

Received 9 June 2023; revised 23 December 2023; accepted 9 January 2024; date of publication 31 January 2024;
date of current version 19 June 2024.

Digital Object Identifier 10.1109/TQE.2024.3358674

Tools for the Analysis of Quantum Protocols Requiring State Generation Within a Time Window

BETHANY DAVIES^{1,2,3}, THOMAS BEAUCHAMP^{1,2,3},
GAYANE VARDYAN^{1,3}, AND STEPHANIE WEHNER^{1,2,3}

¹QuTech, Delft University of Technology, 2628 CJ Delft, The Netherlands

²Kavli Institute of Nanoscience, Delft University of Technology, 2628 CJ Delft, The Netherlands

³Quantum Computer Science, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CJ Delft, The Netherlands

Corresponding authors: Bethany Davies; Stephanie Wehner (e-mail: b.j.davies@tudelft.nl; s.d.c.wehner@tudelft.nl).

The work of Bethany Davies, Thomas Beauchamp, and Stephanie Wehner was supported by the Quantum Internet Alliance through EU Horizon Europe under Grant 101102140. The work of Gayane Vardyan was supported by the NWO ZK QSC Ada Lovelace Fellowship.

ABSTRACT Quantum protocols commonly require a certain number of quantum resource states to be available simultaneously. An important class of examples is quantum network protocols that require a certain number of entangled pairs. Here, we consider a setting in which a process generates a quantum resource state with some probability p in each time step and stores it in a quantum memory that is subject to time-dependent noise. To maintain sufficient quality for an application, each resource state is discarded from the memory after w time steps. Let s be the number of desired resource states required by a protocol. We characterize the probability distribution $X_{(w,s)}$ of the ages of the quantum resource states, once s states have been generated in a window w . Combined with a time-dependent noise model, knowledge of this distribution allows for the calculation of fidelity statistics of the s quantum resources. We also give exact solutions for the first and second moments of the waiting time $\tau_{(w,s)}$ until s resources are produced within a window w , which provides information about the rate of the protocol. Since it is difficult to obtain general closed-form expressions for statistical quantities describing the expected waiting time $\mathbb{E}(\tau_{(w,s)})$ and the distribution $X_{(w,s)}$, we present two novel results that aid their computation in certain parameter regimes. The methods presented in this work can be used to analyze and optimize the execution of quantum protocols. Specifically, with an example of a blind quantum computing protocol, we illustrate how they may be used to infer w and p to optimize the rate of successful protocol execution.

INDEX TERMS Performance analysis, quantum networks, scan statistics.

I. INTRODUCTION

It is common for quantum computing and quantum network protocols to require the simultaneous availability of a certain number of high-quality quantum resource states. In the domain of quantum networks, such resource states are typically entangled pairs of qubits, where the execution of protocols, such as entanglement distillation and many quantum network applications, require multiple entangled pairs to be available at the same time [1], [2], [3]. Another example of a resource state can be found in the domain of quantum computing, where magic state distillation relies on the presence of multiple initial magic states [4].

Here, we study the setting in which resource states are generated using a probabilistic process. In each time step, this

process succeeds in generating one resource state with probability p . If the state is prepared successfully, it is immediately stored in a quantum memory that is subject to time-dependent noise. The process is repeated until all s states required by a protocol are in memory. Such a setting is ubiquitous in quantum networking, and (photonic) quantum computing. A prime example is heralded entanglement generation, which is commonly used in present-day quantum networks (see, e.g., [5], [6], [7], [8], [9], and [10]).

If the noise is time dependent, this means that when a state is placed in a quantum memory its quality will degrade over time. In practice then, in order to deliver states of sufficient quality, one often imposes a window of w time steps within which all s states must be produced. If the states are produced

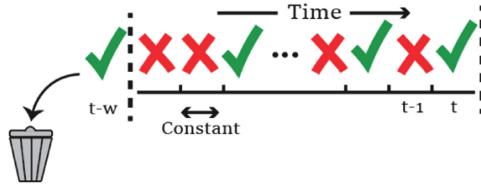


FIGURE 1. Setup. In each time step, a probabilistic process generates a resource state, where p is the probability of success (tick) and $1 - p$ the probability of failure (cross). After generation, the resource state is immediately placed into a quantum memory subject to time-dependent noise. To ensure the states have sufficient quality to enable a quantum protocol, states that are older than a specific window w of time steps are discarded (bin).

within the desired window, the quality of the states is high enough for the application to succeed. Otherwise, the states are typically discarded (see Fig. 1). In the context of quantum repeater protocols, such a window size is also often referred to as a “cutoff time,” and the analysis across multiple nodes is generally nontrivial [11], [12], [13], [14], [15], [16]. In the context of repeater chains, the goal is typically to deliver one state at a high rate. This is different from our case, where the goal is to deliver multiple states. If a protocol requires s quantum resource states of sufficiently high quality to exist simultaneously, this translates to a requirement of s successful generation events within the window of w time steps. The motivation of this work is to quantify the effects of noise on a quantum protocol—we consider a time window because states may be subjected to time-dependent noise in memory, and therefore must be discarded before they are too old. We remark that our methods apply to many different types of hardware, including those with long coherence times [17].

When analyzing the performance of protocols that rely on such a generation of resource states, we are interested in a number of performance metrics. For example, one may be interested in the rate at which we can execute a protocol, the probability that the overall quantum protocol will be successful, or a combined metric that considers the number of successful executions of the quantum protocol per time unit. To understand and optimize such performance metrics, we are interested in understanding a number of quantities related to the system’s ability to prepare the resource states required by the protocol.

Firstly, one may consider the *waiting time* $\tau_{(w,s)}$ until there are s successes within a window of w time steps. We remark that for fixed parameters (w, s, p) this provides us also with information about the rate at which a protocol can be carried out, when executed multiple times. Secondly, we look at the *ending pattern* $X_{(w,s)}$ (see Fig. 2), which contains the ages of the s quantum resources at time $\tau_{(w,s)}$. Combined with a model of decoherence, this can be used to compute the quality (fidelity) of the resources immediately after the last state has been produced, which is when the quantum protocol may be executed. Obtaining the distribution of $X_{(w,s)}$, then, gives the distribution of the fidelities of the resource states.

The goal of this work is to provide tools that may be used to analyze the performance of a given quantum protocol for

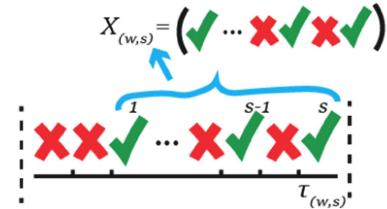


FIGURE 2. Ending patterns. At the first instance $\tau_{(w,s)}$ when a window of w time steps contains s successes, we are interested in how long ago each of the s links were generated. This allows one to quantify the quality of the corresponding resource states. The information of when the s resource states were produced is contained in the ending pattern $X_{(w,s)}$.

specific choices of w , s , and p , as well as to choose a combination of these parameters to optimize its performance.

A. RESULTS

Our main contributions are summarized below.

- 1) For all values of w , s , and p , we provide formulae for the first and second moments of $\tau_{(w,s)}$ (and therefore, its mean and variance), and the full distribution of $X_{(w,s)}$. For $(w, s) = (\infty, s)$, these are in a simple closed form, and similarly for $(w, s) = (w, 2)$. For all other values of (w, s) , we present general formulae, which are in the form of a linear system that may be solved numerically. The dimension of the system scales as w^{s-1} . For large w and s , it is therefore difficult to obtain closed-form expressions from these systems.
- 2) We provide an efficient method to find bounds on the range of w and p for which $\mathbb{E}(\tau_{(w,s)})$ and $X_{(w,s)}$ may be approximated by $\mathbb{E}(\tau_{(\infty,s)})$ and $X_{(\infty,s)}$ to an arbitrary degree of accuracy. In a practical context, this allows one to quickly compute thresholds on the window size such that increasing w further provides no improvement for a protocol rate. Moreover, for appropriate parameter regimes, this approximation is desirable due to the fact that the dimension of the linear system to solve for the expected waiting time $\mathbb{E}(\tau_{(w,s)})$ and $X_{(w,s)}$ scales with w and s , as described above. This is in contrast to the simple closed-form expressions that can be found for the corresponding quantities in the case $w = \infty$.
- 3) We characterize the behavior of $\mathbb{E}(\tau_{(w,s)})$ and $X_{(w,s)}$ in the limit of a small probability of success. In particular, we show that as $p \rightarrow 0$, $\mathbb{E}(\tau_{(w,s)})$ scales as p^{-s} , and that the distribution of $X_{(w,s)}$ becomes uniform. This result may be used to gain intuition about the performance of a quantum application when the resource generation success probability is small, without needing to perform (potentially lengthy) numerical computations.
- 4) We provide a demonstration of how these methods may be used in practice. We consider a blind quantum computation (BQC) protocol [18]. In our model, entanglement is consumed in the transmission of qubits from a

client device to a server device. The model includes noise due to imperfect entangled links and memory decoherence. For a setup involving a computation on four qubits, we provide an example of how the methods from Section III may be used to choose architecture parameters that optimize the rate of the protocol.

B. RELATED WORK

To obtain our results, we draw on methods used in the mathematical literature known as scan statistics [19], [20]. Scan statistics is typically concerned with patterns and clusters in a sequence of random events. This is a field that incorporates techniques from multiple different areas of mathematics. In the quantum context, the problem is different from other areas in caring about the ending pattern distribution. In this work, we therefore use the approach that makes use of martingales, because this allows one to obtain both $\mathbb{E}(\tau_{(w,s)})$ and the distribution of $X_{(w,s)}$ [21]. It is possible to obtain the same quantities with embedded Markov chains [22], but we continue here with the martingale method; firstly because the resulting formula has a smaller dimension (it is therefore faster to compute the quantities of interest), and second because it has a regular structure that allows us to derive asymptotic results for small p , which is an experimentally relevant regime. One of the aims of this work is to characterize both $\tau_{(w,s)}$ and the distribution of the ending pattern $X_{(w,s)}$. However, if one is only interested in $\tau_{(w,s)}$ and not $X_{(w,s)}$, then there exist other methods to compute $\mathbb{E}(\tau_{(w,s)})$, and also in principle the full distribution of $\tau_{(w,s)}$ —see, e.g., [23] or [24], which give formulae to obtain the probability generating function of $\tau_{(w,s)}$. To the best of the authors' knowledge, however, these also result in a large system of equations, and not the ending pattern distribution. We therefore do not provide details of these other methods in this work. Other quantities related to the distribution of $\tau_{(w,s)}$ have been explored in great depth in the scan statistics literature, which may also have relevance to the quantum domain. For example, there exist a number of bounds and approximations for $\mathbb{P}(\tau_{(w,s)} \leq n)$ (see, e.g., [19] for an overview of results), which may prove useful in allocating time for entanglement generation in a quantum network schedule. By contrast, in this work, we focus on the behavior of $\mathbb{E}(\tau_{(w,s)})$ and $X_{(w,s)}$, and their implications for the performance of quantum protocols. To the best of the authors' knowledge, this work is the first to characterize the behavior of the ending pattern distribution in certain parameter regimes, and demonstrate an explicit example of the application of results from scan statistics to a quantum protocol.

C. OUTLINE

The rest of this article is organized as follows. In Section II, the quantities $\tau_{(w,s)}$ and $X_{(w,s)}$ are formally defined. In Sections III-A and III-B, we give formulae for the first and second moments of $\tau_{(w,s)}$, and the distribution of $X_{(w,s)}$. In

Sections III-C and III-D, we present results that aid the understanding and approximation of these quantities. In Section IV-A, the behavior and practical relevance of the results of Section III are outlined, specifically looking at $\mathbb{E}(\tau_{(w,s)})$. In Section IV-B, we give an example of how one may use the results of Section III to choose architecture parameters that optimize the performance of a BQC protocol. Finally, further directions are summarized in Section V.

II. PRELIMINARIES

We view quantum resource generation attempts as a sequence of i.i.d. Bernoulli trials $(Z_i)_{i=1}^\infty$ with success probability $p = \mathbb{P}(Z_1 = 1) > 0$. Then, if a protocol requires $s \leq w$ quantum resources to coexist, the time taken to complete the application is dependent on the *waiting time* $\tau_{(w,s)}$ to produce s successes within a window of size w . We are also interested in the *ending pattern* $X_{(w,s)}$, which completes the process, because this contains the ages of the s quantum resources present at time $\tau_{(w,s)}$. We denote the set of possible ending patterns as $\Omega(w, s)$. This contains every possible configuration of the s successes within the scanning window, so that $X_{(w,s)} \in \Omega(w, s)$. A visualization of how an ending pattern realizes the end of the process is given in Fig. 2. More specifically, we define

$$\Omega_l(s) := \left\{ x \in \{0, 1\}^l : x_1 = x_l = 1 \wedge \sum_{i=1}^l x_i = s \right\} \quad (1)$$

to be the set of all length- l binary strings $x = (x_1, \dots, x_l)$ that contain s successes, two of which occur at either end of the string. Then

$$\Omega(w, s) := \bigcup_{l=s}^w \Omega_l(s) \quad (2)$$

is the set of ending patterns. The set $\Omega(w, s)$ can be thought of as containing all clusters of s successes that were produced within a time less than or equal to w time steps. Note that the number of possible ending patterns is given by

$$|\Omega(w, s)| = \binom{w-1}{s-1}. \quad (3)$$

To see this, consider the fact that each ending pattern in $\Omega(w, s)$ corresponds uniquely to an ending scenario where the s successes are distributed within the window of w time steps, as may be seen from Fig. 2. Since the final quantum resource must always have been prepared at the most recent time step and therefore is fixed, it remains to distribute the remaining $s-1$ successes within $w-1$ time steps, meaning that the number of possible ending patterns is restricted to (3). The waiting time $\tau_{(w,s)}$ is then defined by

$$\tau_{(w,s)} := \min_{x \in \Omega(w,s)} \{\tau_x\} \quad (4)$$

i.e., this is the time until we see the first ending pattern in the sequence of Bernoulli trials. Here, τ_x is the time taken until one particular ending pattern x is first seen, so that for

$$x \in \Omega_l(s) \subset \Omega(w, s)$$

$$\tau_x := \min\{t : (Z_{t-l+1}, Z_{t-l+2}, \dots, Z_t) = x\}. \quad (5)$$

We note that $\tau_{(w,s)}$ is well defined because it is bounded above by a geometric random variable (see Appendix B1). There is also a relationship between $\tau_{(w,s)}$ and the distribution of $X_{(w,s)}$ given by

$$\mathbb{P}(X_{(w,s)} = x) = \mathbb{P}(\tau_{(w,s)} = \tau_x) \quad (6)$$

recalling that $X_{(w,s)}$ takes the value of the ending pattern that completes the process. No two ending patterns can realize the end of the process at the same time since no element of $\Omega(w, s)$ contains another, and so $X_{(w,s)}$ is well defined.

