Distributed Quantum Proofs for Replicated Data

Pierre Fraigniaud
IRIF, CNRS and Université de Paris, France

François Le Gall
Graduate School of Mathematics, Nagoya University, Japan

Harumichi Nishimura
Graduate School of Informatics, Nagoya University, Japan

Ami Paz
Faculty of Computer Science, Universität Wien, Austria

Abstract

This paper tackles the issue of *checking* that all copies of a large data set replicated at several nodes of a network are identical. The fact that the replicas may be located at distant nodes prevents the system from verifying their equality locally, i.e., by having each node consult only nodes in its vicinity. On the other hand, it remains possible to assign *certificates* to the nodes, so that verifying the consistency of the replicas can be achieved locally. However, we show that, as the replicated data is large, classical certification mechanisms, including distributed Merlin-Arthur protocols, cannot guarantee good completeness and soundness simultaneously, unless they use very large certificates. The main result of this paper is a distributed *quantum* Merlin-Arthur protocol enabling the nodes to collectively check the consistency of the replicas, based on small certificates, and in a single round of message exchange between neighbors, with short messages. In particular, the certificate-size is logarithmic in the size of the data set, which gives an exponential advantage over classical certification mechanisms. We propose yet another usage of a fundamental quantum primitive, called the SWAP test, in order to show our main result.

2012 ACM Subject Classification Theory of computation → Quantum communication complexity; Theory of computation → Distributed computing models

Keywords and phrases Quantum Computing, Distributed Network Computing, Algorithmic Aspects of Networks

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.28

Related Version A full version of the paper is available at `https://arxiv.org/abs/2002.10018`.

Funding *Pierre Fraigniaud*: Additional support from the ANR projects *DESCARTES* and *QuDATA*.
François Le Gall: JSPS KAKENHI grants Nos. JP16H01705, JP19H04066, JP20H04139 and JP20H00579 and MEXT Quantum Leap Flagship Program Grant Number JPMXS0120319794.
Harumichi Nishimura: JSPS KAKENHI grants Nos. JP16H01705, JP19H04066 and MEXT Quantum Leap Flagship Program Grant Number JPMXS0120319794.
Ami Paz: We acknowledge the Austrian Science Fund (FWF) and netIDEE SCIENCE project P 33775-N.

Acknowledgements We thank the anonymous reviewer of ITCS 2021 who pointed [44] to us, and Uri Meir for useful discussions.

1 Introduction

In the context of distributed systems, the presence of faults potentially corrupting the individual states of the nodes creates a need to regularly check whether the system is in a global state that is legal with respect to its specification. A basic example is a system storing data, and using replicas in order to support crash failures. In this case, the application managing the data is in charge of regularly checking that the several replicas of the same

12th Innovations in Theoretical Computer Science Conference (ITCS 2021).
Editor: James R. Lee; Article No. 28; pp. 28:1–28:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

data, stored at different nodes scattered in the network, are all identical. Another example is an application maintaining a tree spanning the nodes of a network, e.g., for multicast communication. In this case, every node stores a pointer to its parent in the tree, and the application must regularly check that the collection of pointers forms a spanning tree. This paper addresses the issue of checking the correctness of a distributed system configuration at low cost.

Several mechanisms have been designed for certifying the correctness of the global state of a system in a distributed manner. One popular mechanism is called *locally checkable proofs* [24], and it extends the seminal concept of *proof-labeling schemes* [35]. In these frameworks, the distributed application does not only construct or maintain some distributed data structure (e.g., a spanning tree), but also constructs a distributed *proof* that the data structure is correct. This proof has the form of a *certificate* assigned to each node (the certificates assigned to different nodes do not need to be the same). For collectively checking the legality of the current global system state, the nodes exchange their certificates with their neighbors in the network. Then, based on its own individual state, its certificate, and the certificates of its neighbors, every node accepts or rejects, according to the following specification. If the global state is legal, and if the certificates are assigned properly by the application, then all nodes accept. Conversely, if the global state is illegal, then at least one node rejects, *no matter which certificates are assigned to the nodes.* Such a rejecting node can raise an alarm, or launch a recovery procedure. The main aim of locally checkable proofs is to be *compact*, that is, to use certificates as small as possible, for two reasons: first, to limit the space complexity at each node, and, second, to limit the message complexity of the verification procedure involving communications between neighbors.

For instance, in the case of the Spanning Tree predicate, the application does not only construct a spanning tree T of the network, but also a distributed proof that T is indeed a spanning tree, i.e., that the collection T of pointers forms a cycle-free connected spanning subgraph. It has been known for long [2, 6, 26] that, by assigning to every node a certificate of logarithmic size, the nodes can collectively check whether T is indeed a spanning tree, in a single round of communication between neighboring nodes. The certificate assigned to a node is the identity of the root of the tree, and its distance to this root (both are of logarithmic size as long as the IDs are in a range polynomial in the number of nodes). Every node just checks that it is provided with the same root-ID as all its neighbors in the network, and that the distance given to its parent in its certificate is one less than its own given distance – a node with distance 0 checks that its ID is indeed the root-ID provided in its certificate. Obviously, if the collection T of pointers forms a spanning tree, and if the certificates are assigned properly by the application, then all nodes pass these tests, and accept. On the other hand, it is easy to check that if T is not a spanning tree (it is not connected, or it contains a cycle), then at least one node detects a discrepancy and rejects, no matter which certificates are assigned to the nodes.

Unfortunately, not all boolean predicates on labeled graphs can be distributedly certified using certificates as small as for spanning tree. This is typically the case of the aforementioned scenario of a distributed data storage using replicas, for which one must certify equality. Let us for instance consider the case of two nodes Alice and Bob at the two extremities of a path, that is, the two players are separated by intermediate nodes. Alice and Bob respectively store two n-bit strings x and y, and the objective is to certify that $x = y$. That is, one wants to certify equality (EQ) between *distant* players. A direct reduction from the non-deterministic communication complexity of EQ shows that certifying EQ cannot be achieved with certificates smaller than $\Omega(n)$ bits.

Randomization may help circumventing the difficulty of certifying some boolean predicates on labeled graphs using small certificates. Hence, a weaker form of protocols has been considered, namely *distributed Merlin-Arthur* protocols (dMA), a.k.a. *randomized proof-labeling schemes* [22]. In this latter context, Merlin provides the nodes with a proof, just like in locally checkable proofs, and Arthur performs a *randomized* local verification at each node. Unfortunately, some predicates remain hard in this framework too. In particular, as we show in the paper, there are no classical dMA protocols for (distant) EQ using compact certificates. Recently, several extensions of dMA protocols were proposed, e.g., by allowing more interaction between the prover and the verifier [15, 21, 39]. In this work, we add the quantum aspect, while considering only a single interaction, and only in the prescribed order: Merlin sends a proof to Arthur, and then there is no more interaction between them.

1.1 Our Results

We carry on the recent trend of research consisting of investigating the power of quantum resources in the context of distributed network computing (cf., e.g., [17, 37, 27, 38, 28, 23]), by designing a distributed Quantum Merlin-Arthur (dQMA) protocol for distant EQ, using compact certificates and small messages. While we use the dQMA terminology in order to be consistent with prior work, we emphasize that the structure of the discussed protocols is rather simple: each node is given a quantum state as a certificate, the nodes exchange these states, perform a local computation, and finally accept or reject.

Our main result is the following. A collection of n-bit strings $x_1, \dots, x_t$ are stored at t terminal nodes $u_1, \dots, u_t$ in a network $G = (V, E)$, where node u_i stores x_i. We denote EQ_n^t the problem of checking the equality $x_1 = \dots = x_t$ between the t strings. Let us define the *radius* of a given instance of EQ_n^t as $r = \min_i \max_j \mathsf{dist}_G(u_i, u_j)$, where dist_G denotes the distance in the (unweighted) graph G. Our main result is the design of a dQMA protocol for EQ_n^t, using small certificate. This can be summarized by the following informal statement (the formal statement is in Section 5):

▶ **Main Result.** *There is a distributed Quantum Merlin-Arthur (dQMA) protocol for certifying equality between t binary strings (EQ_n^t) of length n, and located at a radius-r set of t terminals, in a single round of communication between neighboring nodes using certificates of size $O(tr^2 \log n)$ qubits, and messages of size $O(tr^2 \log(n + r))$ qubits.*

It is worth mentioning that, although the dependence in r and t is polynomial, the dependence in the actual size n of the instance remains logarithmic, which is our main concern. Indeed, for applications such as the aforementioned distributed data storage motivating the distant EQ_n^t problem, it is expected that both the number t of replicas, and the maximum distance between the nodes storing these replicas are of several orders of magnitude smaller than the size n of the stored replicated data.

