Q \Longrightarrow s \geq S - \frac{S}{Np} \tag{64}$$

$$\tilde{s} \notin S_{n+1}^Q \Longrightarrow \tilde{s} < S - \frac{S}{Np} \tag{65}$$

which constitutes a *reductio ab absurdum*, given that the system cannot evolve from s_n to $\tilde{s}_{n+1}$ with $\tilde{s} < s$.

References

[1] M. Viscardi, J. Illiano, A. S. Cacciapuoti, and M. Caleffi, "Entanglement distribution in the quantum internet: an optimal decision problem formulation," *IEEE QCE23*, 2023, under review.

[2] J. Illiano, M. Caleffi, A. Manzalini, and A. S. Cacciapuoti, "Quantum internet protocol stack: a comprehensive survey," *Computer Networks*, vol. 213, 2022.

[3] J. Miguel-Ramiro, A. Pirker, and W. Dür, "Genuine quantum networks with superposed tasks and addressing," *npj Quantum Information*, vol. 7, p. 135, 09 2021.

[4] R. V. Meter, R. Satoh, N. Benchasattabuse, K. Teramoto, T. Matsuo, M. Hajdusek, T. Satoh, S. Nagayama, and S. Suzuki, "A quantum internet architecture," *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 341–352, sep 2022.

[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, 2020.

[6] S. Wehner, D. Elkouss, and R. Hanson, "Quantum Internet: a Vision for the Road Ahead," *Science*, vol. 362, no. 6412, 2018.

[7] W. Dür, R. Lamprecht, and S. Heusler, "Towards a quantum internet," *European Journal of Physics*, vol. 38, no. 4, p. 043001, 2017.

[8] H. J. Kimble, "The quantum internet," *Nature*, vol. 453, no. 7198, pp. 1023–1030, 2008.

[9] M. Caleffi, M. Amoretti, D. Ferrari, D. Cuomo, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, "Distributed quantum computing: a survey," *arXiv preprint arXiv:2212.10609*, 2022.

[10] C. Wang, A. Rahman, R. Li, M. Aelmans, and K. Chakraborty, "Application scenarios for the quantum internet," Internet Engineering Task Force, Internet-Draft draft-irtf-qirg-quantum-internet-use-cases-12, 2022, work in Progress.

[11] W. Kozlowski, S. Wehner, R. V. Meter, B. Rijsman, A. S. Cacciapuoti, M. Caleffi, and S. Nagayama, "Architectural Principles for a Quantum Internet," RFC 9340, Mar. 2023.

[12] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, "On the exact analysis of an idealized quantum switch," *ACM SIGMETRICS Performance Evaluation Review*, vol. 48, no. 3, pp. 79–80, 2021.

[13] ——, "On the stochastic analysis of a quantum entanglement distribution switch," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–16, 2021.

[14] Á. G. Iñesta, G. Vardoyan, L. Scavuzzo, and S. Wehner, "Optimal entanglement distribution policies in homogeneous repeater chains with cutoffs," *npj Quantum Information*, vol. 9, no. 1, p. 46, 2023.

[15] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, "Practical figures of merit and thresholds for entanglement distribution in quantum networks," *Phys. Rev. Research*, vol. 1, p. 023032, Sep 2019.

[16] M. Caleffi and A. S. Cacciapuoti, "Quantum switch for the quantum internet: Noiseless communications through noisy channels," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 3, pp. 575–588, 2020.

[17] A. S. Cacciapuoti, M. Caleffi, R. Van Meter, and L. Hanzo, "When entanglement meets classical communications: Quantum teleportation for the quantum internet," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3808–3833, 2020, invited paper.

[18] M. Epping *et al.*, "Multi-partite entanglement can speed up quantum key distribution in networks," *New J. Phys.*, 2017.

[19] G. Avis, F. Rozpedek, and S. Wehner, "Analysis of multipartite entanglement distribution using a central quantum-network node," *Phys. Rev. A*, vol. 107, p. 012609, Jan 2023.

[20] J. Illiano, M. Caleffi, M. Viscardi, and A. S. Cacciapuoti, "Design and analysis of genuine entanglement access control for the quantum internet," *arXiv preprint arXiv:2305.01276*, 2023, under review.

[21] H. Zhou *et al.*, "Parallel and heralded multiqubit entanglement generation for quantum networks," *Phy. Rev. A*, 2023.

[22] L. Bugalho *et al.*, "Distributing multipartite entanglement over noisy quantum networks," *quantum*, vol. 7, p. 920, 2023.

[23] S. Barz *et al.*, "Heralded generation of entangled photon pairs," *Nature photonics*, vol. 4, no. 8, pp. 553–556, 2010.

[24] J. Hofmann *et al.*, "Heralded entanglement between widely separated atoms," *Science*, vol. 337, no. 6090, pp. 72–75, 2012.

[25] C. H. Bennett *et al.*, "Capacities of quantum erasure channels," *Phys. Rev. Lett.*, vol. 78, pp. 3217–3220, Apr 1997.

[26] ——, "Entanglement-assisted classical capacity of noisy quantum channels," *Phys. Rev. Lett.*, vol. 83, Oct 1999.

[27] D. Bruss *et al.*, "Quantum entanglement and classical communication through a depolarizing channel," *J Mod Opt*, 2000.

[28] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2011.

[29] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[30] M. Abdel-Hameed, "Optimality of the one step look-ahead stopping times," *Journal of Applied Probability*, vol. 14, no. 1, pp. 162–169, 1977.

[31] M. Yasuda, "The optimal value of markov stopping problems with one-step look ahead policy," *Journal of Applied Probability*, vol. 25, no. 3, pp. 544–552, 1988.