III. FORMULAE AND APPROXIMATIONS

In the following two sections, we provide exact solutions for the first and second moments of $\tau_{(w,s)}$, and the full distribution of $X_{(w,s)}$. Formulae are provided for all possible values of w and s . In Section III-C, we look at approximating the solutions for a large w . In Section III-D, we characterize the solution behavior for small p .

A. INFINITE WINDOW

Here, we consider the case where no resource states are discarded (or equivalently when $w = \infty$) and give solutions for the first and second moments of $\tau_{(\infty,s)}$, and the distribution of $X_{(\infty,s)}$. This serves as a useful initial study of the problem, providing intuition for the case where w is large and finite.

When no states are discarded, the waiting time to see all of the successes simply becomes a sum of s i.i.d. geometric random variables with parameter p . This is known as a *negative binomial* distribution, and has an exact distribution given by

$$\mathbb{P}(\tau_{(\infty,s)} = n) = \binom{n-1}{s-1} (1-p)^{n-s} p^s \quad (7)$$

and expectation

$$\mathbb{E}(\tau_{(\infty,s)}) = \frac{s}{p}. \quad (8)$$

Note that for $w' > w$, it is always the case that $\tau_{(w',s)} \leq \tau_{(w,s)}$ (increasing the window size must always decrease the time for the process to complete), and so

$$\mathbb{E}(\tau_{(w',s)}) \leq \mathbb{E}(\tau_{(w,s)}), \text{ for } w' > w. \quad (9)$$

In particular, the waiting time for a finite w will always be greater than or equal to the infinite case. Then, using (8) and (9), we obtain a simple lower bound in terms of s and p

$$\mathbb{E}(\tau_{(w,s)}) \geq \frac{s}{p}. \quad (10)$$

The variance of $\tau_{(\infty,s)}$ is given by

$$\text{Var}(\tau_{(\infty,s)}) = \frac{s \cdot (1-p)}{p^2} \quad (11)$$

from which we can see that the standard deviation is reciprocal in p .

It is also possible to derive a simple expression for the distribution of $X_{(\infty,s)}$. For a binary string $x \in \Omega_l$ that lives in the (now infinite) set of ending patterns $\Omega(\infty, s)$

$$\mathbb{P}(X_{(\infty,s)} = x) = (1-p)^{l-s} p^{s-1} \quad (12)$$

which can be seen by considering the probability of generating the remaining $l-1$ entries of B after the first success has been generated. We see that when the window size is infinite, the probability of generating ending patterns of the same length is constant.

B. FINITE WINDOW

1) $s = 2$

When $s = 2$, it is possible again to derive closed-form solutions for the first and second moments of $\tau_{(w,s)}$ and the distribution of $X_{(w,s)}$. We present below the formulae for $\mathbb{E}(\tau_{(w,s)})$ and the ending pattern distribution.

In this case, the ending patterns are determined by the time between the two states, i.e., $|\Omega_l(2)| = 1$. We separate the process of resource generation into two parts: 1) generation of the first state and 2) generation of the second state. Generation of the first link occurs when there is no state stored in memory. This is not limited by the window, and has a generation time described by a geometric distribution with parameter p . To finish the process, the generation of the second state must happen within $w-1$ time steps of the first link being generated. When the process is finished, the time between the two states is then a geometric distribution conditional on this event, which occurs with probability $1 - (1-p)^{w-1}$. Then, letting $L \in \{1, \dots, w-1\}$ be the number of attempts after the first state to generate the second

$$\mathbb{P}(L = n) = \frac{(1-p)^{n-1} p}{1 - (1-p)^{w-1}} \quad (13)$$

which gives the full ending pattern distribution, where $L = n$ corresponds to $X_{(w,s)} \in \Omega_{n+1}(2)$.

Now, let M be the number of times a first state must be generated until the process is finished. Since this is determined by the success of the second state within the time window, we have

$$M \sim \text{Geom}\left(1 - (1-p)^{w-1}\right). \quad (14)$$

The total time is then given in terms of M and L by

$$\tau_{(w,2)} = \sum_{j=1}^M T_j + (M-1)(w-1) + L \quad (15)$$

where the random variables $T_j \sim \text{Geom}(p)$ describe the number of attempts to generate the first state. Now, as shown in Appendix A

$$\mathbb{E}\left(\sum_{j=1}^M T_j\right) = \mathbb{E}(M)\mathbb{E}(T_1) = \frac{1}{(1 - (1-p)^{w-1})} \cdot \frac{1}{p} \quad (16)$$

and

$$\mathbb{E}(L) = \frac{1 - (1-p)^w - wp(1-p)^{w-1}}{p(1 - (1-p)^{w-1})}. \quad (17)$$

The expected waiting time may then be computed as

$$\mathbb{E}(\tau_{(w,2)}) = \mathbb{E}(M)\mathbb{E}(T_1) + (\mathbb{E}(M) - 1)(w - 1) + \mathbb{E}(L) \quad (18)$$

from which we obtain

$$\mathbb{E}(\tau_{(w,2)}) = \frac{1}{p} + \frac{1}{p(1 - (1-p)^{w-1})}. \quad (19)$$

The variance of $\tau_{(w,2)}$ may also be computed by making use of (15). The computation is given in Appendix A.

2) $s > 2$

We now give a formula to exactly compute $\mathbb{E}(\tau_{(w,s)})$ and the full distribution $\{\mathbb{P}(X_{(w,s)} = x) : x \in \Omega(w, s)\}$, for a finite w . This is derived using the method from [21], which makes use of martingales. For completeness, we include an outline of the derivation in Appendix B2, where a gambling analogy is introduced to aid understanding. The resulting formula is in the form of a linear system of size $|\Omega(w, s)| + 1$ that can be solved exactly. Each matrix element defining the linear system can be computed simply and efficiently. Further, in Appendix B4, we give a formula for the second moment of $\tau_{(w,s)}$, which now involves two linear systems of size $|\Omega(w, s)|$. A martingale method is also used for its derivation, and for this we refer to [24]. The second moment of $\tau_{(w,s)}$ can then be used to calculate the variance and standard deviation of $\tau_{(w,s)}$.

Before stating the first formula, we introduce some notation. We define a function $*$ that maps two binary strings $x = (x_1, \dots, x_k)$ and $y = (y_1, \dots, y_m)$ to a scalar value, given by

$$x * y := \sum_{j=1}^{\min(k,m)} \left(\prod_{i=1}^j \delta_{(x_i, y_{m-j+i})} \right) \quad (20)$$

where for $a, b \in \{0, 1\}$, the quantity $\delta_{a,b}$ is defined as

$$\delta_{(a,b)} = \begin{cases} \frac{1}{p_a} & \text{if } a = b \\ 0, & \text{otherwise} \end{cases} \quad (21)$$

where $p_1 := p$ and $p_0 := 1 - p$. From (20), we see that the value of $x * y$ is obtained by comparing the overlap of successive substrings of x and y . If two substrings match exactly, then the corresponding term is included in the sum, and it is weighted by an amount that is dependent on the Bernoulli parameter p . Informally, then, $x * y$ measures how similar the structures of x and y are. A simple example of the action of $*$ is given as follows. We consider the action of $*$ on two ending patterns, recalling (1). Letting $s = 3$ and $w \geq 7$; suppose that $x = 1010001$ and $y = 100011$. Computing (20) then yields

$$x * y = \frac{1}{p_1} + 0 + 0 + 0 + \frac{1}{p_1^2} \cdot \frac{1}{p_0^3} + 0 = \frac{1}{p_1} + \frac{1}{p_1^2 p_0^3}.$$

Since all elements of $\Omega(w, s)$ start and finish with a success by their definition in (1), the same initial $1/p$ term will be present for any pair of ending patterns. Whether or not there are higher order terms depends on the overlap of the successive substrings. In particular, for two ending patterns $x, y \in \Omega(w, s)$, the quantity $x * y$ will be of order $1/p^s$ if and only if $x = y$.

Equipped with these definitions, we now give the formula for the expected waiting time and the ending pattern distribution.

Theorem 1: Let $N := |\Omega(w, s)|$. After enumerating the ending patterns as $\Omega(w, s) \equiv \{x^{(i)} : i = 1, \dots, N\}$, let

$$\vec{v} = \begin{pmatrix} \mathbb{E}(\tau_{(w,s)}) \\ \mathbb{P}(X = x^{(1)}) \\ \mathbb{P}(X = x^{(2)}) \\ \vdots \\ \vdots \\ \mathbb{P}(X = x^{(N)}) \end{pmatrix}. \quad (22)$$

Then

$$A\vec{v} = \vec{e}_1 \quad (23)$$

where $\vec{e}_1 := (1, 0, \dots, 0)^T$ is a vector of length $N + 1$, and

$$A := \begin{pmatrix} 0 & 1 & 1 & \dots & 1 \\ -1 & x^{(1)} * x^{(1)} & x^{(1)} * x^{(2)} & \dots & x^{(1)} * x^{(N)} \\ -1 & x^{(2)} * x^{(1)} & x^{(2)} * x^{(2)} & \dots & x^{(2)} * x^{(N)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ -1 & x^{(N)} * x^{(1)} & \dots & x^{(N)} * x^{(N)} \end{pmatrix}. \quad (24)$$

The matrix A is invertible since no element of $\Omega(w, s)$ contains another [21]. The solution for \vec{v} therefore always exists and is unique. One may then solve the linear system (23) to obtain $\mathbb{E}(\tau_{(w,s)})$ and the ending pattern distribution. Code is provided in [25] that performs this task. The matrix A is fully determined by the success probability p and the parameters w and s . From (3), the dimension of the square matrix A is $|\Omega(w, s)| + 1 = \mathcal{O}(w^{s-1})$, and so the complexity of this task is increasingly difficult for large w and s . In the following sections, we derive results that aid the understanding and computation of the corresponding results in the two characteristic regimes of large w and small p . Moreover, by the definition of the star product as given in (20), we see that each entry of A is a polynomial in $1/p$ and $1/(1-p)$. As discussed previously, each entry of the submatrix formed by removing the first row and column is greater than or equal to $1/p$, due to properties of the operator $*$. Entries that do not take this exact value contain higher order terms in $1/p$, due to the fact that there is a greater overlap of the ending patterns corresponding to the row and column indices of such an entry. The entries of the highest power in $1/p$ are exactly the diagonal elements and are of order s , because a string overlaps completely with itself and contains s successes. We note that in principle, the solutions for \vec{v} can be computed

analytically as functions of p by inverting A directly. However, due to the scaling of the system with w and s , this is in practice computationally laborious.

C. APPROXIMATING WITH AN INFINITE WINDOW

Now, one might ask: how large must the window size be for the approximation $w = \infty$ to be accurate? This is desirable due to the simple analytical form of the results for the distributions of $\tau_{(\infty,s)}$ and $X_{(\infty,s)}$, as seen in Section III-A. This is in contrast to the solutions presented in Section III-B for the case of a finite w , which scale with w and s . The approximation becomes valid when the window size has “saturated” the process, so that increasing the window size does not provide any significant improvement for the rate. Alternatively, the approximation becomes accurate when $\mathbb{P}(\tau_{(w,s)} > w)$ is small. This intuition is formalized with the following theorem.

Theorem 2: Let $\tau_{(w,s)}$ be the waiting time for s successes in a w -window. Let $X_{(w,s)}$ be the corresponding ending pattern. Let p denote the success probability of each trial. Let $\epsilon(w, s, p) := \mathbb{P}(\tau_{(w,s)} > w)$. Suppose that $0 < p < 1$ and $w < \infty$. Then

$$\frac{\mathbb{E}(\tau_{(w,s)}) - \mathbb{E}(\tau_{(\infty,s)})}{\mathbb{E}(\tau_{(w,s)})} < \epsilon(w, s, p) \quad (25)$$

and

$$\sum_{x \in \Omega(w,s)} |\mathbb{P}(X_{(w,s)} = x) - \mathbb{P}(X_{(\infty,s)} = x)| < 2\epsilon(w, s, p). \quad (26)$$

We now look to evaluate $\epsilon(w, s, p)$. Looking back at the identity (7) for the distribution of $\tau_{(\infty,s)}$, we have

$$\mathbb{P}(\tau_{(w,s)} > w) = \mathbb{P}(\tau_{(\infty,s)} > w) \quad (27)$$

$$= \sum_{n=w+1}^{\infty} \binom{n-1}{s-1} (1-p)^{n-s} p^s. \quad (28)$$

To evaluate the right-hand side of (25), it is convenient to rewrite this as a finite sum, as provided by the following lemma. The proof of this is given in Appendix C1.

Lemma 1: Let $\tau_{(w,s)}$ be the waiting time for s successes in a w -window, as defined in (4). Suppose that $0 < p < 1$ and $w < \infty$. Then

$$\mathbb{P}(\tau_{(w,s)} > w) = \sum_{i=0}^{s-1} \binom{w}{i} (1-p)^{w-i} p^i. \quad (29)$$

This sum is simple and efficient to evaluate for constant s , and can then be used to find a range of w for which the two expectations are close. For example, if one is interested in evaluating $\mathbb{E}(\tau_{(w,s)})$, when in fact w is large enough such that it may be reasonably approximated by $\mathbb{E}(\tau_{(\infty,s)}) = s/p$, it is possible avoid solving a large system of equations with the following method. Demanding some desired error δ , one may quickly compute

$$w^* = \min \{w : \epsilon(w, s, p) < \delta\}. \quad (30)$$

Then, for all $w \geq w^*$, one may approximate $\mathbb{E}(\tau_{(w,s)})$ with $\mathbb{E}(\tau_{(\infty,s)}) = s/p$ with accuracy on the order of δ . The same can be done with the ending pattern distribution: if one is interested in the expectation of some fidelity quantity $F(X_{(w,s)})$, one may also approximate $\mathbb{E}(F(X_{(w,s)}))$ with $\mathbb{E}(F(X_{(\infty,s)}))$ with the same accuracy.