It is also important to note that our protocol satisfies the basic requirement of *reusability*, as one aims for protocols enabling regular and frequent verifications that the data are not corrupted. Specifically, the quantum operations performed on the certificates during the local verification phase operated between neighboring nodes preserve the quantum nature of these certificates. That is, if EQ_n^t is satisfied, i.e., if all the replicas x_i's are equal, then, up to an elementary local relocation of the quantum certificates, these certificates are available for a next test. If EQ_n^t is not satisfied, i.e., if there exists a pair of replicas $x_i \neq x_j$, then the certificates do not need to be preserved as this scenario corresponds to the case where the correctness of the data structure is violated, requiring the activation of recovery procedures for fixing the bug, and reassigning certificates to the nodes.

Our quantum protocol is based on the SWAP test [12], which is a basic tool in the theory of quantum computation and quantum information. This test allows to check if a quantum state is symmetric, and has several applications, such as estimating the inner product of two states (e.g., [12, 9, 47]), checking whether a given state (or a reduced state of it) is pure or entangled with the environment system (e.g., [1, 33, 25, 32]), and more. In this paper, we use the SWAP test in yet another way: *for checking if two of the reduced states of a given state are close.* A similar use was done by Rosgen [44] in a different context – transforming quantum circuits to shallow ones in a hardness reduction proof.

Finally, observe that our logarithmic upper bound for dQMA protocols is in contrast to the linear lower bound that can be shown for classical dMA protocols even for $t = 2$ on a path of 4 nodes and even for the case where communication between the neighboring nodes is extended to multiple rounds (see precise statement and proof in Section 6). Our results thus show that quantum certification mechanism can provide an exponential advantage over classical certification mechanisms.

1.2 Related Work

The concept of distributed proofs is a part of the framework of distributed network computing since the early works on fault-tolerance (see, e.g., [2, 6, 26]). Proof-labeling schemes were introduced in [35], and variants have been studied in [24, 20]. Randomized proof-labeling schemes have been studied in [22]. Extensions of distributed proofs to a hierarchy of decision mechanisms have been studied in [18] and [7]. Frameworks like cloud computing recently enabled envisioning systems in which the nodes of the network could interact with a third party, leading to the concept of *distributed interactive proofs* [34]. There, each node can interact with an *oracle* who has a complete view of the system, is computationally unbounded, but is not trustable. For instance, in Arthur-Merlin (dAM) protocols, the nodes start by querying the oracle Merlin, which provides them with answers in their certificates. There is a simple classical compact dAM protocol for distant EQ, where the two players stand at the extremities of a path (see Section 3). We refer to [15, 21, 39] for recent developments in the framework of distributed interactive proofs. While distributed Arthur-Merlin protocols and their extensions provide an appealing theoretical framework for studying the power of interactive proofs in the distributed setting, the practical implementation of such protocols remains questionable, since they all require the existence of a know-all oracle, Merlin, and it is unclear if a Cloud could play this role. On the other hand, in dMA and dQMA protocols, interaction with an external party is not required, but only a one-time assignment of certificates is needed, which are then reusable for regular verification. As in the classical proof-labeling schemes setting, these certificates can actually be *created* by the nodes themselves during a pre-processing phase, making the reliance on a know-all oracle unnecessary.

After a few early works [8, 17, 23, 45] that shed light on the potential and limitations of quantum distributed computing (see also [5, 11, 16] for general discussions), evidence of the advantage of quantum distributed computing over classical distributed computing have been obtained recently for three fundamental models of (synchronous fault-free) distributed network computing: the CONGEST model [28, 37], the CONGEST-CLIQUE model [27] and the LOCAL model [38]. The present paper adds to this list another important task for which quantum distributed computing significantly outperforms classical distributed computing, namely, distributed certification.

Note that while this paper is the first to study quantum Merlin-Arthur protocols in a distributed computing framework, there are a number of prior works studying them in communication complexity [43, 30, 31, 10]. In particular, quantum Merlin-Arthur protocols are shown to improve some computational measure (say, the total length of the messages from the prover to Alice, and of the messages between Alice and Bob) exponentially compared to Merlin-Arthur protocols where the messages from the prover are classical [43, 31].

The question of computing functions on inputs that are given to graph nodes was also studied in the context of communication complexity. The equality function was studied for the case where all nodes have inputs [4]. Other works considered a setting similar to ours, i.e., where only some nodes have inputs [13, 14], but did not study the equality problem.

2 Model and Definitions

Distributed verification on graphs. Let $t \geq 2$, and let $f\colon (\{0,1\}^n)^t \to \{0,1\}$ be a function. The aim of the nodes is to collectively decide whether $f(x_1,\dots,x_t) = 1$ or not, where $x_1,\dots,x_t$ are assigned to t nodes of a graph. Specifically, an instance of the problem f is a t-tuple $(x_1,\dots,x_t) \in \{0,1\}^n \times \cdots \times \{0,1\}^n$, a connected graph $G = (V,E)$, and an ordered sequence $v_1,\dots,v_t$ of distinct nodes of G. The node v_i is given x_i as input, for $i = 1,\dots,t$. All the other nodes receive no inputs. We consider distributed Merlin-Arthur (dMA) protocols for deciding whether $f(x_1,\dots,x_t) = 1$, in which a non-trustable *prover* (Merlin) assigns (or "sends") *certificates* to the nodes, and then the nodes (Arthur) perform a 1-round randomized verification algorithm. The verification algorithm consists of each node simultaneously sending messages to all its immediate neighbors, receiving messages from them, then performing a local computation, and finally accepting or rejecting locally.[1] We say that a dMA protocol has *completeness* a and *soundness* b for a function f if the following holds for every $(x_1,\dots,x_t) \in \{0,1\}^n \times \cdots \times \{0,1\}^n$, every connected graph G, and every ordered sequence $v_1,\dots,v_t$ of distinct nodes in G:

(completeness) if $f(x_1,\dots,x_t) = 1$, then the prover can assign certificates to the nodes such that $\Pr[\text{all nodes accept}] \geq a$;

(soundness) if $f(x_1,\dots,x_t) = 0$, then, for every certificate assignment by the prover, $\Pr[\text{all nodes accept}] \leq b$.

The completeness condition guarantees that, when the system is in a "legal" state (specified by $f(x_1,\dots,x_t) = 1$), with probability at least a all nodes accept. The soundness condition guarantees that, when the system is in an "illegal" state (specified by $f(x_1,\dots,x_t) = 0$), with probability at least $1-b$ at least one node rejects. The value b represents the error probability of the protocol on an illegal instance, and thus we sometimes refer to it as the *soundness error*. A node detecting illegality of the state can raise an alarm, or launch a recovery procedure. Protocols with completeness 1 are called 1-sided protocols, or protocols with perfect completeness. Similarly to prior works on distributed verification, the certificate size of the protocol is measured as the maximum size (over all the nodes of the network) of the certificate sent by the prover to one of the nodes, and the message size of the protocol is measured as the maximum size (over all pairs of adjacent nodes) of the message exchanged between two adjacent nodes. Specifically, we will consider the multi-party version of the equality function, EQ_n^t, which is the boolean-valued function from $(\{0,1\}^n)^t$ such that $\mathsf{EQ}_n^t(x_1,\dots,x_t) = 1 \iff x_1 = \cdots = x_t$.

[1] We can naturally extend this definition to define dMA protocols with μ rounds of communication among neighbors, for any integer $\mu \geq 1$. In this paper, however, we focus on the case $\mu = 1$ since all the protocols we design use only 1-round verification algorithms. The only exception is Section 6, where we show classical lower bounds that hold even for $\mu > 1$.

In this work, we extend the framework of dMA protocols, to consider also cases where the certificates given to the nodes can contain qubits (although they may also contain classical bits) and the nodes can exchange messages consisting of qubits. These will be called *distributed Quantum Merlin-Arthur* (dQMA) protocols. More precisely, in a dQMA protocol for a function f, a non-trustable prover first sends a certificate to each node, which consists of a quantum state and classical bits; the quantum states may be entangled, even though all our quantum protocols do not require any prior entanglement, nor any shared classical random bits. Then the nodes perform a 1-round quantum verification algorithm, where each node simultaneously sends a quantum message to all its immediate neighbors, receives quantum messages from them, then performs a local computation, and finally accepts or rejects locally. Note that, as opposed to the classical setting, we cannot assume that a node simply broadcasts its certificate to all its neighbors, as quantum states cannot be duplicated. However, a node can still send copies of the classical parts of the certificate. We define completeness and soundness of dQMA protocols as for dMA protocols.

▶ Remark. A special case of interest is when the graph G is a path $v_0, \dots, v_r$, $r \geq 1$, where the left-end node v_0 has an n-bit string x as input, the right-end node v_r has an n-bit string y as input, and the intermediate nodes $v_1, \dots, v_{r-1}$ have no inputs. That is, $t = 2$. Given a function $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, the aim of the nodes is to collectively decide whether $f(x,y) = 1$ or not. This setting is very much related to communication complexity.

Classical two-party communication complexity. We refer to [36] for the basic concepts of two-party communication complexity. In this paper we will only consider two-party one-way communication complexity. In this model two parties, denoted Alice and Bob, each receives an input $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$, respectively. The goal is for Bob to output the value $f(x,y)$ for some known Boolean function $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Only Alice can send a message to Bob. The one-way two-sided-error communication complexity of f is the minimum number of bits that have to be sent on the worst input in a protocol that outputs the correct answer with probability at least $2/3$. The one-way one-sided-error communication complexity of f is the minimum number of bits that have to be sent on the worst input in a protocol that outputs the correct answer with probability 1 on any 1-input, and outputs the correct answer with probability at least $2/3$ on any 0-input.

We shall especially consider the following two functions. The equality function EQ_n is defined as $\mathsf{EQ}_n(x,y) = 1$ when $x = y$ and $\mathsf{EQ}_n(x,y) = 0$ otherwise, for any $x, y \in \{0,1\}^n$. Its one-way one-sided-error communication complexity is $O(\log n)$ – see, e.g., [36]. For any integer $d \geq 0$, the Hamming distance function HAM_n^d is defined as follows: for any $x, y \in \{0,1\}^n$, $\mathsf{HAM}_n^d(x,y) = 1$ if the Hamming distance between x and y is at most d, and $\mathsf{HAM}_n^d(x,y) = 0$ otherwise. It is known [47] that, for d constant, the one-way two-sided-error communication complexity of HAM_n^d is $O(\log n)$.

For any Boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, a set $S \subseteq \{0,1\}^n \times \{0,1\}^n$ is a 1-*fooling set* for f if, on the one hand, for every $(x,y) \in S$, $f(x,y) = 1$, and, on the other hand, for every two pairs $(x_1,y_1) \neq (x_2,y_2)$ in $S \times S$, $f(x_1,y_2) = 0$ or $f(x_2,y_1) = 0$.

Quantum two-party communication complexity. We assume the reader is familiar with the basics of quantum computation, in particular the notion of qubits, Dirac notation such as $|\psi\rangle$ and $\langle\psi| := (|\psi\rangle)^\dagger$, and the quantum circuit model (see Sections 2 and 4 in Ref. [40], for instance). In the full version of this paper, we present more advanced concepts such as mixed states that will be used in some of our proofs.

Quantum two-party communication complexity, first introduced by Yao [46], is defined similarly to the classical version. The only difference is that the players are allowed to exchange qubits instead of bits (the cost of a quantum protocol is the number of qubits sent by the protocol). Note that since quantum protocols can trivially simulate classical protocols, the quantum communication complexity of a function is never larger than its classical communication complexity. More precisely, an m-qubit one-way quantum protocol π for the function f can be described in its most general form as follows. Alice prepares an m-qubit (pure) quantum state $|h_x\rangle$ and sends it to Bob.[2] Bob then makes a measurement on the state $|h_x\rangle$, which gives an outcome $b \in \{0,1\}$. Finally, Bob outputs b. Since Bob's measurement in the above description depends only on his input y, it can be mathematically described, for each $y \in \{0,1\}^n$, by two positive semi-definite matrices $M_{y,0}$ and $M_{y,1}$ such that $M_{y,0} + M_{y,1} = I$. This pair $\{M_{y,0}, M_{y,1}\}$ is called a POVM measurement (POVM measurements are the most general form of measurements allowed by quantum mechanics). If $|h_x\rangle$ is measured by the POVM $\{M_{y,0}, M_{y,1}\}$, the probability that $b = 0$ is $\mathrm{tr}(M_{y,0}(|h_x\rangle\langle h_x|))$, while the probability that $b = 1$ is $\mathrm{tr}(M_{y,1}(|h_x\rangle\langle h_x|))$.

3 General Overview of our Techniques

Let us provide an intuition of our protocol in the case of EQ_n^2 over a path $v_0, \ldots, v_r$ of length $r \geq 1$ in which the terminals are the two nodes v_0 and v_r (that we rename Alice and Bob, for convenience). Let us call x and y the n-bit strings owned by Alice and Bob, respectively. There is a simple classical protocol for distant equality in a somewhat similar setting, where the verifier (Arthur, consisting of all the graph nodes) can send random bits to the prover (Merlin) before receiving the certificates; this is called a dAM protocol. In this protocol, Alice picks a hash function h at random in an appropriate family of hash functions (i.e., a family such that both h and $h(x)$ can be encoded using $O(\log n)$ bits and such that the probability that $h(x) \neq h(y)$ is high when $x \neq y$). Merlin provides every node with the certificate $(h, h(x))$, each node checks it received the same certificates as its neighbors, and Bob additionally checks whether $h(x) = h(y)$. Obviously, one cannot switch the order of Arthur and Merlin, as letting Merlin choose the hash function would enable him to fool Arthur on illegal instances by picking h that hashes identically the distinct input strings x and y. The main idea of our dQMA protocol is to ask Merlin to provide the nodes with a quantum certificate consisting of the *quantum superposition* of all the possible hashes.

Entering slightly more into the details, for any $x \in \{0,1\}^n$ we consider the *quantum fingerprint* $|h_x\rangle = \frac{1}{\sqrt{K}} \sum_h |h\rangle|h(x)\rangle$, where the sum is over all the hash functions, and $\frac{1}{\sqrt{K}}$ is the normalization factor of the quantum state. By using the same family of hash functions as in the aforementioned dAM protocol, these fingerprints can be constructed in such a way that their length is $O(\log n)$ qubits, and $|h_x\rangle$ and $|h_y\rangle$ are very far (more precisely, almost orthogonal) when $x \neq y$. Checking whether the two quantum fingerprints $|h_x\rangle$ and $|h_y\rangle$ are either equal or far apart can be achieved by a quantum test called the SWAP test [12]. Formally, the probability that the SWAP test accepts is $1/2 + |\langle h_x|h_y\rangle|^2/2$, where $\langle h_x|h_y\rangle$ denotes the inner product between the two quantum states $|h_x\rangle$ and $|h_y\rangle$.

Let us now describe the outline of our dQMA protocol. In the protocol each intermediate node $v_1, \ldots, v_{r-1}$ expects to receive the quantum fingerprint $|h_x\rangle$. Alice, who does not receive any certificate, creates by herself the fingerprint $|h_x\rangle$, which depends only on x. Similarly, Bob creates by himself the fingerprint $|h_y\rangle$. The checking procedure simply checks whether

[2] Without loss of generality, we assume that Alice does not use any mixed state (i.e., a probability distribution on pure states) in her message, as she can simulate it using a pure state called the *purification* [40] whose length is at most twice the one of the mixed state.

all these $(r+1)$ fingerprints are equal. This is done by applying the SWAP test to check whether the fingerprints owned by adjacent nodes are equal or not. There are however a few subtleties. In particular, since our analysis crucially requires that the SWAP tests do not overlap, for each node we need to decide whether it will perform the SWAP test with its right neighbor or its left neighbor. We do it in a randomized way and deal carefully with the conflicting choices that appear. For the case $x = y$ all the SWAP tests then succeed with probability 1 and thus all the nodes accept.

For the case $x \neq y$, let us provide some intuition about why a prover cannot fool the nodes for convincing them to all accept. To simplify the description we assume below that $|h_x\rangle$ and $|h_y\rangle$ are orthogonal (instead of only almost orthogonal). If the prover was forced to send certificates restricted to *product states* of the form $|g_1\rangle \otimes |g_2\rangle \otimes \cdots |g_{r-1}\rangle$ where $|g_j\rangle$ is the state to the jth node, then a fairly straightforward argument would guarantee that, with large probability, at least one node rejects. Indeed, under the product states restriction, intuitively the best strategy for the prover to cheat is to send states "intermediate" between $|h_x\rangle$ and $|h_y\rangle$, namely, to send the state $|g_j\rangle = \cos(\pi j/2r)|h_x\rangle + \sin(\pi j/2r)|h_y\rangle$ to node v_j for each $j \in \{1, \ldots, r-1\}$. Then, the probability that all nodes accept when performing the SWAP tests would be roughly $\prod_{j=1}^{r-1}(1/2 + |\langle g_j|g_{j+1}\rangle|^2/2) = 1 - \Omega(1/r)$. The cheating prover could then be caught with probability $\Omega(1/r)$, and this probability can be amplified to $\Omega(1)$ by asking the prover to send several copies of the certificates (amplification is possible since our protocol has perfect completeness).

The formal analysis of the protocol however faces several difficulties, which are mostly due to the nature of quantum computation, and are especially challenging to handle in the framework of distributed computation. For instance, quantum states cannot be duplicated (the "no-cloning Theorem"), which implies that a same quantum state cannot be used for parallel tests. Additionally, even sequential tests face the difficulty that the first test may collapse the quantum state, making the second test impossible to perform (or at least significantly complicating the analysis of the second test). Thus node v_i cannot perform the SWAP test with its two neighbors v_{i-1} and v_{i+1} simultaneously and (as already mentioned) we have to design carefully the protocol so that the SWAP tests do not overlap. A second, and much more problematic issue is that the non-trustable Merlin can send arbitrary certificates to the nodes for fooling them. In particular it is not restricted to send certificates that are product states. A priori, it may seem that the SWAP test is not strong enough to handle fooling strategies beyond product states. In this work we show that the SWAP test can actually detect such fooling strategies.