D. ASYMPTOTIC BEHAVIOR OF THE EXPECTATION

From (9), an upper bound for $\mathbb{E}(\tau_{(w,s)})$ is given by $\mathbb{E}(\tau_{(s,s)})$, which can be written in a simple analytical form. In the case $w = s$, there is only one ending pattern x , which corresponds to the case of s consecutive successes. From (23), we then have

$$\mathbb{E}(\tau_{(s,s)}) = x * x = \sum_{j=1}^s \frac{1}{p^j} = \frac{1/p^s - 1}{1 - p} \quad (31)$$

which for small p satisfies $\mathbb{E}(\tau_{(s,s)}) \sim 1/p^s$. In comparison, from (8), the scaling of the expectation for $w = \infty$ is reciprocal in p . Further, from the form of A given in (24), all entries of \vec{v} will be ratios of polynomials in $1/p$. Looking at the first component of \vec{v} , which is the waiting time expectation, this tells us that there is some integer value $\alpha_s(w)$ that dominates the scaling for small p , so that

$$\mathbb{E}(\tau_{(w,s)}) \sim \frac{c(w, s)}{p^{\alpha_s(w)}} \quad (32)$$

where $c(w, s)$ is a constant. Now, recalling from (9) that $\mathbb{E}(\tau_{(w,s)})$ is a decreasing function of w , we therefore expect the same of $\alpha_s(w)$, which satisfies $\alpha_s(s) = s$ and $\alpha_s(\infty) = 1$. Below we show that for $w < \infty$, $\alpha_s(w)$ is always equal to s , and also derive the scaling factor $c(w, s)$.

Theorem 3: Let $\tau_{(w,s)}$ be the waiting time for s successes in a w -window, as defined in (4). Let $X_{(w,s)}$ be the corresponding ending pattern. Let p be the success probability of the process. Then, in the limit $p \rightarrow 0$

$$\mathbb{E}(\tau_{(w,s)}) \sim \frac{1}{|\Omega(w, s)| p^s} \quad (33)$$

and for all $x \in \Omega(w, s)$

$$\mathbb{P}(X_{(w,s)} = x) \rightarrow \frac{1}{|\Omega(w, s)|} \quad (34)$$

where $|\Omega(w, s)| = \binom{w-1}{s-1}$ is the number of possible ending patterns.

A proof of Theorem 3 is given in Appendix C2. It is interesting future work to quantify the speed of convergence of (33) and (34).

As intuition for (34), note that for very small p , the probability of having w failures preceding the ending pattern is high. In this case, the ending pattern distribution is equivalent to the ending pattern distribution given we succeed in w attempts, which in the limit of small p converges to the uniform distribution.

The behavior captured by Theorems 2 and 3 may be viewed as two limiting behaviors of the problem in the regimes of small and large p , respectively. In particular, we

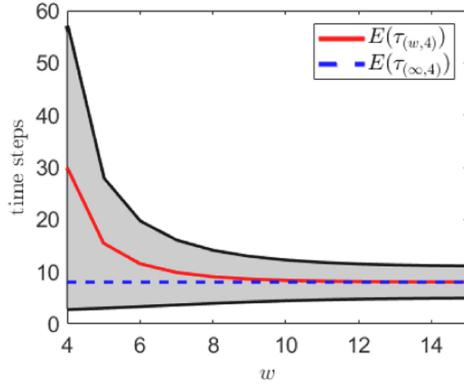


FIGURE 3. How $\mathbb{E}(\tau_{(w,4)})$ varies with w . We see that $\mathbb{E}(\tau_{(w,4)})$ (red line) converges to the lower bound $\mathbb{E}(\tau_{(\infty,4)}) = 4/p$ (blue line) as w becomes large. The gray region is one standard deviation of $\tau_{(w,4)}$ above and below its expectation (for $w < \infty$). All quantities are evaluated with a success probability $p = 0.5$.

expect that the formula provided by Theorem 1 becomes useful in neither regime, i.e., when p is neither too small nor too large to apply either approximation. Moreover, it is important to keep in mind that such a regime will depend on the choices of w and s .

IV. ILLUSTRATION AND APPLICATION

We expect the methods presented in the above sections to be useful in choosing the optimal window size for a quantum protocol. To this end, we firstly analyze in more detail the behavior of $\mathbb{E}(\tau_{(w,s)})$. We then demonstrate how these methods may be used to optimize the performance of a BQC protocol.

A. ILLUSTRATION

We fix $s = 4$ as an example to showcase the characteristic behaviors of the expected waiting time. From our investigations, the solutions for other values of s display the same qualitative behavior. To produce each figure, we compute $\mathbb{E}(\tau_{(w,s)})$ by numerically solving the linear system (23) for the specific choices of w , s , and p . Recall that the size of this linear system scales as $\mathcal{O}(w^{s-1})$. The value $s = 4$ is small enough so that for the w values that we consider, the complexity of the problem is not too large to be solved on a laptop.

In Fig. 3, $\mathbb{E}(\tau_{(w,4)})$ is plotted against w , with the success probability set to $p = 0.5$. We notice the convergence to the $w = \infty$ lower bound. The gray region is that given by one standard deviation above and below the expectation. Note that one also expects the standard deviation to converge to that of $\tau_{(\infty,4)}$, which is given in a closed form by (11).

In Fig. 4, $\mathbb{E}(\tau_{(w,4)})$ is again plotted against w , but this time for three different values of the success probability. In each case, the solution again approaches the corresponding $w = \infty$ lower bound. Each line starts at $\mathbb{E}(\tau_{(4,4)}) = (1/p^s - 1)/(1 - p)$, corresponding to $w = s$, and converges to the $w = \infty$ limit. This convergence is an important feature, because at some point increasing w provides no significant

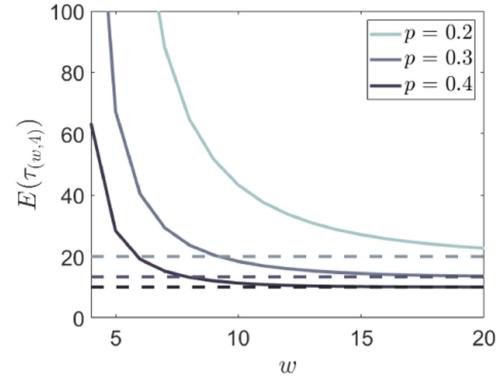


FIGURE 4. How $\mathbb{E}(\tau_{(w,4)})$ varies with w and p . We see that for larger p , $\mathbb{E}(\tau_{(w,4)})$ (solid line) approaches the lower bound $\mathbb{E}(\tau_{(\infty,4)}) = 4/p$ (dashed line) at a higher rate.

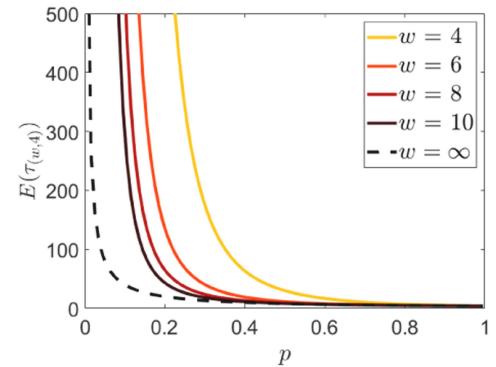


FIGURE 5. How $\mathbb{E}(\tau_{(w,4)})$ varies with p . We see that $\mathbb{E}(\tau_{(w,4)})$ (solid lines) demonstrates the reciprocal scaling as $p \rightarrow 0$, as encapsulated by Theorem 3. There is convergence to $\mathbb{E}(\tau_{(\infty,4)})$ (dashed line). This plot was made by discretizing p into 100 points, evenly spaced in the range $(0, 1)$.

improvement for the protocol rate. As one would expect intuitively, the convergence occurs more quickly for a larger p , as increasing the window size effectively saturates the problem more easily. To quantify this, we can use the arguments of Section III-C. For example, taking the desired margin of error to be 2%, define

$$w^* = \min \{w : \epsilon(w, s, p) < 0.02\} \quad (35)$$

where $\epsilon(w, s, p)$ is given by (29). By Theorem 2 and Lemma 1, the approximation $\mathbb{E}(\tau_{(w,s)}) \approx \mathbb{E}(\tau_{(\infty,s)})$ is then valid to the same margin of error for all $w > w^*$. It is interesting to see how this compares to the smallest window size w_{true}^* for which the same approximation can be made, which is defined formally as

$$w_{\text{true}}^* = \min \left\{ w : \frac{\mathbb{E}(\tau_{(w,s)}) - \mathbb{E}(\tau_{(\infty,s)})}{\mathbb{E}(\tau_{(w,s)})} < 0.02 \right\}. \quad (36)$$

The value w^* is then an upper bound for w_{true}^* . For example, letting $p = 0.5$ and $s = 4$ yields $w^* = 15$, and checking with the exact solutions gives $w_{\text{true}}^* = 12$. These are plotted for more values of p in Fig. 5. We see from the plot that as p increases, the bound appears to become tighter. The value

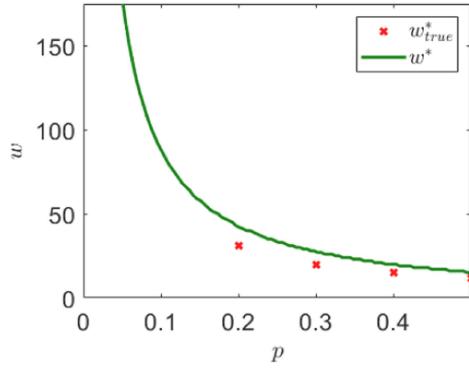


FIGURE 6. Comparison of thresholds on w for the infinite window approximation. One may use w^* (green line) as a threshold, which is more easily computable than w_{true}^* (red cross). See (35) and (36) for the definition of these quantities. Here, we assume a desired error of 2%.

w_{true}^* was plotted for only a few select values of p because its calculation is computationally intensive.

In Fig. 6, $\mathbb{E}(\tau_{(w,4)})$ is plotted against p for five different values of the window size. One can see that the scaling of $\mathbb{E}(\tau_{(w,4)})$ occurs more slowly for a larger w . This indicates the reciprocal behavior as given in (33), where $\mathbb{E}(\tau_{(w,4)}) \sim 1/|\Omega(w, 4)|p^4$, and in the case of a larger w the constant $|\Omega(w, 4)|$ suppresses the scaling. As $p \rightarrow 1$, all plots simply converge to $s = 4$, because in this case the process is deterministic. Further, we see again the convergence of the expectation to the infinite window limit. If p becomes large, we expect the problem to “saturate” in the same sense as before, so that $\mathbb{E}(\tau_{(w,s)}) \approx \mathbb{E}(\tau_{(\infty,s)})$. The speed of this convergence can again be quantified using the results of Section III-C. Demanding the same error of 2%, we take the p^* that satisfies

$$p^* = \inf \{p : \epsilon(w, s, p) < 0.02\} \quad (37)$$

or equivalently, p^* is the unique value of p such that $\epsilon(w, s, p^*) = 0.02$. The value p^* is an upper bound for the true threshold p_{true}^*

$$p_{\text{true}}^* := \inf \left\{ p : \left(\frac{\mathbb{E}(\tau_{(w,s)}) - \mathbb{E}(\tau_{(\infty,s)})}{\mathbb{E}(\tau_{(w,s)})} \right) \Big|_p < 0.02 \right\} \quad (38)$$

where we now include dependence of the expectations on the success probability p . In Fig. 7, p^* and p_{true}^* are plotted against w . The value p_{true}^* is computationally intensive to find for large values of w , and has therefore only been plotted for selected small values of w . The bound p^* , however, is efficient to compute. We observe that the bound appears to be tighter for smaller w .

B. APPLICATION TO A BQC PROTOCOL

In the following, we provide an example of how the results from Section III may be used in the performance analysis of a quantum network application. We consider a verifiable BQC protocol [18]. This involves a client, who uses a more powerful server device to carry out a bounded error quantum

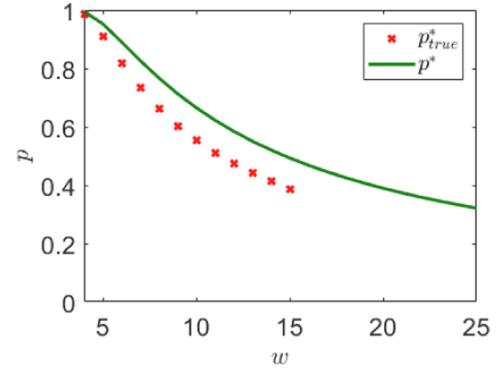


FIGURE 7. Comparison of thresholds on p for the infinite window approximation. One may use p^* (green line) as a threshold, which is more easily computable than p_{true}^* (red cross). See (37) and (38) for the definition of these quantities. Here, we assume a desired error of 2%.

polynomial-time (BQP) computation [26], which is specified in the measurement-based formalism [27]. In this formalism, the computation is defined with respect to a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. The computation is performed by firstly creating a graph state corresponding to G , and then applying a series of measurements (“measurement flow”) to a subset of qubits. The BQC protocol is designed such that the server remains ignorant of the client’s desired computation (blindness). Further, it ensures that the client can validate that the outcome is correct, even in the presence of some amount of noise or a malicious server (verifiability). These properties are stated precisely in terms of the composable security properties of the protocol [28]. For the protocol in full detail, we refer to [18]. Here, we provide a short outline of the BQC protocol, and a simple model of how it is carried out. We then apply the results of Section III to study the performance of the protocol.

1) PROTOCOL FEASIBILITY

The BQC protocol involves a series of rounds. In each round, the client sends $|V|$ qubits to the server, and also a description of the measurement flow it should carry out. If the server is honest, it will then create a graph state by applying entangling gates corresponding to edges in E , carry out the corresponding measurement flow, and send the measurement outcomes back to the client.

The protocol involves interweaving two types of rounds: *computation* and *test* rounds. The computation rounds are used to carry out the client’s desired computation. In these rounds, the computation measurement flow is encrypted in order to maintain blindness. The function of the test rounds is to check for deviations from the client’s specified operations. Deviations could be due to noise, or the server being malicious. Each test round has the outcome of either pass or fail, and the protocol is aborted if the ratio of failed test rounds lies above a certain threshold.

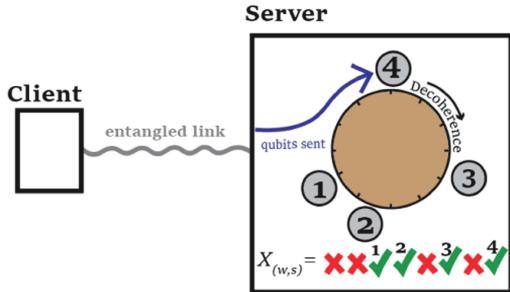


FIGURE 8. Scenario considered for the model of BQC. Generation of the entangled link is attempted sequentially, with success probability p . Upon success, the entangled link is immediately used to transmit qubits from the client to the server. While in the server, qubits (numbered gray circles) undergo decoherence (brown arrow). Qubits are discarded from the server after they have existed for w time steps.