Specifically, our approach consists in considering the so-called *reduced states*, and to establish the following property of the SWAP test (cf. Lemma 5 in Section 4). If the SWAP test accepts with high probability when applied on the part of any two adjacent nodes in a (possibly non-product) global quantum state resulting from the certificates, then the two reduced states of that part (which is a bipartite state) must be close. As the two states $|h_x\rangle$ and $|h_y\rangle$ are very far apart when $x \neq y$, we can thus use this result to show that there is a good probability that the SWAP test rejects at some node. Moreover, using reduced states allows us to overcome other technical difficulties in the analysis of the (non-overlapping) SWAP tests we consider. Indeed, some form of average-case success probability of all the SWAP tests can be considered, instead of having to argue about the probability that all the SWAP tests individually accept.

4 Quantum Distributed Proofs on Paths

In this section we restrict ourselves to the case of a path $v_0, \ldots, v_r$ of length $r \geq 1$, in which only the two extremities v_0 and v_r are given inputs. This framework allows us to elaborate our main technique, that will be extended to arbitrary graphs in Section 5. Let $x \in \{0,1\}^n$ be the input to v_0, and $y \in \{0,1\}^n$ be the input to v_r. Our goal is to design a dQMA protocol to decide whether $f(x,y) = 1$ or not, for some given Boolean function on $\{0,1\}^n \times \{0,1\}^n$.

We show the following general theorem that converts a one-way quantum communication complexity protocol into a quantum Merlin-Arthur protocol for the corresponding long-distance problem on the path. This theorem applies not only to one-sided-error protocols, but also to the two-sided-error case (with a logarithmic additional factor in the complexity).

▶ **Theorem 1.** *Let $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a Boolean function.*

- *If f has a quantum one-way one-sided-error communication protocol transmitting at most q qubits, then there exists a 1-sided distributed quantum Merlin-Arthur protocol for f on the path of length r, with soundness $1/3$, using certificates of size $O(r^2 q)$ qubits, and exchanging messages of length $O(r^2(q + \log r))$ qubits.*
- *If f has a quantum one-way two-sided-error communication protocol transmitting at most q qubits, then, for any constant c, there exists a distributed quantum Merlin-Arthur protocol for f on the path of length r with completeness $1 - 1/n^c$, soundness $1/3$, using certificates of size $O(r^2 q \log(n + r))$ qubits, and exchanging messages of length $O(r^2 q \log(n + r))$ qubits.*

Using known results (cf. Section 2) about one-way communication complexities of EQ_n and HAM_n^d, the following two results are direct applications of Theorem 1.

▶ **Corollary 2.** *There exists a one-sided quantum Merlin-Arthur protocol for EQ_n in the path of length r with soundness $1/3$, using certificates of size $O(r^2 \log n)$ qubits, and exchanging messages of length $O(r^2 \log(n + r))$ qubits.*[3]

▶ **Corollary 3.** *For any $c > 0$ and $d > 0$, there exists a quantum Merlin-Arthur protocol for HAM_n^d in the path of length r with completeness $1 - 1/n^c$, soundness $1/3$, using certificates of length $O(r^2 (\log n) \log(n+r))$ qubits, and exchanging messages of length $O(r^2 (\log n) \log(n+r))$ qubits.*

The rest of this section is dedicated to proving Theorem 1. Let us first give an overview of the proof. In our dQMA protocol in the path, the verification algorithm performed by the nodes on the line is merely a simulation of a two-party one-way quantum communication complexity protocol π between Alice and Bob for the function $f(x,y)$, with the help of certificates provided by the prover. Specifically, every intermediate node $v_1, \ldots, v_{r-1}$ expects to receive the quantum state sent by Alice to Bob in π, as certificate. Let us denote by $|h_x\rangle$ this state, which depends on x. The right-end node v_r simulates the two-party protocol π using $|h_x\rangle$ received from the left neighbor v_{r-1}, and applying Bob's measurement (i.e., the POVM measurement). If $f(x,y) = 1$, the prover honestly sends the desired state, and v_r accepts as it does receive $|h_x\rangle$. However, if $f(x,y) = 0$, then the malicious prover does not necessarily send a desired state. To catch the potentially malicious behavior of the prover on "illegal" instances (i.e., those for which $f(x,y) = 0$), each intermediate node checks whether its local proof is "close to" the one of its right neighbor. This is performed by an application of the SWAP test.

[3] Here we are using the fact that $\log n + \log r$ is of the same order as $\log(n + r)$ for conciseness.

Section 4.1 below describes in more detail how to construct the distributed quantum Merlin-Arthur protocol, denoted $\mathcal{P}_\pi$, from an arbitrary one-way quantum communication protocol π for the function f. Section 4.2 analyzes the completeness and the soundness of the protocol $\mathcal{P}_\pi$. Finally, Section 4.3 shows how to reduce the soundness error using "parallel repetitions" and how to apply this analysis to prove Theorem 1.

4.1 A dQMA Protocol for the Path

Let $\varepsilon \geq 0$ be a constant, which will be fixed small enough later in the proof. Let π be a quantum one-way communication protocol for f transmitting at most q qubits, such that, for every input pair (x, y), if $f(x, y) = 1$ then π outputs 1 with probability at least $1 - \varepsilon$, and if $f(x, y) = 0$ then π outputs 0 with probability at least 2/3. Let $|h_x\rangle$ be the q-qubit (pure) state sent from Alice to Bob, and let $\{M_{y,1}, M_{y,0}\}$ be the POVM measurement performed by Bob on $|h_x\rangle$, where $M_{y,1}$ corresponds to the measurement result 1 (accept) and $M_{y,0}$ to the measurement result 0 (reject). Our quantum Merlin-Arthur protocol $\mathcal{P}_\pi$ is as follows.

Protocol $\mathcal{P}_\pi$ for function f on input pair (x, y) in path $v_0, \ldots, v_r$:

1. If $f(x, y) = 1$ then the prover sends the quantum register R_j that has the state $|h_x\rangle$ (or $|h_x\rangle\langle h_x|$ as the mixed state representation) as certificate to each of the intermediate nodes v_j, $j \in \{1, \ldots, r-1\}$.
2. The left-end node v_0 prepares the state $\rho_0 = |h_x\rangle\langle h_x|$ in quantum register R_0.
3. For every $j = 0, \ldots, r-1$, the node v_j chooses a bit b_j uniformly at random, and sends its quantum register R_j to the right neighbor v_{j+1} whenever $b_j = 0$.
4. For every $j = 1, \ldots, r-1$, if v_j receives a quantum register from its left neighbor v_{j-1}, and if $b_j = 1$, then v_j performs the SWAP test on the registers (R_{j-1}, R_j), and accepts or rejects accordingly; Otherwise, v_j accepts.
5. If the right-end node v_r receives a quantum register R_{r-1} from its left neighbor, then v_r performs the POVM measurement $\{M_{y,1}, M_{y,0}\}$ corresponding to π applied to the state in R_{r-1}, and accepts or rejects accordingly; Otherwise, v_r accepts.

In the above protocol $\mathcal{P}_\pi$, the size of the quantum certificate that each node receives from the prover is at most q, and the length of the quantum message that each node sends to the neighbor is also at most q. In the next subsection, we prove that the above protocol has completeness $1 - \varepsilon/2$ and soundness $1 - 1/42r^2$.

4.2 Analysis of Protocol $\mathcal{P}_\pi$

For the analysis, we recall the SWAP test. The test is a protocol with a given input state on $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$, where $\mathcal{H}_1$ and $\mathcal{H}_2$ are complex Euclidian spaces. Here, we consider $\mathcal{H}_1$ and $\mathcal{H}_2$ as quantum registers R_1 and R_2.

SWAP test on a pure state $|\psi\rangle$ on $\mathcal{H}$, which is given in registers (R_1, R_2).

1. Prepare the single-qubit state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ in register R_0.
2. If the content of R_0 is 1, then apply the swap operator S on the state $|\psi\rangle$ in registers (R_1, R_2), where S is defined by $S(|j_1\rangle|j_2\rangle) = |j_2\rangle|j_1\rangle$ (namely, S swaps register R_1 and register R_2).
3. Apply the Hadamard operator $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ on the state in register R_0, and measure the content in the standard basis. Accept if the content is 0, and reject otherwise.

Completeness. The following lemma is a direct consequence of the definition of the SWAP test. Here, $\mathcal{H}_S$ is the symmetric subspace in $\mathcal{H}$ (namely, the subspace spanned by the states invariant by the swap operator S, or equivalently, the eigenstates of S with eigenvalue 1), and $\mathcal{H}_A$ is the anti-symmetric subspace in $\mathcal{H}$ (namely, the subspace spanned by the eigenstates of S with eigenvalue -1). Note that any state in $\mathcal{H}$ is represented as the superposition of a state in $\mathcal{H}_S$ (symmetric state) and a state in $\mathcal{H}_A$ (anti-symmetric state) as the swap operator S is a Hermitian matrix that only has $+1$ and -1 eigenvalues.

▶ **Lemma 4.** *Assume that $|\psi\rangle = \alpha|\psi_S\rangle + \beta|\psi_A\rangle$ where $|\psi_S\rangle \in \mathcal{H}_S$ and $|\psi_A\rangle \in \mathcal{H}_A$. Then, the SWAP test on input $|\psi\rangle$ accepts with probability $|\alpha|^2$.*

For the completeness, assume $f(x,y) = 1$. The prover then sends $|h_x\rangle$ to all the intermediate nodes. Then, all the nodes except the right-end node have $|h_x\rangle$. By Lemma 4, all the SWAP tests done in Step 4 are accepted with probability 1 (note that $|h_x\rangle \otimes |h_x\rangle$ is a symmetric state). Furthermore, the right-end node accepts with probability at least $(1-\varepsilon)/2 + 1/2 = 1 - \varepsilon/2$ as v_r can receive $|h_x\rangle$ and simulate π with probability 1/2 and accepts otherwise in Step 5.

Soundness. The following lemma presents a crucial property of the SWAP test: its applicability for checking whether the two *reduced* states are close. It is a trace-distance version of a lemma by Rosgen [44, Lemma 5.1]. Here, a reduced state intuitively represents the local information on its own quantum system, by disregarding the outside systems. Note that the trace distance between two quantum states ρ and σ is characterized as $\mathsf{dist}(\rho,\sigma) = \max_M \mathrm{tr}(M(\rho-\sigma))$, where the maximization is taken over all positive semi-definite matrices M such that $M \leq I$.

▶ **Lemma 5.** *Let $z \geq 1$, and assume that the SWAP test on input ρ in the input register (R_1, R_2) accepts with probability $1 - 1/z$. Then, $\mathsf{dist}(\rho_1,\rho_2) \leq 2/\sqrt{z} + 1/z$, where ρ_j is the reduced state on R_j of ρ. Moreover, if the SWAP test on input ρ accepts with probability 1, then $\rho_1 = \rho_2$ (and hence $\mathsf{dist}(\rho_1,\rho_2) = 0$).*

For the soundness, let (x,y) be any pair such that $f(x,y) = 0$.

▶ **Lemma 6.** *For every $j \in \{1,\ldots,r\}$, let F_j be the event that v_j performs the local test (SWAP or POVM) in Protocol $\mathcal{P}_\pi$, and let E_j be the event that this local test rejects. Then we have $\sum_{j=1}^{r} \Pr[E_j|F_j] \geq \frac{1}{21r}$.*

Proof. Let $\alpha_j = \Pr[E_j|F_j]$. Then, for every $j \in \{1,\ldots,r\}$, $\Pr[\overline{E_j}|F_j] = 1 - \alpha_j$, where we note that the complementary event $\overline{E_j}$ for $j = 1,\ldots,r-1$ is the event where the SWAP test on the two q-qubit states ρ_{j-1} and ρ_j accepts, and $\overline{E_r}$ represents the event that the result of the POVM measurement is 1 (accept), which corresponds to the POVM element $M_{y,1}$. By Lemma 5, the trace distance dist between the reduced q-qubit states ρ_{j-1} on v_{j-1} and ρ_j on v_j is

$$\mathsf{dist}(\rho_{j-1},\rho_j) \leq \begin{cases} \frac{2}{\sqrt{1/\alpha_j}} + \frac{1}{1/\alpha_j} & \text{if } \alpha_j \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

and thus $\mathsf{dist}(\rho_{j-1},\rho_j) \leq 3\sqrt{\alpha_j}$. Then, by the triangle inequality,

$$\mathsf{dist}(\rho_0,\rho_{r-1}) \leq \sum_{j=1}^{r-1} \mathsf{dist}(\rho_{j-1},\rho_j) \leq 3\sum_{j=1}^{r-1} \sqrt{\alpha_j}.$$

For $\Pr[E_r|F_r]$, the soundness of π promises that the probability that the test $\{M_{y,1}, M_{y,0}\}$ rejects $\rho_0 = |h_x\rangle\langle h_x|$ is at least $2/3$, i.e., $\mathrm{tr}(M_{y,0}\rho_0) \geq 2/3$. Hence,

$$\alpha_r = \Pr[E_r|F_r] = \mathrm{tr}(M_{y,0}\rho_{r-1}) \geq \frac{2}{3} - \mathsf{dist}(\rho_0, \rho_{r-1}) \geq \frac{2}{3} - 3\sum_{j=1}^{r-1}\sqrt{\alpha_j},$$

where the first inequality comes from the characterization of dist on indistinguishability of two states, that is, $|\,\mathrm{tr}(M_{y,0}\rho_0) - \mathrm{tr}(M_{y,0}\rho_{r-1})| \leq \mathsf{dist}(\rho_0, \rho_{r-1})$. Thus, we have

$$3\sum_{j=1}^{r}\sqrt{\alpha_j} \geq \alpha_r + 3\sum_{j=1}^{r-1}\sqrt{\alpha_j} \geq \frac{2}{3}.$$

By the Cauchy-Schwarz inequality,

$$\sqrt{r}\sqrt{\sum_{j=1}^{r}\alpha_j} \geq \sum_{j=1}^{r}\sqrt{\alpha_j},$$

and thus we have

$$\sum_{j=1}^{r}\alpha_j \geq \left(\frac{2}{9\sqrt{r}}\right)^2 \geq \frac{1}{21r}.$$ ◀

In Steps 4 and 5, node v_j, $1 \leq j \leq r$, performs the local test (SWAP or POVM) with probability at least $1/4$ (more precisely, v_j, $1 \leq j \leq r-1$, performs the SWAP test with probability $1/4$ and v_r performs the POVM with probability $1/2$). It follows that, for every $j \in \{1, \ldots, r\}$, the event F_j occurs in at least $(1/4) \times 2^r$ outcomes of all the 2^r possible outcomes $b_0 \cdots b_{r-1}$ that can be obtained in Step 3. Here, we consider any fixed outcomes $b_0 \cdots b_{r-1}$ that induce k events $F_{j_1}, F_{j_2}, \ldots, F_{j_k}$ with $k \neq 0$ where we note that $0 \leq k \leq \lfloor r/2 \rfloor$ in general. The probability that some node rejects in Steps 4 or 5 *under this outcome* is

$$\Pr[\vee_{t=1}^{k}E_{j_t}|\wedge_{i'=1}^{k}F_{j_{i'}}] \geq \frac{1}{\lfloor r/2 \rfloor}\sum_{i=1}^{k}\Pr[E_{j_i}|\wedge_{i'=1}^{k}F_{j_{i'}}] = \frac{1}{\lfloor r/2 \rfloor}\sum_{i=1}^{k}\Pr[E_{j_i}|F_{j_i}],$$

where the inequality follows from $k\Pr[\vee_{t=1}^{k}E_{j_t}] = \sum_{i=1}^{k}\Pr[\vee_{t=1}^{k}E_{j_t}] \geq \sum_{i=1}^{k}\Pr[E_{j_i}]$ and $k \leq \lfloor r/2 \rfloor$, and the equality comes from the fact that each F_{j_i} and E_{j_i} are independent from all the other event $F_{j_{i'}}$ with $i' \neq i$ (note that $|j_{i'} - j_i| \geq 2$ since F_{j-1} and F_j never occur at the same time). As each outcome occurs with probability $1/2^r$, the probability that some node rejects in Steps 4 or 5 is at least

$$\frac{1}{2^r}\cdot[(1/4)\cdot 2^r]\cdot\frac{1}{\lfloor r/2 \rfloor}\sum_{j=1}^{r}\Pr[E_j|F_j] \;\geq\; \frac{1}{2r}\sum_{j=1}^{r}\Pr[E_j|F_j] \;\geq\; \frac{1}{2r}\cdot\frac{1}{21r} = \frac{1}{42r^2},$$

where the second last inequality comes from Lemma 6.

4.3 Proof of Theorem 1

So far, we have shown that the protocol $\mathcal{P}_\pi$ has a completeness parameter very close to 1, but high soundness error. To establish Theorem 1, we need to reduce the soundness error without degrading the completeness too much. This is achieved via a form of parallel repetition of $\mathcal{P}_\pi$, by taking the logical conjunction of the outputs obtained by repetitions. The protocol resulting from k repetitions of $\mathcal{P}_\pi$ is denoted by $\mathcal{P}_\pi[k]$, and works as follows.