Assuming the test round outcomes are i.i.d., the sufficient condition for verifiability that we will consider is given by

$$p_{av} < \frac{2\gamma - 1}{k(2\gamma - 2)} \quad (39)$$

as shown in [29]. Here, p_{av} is the average probability of failure of a test round, and γ is the inherent error probability of the BQP computation. The value k is an integer and is corresponding to the k -coloring chosen by the client. This is a partition of the set of vertices into k subsets, known as colors, such that there is no edge between two vertices of the same color. For the relevance of this to the BQC protocol, see Appendix E1 for a description of test rounds. For deterministic computations ($\gamma = 0$), (39) simplifies to

$$p_{av} < \frac{1}{2k}. \quad (40)$$

When the server is honest, the quantity p_{av} is determined on the amount by noise, which could for example arise from imperfect local operations and measurements, or imperfect memory in the server. Further, in a networked setting where the client and server are distantly separated, the client may send its qubits to the server by making use of entanglement that has been established between the two parties. In this way, the performance of the protocol is directly dependent on properties of the quantum network architecture connecting client and server. In such architectures, however, there is in general a tradeoff between rate and quality. In the case of this BQC protocol, demanding that the condition (39) is met then effectively places an upper bound on the rate of the protocol. In the following, we consider a simple model of the network and device architectures, and provide a demonstration of how the methods presented in Section III may be used to find architecture parameters that maximize the protocol rate, given the constraint (39).

2) MODEL OF NETWORK ARCHITECTURE

Our model of the quantum network architecture on which the BQC protocol is carried out is summarized in the following. A depiction is in Fig. 8.

- 1) The server is honest, meaning that it carries out all tasks specified by the client. The BQC protocol protects against malicious server activity, as well as being robust to noise. Here, we solely aim to quantify the effect of noise on the protocol.
- 2) Entanglement generation between client and server is performed with sequential attempts. Each attempt succeeds with probability p .
- 3) Upon entanglement success, a qubit transmission procedure takes place. We assume that each qubit comes into existence in the server memory at the end of the corresponding time step.
- 4) Immediately after transmission, each qubit is established with fidelity $F_{est}(p)$, where $F_{est} : [0, 1] \rightarrow [0, 1]$ is a decreasing function. In this way, we include a tradeoff between rate and fidelity that is inherent to the entanglement generation process occurring between client and server. In the following, we choose this to be $F_{est}(p) = 1 - \lambda p$. Motivation for this choice of F_{est} is given in Appendix D.
- 5) While they are stored in the server, qubits are subject to depolarizing noise with a memory lifetime of T time steps. For a d -dimensional density matrix $\rho \in \mathcal{D}(\mathcal{H}_d)$, this has action

$$\rho \rightarrow e^{-\frac{t}{T}} \rho + \left(1 - e^{-\frac{t}{T}}\right) \frac{\mathbb{I}_d}{d} \quad (41)$$

where \mathbb{I}_d is the d -dimensional identity matrix and t is the number of time steps for which ρ has existed at the server. For the case of a qubit, i.e., $d = 2$, the fidelity then decays as

$$F_{est} \rightarrow \left(F_{est} - \frac{1}{2}\right) e^{-\frac{t}{T}} + \frac{1}{2}. \quad (42)$$

- 6) To reduce decoherence, the server discards a qubit once it has been in memory for w time steps.
- 7) All local operations and measurements by the client and server devices are perfect and instantaneous. In particular, once all qubits required for the round are present in the server, it immediately and perfectly applies the measurement flow that has been specified by the client.
- 8) Before each round, the client chooses an element of V uniformly at random. The corresponding qubit is the first one sent. The client then cycles through the qubits from V in some predefined order. With this added randomness, the resulting order of the qubit ages will appear completely random. We continue with this assumption because it simplifies the resulting calculation of p_{av} , by removing any dependence of the qubit ages on events that occurred beyond the last w time steps. More details of protocol test rounds are given in Appendix E.

In our model, then, the fidelity of a qubit in the server depends only on the amount of time it has been stored there

and the entanglement generation success probability p . Notice that our setup consists of the sequential attempted establishment of qubits at the server and the discarding of these qubits after they have existed for a predefined number of time steps. We then have a situation analogous to that considered in Sections II–IV of this work, where the qubits function as the corresponding quantum resources. The methods given in Section III can then be applied to study this situation: the time taken to complete a round is $\tau_{(w,s)}$ time steps, where $s = |V|$ is the number of qubits required to produce a graph state, and $\tau_{(w,s)}$ is the waiting time as defined in Section II. Furthermore, the qubit fidelities at the time when the server applies its entangling gates and measurements are determined by the ending pattern $X_{(w,s)}$, which finishes the process. More specifically, it is possible to calculate p_{av} exactly using the ending pattern distribution. We briefly describe this now.

Suppose that during a particular test round, at the time the server will carry out its local operations and measurements, the fidelities of the server qubits are $\vec{F} = (F_1, F_2, \dots, F_{|V|})$. Then, given the model described in the previous section, it is possible to find a function that tells us the probability of error of a test round, $P_G(\vec{F})$. This is a polynomial in the values F_i , and has a form dependent on the graph G and the choice of k -coloring. The details of how to obtain this function are given in Appendix E. An expression for the *average* probability of error of a test round is then

$$p_{\text{av}} = \sum_{\vec{F}} \mathbb{P}(\vec{F}) P_G(\vec{F}) \quad (43)$$

where $\mathbb{P}(\vec{F})$ is the probability of obtaining the particular fidelity vector \vec{F} . Note that in the model introduced in the previous section, the qubit fidelities \vec{F} are determined by the amount of time for which the qubits have been stored in the server. Moreover, recall that the ages of the links are contained exactly in the ending pattern $X_{(w,s)}$. Writing this dependence as

$$\vec{F} = \vec{F}(X_{(w,s)}) \quad (44)$$

we then rewrite (43) to obtain an expression for the average probability of test round failure

$$p_{\text{av}} = \sum_{x \in \Omega(w,s)} \mathbb{P}(X_{(w,s)} = x) P_G(\vec{F}(x)). \quad (45)$$

This is a quantity that we can now evaluate using the methods introduced in Section III. In this way, the tools from Section III allow for the direct connection between the feasibility of the BQC protocol, as determined by p_{av} , to its rate. Since the above formula for p_{av} is dependent on the graph structure and k -coloring, some parameter regimes may be sufficient for some calculations but not others. For example, for more complicated graphs that require a larger k , the condition (39) is more strict. Further, if one chooses a different graph or k -coloring for the calculation, the polynomial P_G may differ.

3) NUMERICAL EVALUATION

We now aim to find optimal values of the architecture parameters p and w for one round of the protocol. By *optimality*, we mean that the expected time taken to carry out a round is minimized, while ensuring that the protocol is still feasible. Note that this does not necessarily mean optimality for the full protocol, which is comprised of multiple rounds. To optimize over the full protocol, one would do a further optimization over more protocol parameters (for example, the ratio of computation and test rounds), which we do not consider in this work.

There is a combination of tradeoffs between rate and fidelity present in our scenario: first due to varying the success probability, and second due to varying the window size. An increase in p increases the rate at which successful links are generated, but decreases the initial fidelity of qubits in the server by an amount determined by $F_{\text{est}}(p)$. We would therefore expect that a smaller value of w is required to minimize decoherence at the server, to ensure that the condition (39) is met. This in turn increases the expected time taken to generate all necessary entangled links within the time window. More formally, given a fixed p , we may find the minimal expected time for one round with the following procedure.

- 1) Find the maximum value of w such that the protocol is still feasible for this value of p

$$w_{\text{max}}(p) := \max \left\{ w : p_{\text{av}} < \frac{2\gamma - 1}{k(2\gamma - 2)} \right\}. \quad (46)$$

- 2) Compute $\mathbb{E}(\tau_{(w_{\text{max}}(p),s)})|_p$. This is the minimum expected time for one round.

As an example of this method put into practice, we consider the case where the client would like to perform a BQP calculation on a square graph, so that $|V| = 4$. This requires $s = 4$ entangled pairs to be produced within the time window. For simplicity, we will consider deterministic computations, so that the requirement on the probability of error is $p_{\text{av}} < 1/2k$. In this case, k can be chosen to be 2 (see Appendix E for an example of a 2-coloring of a square graph), and the sufficient condition becomes $p_{\text{av}} < 1/4$.

In Fig. 9, the minimum expected time to carry out a round is plotted against p for three different values of the memory lifetime parameter T , and $F_0(p) = 1 - \lambda p$. The code used to produce Figs. 9 and 10 is provided in [25]. We choose $\lambda = \frac{1}{2}$ in order to best display the behavior of the solution, given our computational resources. In particular, the range of p that we plot is chosen to clearly show the region of the optimal combination of the two tradeoffs. For small p , the expected waiting time is high due to the small entanglement generation probability. For large p , it is high due to the small window size required due to the decrease in F_{est} . We therefore see a region in the middle of the plot where the average waiting time is minimal, or equivalently, the rate at which rounds can be carried out is maximal. For larger T , the decoherence of qubits in memory is reduced, and so it is possible to have a

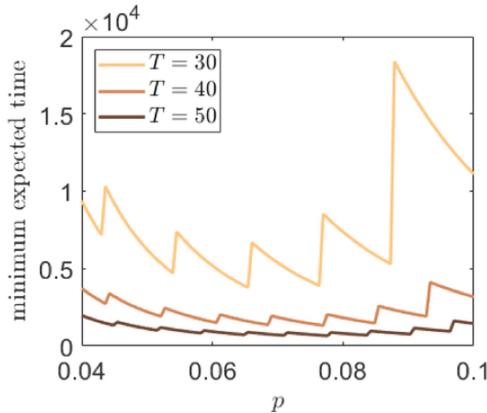


FIGURE 9. Minimum expected time for one round of a BQC calculation versus success probability of entanglement generation. For a given value of p , we find the maximum window size (46), and then use that to compute the minimum expected waiting time. Here, p is discretized into 100 values of p that are evenly spaced in the range $[0.04, 0.1]$. We assume calculations on a square graph, which requires four entangled links to be produced within a time window.

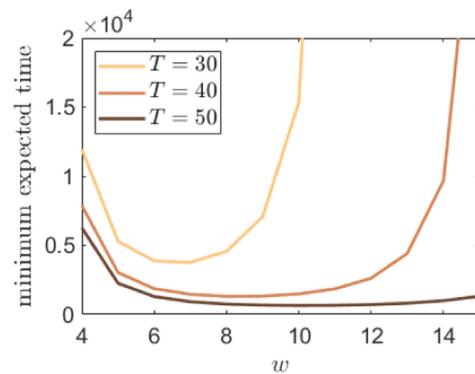


FIGURE 10. Minimum expected time for one round of a BQC calculation versus window size. For each window size $4 \leq w \leq 15$, we find the maximum success probability (47), and then use this to compute the expected waiting time. We assume calculations on a square graph, which requires four entangled links to be produced within a time window.

larger window size without disrupting the condition on p_{av} . We thus see that the expected time for one round decreases with T . Further, there are sharp peaks in the plots for each T , which are due to the discrete nature of w . This can be explained as follows: in the middle of two peaks, it is possible to increase the value of p without disrupting the condition (39). However, there will come a point where this condition is in fact an equality, which is when the window cutoff must be decreased in order to maintain the minimum quality of qubits in the server. Since w is a discrete parameter, this causes a jump to a higher expected time.

One can do something similar when varying the window size. Given a fixed w , we find the minimum expected time for one round with the following steps.

- 1) Find the maximum value of p such that the protocol is still feasible

$$p_{\text{max}}(w) := \sup \left\{ p : p_{\text{av}} < \frac{2\gamma - 1}{k(2\gamma - 2)} \right\}. \quad (47)$$

- 2) Compute $\mathbb{E}(\tau_{(w,s)})|_{p_{\text{max}}(w)}$.

In Fig. 10, the minimum expected time to carry out a round is plotted against the window size. This is again in the case of a square graph, for the same three values of the memory lifetime parameter T . We see a similar behavior as when varying the success probability: a smaller w induces a larger expected time. When s is larger, qubits are subject to more decoherence, and in order to keep condition (39), it is necessary to decrease the success probability. This is what induces a larger waiting time for larger w . We therefore again see an optimal region of w for which the expected time to carry out one round of the protocol is minimized.

Finally, we note that in practice, in order to optimize the full BQC protocol, one would need to consider how other aspects of the setup, such as hardware, architecture, and protocol, affect the performance. The simple scenario chosen in this work was to highlight the application of the results of Section III. We see from Figs. 9 and 10 that for such values of T and s , the methods from Section III enable one to make a careful choice of (w, p) that can improve the rate of rounds of the protocol by two or three times, in comparison to other nonoptimal choices of (w, p) that are also sufficient for protocol feasibility.

V. FURTHER DIRECTIONS

With the methods presented in this work, we focus on computing both the first and second moments of $\tau_{(w,s)}$, and the full distribution of $X_{(w,s)}$. We have seen that for w finite and $s > 2$, the formulae given here to compute $\mathbb{E}(\tau_{(w,s)})$ and the distribution of $X_{(w,s)}$ are in the form of linear systems that scale as $|\Omega(w, s)| + 1$. If one would like to compute the full ending pattern distribution, then this seems to be a good scaling, since the outcome is comprised of $|\Omega(w, s)|$ probabilities. However, if one is for example only interested in $\mathbb{E}(\tau_{(w,s)})$ (e.g., for computing a protocol rate), then for certain regimes of w and p it may be useful to consider a continuous approximation, where the time between successful resource generation attempt is exponentially distributed. Such a case is often considered in the scan statistics literature (for example, see [19]). However, how to study the ending pattern distribution in the continuous case is not immediately clear.

We also note that a useful tool of approximation would be to further understand the asymptotic scaling highlighted by Theorem 3. More specifically, it would be interesting to know exactly how fast is the approach of (33) and (34), in terms of s and w .

In the setup of the problem, one could also consider a more realistic model of a quantum network architecture. For example, there may be parameter drift, when the success probability decreases over time due to increased noise. Further, in the more general case where the sequential attempts are not necessarily independent but Markovian, methods similar to those used in this article may again be applied to the problem—see [20], for example.

APPENDIX A IDENTITIES FOR THE CASE OF TWO RESOURCE STATES

1) EVALUATION OF $\mathbb{E}(\tau_{(w,2)})$

Evaluation of $\mathbb{E}(L)$: Recalling from (13) the distribution of L , we have

$$\begin{aligned}\mathbb{E}(L) &= \sum_{n=1}^{w-1} \frac{n(1-p)^{n-1}p}{1-(1-p)^{w-1}} \\ &= \frac{p}{1-(1-p)^{w-1}} \sum_{n=1}^{w-1} n(1-p)^{n-1} \\ &= \frac{p}{1-(1-p)^{w-1}} \cdot -\frac{d}{dp} \sum_{n=1}^{w-1} (1-p)^n \\ &= \frac{p}{1-(1-p)^{w-1}} \cdot -\frac{d}{dp} \frac{1-(1-p)^w}{1-p} \\ &= \frac{p}{1-(1-p)^{w-1}} \cdot \frac{1-(1-p)^w - wp(1-p)^{w-1}}{p^2} \\ &= \frac{1-(1-p)^w - wp(1-p)^{w-1}}{p(1-(1-p)^{w-1})}\end{aligned}$$

where to evaluate the sum we have used the identity for a geometric series.