Protocol $\mathcal{P}_\pi[k]$: Soundness reduction of Protocol $\mathcal{P}_\pi$

1. If $f(x,y)=1$ then the prover sends the k quantum registers $R_{j,i}$ $(i=1,\dots,k)$, each of which has a state $|h_x\rangle$ as certificate, to each of the intermediate nodes v_j, $j\in\{1,\dots,r-1\}$.
2. The left-end node v_0 prepares the k quantum registers $R_{0,i}$, each of which has $|h_x\rangle$.
3. For every $j=0,\dots,r-1$, the node v_j chooses a k-bit string $b_{j,1}\cdots b_{j,k}$ uniformly at random, and sends $R_{j,i}$, together with the index i, to its right neighbor v_{j+1} whenever $b_{j,i}=0$.
4. For every $j=1,\dots,r-1$ and for every $i=1,\dots,k$, if the node v_j receives an index i, and if $b_{j,i}=1$, then v_j performs the SWAP test on $(R_{j-1,i},R_{j,i})$. Node v_j rejects whenever at least one of the performed SWAP tests rejects, and it accepts otherwise.
5. If the right-end node v_r receives an index $i\in\{1,\dots,k\}$, then it performs the POVM measurement $\{M_{y,1},M_{y,0}\}$ corresponding to π applied to the state in $R_{r-1,i}$. Node v_r rejects if at least one of the performed POVM measurements rejects, and it accepts otherwise.

Protocol $\mathcal{P}_\pi[k]$ has completeness $(1-\varepsilon/2)^k$, that is, the completeness has not reduced much whenever ε is small. By a similar analysis of standard error reduction techniques for quantum Merlin-Arthur games as the analysis in [3, 29], one can show that $\mathcal{P}_\pi[k]$ has soundness $(1-1/42r^2)^k$. By choosing $k=84r^2$, the resulting protocol $\mathcal{P}_\pi[k]$ has completeness $1-42r^2\varepsilon$ and soundness error $(1/e)^2<1/3$, while the size of the certificates is $O(r^2q)$ qubits, and the length of the message exchanged between neighbors is $O(r^2(q+\log r))$ (where the additional term $\log r$ comes from the index to the right neighbor in Step 3 of $\mathcal{P}_\pi[k]$).

Theorem 1 can now be easily derived from the above analysis. For the first part of the theorem, where f is having a one-sided-error one-way protocol π, simply use the protocol $\mathcal{P}_\pi$ from Section 4.1 with $\varepsilon=0$. The result then follows from the analysis of Section 4.2 and from the above discussion about soundness reduction.

For the case of second part of the theorem, where f is having a two-sided-error one-way protocol, we repeat the protocol π for $O(\log(n+r))$ times and using *majority voting* to get a protocol that correctly computes the value of the function with probability at least $1-1/42n^cr^2$. The protocol π of Section 4.1 can thus be chosen with $\varepsilon=1/42n^cr^2$, with message size $O(q\log(n+r))$. The result then follows similarly. ◀

5 Certifying Equality in General Graphs

We now extend our protocol for checking equality between n-bit strings $x_1,\dots,x_t$ stored at $t\geq 2$ distinct nodes $u_1,\dots,u_t$ of a connected simple graph G. We first show how to reduce the problem to trees of a specific structure, and then present a protocol for trees.

5.1 Reduction to Trees

Let $G=(V,E)$ be a connected simple graph, and let $u_1,\dots,u_t$ be $t\geq 2$ distinct nodes of G. Assume, without loss of generality, that u_1 is the most central node among them, i.e., it satisfies $\max_{i=1,\dots,t}\mathsf{dist}_G(u_1,u_i)=\min_{j=1,\dots,t}\max_{i=1,\dots,t}\mathsf{dist}_G(u_j,u_i)$. Let $r=\max_{i=1,\dots,t}\mathsf{dist}_G(u_1,u_i)$ be the *radius* of the t terminals $u_1,\dots,u_t$. We construct a tree T rooted at u_1, that has all terminals as leaves, maximum degree t and depth at most $r+1$ (see Figure 1). To this end, start with a BFS tree T' in G, rooted at u_1. Truncate the tree at each terminal u_i that does not have any terminal as successors, thus limiting

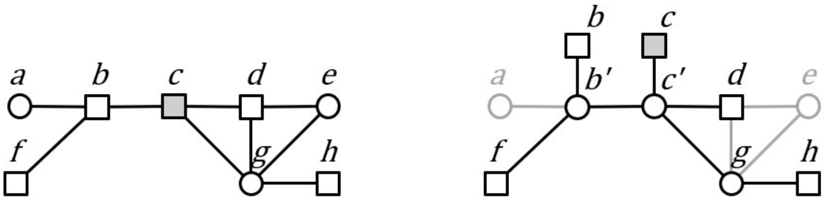

Figure 1 The construction of a spanning tree: a graph G (left) and its corresponding spanning tree T (right). Terminals are marked by squares, and c is the root. Node b (resp. c), which was a terminal, is replaced by a non-terminal node b' (resp. c'), while the other terminals are leaves in the tree so they remain unmodified.

the depth to r and the degree to t. For every terminal u_i that is not a leaf, including u_1, replace u_i with a node u'_i, and connect u_i to u'_i as a leaf, where the input x_i stays at u_i – this guarantees that all inputs are now on leaves, the same degree bound holds, and the depth is increased by at most 1.

While T is not a sub-tree of G, we can easily emulate an algorithm or a labeling scheme designed for T, in G (specifically, in T'). To this end, every internal terminal u_i in T' simulates the behavior of u_i itself, and also of u'_i. The following lemma is using classical assumptions of network computing (see, e.g., [42]) and can be proved using standard techniques (see, e.g., [35]). We refer to the tree T in the construction described above.

▶ **Lemma 7.** *For any graph $G = (V, E)$ with nodes IDs taken in a range polynomial in $|V|$, there is a deterministic distributed Merlin-Arthur protocol for the tree T using certificates on $O(\log |V|)$ bits.*

The term *deterministic* in the above lemma means that the verification process is deterministic, which implies perfect completeness and perfect soundness (i.e., soundness error 0). Roughly speaking, in this protocol each non-tree node will have a (non-quantum) label indicating its distance from the tree, and each tree node will have as label its depth in the tree, the ID of its parent, and the ID of the root.

5.2 Certifying Equality in Trees

Based on our tree construction from a graph and Lemma 7, we can restrict our attention to the case in which the t terminals $u_1, \ldots, u_t$, who hold the n-bit strings $x_1, \ldots, x_t$, belong to a tree T rooted at u_1, of depth equal to $r + 1$, where r is the radius of the terminals, with maximum degree t, and with leaves $u_2, \ldots, u_t$. Moreover, we assume that the root u_1 itself is of degree 1 due to our tree construction. We present a distributed quantum Merlin-Arthur protocol for the equality function EQ_n^t in this setting, and hence prove our main result.

▶ **Theorem 8.** *There is a distributed quantum Merlin-Arthur protocol on T for EQ_n^t between t terminals of radius r, with perfect completeness, soundness $1/3$, certificate size $O(t\,r^2 \log n)$ qubits, and message length $O(t\,r^2 \log(n + r))$ qubits.*

Proof. Let π be a one-way communication protocol for EQ_n transmitting $m = O(\log n)$ qubits such that, for every input pair (x, y), if $x = y$ then π outputs 1 with probability 1 and if $x \neq y$ then π outputs 0 with probability at least $2/3$ (such a protocol exists, as mentioned in Section 2). For input (x, y), let $|h_x\rangle$ be the quantum message from Alice to Bob in π, and let

$\{M_{y,1}, M_{y,0}\}$ be the POVM measurement done by Bob on $|h_x\rangle$ in π, where $M_{y,1}$ corresponds to the measurement result 1 (accept), and $M_{y,0}$ corresponds to the measurement result 0 (reject), respectively. Our quantum Merlin-Arthur protocol is as follows.

Protocol $\mathcal{P}(\mathsf{EQ}_n^t)$ for equality in trees

1. If $\mathsf{EQ}_n^t(x_1, \ldots, x_t) = 1$ then the prover sends an m-qubit state equal to $|h_{x_1}\rangle$ to each of the nodes v that have no input.
2. For every $i \in \{1, \ldots, t\}$, node u_i prepares the m-qubit state $|h_{x_i}\rangle$.
3. Every non-root node v of the tree chooses a bit b_v uniformly at random. If $b_v = 0$, then v sends its m-qubit state to its parent in T.
4. For every non-terminal node v, if v receives a state from the children, and if $b_v = 1$, then v performs the SWAP test on the $2m$-qubit state that consists of the m-qubit state received from the prover and an m-qubit state received from the children, which is chosen uniformly at random if he/she receives multiple m-qubit states from the children, and accepts or rejects accordingly. Otherwise, v accepts.
5. If the root node u_1 receives a state from its children, then u_1 performs the POVM measurement $\{M_{x_1,1}, M_{x_1,0}\}$ on an m-qubit state received from the children, which is chosen uniformly at random if he/she receives multiple m-qubit states from the children, and accepts or rejects accordingly. Otherwise, u_1 accepts.