Proof that $\mathbb{E}(\sum_{j=1}^M T_j) = \mathbb{E}(M)\mathbb{E}(T_1)$: This is used to evaluate the expectation $\mathbb{E}(\tau_{(w,2)})$. The random variables M and $\{T_j\}$ are independent, and since the $\{T_j\}$ are identically distributed

$$\begin{aligned}\mathbb{E}\left(\sum_{j=1}^M T_j\right) &= \sum_{m=1}^{\infty} \mathbb{E}\left(\sum_{j=1}^m T_j\right) \mathbb{P}(M=m) \\ &= \sum_{m=1}^{\infty} \mathbb{E}(T_1) \cdot m \mathbb{P}(M=m) \\ &= \mathbb{E}(T_1)\mathbb{E}(M).\end{aligned}$$

■

2) EVALUATION OF $\text{Var}(\tau_{(w,2)})$

Recall M , T_j , and L , as given in Section III-B1. These are independent, M and T have distributions $M \sim \text{Geom}(1 - (1 - p)^{w-1})$, $T_j \sim \text{Geom}(p)$, and the distribution of L is given in

(13). From (15), we have

$$\text{Var}(\tau_{(w,2)}) = \text{Var}\left(\sum_{j=1}^M T_j + (M-1)(w-1)\right) + \text{Var}(L) \quad (48)$$

since L is independent of M and T_j . Now, letting $\text{Cov}(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$ be the covariance, we have

$$\begin{aligned}\text{Var}\left(\sum_{j=1}^M T_j + (M-1)(w-1)\right) \\ = \text{Var}\left(\sum_{j=1}^M T_j\right) + \text{Var}((M-1)(w-1)) \\ + 2 \cdot \text{Cov}\left(\sum_{j=1}^M T_j, (M-1)(w-1)\right) \quad (49)\end{aligned}$$

where we have used the identity $\text{Var}(X+Y) = \text{Var}(X) + \text{Var}(Y) + 2 \cdot \text{Cov}(X, Y)$. We now evaluate (49) term by term. Firstly

$$\begin{aligned}\mathbb{E}\left(\left(\sum_{j=1}^M T_j\right)^2\right) &= \sum_m \mathbb{E}\left(\left(\sum_{j=1}^m T_j\right)^2\right) \mathbb{P}(M=m) \\ &= \sum_m \mathbb{E}\left(\sum_{j=1}^m T_j^2 + \sum_{i \neq j} T_i T_j\right) \mathbb{P}(M=m) \\ &= \sum_m \left(m\mathbb{E}(T_1^2) + m(m-1)\mathbb{E}(T_1)^2\right) \mathbb{P}(M=m) \\ &= \mathbb{E}(M)\mathbb{E}(T_1^2) + \left(\mathbb{E}(M^2) - \mathbb{E}(M)\right)\mathbb{E}(T_1)^2.\end{aligned}$$

Subtracting $\mathbb{E}(\sum_{j=1}^M T_j)^2 = \mathbb{E}(M)^2\mathbb{E}(T_1)^2$ then yields

$$\text{Var}\left(\sum_{j=1}^M T_j\right) = \mathbb{E}(M)\text{Var}(T_1) + \text{Var}(M)\mathbb{E}(T_1)^2. \quad (50)$$

Secondly

$$\text{Var}((M-1)(w-1)) = (w-1)^2\text{Var}(M). \quad (51)$$

Thirdly

$$\text{Cov}\left(\sum_{j=1}^M T_j, (M-1)(w-1)\right) \quad (52)$$

$$= (w-1)\text{Cov}\left(\sum_{j=1}^M T_j, M\right) \quad (53)$$

$$= (w-1) \left(\sum_m m^2 \mathbb{E}(T_1) \mathbb{P}(M=m) - \mathbb{E}(M)^2 \mathbb{E}(T_1) \right) \quad (54)$$

$$= (w-1)\text{Var}(M)\mathbb{E}(T_1). \quad (55)$$

It now remains to evaluate $\text{Var}(L)$. We firstly calculate

$$\mathbb{E}(L^2) = \sum_{n=1}^{w-1} \frac{n^2(1-p)^{n-1}p}{1-(1-p)^{w-1}} \quad (56)$$

$$= \mathbb{E}(L) + \frac{p}{1-(1-p)^{w-1}} \sum_{n=1}^{w-1} n(n-1)(1-p)^{n-1}. \quad (57)$$

Now

$$\begin{aligned} \sum_{n=1}^{w-1} n(n-1)(1-p)^{n-1} &= (1-p) \frac{d^2}{dp^2} \sum_{n=0}^{w-1} (1-p)^n \\ &= (1-p) \frac{d^2}{dp^2} \left(\frac{1-(1-p)^w}{p} \right) \end{aligned}$$

so

$$\begin{aligned} \text{Var}(L) &= \frac{p(1-p)}{1-(1-p)^{w-1}} \frac{d^2}{dp^2} \left(\frac{1-(1-p)^w}{p} \right) \\ &\quad + \mathbb{E}(L) - \mathbb{E}(L)^2. \end{aligned} \quad (58)$$

We are now equipped to compute the full variance of $\tau_{(w,2)}$

$$\begin{aligned} \text{Var}(\tau_{(w,2)}) &= \mathbb{E}(M)\text{Var}(T_1) + \text{Var}(M)\mathbb{E}(T_1)^2 \\ &\quad + 2(w-1)\text{Var}(M)\mathbb{E}(T_1) + (w-1)^2\text{Var}(M) \\ &\quad + \text{Var}(L) \end{aligned} \quad (59)$$

where one may find a closed-form expression by inputting the standard identities for a geometric random variable, which are

$$\mathbb{E}(T_1) = \frac{1}{p} \quad (60)$$

$$\text{Var}(T_1) = \frac{1-p}{p^2} \quad (61)$$

$$\mathbb{E}(M) = \frac{1}{1-(1-p)^{w-1}} \quad (62)$$

$$\text{Var}(M) = \frac{(1-p)^{w-1}}{(1-(1-p)^{w-1})^2}. \quad (63)$$

APPENDIX B

ENDING PATTERN DISTRIBUTION AND WAITING TIME MOMENTS IN THE CASE OF A FINITE WINDOW SIZE

1) WAITING TIME IS WELL DEFINED

We show here that τ_x can be bounded above by a geometrically distributed random variable. Using the notation $p_1 = p$, $p_0 = 1 - p$ for an ending pattern $x \in \Omega_l(s)$, this exact sequence will appear in any given l consecutive trials Z_{i-l+1}, \dots, Z_i with probability $\gamma_x := p_{x_1} \dots p_{x_l}$. Defining a new sequence of random variables $(Y_n)_{n=1}^{\infty}$

$$Y_n = \begin{cases} 1 & \text{if } Z_i = x_i \text{ for all } (n-1)l < i \leq nl \\ 0, & \text{otherwise.} \end{cases} \quad (64)$$

Each Y_n is then Bernoulli with parameter γ_x . It takes the value 1 if the n th segment of l trials exactly matches with x . There is then an associated waiting time random variable $\tilde{\tau}_x$ that is geometric with parameter γ_x

$$\tilde{\tau}_x := \min\{n : Y_n = 1\}. \quad (65)$$

Moreover, the waiting time to see x satisfies $\tau_x \leq \tilde{\tau}_x \cdot l$. Taking expectations yields

$$\mathbb{E}(\tau_{(w,s)}) \leq \mathbb{E}(\tau_x) \leq \mathbb{E}(\tilde{\tau}_x \cdot l) = \frac{l}{\gamma_x} < \infty \quad (66)$$

which completes our proof. We note that the same method can be used to show that all moments of $\tau_{(w,s)}$ are finite.

2) EXPECTED WAITING TIME OF A SIMPLE PATTERN

Using the theory of martingales and a helpful gambling analogy to aid understanding, we now derive a way to numerically compute the ending pattern distribution $\{\mathbb{P}(x) : x \in \Omega(w, s)\}$, and the first and second moments of the waiting time $\tau_{(w,s)}$, in the case of a finite window size. The result of this is Theorem 1 in the main text. The method was introduced in [30], where Li considered the more abstract case of a general sequence of discrete i.i.d. random variables, and a general set of ending patterns. Here, due to its relevance to the subject of the main text, we continue with the case of i.i.d. Bernoulli trials.

It will be useful to first of all consider the case where we wait for an instance of a single pattern $x = (x_1, \dots, x_l) \in \{0, 1\}^l$, instead of waiting for any instance of the set $\Omega(w, s)$. The former case is referred to as a *simple pattern* and the latter as a *compound pattern*. In this section, we will find an exact expression for $\mathbb{E}(\tau_x)$. Here, τ_x refers to the waiting time until seeing the pattern x , and is defined in (5).

To provide intuition, we introduce the following scenario of gamblers in a casino. Suppose that just before the first trial is realized, a gambler, hereinafter referred to as Gambler 1, bets € 1 on the outcome $\{Z_1 = x_1\}$. We also suppose that the odds are fair, so that if this is the case then she wins $\frac{1}{p_{x_1}}$.¹ Moreover, if she wins, then she straight away bets all of these winnings on the outcome $\{Z_2 = x_2\}$. If not, the casino keeps her € 1 and she does not place any more bets. For a general n , Gambler 1 then proceeds at the n th trial in a similar way: if she has yet to lose, she bets all of her winnings on the outcome $\{Z_n = x_n\}$, and if not, she does not place any bet. Furthermore, at every trial we introduce a new gambler who behaves in exactly the same way, so that Gambler 2 bets € 1 on the outcome $\{Z_2 = x_1\}$, and continues betting all of her winnings on the subsequent rounds being equal to the next entry of x , up until she loses a round. Gambler j bets € 1 on the outcome $\{Z_j = x_1\}$ and continues with exactly the same strategy. The game stops when the sequence x first appears, which by definition is at the τ_x th trial.

¹i.e., the expected net gain of the gambler is zero. Calling this G , we can verify explicitly by writing $\mathbb{E}(G) = (1/p_{x_1} - 1) \cdot p_{x_1} + (-1) \cdot (1 - p_{x_1}) = 0$.

Our aim now is to write down an expression for the combined net gain of the gamblers after the n th trial, for a general n . In order to do this concisely, we recall the definition (21) of the quantities $\delta_{(a,b)}$. Given a realization $C_n := (c_1, \dots, c_n)$ of the first n trials, the winnings of Gambler j after the n th round can then be written as

$$W^{(j)}(C_n) = \begin{cases} \delta_{(x_1, c_j)} \delta_{(x_2, c_{j+1})} \dots \delta_{(x_{n-j+1}, c_n)} & \text{for } n \\ -l + 1 \leq j \leq n; 0, \text{ otherwise.} \end{cases} \quad (67)$$

Here, we see that the functions $\delta_{(a,b)}$ allow us to elegantly write down a gambler's winnings. With this in mind, the *combined* winnings of the gamblers after the n th trial is

$$W(C_n) := \sum_{j=1}^n W^{(j)}(C_n) = \sum_{j=n-l+1}^n W_n^{(j)} \equiv x * C_n \quad (68)$$

where we have introduced the function $*$ that was defined in (20), and that maps two binary strings to a scalar value. From (67), we see that the *net gain* of the j th gambler after the n th time step is simply

$$G^{(j)}(C_n) := W^{(j)}(C_n) - 1 \quad (69)$$

and similarly, the *total net gain* of the gamblers after the n th trial is

$$G(C_n) := \sum_{j=1}^n G^{(j)}(C_n) = x * C_n - n. \quad (70)$$

We can now define a sequence of random variables $(G_n)_{n \geq 0}$

$$G_n := G(C_n) \quad (71)$$

which take the value of the total net gain of the gamblers after each round. In particular, after the game ends, the total net gain is

$$G_{\tau_x} = x * x - \tau_x. \quad (72)$$

Note that $x * x$ is a quantity that is only dependent on the pattern x . Since the game is defined to be fair at every round, the expected total net gain when the game finishes would intuitively be equal to zero, i.e.,

$$\mathbb{E}(G_{\tau_x}) = 0. \quad (73)$$

A neat expression for the expected waiting time to see the sequence B follows by making use of the linearity of expectation:

$$\mathbb{E}(\tau_x) = x * x. \quad (74)$$

To prove (73), we make use of the fact that $(G_n)_{n \geq 0}$ is a *martingale*, for which the following properties must hold:

- (i) $\mathbb{E}(|G_n|) < \infty$;
- (ii) $\mathbb{E}(G_{n+1}|G_n, \dots, G_1) = G_n$.

To show (i), we use the definition (72), and see that

$$\mathbb{E}(|G_n|) \leq x * C_n + \mathbb{E}(\tau_x) < \infty \quad (75)$$

since the waiting time τ_x is well defined, and $x * C_n$ is bounded. To show condition (ii), we use the fact that the

game is fair at each round. Suppose that we have the maximum amount of information about what has happened in the first n trials, i.e., we know that they have taken the values (c_1, \dots, c_n) . Then, the conditional expectation of G_{n+1} satisfies

$$\begin{aligned} & \mathbb{E}(G_{n+1}|Z_1 = c_1, \dots, Z_n = c_n) \\ &= \sum_{c_{n+1} \in \{0,1\}} G(C_n, c_{n+1}) \mathbb{P}(Z_{n+1} = c_{n+1}) \\ &= \sum_{j=1}^{n+1} \sum_{c_{n+1} \in \{0,1\}} G^{(j)}(C_n, c_{n+1}) p_{c_{n+1}} \\ &= \sum_{j=1}^{n+1} G^{(j)}(C_n) = G_n \end{aligned}$$

where to go to the final line, we have made use of the definition of $G^{(j)}$. Since the realizations of (Z_1, \dots, Z_n) completely determine the values of G_1, \dots, G_n , this also shows (ii).

We now know that $(G_n)_{n \geq 0}$ is a martingale. However, this is not quite enough to show (73), which is what is required to obtain the final simple form for $\mathbb{E}(\tau_x)$. In particular, some extra machinery is needed, in the form of Doob's *optional stopping theorem*, a proof of which can be found in [31]. A version of this is stated below.

Theorem 4 (Optional Stopping): Let G_n be a martingale and τ be a stopping time. Suppose that there exists a constant K such that $|G_n - G_{n-1}| < K$ for all n . Suppose also that τ is a.s. finite. Then, $\mathbb{E}(G_\tau) = \mathbb{E}(G_1)$. \blacksquare

All that remains to be done is to show that the martingale defined in (72) satisfies the required properties to satisfy Theorem 4. Firstly, we have

$$|G_n - G_{n-1}| < x * C_n + x * C_{n-1} + 1 \leq K \quad (76)$$

where

$$K = 2 \cdot \max_{C \in \{0,1\}^l} \{x * C\} + 1. \quad (77)$$

Secondly, we see that since τ_x is bounded above by a geometric random variable, it is a.s. finite. This gives us (73).

3) STARTING FROM ANOTHER PATTERN:

We now adapt the results above in order to find the expected time to see y , given that we start already with some pattern y . We extend the gambling analogy in order to illustrate this concept, and suppose that we want to calculate the expected time until seeing y only *after* some number of rounds, m , say, have been realized. In particular, after the m th round, we know the first m realizations of the i.i.d. Bernoulli sequence, and we call these $y = (y_1, \dots, y_m)$. At this point, the net gain of the gamblers is, thus, $G_m = x * y - m$. We will evaluate the net gain of the gamblers *compared* with this point after each of the n trials, which for $n \geq m$, we denote by \tilde{G}_n . This is simply given by

$$\tilde{G}_n = G_n - G_m = (x * C_n - n) - (x * y - m) \quad (78)$$

$$= x * C_n - x * y - (n - m) \quad (79)$$

where C_n is no longer completely general as its first m entries must correspond to y . Using the same reasoning as before, one can show that $(\tilde{G}_n)_{n \geq 0}$ is a martingale. Then, defining τ_{xy} as the waiting time to see x given that we have already seen pattern y , it is again possible to use Theorem 4 to show that

$$0 = \mathbb{E}(\tilde{G}_{\tau_{xy}}) = \mathbb{E}(x * x - x * y - \tau_{yx})$$

and so by the linearity of expectations

$$\mathbb{E}(\tau_{xy}) = x * x - x * y. \quad (80)$$

We may now use the results derived above to derive a formula for $\mathbb{E}(\tau_{(w,s)})$ and the distribution of $X_{(w,s)}$. Given $x \in \Omega(w, s)$, we write

$$\begin{aligned} & \mathbb{E}(\tau_x) \\ &= \mathbb{E}(\tau_{(w,s)}) + \mathbb{E}(\tau_x - \tau_{(w,s)}) \\ &= \mathbb{E}(\tau_{(w,s)}) + \sum_{y \in \Omega(w,s)} \mathbb{P}(X_{(w,s)} = y) \mathbb{E}(\tau_x - \tau_{(w,s)} | X_{(w,s)} = y) \\ &= \mathbb{E}(\tau_{(w,s)}) + \sum_{y \in \Omega(w,s)} \mathbb{P}(X_{(w,s)} = y) (x * x - x * y) \end{aligned}$$

where we have noticed that $\mathbb{E}(\tau_x - \tau_{(w,s)} | X_{(w,s)} = y) = \mathbb{E}(\tau_{xy})$. Applying Theorem 4 and enforcing the condition that the ending pattern probabilities must sum to one, then, yields the formula (23).

4) FORMULA FOR THE SECOND MOMENT OF THE WAITING TIME

An extension to the gambling analogy given above can be used to derive the formula for the second moment of the waiting time, for which we refer to [20]. Here, we will only state the formula. We first of all define a new operation \dagger that maps two elements $x, y \in \Omega(w, s)$ to a real number. If $x = (x_1, \dots, x_k)$ and $y = (y_1, \dots, y_m)$

$$x \dagger y := \sum_{j=1}^{\min(k,m)} (1-j) \prod_{i=1}^j \delta_{(x_i, y_{m-j+i})}. \quad (81)$$

Letting $\Omega \equiv \Omega(w, s)$, the second moment of $\tau_{(w,s)}$ can be found through solving the following systems.

Theorem 5: Let $\{u_j\}_{1 \leq j \leq |\Omega|}$ and $\{v_j\}_{1 \leq j \leq |\Omega|}$. Solve the linear systems

$$\sum_{j=1}^{|\Omega|} W_{ij} u_j = 1, \text{ for } 1 \leq i \leq |\Omega| \quad (82)$$

$$\sum_{j=1}^{|\Omega|} (N_{ij} u_j + W_{ij} v_j) = 1, \text{ for } 1 \leq i \leq |\Omega| \quad (83)$$

with $W_{ij} := x^{(i)} * x^{(j)}$ and $N_{ij} := x^{(i)} \dagger x^{(j)}$. Then

$$\mathbb{E}(\tau_{(w,s)}^2) = \frac{1 + \left(1 - \sum_j v_j - \sum_j u_j/2\right) \cdot \mathbb{E}(\tau_{(w,s)})}{\sum_j u_j/2}. \quad (84)$$

■

Code that makes use of this formula to compute $\mathbb{E}(\tau_{(w,s)}^2)$ is provided in [25].

APPENDIX C APPROXIMATIONS

1) INFINITE WINDOW SIZE APPROXIMATION

Proof of Theorem 2: Letting $\epsilon \equiv \epsilon(w, s, p) = \mathbb{P}(\tau_{(w,s)} > w)$, the expectation of $\tau_{(w,s)}$ can be rewritten as

$$\begin{aligned} \mathbb{E}(\tau_{(w,s)}) &= (1 - \epsilon) \mathbb{E}(\tau_{(w,s)} | \tau_{(w,s)} \leq w) \\ &\quad + \epsilon \mathbb{E}(\tau_{(w,s)} | \tau_{(w,s)} > w). \end{aligned} \quad (85)$$

Now, note that for $n \leq w$

$$\mathbb{P}(\tau_{(w,s)} = n) = \mathbb{P}(\tau_{(\infty,s)} = n)$$

i.e., for this range of n , the distributions of $\tau_{(w,s)}$ and $\tau_{(\infty,s)}$ exactly match. We can thus rewrite (85) as

$$\begin{aligned} & (1 - \epsilon) \mathbb{E}(\tau_{(\infty,s)} | \tau_{(\infty,s)} \leq w) + \epsilon \mathbb{E}(\tau_{(w,s)} | \tau_{(w,s)} > w) \\ &= \mathbb{E}(\tau_{(\infty,s)}) - \epsilon \mathbb{E}(\tau_{(\infty,s)} | \tau_{(\infty,s)} > w) \\ &\quad + \epsilon \mathbb{E}(\tau_{(w,s)} | \tau_{(w,s)} > w) \end{aligned}$$

where to obtain the last equality we have expanded $\mathbb{E}(\tau_{(\infty,s)})$ in the same way as (85). Now, if one considers starting the whole process again after the first w time steps, we see that $\mathbb{E}(\tau_{(w,s)} | \tau_{(w,s)} > w) \leq w + \mathbb{E}(\tau_{(w,s)})$. Combining this with the fact that $\mathbb{E}(\tau_{(\infty,s)} | \tau_{(\infty,s)} > w) > w$, we find that

$$\mathbb{E}(\tau_{(w,s)}) - \mathbb{E}(\tau_{(\infty,s)}) \leq \epsilon \cdot (w + \mathbb{E}(\tau_{(w,s)}) - w) \quad (86)$$

from which (25) follows.

We further bound the distance between the ending pattern distributions. Making use of

$$\begin{aligned} \mathbb{P}(X_{(w,s)} = x) &= \mathbb{P}(X_{(w,s)} = x | \tau_{(w,s)} \leq w) (1 - \epsilon) \\ &\quad + \mathbb{P}(X_{(w,s)} = x | \tau_{(w,s)} > w) \epsilon \end{aligned}$$

we have

$$\mathbb{P}(X_{(w,s)} = x) - \mathbb{P}(X_{(\infty,s)} = x)$$

$$= (\mathbb{P}(X_{(w,s)} = x | \tau_{(w,s)} > w) - \mathbb{P}(X_{(\infty,s)} = x | \tau_{(w,s)} > w)) \epsilon$$

and so, letting $\Omega \equiv \Omega(\infty, s)$

$$\begin{aligned} & \sum_{x \in \Omega} |\mathbb{P}(X_{(w,s)} = x) - \mathbb{P}(X_{(\infty,s)} = x)| \\ & < \sum_{x \in \Omega} (\mathbb{P}(X_{(w,s)} = x | \tau_{(w,s)} > w) \\ & \quad + \mathbb{P}(X_{(\infty,s)} = x | \tau_{(w,s)} > w)) \epsilon \\ & = 2\epsilon(w, s, p). \end{aligned}$$

Proof of Lemma 1: Here, we show the identity (29) for $\epsilon(w, s, p)$. Letting $q = 1 - p$, we have

$$\begin{aligned}
 \epsilon(w, s, p) &= \sum_{n=w+1}^{\infty} \binom{n-1}{s-1} q^{n-s} p^s \\
 &= \frac{p^s}{(s-1)!} \sum_{n=w+1}^{\infty} \frac{(n-1)!}{(n-s)!} q^{n-s} \\
 &= \frac{p^s}{(s-1)!} \sum_{n=w+1}^{\infty} \frac{d^{s-1}}{dq^{s-1}} (q^{n-1}) \\
 &= \frac{p^s}{(s-1)!} \frac{d^{s-1}}{dq^{s-1}} \left(\sum_{n=w+1}^{\infty} q^{n-1} \right) \\
 &= \frac{p^s}{(s-1)!} \frac{d^{s-1}}{dq^{s-1}} \left(\frac{q^w}{1-q} \right) \\
 &= \frac{p^s}{(s-1)!} \sum_{i=1}^{s-1} \binom{s-1}{i} \frac{d^i}{dq^i} (q^w) \frac{d^{s-1-i}}{dq^{s-1-i}} ((1-q)^{-1}) \\
 &= \frac{p^s}{(s-1)!} \sum_{i=1}^{s-1} \frac{(s-1)!}{i!(s-i-1)!} \frac{w!}{(w-i)!} q^{w-i} \frac{(s-i-1)!}{(1-q)^{s-i}} \\
 &= \sum_{i=1}^{s-1} \binom{w}{i} q^{w-i} p^i.
 \end{aligned}$$

2) ASYMPTOTIC BEHAVIOR OF THE EXPECTED WAITING TIME AND ENDING PATTERN

Proof of Theorem 3: For conciseness, here we take $\Omega \equiv \Omega(w, s)$. A formula for the inverse of A is given in terms of its adjugate matrix $\text{adj } A$ [32]

$$[\text{adj } A]_{ij} := (-1)^{(i+j)} \det M_{ji} \quad (87)$$

where M_{ij} is the $|\Omega| \times |\Omega|$ matrix obtained by removing row i and column j from A . Since A is invertible, the inverse is

$$A^{-1} = \frac{\text{adj } A}{\det A}. \quad (88)$$

Now consider the system (23). If we consider solving for \vec{v} by multiplying through by A^{-1} , we see that its first element is

$$\mathbb{E}(\tau_{(w,s)}) = \frac{\det B}{\det A} \quad (89)$$

where B is the $|\Omega| \times |\Omega|$ matrix obtained by removing the first row and column from A , so that $B_{ij} = x^i * x^j$. Since all the entries of A are polynomials in $1/p$ and $1/q$, so are $\det B$ and $\det A$.

To proceed with showing (33), we characterize the scaling of $\det B$ and $\det A$ for small p . Since $q = 1 - p$ is close to 1

for small p , it suffices to only consider the powers of $1/p$ for the analysis of the asymptotic scaling as $p \rightarrow 0$ [recalling the definition of the star product (20)]. We first consider $\det B$. With the observation that the higher order terms in $1/p$ are given by the star products on the diagonal, and moreover, that these each have leading order term given by $1/p^s$. The form of $\det B$, then, is a polynomial of maximum degree $1/p^{s|\Omega|}$. In fact, this is exactly the degree. One can compute this contribution by considering the matrix \tilde{B} of highest powers: letting $r \equiv 1/p^s$, we have $\det B \sim \det \tilde{B}$, where

$$\tilde{B} = \begin{pmatrix} r & 0 & \dots & 0 \\ 0 & r & \dots & 0 \\ \vdots & & \ddots & 0 \\ 0 & 0 & \dots & r \end{pmatrix} \quad (90)$$

and hence, $\det B \sim r^{|\Omega|} = 1/p^{s|\Omega|}$. We then do the same with A . In this case, $\det A \sim \det \tilde{A}$, where

$$\tilde{A} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ -1 & r & 0 & \dots & 0 \\ -1 & 0 & r & \dots & 0 \\ \vdots & \vdots & & \ddots & 0 \\ -1 & 0 & 0 & \dots & r \end{pmatrix} \quad (91)$$

where the existence of a 0 in the top left-hand corner now disrupts the evaluation of $\det \tilde{A}$ by multiplying along the diagonal, as we did above. Our next step is to evaluate $\det \tilde{A}$ by expanding along the top row

$$\det \tilde{A} = \sum_{k=1}^{|\Omega|} (-1)^k \det \tilde{A}_k \quad (92)$$

where \tilde{A}_k is the $|\Omega| \times |\Omega|$ matrix formed by removing the first row and the k th column from \tilde{A}

$$\tilde{A}_k = \begin{pmatrix} -1 & r & 0 & \dots & 0 \\ \vdots & 0 & \ddots & & \vdots \\ & & & r & 0 \\ & & & 0 & 0 \\ & & & 0 & r \\ \vdots & & & & \ddots & 0 \\ -1 & 0 & \dots & \dots & 0 & r \end{pmatrix}. \quad (93)$$

In \tilde{A}_k , the r 's are placed above the diagonal in rows $1, \dots, k-1$, and on the diagonal in rows $k+1, \dots, |\Omega|$. The determinant of \tilde{A}_1 may be calculated by simply multiplying the diagonal elements, to obtain

$$\det \tilde{A}_1 = -r^{|\Omega|-1}. \quad (94)$$

We then notice that any \tilde{A}_k can be transformed into \tilde{A}_1 by moving the k th row to the top row. This can be achieved by performing $k-1$ row operations, if it is moved by successively exchanging with the row above it $k-1$ times. Then, since each row operation incurs a factor of $(-1)^k$

$$\det \tilde{A}_k = (-1)^{k-1} \det \tilde{A}_1 = (-1)^k r^{|\Omega|-1}. \quad (95)$$

With (92), we then see that

$$\det \tilde{A} = |\Omega| r^{|\Omega|-1} \quad (96)$$

and so $\det A \sim |\Omega|/p^{s(|\Omega|-1)}$. Substituting into (89), we find

$$\mathbb{E}(\tau_{(w,s)}) \sim \frac{p^{s(|\Omega|-1)}}{|\Omega| p^{s|\Omega|}} = \frac{1}{|\Omega| p^s}. \quad (97)$$

To show (34), we employ a similar method. We have from (88) that

$$\mathbb{P}(X_{(w,s)} = x^{(k)}) = (-1)^{1+k} \frac{\det M_{k0}}{\det A} \quad (98)$$

$$\sim (-1)^{1+k} \frac{\det C_k}{\det X} \quad (99)$$

where C_k is obtained by removing the first column and k th row from A

$$C_k = \begin{pmatrix} 1 & \dots & & \dots & 1 \\ r & 0 & \dots & \dots & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & r & 0 & 0 \\ & & 0 & 0 & r \\ \vdots & & & & \ddots & 0 \\ 0 & \dots & & \dots & 0 & r \end{pmatrix} \quad (100)$$

where in C_k the x -entries are placed below the diagonal in columns $1, \dots, k-1$, and on the diagonal in columns $k+1, \dots, |\Omega|$. We note that C_k is the transpose of B_k , with the first column multiplied by a scaling factor of -1 . Therefore, $\det C_k = -\det B_k = (-1)^{k+1} r^{|\Omega|-1}$, and making use again of (96)

$$\mathbb{P}(X_{(w,s)} = x^{(k)}) \sim \frac{r^{|\Omega|-1}}{|\Omega| r^{|\Omega|-1}} \quad (101)$$

from which the result follows. \blacksquare

APPENDIX D TRADEOFF FUNCTION DUE TO ENTANGLEMENT GENERATION SCHEME

In this Appendix, we motivate the linear tradeoff function $F_{\text{est}} = 1 - \lambda p$ used in the BQC analysis, which describes the fidelity of qubits in the server immediately after transmission.