The perfect completeness trivially holds since every local test yields acceptance with certainty. For the soundness, if $\mathsf{EQ}_n^t(x_1, \ldots, x_t) = 0$ then there is a leaf u_i, $i > 1$, with $x_i \neq x_1$. Then, we can perform almost the same analysis as in Section 4, but for the path connecting u_1 and u_i in T. The only difference is the probability that each local test occurs; it is at least $1/4$ in the analysis of $\mathcal{P}_\pi$ done in Section 4, while it is at least $(1/4) \cdot (1/t)$ in the protocol $\mathcal{P}(\mathsf{EQ}_n^t)$ we are now considering, as every non-terminal node v_j or u_1 on the path chooses the m-qubit state from the child on the path uniformly at random from the multiple m-qubit states (possibly) sent from all the children. Hence, $\mathcal{P}(\mathsf{EQ}_n^t)$ has soundness error $1 - O(1/tr^2)$. The proof is completed by performing $O(tr^2)$ parallel repetitions of $\mathcal{P}(\mathsf{EQ}_n^t)$ for error reduction, similarly to the $O(r^2)$ parallel repetitions of $\mathcal{P}_\pi$ in Section 4. ◀

▶ Remark. Using Lemma 7, we get that, up to adding $O(\log|V|)$ classical bits in the certificates of the nodes, Theorem 8 can be extended to the case where the terminals are in a connected graph $G = (V, E)$.

6 Classical Lower Bounds

In this section, we show that non-quantum distributed Merlin-Arthur (dMA) protocols for distant EQ require certificates of linear size. In fact, we establish a more general lower bound which applies to all functions f with large fooling set, even using shared randomness. In addition, the bound holds for settings which allow the graph nodes to have multiple communication rounds among them, after receiving the certificates and before deciding if they finally accept (see, e.g., [19, 41]).

For the lower bound, it is sufficient to consider the path $v_0, \ldots, v_r$ in which v_0 and v_r are provided with inputs x and y, respectively.

▶ **Theorem 9.** *Let $r \geq 2\mu + 1$, and let $f(x, y)$ be any Boolean function with a 1-fooling set of size at least k. Let $\mathcal{P}$ be a classical Merlin-Arthur protocol for f in a path of r edges, with μ rounds of communication among the nodes, shared randomness, certificates of size $\lfloor \frac{1}{2\mu} \log(k-1) \rfloor$ bits, and completeness $1 - p$. Then $\mathcal{P}$ has soundness error at least $1 - 2p$.*

Proof. Consider the path $v_0, \ldots, v_r$ with a fixed identifier assignment, and inputs x and y given to v_0 and v_r, respectively. Since f has large 1-fooling set but small certificates, there exist two distinct pairs of "fooling" inputs that have the same certificate assignments on the 2μ node $v_1, \ldots, v_{2\mu}$. That is, we can fix two input pairs (x, y) and (x', y'), with $f(x, y) = f(x', y') = 1$, and, w.l.o.g., $f(x, y') = 0$, with corresponding certificate assignments w and w', such that

$$w(v_i) = w'(v_i) \text{ for every } i \in \{1, \ldots, 2\mu\},$$

where $w(v_i)$ and $w'(v_i)$ are the certificate assigned to the node v_i in the assignments w and w', respectively.

We interpret the outputs as Boolean values, where accept = 1 and reject = 0, and denote by $\mathsf{out}_i(x, y, w)$ the output of v_i when the inputs are x and y and the certificate assignment is w. Since $\mathcal{P}$ has completeness $1 - p$, we have

$$\Pr_s \Big[\bigwedge_{i \le \mu} \mathsf{out}_i(x, y, w) = 1 \wedge \bigwedge_{i \ge \mu+1} \mathsf{out}_i(x, y, w) = 1 \Big] \ge 1 - p,$$

and the same holds for (x', y', w'). Hence,

$$\Pr_s \Big[\bigwedge_{i \le \mu} \mathsf{out}_i(x, y, w) = 1 \Big] \ge 1 - p, \text{ and } \Pr_s \Big[\bigwedge_{i \ge \mu+1} \mathsf{out}_i(x', y', w') = 1 \Big] \ge 1 - p.$$

The output out_i of every node v_i is a function of its own identifier and certificate, the certificates of the nodes in its distance-μ neighborhood, and the public random string s. In addition, the outputs of $v_0, v_1, \ldots, v_\mu$ may also depend on the input x to v_0, and the outputs of $v_{r-\mu}, \ldots, v_r$ may also depend on the input y to v_r. Formally speaking, the outputs can also depend on the identifiers of the neighbors, but these are fixed given the node's identifier, so we ignore them henceforth.

Let w'' be the certificate assignment defined by

$$\begin{array}{ll} w''(v_0) = w(v_0); & \\ w''(v_i) = w(v_i) = w'(v_i) & \text{for } i \in \{1, \ldots, 2\mu\}; \\ w''(v_i) = w'(v_i) & \text{for } i \in \{2\mu + 1, \ldots, r\}. \end{array}$$

Consider the input assignment (x, y') combined with the certificate assignment w''. The definition of w'' implies that nodes $v_0, \ldots, v_{2\mu}$ receive the same certificates as in w, and thus nodes $v_0, \ldots, v_\mu$ cannot distinguish this form the input assignment (x, y) with certificates assignment w. On the other hand, nodes $v_1, \ldots, v_r$ receive the same certificates as in w', so the nodes $v_{\mu+1}, \ldots, v_r$ cannot distinguish this form the input assignment (x', y') with certificates assignment w'.

A union bound finishes the proof:

$$\begin{aligned} &\Pr_s \Big[\bigwedge_{i \le \mu} \mathsf{out}_i(x, y', w'') = 1 \wedge \bigwedge_{i \ge \mu+1} \mathsf{out}_i(x, y', w'') = 1 \Big] \\ =&\Pr_s \Big[\bigwedge_{i \le \mu} \mathsf{out}_i(x, y, w) = 1 \wedge \bigwedge_{i \ge \mu+1} \mathsf{out}_i(x', y', w') = 1 \Big] \\ \ge& 1 - \Pr_s \Big[\neg \bigwedge_{i \le \mu} \mathsf{out}_i(x, y, w) = 1 \Big] - \Pr_s \Big[\neg \bigwedge_{i \ge \mu+1} \mathsf{out}_i(x', y', w') = 1 \Big] \\ \ge& \; 1 - 2p. \end{aligned}$$

That is, we found a certificate assignment w'' for the input (x, y') which makes all node accept with probability at least $1 - 2p$, even though $f(x, y') = 0$. Hence, the soundness error is at least $1 - 2p$, as claimed. ◀

Since EQ_n has a 1-fooling set of size 2^n, the corollary below follows directly from Theorem 9. The case $r = 3$ and $\mu = 1$ gives a linear lower bound for dMA protocols on a 4-node path.

▶ **Corollary 10.** *For every $r \geq 2\mu + 1$, every distributed (classical) Merlin-Arthur protocol for EQ_n^2 with μ rounds of communication among the nodes in the path of r edges with certificates of size at most $\lfloor (n-1)/2\mu \rfloor$ bits, and completeness $1 - p$ has soundness error at least $1 - 2p$.*

The requirements for a good protocol is to have a high completeness (i.e., small value for p, ideally $p = 0$) and a reasonably small soundness error. Corollary 10 precisely shows that for the equality function such protocols cannot exist in the classical setting unless the certificate size is linear in n.

▶ Remark. The completeness-soundness gap of Theorem 9 is optimal in general, in the sense that it cannot be improved for EQ_1^2, i.e., distant equality between two input bits. Consider the following protocol $\mathcal{P}$ for EQ_1^2, on input $(x, y) \in \{0, 1\} \times \{0, 1\}$. It uses a shared random variable $X \in \{-1, 0, 1\}$ with $\Pr[X = 0] = \Pr[X = 1] = p$, and $\Pr[X = -1] = 1 - 2p$. In the case $X = -1$, all the nodes accept. In the case $X \in \{0, 1\}$, v_0 accepts whenever $X = x$, v_r accepts whenever $X = y$, and all the other nodes accept. If $x = y$, the probability that all the nodes accept is $1 - 2p + (1/2) \cdot (2p) = 1 - p$. If $x \neq y$, then either v_0 or v_r systematically rejects for $X \neq -1$, and thus the probability that all the nodes accept is $1 - 2p$.

7 Conclusion

In this paper, we extended the notion of randomized proof-labeling scheme to the quantum setting. We showed the efficiency of distributed quantum certification mechanisms by designing a distributed quantum Merlin-Arthur protocol for EQ_n^t between t parties spread out in a graph, using certificates and messages whose size depend logarithmically on n, the size of the data. This is in contrast to classical distributed Merlin-Arthur protocols, which require certificates of size linear in n, even when messages of unbounded size can be used. Our result was obtained by using an interesting property of the SWAP test: it can be applied for checking proximity properties between reduced states.

Which other Boolean predicates on labeled graphs, beyond equality, could take benefit from quantum resources for the design of compact distributed certification schemes is an intriguing question. Theorem 1 gives a partial answer on the path. A complete answer to this question would significantly help improving our comprehension of the power of quantum computing in the distributed setting.

References

1 Scott Aaronson, Salman Beigi, Andrew Drucker, Bill Fefferman, and Peter W. Shor. The power of unentanglement. *Theory of Computing*, 5:1:1–1:42, 2009. doi:10.4086/toc.2009.v005a001.