Let σ denote the noisy two-qubit entangled state that is produced between the client and server when there is a successful attempt. When there is a success, the client and server perform some qubit transmission procedure Λ_{σ} , which could for example be teleportation, or a remote state preparation protocol. We assume that this protocol establishes all qubits in the server with the same fidelity F_{est} . For example, this is the case if the noisy entangled state is depolarized

$$\sigma = \frac{4F_0 - 1}{3} |\Phi^+\rangle\langle\Phi^+| + \frac{1 - F_0}{3} \mathbb{I}_4 \quad (102)$$

and the standard teleportation protocol from [33] is applied. Here, $F_0 = \langle\Phi^+|\sigma|\Phi^+\rangle$ is the fidelity of σ to the target state. This involves performing a full measurement in the Bell basis $\{|\Phi_{ij}\rangle\}$ and applying the corresponding Pauli corrections. If $|\psi\rangle$ is the qubit state to be teleported, its action is given by Λ^{st}

$$\Lambda_{\sigma}^{\text{st}}(|\psi\rangle\langle\psi|) := \sum_{i,j} X^i Z^j \langle\Phi_{ij}|(|\psi\rangle\langle\psi| \otimes \sigma)|\Phi_{ij}\rangle Z^j X^i \quad (103)$$

where the Bell measurement acts on the registers containing the qubit state $|\psi\rangle$ and the first qubit of σ . Suppose that the entangled state and qubit transmission procedure are given by (102) and (103), respectively. Then, after transmitting any qubit $|\psi\rangle$, the resulting fidelity is [34]

$$F_{\text{est}} = \frac{2F_0 + 1}{3}. \quad (104)$$

Now, one can incorporate a general rate-fidelity tradeoff inherent to the entanglement generation protocol by specifying that F_{est} is a decreasing function of p . In particular, we draw here on an example from the single-photon scheme for entanglement generation. When implementing a single-photon scheme [6], the fidelity of generated states is

$$F_0(p_{\text{suc}}) = 1 - \frac{p_{\text{suc}}}{2p_{\text{det}}} \quad (105)$$

where p_{suc} is the success probability of a physical entanglement attempt, and p_{det} is the probability of detecting an emitted photon. In the case of a very small p_{suc} , one might want to perform entanglement attempts in *batches* in order to minimize overhead due to communication with higher layers of the software stack (which must be notified when there is, or is not, a success). This scheme has been implemented with nitrogen-vacancy centers in diamond, where typically $p_{\text{suc}} \ll 1$ [35]. If this is the case, choosing one time step to correspond to a batch of $M \ll 1/(p_{\text{suc}})^2$ attempts, the probability of producing at least one entangled link in a time step is

$$p = 1 - (1 - p_{\text{suc}})^M \approx Mp_{\text{suc}}. \quad (106)$$

Substituting this into (104) and (105), we obtain a tradeoff function of

$$F_{\text{est}}(p) = \frac{2\left(1 - \frac{p}{2Mp_{\text{det}}}\right) + 1}{3} = 1 - \lambda p \quad (107)$$

where $\lambda := 1/(3Mp_{\text{det}})$. Since M is a freely adjustable parameter, then so is λ . The simple relationship (107) is also a general first-order behavior for a decreasing function in p such that $F_{\text{est}}(0) = 1$, which justifies the choice as potentially applicable to other hardware and entanglement generation protocols.

APPENDIX E COMPUTING THE ERROR PROBABILITY OF A BQC TEST ROUND

1) TEST ROUNDS

As mentioned previously, the protocol involves interweaving test rounds at random with computation rounds. It is the test rounds that provide verifiability of the protocol, because they allow the client to check for deviations from the ideal measurement outcomes. Recall that the goal of the client in the BQC protocol is to perform a BQP computation, which is defined in the measurement-based formalism with respect to a graph $G = (V, E)$. In one round of the protocol, the client transmits $|V|$ qubits to the server, which (if it is honest) creates a graph state by applying CZ -gates to pairs of qubits as given in the set of edges E . Before carrying out the protocol, the client chooses some k -coloring $\{V_j : j = 1, \dots, k\}$, which is a partition of the set of vertices V into different subsets, known as *colors*, such that there is no edge between two vertices of the same color. This k then corresponds to the k in the feasibility condition (39).

Before each test round, the client chooses a color V_j uniformly at random to be the *trap color*. A qubit corresponding to vertices from this set is then referred to as a *trap qubit*. Any other qubit is referred to as a *dummy qubit*. Each trap qubit $v \in V_j$ will be $|+\theta_v\rangle := (|0\rangle + e^{i\theta_v}|1\rangle)/\sqrt{2}$, for some angle θ_v that is chosen uniformly at random from $\Theta = \{\frac{k\pi}{4} : k = 0, 1, \dots, 7\}$. Each dummy qubit $v \in V \setminus V_j$ will be $|d_v\rangle$, where $d_v \in \{0, 1\}$ is chosen uniformly at random. Then, the effect of the server applying its entangling gates is to flip each trap qubit to the orthogonal basis vector a number of times that corresponds to the sum (modulo 2) of the neighboring dummies. This is a quantity that the client can compute. After constructing the graph state, the server measures its qubits and sends the outcome to the client. The trap qubit measurement basis that is specified by the client is $\{| \pm \delta_v \rangle\}$, for each trap $v \in V_j$, where $\delta_v = \theta_v + r_v\pi$, and $r_v \in \{0, 1\}$ is chosen uniformly at random. The client compares the outcomes of the trap qubits to what is expected if all states and local operations are perfect, declaring the test round to be a failure if there is at least one trap measurement that is incorrect. A depiction of a graph state, a choice of k -coloring, and a choice of qubits for a test round are given in Fig. 11, for the case of a square graph.

2) ERROR PROBABILITY FOR A GENERAL GRAPH

We suppose that the client would like to know the outcome of a BQP calculation, which has corresponding graph $G = (V, E)$, and that the client has chosen a k -coloring $\{V_j\}_{j=1}^k$. Then, given that the vector of fidelities at the time the server applies its operations is $\vec{F} = (F_1, \dots, F_{|V|})$, here we obtain a general form for the probability of error of the test round, $P_G(\vec{F})$. This is a generalization of what can be found in [29], where BQC with two qubits is considered.

We first find the probability of error, *given* that the client has chosen trap color V_j . Call this P_{V_j} . The client, chooses

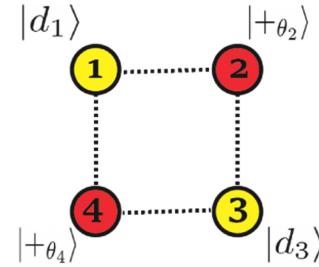


FIGURE 11. Test rounds in the BQC protocol. Here, we choose an example where the computation is performed on a square graph. The client chooses a k -coloring: here $k = 2$, and the two sets are colored in yellow and red. In this case, the red vertices correspond to trap qubits, and the yellow vertices correspond to dummy qubits.

to send the trap qubits $v \in V_j$ as states $|+\theta_v\rangle$. Then, at the time when the test round is carried out, the trap qubits corresponding to vertices $v \in V_j$ are each in a state ρ_v , where

$$\rho_v = F_v|+\theta_v\rangle\langle+\theta_v| + (1 - F_v)|-\theta_v\rangle\langle-\theta_v| + (\text{o.d.1}) \quad (108)$$

where we use (o.d.1) to write the off-diagonal elements (with respect to the basis $\{|+\theta_v\rangle, |-\theta_v\rangle\}$). We do not write them out in full because these end up making no contribution to $P_{V_j}(\vec{F})$, as we will see later. Similarly, the dummy qubits $v \in V \setminus V_j$ will be in the state

$$\rho_v = F_v|d_v\rangle\langle d_v| + (1 - F_v)|d_v \oplus 1\rangle\langle d_v \oplus 1| + (\text{o.d.2}) \quad (109)$$

where we use (o.d.2) to write the off-diagonal elements, this time with respect to the computational basis. The state of the server is then given by the tensor product of all of these states

$$\rho_{\text{server}} = \bigotimes_{v \in V_j} \rho_v \bigotimes_{w \in W_j} \rho_w \bigotimes_{u \in V \setminus (V_j \cup W_j)} \rho_u \quad (110)$$

where we have defined $W_j \subset V \setminus V_j$ to be the set of all vertices that share an edge with a trap qubit. The server then proceeds with the next step of the BQC protocol, and applies CZ gates to all pairs of qubits corresponding to edges in E , resulting in the state

$$\rho'_{\text{server}} = U \rho_{\text{server}} U^\dagger \quad (111)$$

where $U := \prod_{(w,v) \in E} CZ_{(w,v)}$. Recall that we are interested in the probability $P_{V_j}(\vec{F})$ that results in an error. In fact, is it simpler to find a form for the *success probability* $Q_{V_j}(\vec{F}) = 1 - P_{V_j}(\vec{F})$. An error occurs when at least one of the trap qubit measurements does not match the result that would be obtained if all states were perfect. In particular, if everything were perfect, then the client would expect the measurement outcome corresponding to the trap qubit v to be $r_v \oplus D_v$, where

$$D_v = \bigoplus_{w \in W_j : (v,w) \in E} d_w \quad (112)$$

i.e., the sum (modulo 2) of all the dummy variables d_w that surround the trap qubit. The success probability is then given

by

$$Q_{V_j}(\vec{F}) = \text{Tr}_V \left(\left(\bigotimes_{v \in V_j} |(-1)_{\delta_v}^{r_v \oplus D_v} \rangle \langle (-1)_{\delta_v}^{r_v \oplus D_v}| \right) \rho'_{\text{server}} \right) \quad (113)$$

where for convenience, we are using the notation $|(+1)_\theta\rangle \equiv |+\theta\rangle$ and $|(-1)_\theta\rangle \equiv |-\theta\rangle$. Rewriting $|(-1)_{\delta_v}^{r_v \oplus D_v}\rangle = |(-1)_{\theta_v}^{D_v}\rangle$, and after examining (110) and (111), we see that the qubit registers corresponding to vertices $v \in V \setminus (V_j \cup W_j)$ will make no contribution to this quantity, so that

$$Q_{V_j}(\vec{F}) = \text{Tr}_{V_j} \text{Tr}_{W_j} \left(\left(\bigotimes_{v \in V_j} |(-1)_{\theta_v}^{D_v}\rangle \langle (-1)_{\theta_v}^{D_v}| \right) \sigma'_{\text{server}} \right) \quad (114)$$

with

$$\sigma'_{\text{server}} := \tilde{U} \sigma_{\text{server}} \tilde{U}^\dagger \quad (115)$$

where we have defined $\sigma_{\text{server}} := \bigotimes_{v \in V_j} \rho_v \bigotimes_{w \in W_j} \rho_w$ and $\tilde{U} := \prod_{(w,v) \in E_j} \text{CZ}_{(w,v)}$, and $E_j := \{(v, w) \in E : v \in V_j\}$ to be the set of all edges between any element of V_j and any other vertex. Recalling the states of our qubits as given in (108) and (109), and defining $F^{(0)} := F$, $F^{(1)} := 1 - F$ to be used as a more concise way to write some of the terms, we can then write

$$\begin{aligned} \sigma_{\text{server}} &= \bigotimes_{v \in V_j} \left(\sum_{x_v \in \{0,1\}} F_v^{(x_v)} |(-1)_{\theta_v}^{x_v}\rangle \langle (-1)_{\theta_v}^{x_v}| \right) \cdot \\ &\quad \bigotimes_{w \in W_j} \left(\sum_{y_w \in \{0,1\}} F_w^{(y_w)} |d_w + y_w\rangle \langle d_w + y_w| \right) \\ &= \bigotimes_{v \in V_j} \left(\sum_{x_v \in \{0,1\}} F_v^{(x_v)} |(-1)_{\theta_v}^{x_v}\rangle \langle (-1)_{\theta_v}^{x_v}| \right) \cdot \\ &\quad \bigotimes_{\vec{y} \in \{0,1\}^{|W_j|}} \left(\prod_{w \in W_j} F_w^{(y_w)} |\vec{d} + \vec{y}\rangle \langle \vec{d} + \vec{y}| \right) \end{aligned} \quad (116)$$

where in (116) we have rewritten the sum to be over all length- $|W_j|$ binary strings $\vec{y} \equiv (y_w)_{w \in W_j} \in \{0,1\}^{|W_j|}$. We have also stored the dummy variables in a vector \vec{d} , so that $(\vec{d} + \vec{y})_w = d_w + y_w$. Again, we are not writing out the off-diagonal terms because these all disappear when we take the trace, and therefore make no contribution to the final expression. Applying the unitary operator \tilde{U} yields

$$\begin{aligned} \sigma'_{\text{server}} &= \sum_{\vec{y} \in \{0,1\}^{|W_j|}} \prod_{w \in W_j} F_w^{(y_w)} |\vec{d} + \vec{y}\rangle \langle \vec{d} + \vec{y}| \cdot \\ &\quad \bigotimes_{v \in V_j} \left(\sum_{x_v \in \{0,1\}} F_v^{(x_v)} |(-1)_{\theta_v}^{x_v + s_v(\vec{y}) + D_v}\rangle \langle (-1)_{\theta_v}^{x_v + s_v(\vec{y}) + D_v}| \right) \end{aligned}$$

where for a trap $v \in V_j$, we have defined

$$s_v(\vec{y}) := \sum_{w \in W_j : (w,v) \in E_j} y_w \quad (117)$$

which is the sum of the binary variables y_w over all vertices neighboring v . We can now start to trace out registers in order to find a final expression for $Q_{V_j}(\vec{F})$ in terms of the qubit fidelities. Taking the inner product $\langle (-1)_{\delta_v}^{D_v} | \dots | (-1)_{\delta_v}^{D_v} \rangle$ for all trap qubits $v \in V_j$ and tracing out W_j yields our final expression for the success probability, as introduced in (113)

$$Q_{V_j}(\vec{F}) = \sum_{\vec{y} \in \{0,1\}^{|W_j|}} \prod_{w \in W_j} F_w^{(y_w)} \prod_{v \in V_j} F_v^{(s_v(\vec{y}))}. \quad (118)$$

This is a polynomial in the fidelities $\vec{F} = (F_1, \dots, F_{|V|})$, with a form that is completely determined by the graph structure and choice of trap color V_j . The same thus holds for the error probability $P_{V_j}(\vec{F})$. In our model as given in Section IV-B2, it is further necessary to incorporate the fact that the first qubit to be sent is chosen at random. The probability of error is then effectively symmetrized over the $|V|$ possible starting qubits in the following way. Without loss of generality, letting the order in which the qubits are sent to be lexicographical, the probability of error is then

$$P_{V_j}(\vec{F}) \rightarrow \tilde{P}_{V_j}(\vec{F}) := \frac{1}{|V|} \sum_j P_{V_j}(\sigma^j \vec{F}) \quad (119)$$

where σ is the permutation that moves the vector elements one place to the left, i.e., $\sigma(F_1, \dots, F_{|V|}) = (F_2, \dots, F_{|V|}, F_1)$. To obtain the final probability of error, it remains to average over the choice of trap color, recalling that this is chosen uniformly at random. This gives us a final expression for $P_G(\vec{F})$

$$P_G(\vec{F}) = \frac{1}{k} \sum_{j=1}^k \tilde{P}_{V_j}(\vec{F}). \quad (120)$$

3) ERROR PROBABILITY FOR A SQUARE GRAPH

An example of such a polynomial for the case of a square graph is as follows. Consider the k -coloring, as in Fig. 11, with red as the choice of trap color. Suppose that when the server applies its gates and measurements, the qubits have fidelities $\vec{F} = (F_1, F_2, F_3, F_4)$. Then, according to (118), the success probability is given by

$$\begin{aligned} Q_{\text{red}}(\vec{F}) &= F_1 F_2 F_3 F_4 + F_1 (1 - F_2) (1 - F_3) (1 - F_4) \\ &\quad + (1 - F_1) (1 - F_2) F_3 (1 - F_4) + (1 - F_1) F_2 (1 - F_3) F_4 \end{aligned}$$

and the error probability is then

$$P_{\text{red}}(\vec{F}) = 1 - Q_{\text{red}}(\vec{F}). \quad (121)$$

By symmetry of the square graph, the error probability $P_{\text{yellow}}(\vec{F})$ is obtained by exchanging the indices $1 \leftrightarrow 2, 3 \leftrightarrow 4$ in (121). In the case of the square graph, the symmetrization

maps $P_{\text{red}}(\vec{F}) \rightarrow \tilde{P}_{\text{red}}(\vec{F})$, where

$$\begin{aligned}\tilde{P}_{\text{red}}(\vec{F}) = \frac{1}{4} & [P_{\text{red}}(F_1, F_2, F_3, F_4) + P_{\text{red}}(F_2, F_3, F_4, F_1) \\ & + P_{\text{red}}(F_3, F_4, F_1, F_2) + P_{\text{red}}(F_4, F_1, F_2, F_3)].\end{aligned}$$

Note that the symmetries of the error functions P_{red} and P_{yellow} reflect the symmetries of the graph, i.e., they are symmetric under the interchange of $1 \leftrightarrow 3$ or $2 \leftrightarrow 4$. Then

$$\tilde{P}_{\text{red}}(\vec{F}) = \frac{1}{2} \left(P_{\text{red}}(\vec{F}) + P_{\text{yellow}}(\vec{F}) \right). \quad (122)$$

The other error function P_{yellow} maps to the same after the symmetrization (119), i.e., $\tilde{P}_{\text{yellow}}(\vec{F}) = \tilde{P}_{\text{red}}(\vec{F})$. The probability of error, then, is given by

$$P_{\text{square}}(\vec{F}) = \frac{1}{2} \left(\tilde{P}_{\text{yellow}}(\vec{F}) + \tilde{P}_{\text{red}}(\vec{F}) \right) \quad (123)$$

$$= \frac{1}{2} \left(P_{\text{red}}(\vec{F}) + P_{\text{yellow}}(\vec{F}) \right). \quad (124)$$

This is the function that we use with (45) to calculate p_{av} for our model, and compute the results for an example of a square graph in Section IV-B3.

ACKNOWLEDGMENT

The authors would like to thank Francisco Ferreira da Silva, Janice van Dam, and Álvaro G. Iñesta for critical feedback on this article. Bethany Davies would like to thank the KNAW Ammodo award (SW).

REFERENCES

- C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, "Purification of noisy entanglement and faithful teleportation via noisy channels," *Phys. Rev. Lett.*, vol. 76, no. 5, 1996, Art. no. 722, doi: [10.1103/PhysRevLett.76.722](https://doi.org/10.1103/PhysRevLett.76.722).
- D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, "Quantum privacy amplification and the security of quantum cryptography over noisy channels," *Phys. Rev. Lett.*, vol. 77, no. 13, 1996, Art. no. 2818, doi: [10.1103/PhysRevLett.77.2818](https://doi.org/10.1103/PhysRevLett.77.2818).
- S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, 2018, Art. no. eaam9288, doi: [10.1126/science.aam9288](https://doi.org/10.1126/science.aam9288).
- S. Bravyi and A. Kitaev, "Universal quantum computation with ideal Clifford gates and noisy ancillas," *Phys. Rev. A*, vol. 71, no. 22005, Art. no. 022316, doi: [10.1103/PhysRevA.71.022316](https://doi.org/10.1103/PhysRevA.71.022316).
- S. D. Barrett and P. Kok, "Efficient high-fidelity quantum computation using matter qubits and linear optics," *Phys. Rev. A*, vol. 71, no. 6, 2005, Art. no. 060310, doi: [10.1103/PhysRevA.71.060310](https://doi.org/10.1103/PhysRevA.71.060310).
- C. Cabrillo, J. I. Cirac, P. García-Fernandez, and P. Zoller, "Creation of entangled states of distant atoms by interference," *Phys. Rev. A*, vol. 59, no. 2, 1999, Art. no. 1025, doi: [10.1103/PhysRevA.59.1025](https://doi.org/10.1103/PhysRevA.59.1025).
- S. Baier et al., "Realization of a multi-node quantum network of remote solid-state qubits," in *Quantum Information and Measurement*. Washington, DC, USA: Optica Publishing Group, 2021, pp. M2A-2, doi: [10.1126/science.abg1919](https://doi.org/10.1126/science.abg1919).
- M. Pompili et al., "Experimental demonstration of entanglement delivery using a quantum network stack," 2021, *arXiv:2111.11332*, doi: [10.4121/16912522](https://doi.org/10.4121/16912522).
- V. Krutyanskiy, "Entanglement of trapped-ion qubits separated by 230 meters," *Phys. Rev. Lett.*, vol. 130, Feb. 2023, Art. no. 050803, doi: [10.1103/PhysRevLett.130.050803](https://doi.org/10.1103/PhysRevLett.130.050803).
- T. van Leent et al., "Long-distance distribution of atom-photon entanglement at telecom wavelength," *Phys. Rev. Lett.*, vol. 124, no. 1, 2020, Art. no. 0010510, doi: [10.1103/PhysRevLett.124.010510](https://doi.org/10.1103/PhysRevLett.124.010510).
- B. Li, T. Coopmans, and D. Elkouss, "Efficient optimization of cut-offs in quantum repeater chains," in *Proc. IEEE Int. Conf. Quantum Comput. Eng.*, 2020, pp. 158–168, doi: [10.1109/TQE.2021.3099003](https://doi.org/10.1109/TQE.2021.3099003).
- S. Khatri, "Policies for elementary links in a quantum network," *Quantum*, vol. 5, Sep. 2021, Art. no. 537, doi: [10.22331/q-2021-09-07-537](https://doi.org/10.22331/q-2021-09-07-537).
- L. Praxmeyer, "Reposition time in probabilistic imperfect memories," 2013, *arXiv:1309.3407*, doi: [10.48550/arXiv.1309.3407](https://doi.org/10.48550/arXiv.1309.3407).
- E. Shchukin, F. Schmidt, and P. van Loock, "Waiting time in quantum repeaters with probabilistic entanglement swapping," *Phys. Rev. A*, vol. 100, no. 3, 2019, Art. no. 0032322, doi: [10.1103/PhysRevA.100.032322](https://doi.org/10.1103/PhysRevA.100.032322).
- A. G. Iñesta, G. Vardoyan, L. Scavuzzo, and S. Wehner, "Optimal entanglement distribution policies in homogeneous repeater chains with cut-offs," 2022, *arXiv:2207.06533*, doi: [10.1038/s41534-023-00713-9](https://doi.org/10.1038/s41534-023-00713-9).
- F. Rozpędek et al., "Parameter regimes for a single sequential quantum repeater," *Quantum Sci. Technol.*, vol. 3, no. 32018, Art. no. 034002, doi: [10.1088/2058-9565/aab31b](https://doi.org/10.1088/2058-9565/aab31b).
- Y. Ma, Y.-Z. Ma, Z.-Q. Zhou, C.-F. Li, and G.-C. Guo, "One-hour coherent optical storage in an atomic frequency comb memory," *Nature Commun.*, vol. 12, no. 1, 2021, Art. no. 2381, doi: [10.1038/s41467-021-22706-y](https://doi.org/10.1038/s41467-021-22706-y).
- D. Leichtle, L. Music, E. Kashefi, and H. Ollivier, "Verifying BQP computations on noisy devices with minimal overhead," *PRX Quantum*, vol. 2, no. 4, 2021, Art. no. 040302, doi: [10.1103/PRXQuantum.2.040302](https://doi.org/10.1103/PRXQuantum.2.040302).
- J. Glaz, J. I. Naus, and S. Wallenstein, *Scan Statistics*. Berlin, Germany: Springer, 2001, doi: [10.1007/978-1-4757-3460-7](https://doi.org/10.1007/978-1-4757-3460-7).
- J. Glaz, V. Pozdnyakov, and S. Wallenstein, *Scan Statistics: Methods and Applications*. Boston, MA, USA: Birkhäuser, 2009, doi: [10.1007/978-0-8176-4749-0](https://doi.org/10.1007/978-0-8176-4749-0).
- S.-Y. R. Li, "A martingale approach to the study of occurrence of sequence patterns in repeated experiments," *Ann. Probability*, vol. 8, no. 6, pp. 1171–1176, 1980, doi: [10.1214/aop/1176994578](https://doi.org/10.1214/aop/1176994578).
- J. C. Fu and W. W. Lou, *Distribution Theory of Runs and Patterns and Its Applications: A Finite Markov Chain Imbedding Approach*. River Edge, NJ, USA: World Scientific, 2003, doi: [10.1142/4669](https://doi.org/10.1142/4669).
- M. Ebneshrashroob, T. Gao, and M. Sobel, "Sequential window problems," *Sequential Anal.*, vol. 24, no. 2, pp. 159–175, 2005, doi: [10.1081/SQA-200056194](https://doi.org/10.1081/SQA-200056194).
- V. Pozdnyakov and J. M. Steele, "Martingale methods for patterns and scan statistics," in *Scan Statistics: Methods and Applications*. Boston, MA, USA: Birkhäuser, 2009, pp. 289–317, doi: [10.1007/978-0-8176-4749-0_14](https://doi.org/10.1007/978-0-8176-4749-0_14).
- [Online]. Available: <https://github.com/bethanydavies234/tools-paper>
- M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge, MA, USA: Cambridge Univ. Press, 2002, doi: [10.1017/CBO9780511976667](https://doi.org/10.1017/CBO9780511976667).
- R. Raussendorf and H. J. Briegel, "A one-way quantum computer," *Phys. Rev. Lett.*, vol. 86, no. 22, 2001, Art. no. 5188, doi: [10.1103/PhysRevLett.86.5188](https://doi.org/10.1103/PhysRevLett.86.5188).
- V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner, "Composable security of delegated quantum computation," in *Proc. 20th Int. Conf. Theory Appl. Cryptol. Inf. Secur. Adv. Cryptol.*, 2014, pp. 406–425, doi: [10.1007/978-3-662-45608-8_22](https://doi.org/10.1007/978-3-662-45608-8_22).
- G. Avis et al., "Requirements for a processing-node quantum repeater on a real-world fiber grid," *npj Quantum Inf.*, vol. 9, 2023, Art. no. 100, doi: [10.1038/s41534-023-00765-x](https://doi.org/10.1038/s41534-023-00765-x).
- S.-Y. R. Li, "A martingale approach to the study of occurrence of sequence patterns in repeated experiments," *Ann. Probability*, vol. 8, no. 6, pp. 1171–1176, 1980, doi: [10.1214/aop/1176994578](https://doi.org/10.1214/aop/1176994578).
- D. Williams, *Probability With Martingales*. Cambridge, U.K.: Cambridge Univ. Press, Feb. 1991, doi: [10.1017/CBO9780511813658](https://doi.org/10.1017/CBO9780511813658).
- R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985, doi: [10.1017/CBO9780511810817](https://doi.org/10.1017/CBO9780511810817).
- 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, Mar. 1993, doi: [10.1103/PhysRevLett.70.1895](https://doi.org/10.1103/PhysRevLett.70.1895).
- M. Horodecki, P. Horodecki, and R. Horodecki, "General teleportation channel, singlet fraction, and quasidistillation," *Phys. Rev. A*, vol. 60, no. 3, pp. 1888–1898, Sep. 1999, doi: [10.1103/PhysRevA.60.1888](https://doi.org/10.1103/PhysRevA.60.1888).
- S. L. N. Hermans et al., "Entangling remote qubits using the single-photon protocol: An in-depth theoretical and experimental study," *New J. Phys.*, vol. 25, 2023, Art. no. 013011, doi: [10.1088/1367-2630/acb004](https://doi.org/10.1088/1367-2630/acb004).