2 Yehuda Afek, Shay Kutten, and Moti Yung. The local detection paradigm and its application to self-stabilization. *Theoretical Computer Science*, 186(1-2):199–229, 1997. doi:10.1016/S0304-3975(96)00286-1.

3 Dorit Aharonov and Tomer Naveh. Quantum NP – a survey. arXiv:quant-ph/0210077v1, 2002. arXiv:quant-ph/0210077.

4 Noga Alon, Klim Efremenko, and Benny Sudakov. Testing equality in communication graphs. *IEEE Transactions on Information Theory*, 63(11):7569–7574, 2017. doi:10.1109/TIT.2017.2744608.

5 Heger Arfaoui and Pierre Fraigniaud. What can be computed without communications? *SIGACT News*, 45(3):82–104, 2014. doi:10.1145/2670418.2670440.

6 Baruch Awerbuch, Boaz Patt-Shamir, and George Varghese. Self-stabilization by local checking and correction (extended abstract). In *32nd Symposium on Foundations of Computer Science (FOCS)*, pages 268–277, 1991. doi:10.1109/SFCS.1991.185378.

7 Alkida Balliu, Gianlorenzo D'Angelo, Pierre Fraigniaud, and Dennis Olivetti. What can be verified locally? *Journal of Computer System and Sciences*, 97:106–120, 2018. doi:10.1016/j.jcss.2018.05.004.

8 Michael Ben-Or and Avinatan Hassidim. Fast quantum byzantine agreement. In *37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 481–485, 2005. doi:10.1145/1060590.1060662.

9 Hugue Blier and Alain Tapp. A quantum characterization of NP. *Computational Complexity*, 21(3):499–510, 2012. doi:10.1007/s00037-011-0016-2.

10 Ralph Bottesch, Dmitry Gavinsky, and Hartmut Klauck. Equality, revisited. In *40th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 9235 of *LNCS*, pages 127–138. Springer, 2015. doi:10.1007/978-3-662-48054-0_11.

11 Anne Broadbent and Alain Tapp. Can quantum mechanics help distributed computing? *SIGACT News*, 39(3):67–76, 2008. doi:10.1145/1412700.1412717.

12 Harry Buhrman, Richard Cleve, John Watrous, and Ronald de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902:1–167902:4, 2001. doi:10.1103/PhysRevLett.87.167902.

13 Arkadev Chattopadhyay, Jaikumar Radhakrishnan, and Atri Rudra. Topology matters in communication. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 631–640, 2014. doi:10.1109/FOCS.2014.73.

14 Arkadev Chattopadhyay and Atri Rudra. The range of topological effects on communication. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP*, pages 540–551, 2015. doi:10.1007/978-3-662-47666-6_43.

15 Pierluigi Crescenzi, Pierre Fraigniaud, and Ami Paz. Trade-offs in distributed interactive proofs. In *33rd International Symposium on Distributed Computing (DISC)*, pages 13:1–13:17, 2019. doi:10.4230/LIPIcs.DISC.2019.13.

16 Vasil S. Denchev and Gopal Pandurangan. Distributed quantum computing: a new frontier in distributed systems or science fiction? *SIGACT News*, 39(3):77–95, 2008. doi:10.1145/1412700.1412718.

17 Michael Elkin, Hartmut Klauck, Danupon Nanongkai, and Gopal Pandurangan. Can quantum communication speed up distributed computation? In *33rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 166–175, 2014. doi:10.1145/2611462.2611488.

18 Laurent Feuilloley, Pierre Fraigniaud, and Juho Hirvonen. A hierarchy of local decision. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 118:1–118:15, 2016. doi:10.4230/LIPIcs.ICALP.2016.118.

19 Laurent Feuilloley, Pierre Fraigniaud, Juho Hirvonen, Ami Paz, and Mor Perry. Redundancy in distributed proofs. In *32nd International Symposium on Distributed Computing, DISC*, pages 24:1–24:18, 2018. doi:10.4230/LIPIcs.DISC.2018.24.

20 Pierre Fraigniaud, Amos Korman, and David Peleg. Towards a complexity theory for local distributed computing. *Journal of the ACM*, 60(5):35:1–35:26, 2013. doi:10.1145/2499228.

21 Pierre Fraigniaud, Pedro Montealegre, Rotem Oshman, Ivan Rapaport, and Ioan Todinca. On distributed Merlin-Arthur decision protocols. In *26th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, volume 11639 of *LNCS*, pages 230–245. Springer, 2019. doi:10.1007/978-3-030-24922-9_16.

22 Pierre Fraigniaud, Boaz Patt-Shamir, and Mor Perry. Randomized proof-labeling schemes. *Distributed Computing*, 32(3):217–234, 2019. doi:10.1007/s00446-018-0340-8.

23 Cyril Gavoille, Adrian Kosowski, and Marcin Markiewicz. What can be observed locally? In *23rd International Symposium on Distributed Computing (DISC)*, volume 5805 of *LNCS*, pages 243–257. Springer, 2009. doi:10.1007/978-3-642-04355-0_26.

24 Mika Göös and Jukka Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12:19:1–19:33, 2016. doi:10.4086/toc.2016.v012a019.

25 Aram W. Harrow and Ashley Montanaro. Testing product states, quantum Merlin-Arthur games and tensor optimization. *Journal of the ACM*, 60(1):3:1–3:43, 2013. doi:10.1145/2432622.2432625.

26 Gene Itkis and Leonid A. Levin. Fast and lean self-stabilizing asynchronous protocols. In *35th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 226–239, 1994. doi:10.1109/SFCS.1994.365691.

27 Taisuke Izumi and François Le Gall. Quantum distributed algorithm for the All-Pairs Shortest Path problem in the CONGEST-CLIQUE model. In *38th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 84–93, 2019. doi:10.1145/3293611.3331628.

28 Taisuke Izumi, François Le Gall, and Frédéric Magniez. Quantum distributed algorithm for triangle finding in the CONGEST model. In *37th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 23:1–23:13, 2020. doi:10.4230/LIPIcs.STACS.2020.23.

29 Alexei Yu. Kitaev, Alexander H. Shen, and Mikhail N. Vyalyi. *Classical and Quantum Computation.* Graduate Studies in Mathematics 47. American Mathematical Society, 2002.

30 Hartmut Klauck. On Arthur Merlin games in communication complexity. In *26th Annual IEEE Conference on Computational Complexity (CCC)*, pages 189–199, 2011. doi:10.1109/CCC.2011.33.

31 Hartmut Klauck and Supartha Podder. Two results about quantum messages. In *39th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 8635 of *LNCS*, pages 445–456. Springer, 2014. doi:10.1007/978-3-662-44465-8_38.

32 Hirotada Kobayashi, François Le Gall, and Harumichi Nishimura. Stronger methods of making quantum interactive proofs perfectly complete. *SIAM Journal on Computing*, 44(2):243–289, 2015. doi:10.1137/140971944.

33 Hirotada Kobayashi, Keiji Matsumoto, and Tomoyuki Yamakami. Quantum Merlin-Arthur proof systems: Are multiple Merlins more helpful to Arthur? *Chicago Journal of Theoretical Computer Science*, 2009:3:1–3:19, 2009. doi:10.4086/cjtcs.2009.003.

34 Gillat Kol, Rotem Oshman, and Raghuvansh R. Saxena. Interactive distributed proofs. In *37th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 255–264, 2018. doi:10.1145/3212734.3212771.

35 Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. *Distributed Computing*, 22(4):215–233, 2010. doi:10.1007/s00446-010-0095-3.

36 Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, 1997.

37 François Le Gall and Frédéric Magniez. Sublinear-time quantum computation of the diameter in CONGEST networks. In *37th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 337–346, 2018. doi:10.1145/3212734.3212744.

38 François Le Gall, Harumichi Nishimura, and Ansis Rosmanis. Quantum advantage for the LOCAL model in distributed computing. In *36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 49:1–49:14, 2019. doi:10.4230/LIPIcs.STACS.2019.49.

39 Moni Naor, Merav Parter, and Eylon Yogev. The power of distributed verifiers in interactive proofs. In *31st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1096–115, 2020. doi:10.1137/1.9781611975994.67.

40 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

41 Rafail Ostrovsky, Mor Perry, and Will Rosenbaum. Space-time tradeoffs for distributed verification. In *Structural Information and Communication Complexity - 24th International Colloquium, SIROCCO*, pages 53–70, 2017. doi:10.1007/978-3-319-72050-0_4.

42 David Peleg. *Distributed Computing: A Locality-Sensitive Approach.* SIAM, 2000.

43 Ran Raz and Amir Shpilka. On the power of quantum proofs. In *19th IEEE Conference on Computational Complexity (CCC)*, pages 260–274, 2004. doi:10.1109/CCC.2004.1313849.

44 Bill Rosgen. Distinguishing short quantum computations. In *25th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 597–608, 2008. doi:10.4230/LIPIcs.STACS.2008.1322.

45 Seiichiro Tani, Hirotada Kobayashi, and Keiji Matsumoto. Exact quantum algorithms for the leader election problem. *ACM Transactions on Computation Theory*, 4(1):1:1–1:24, 2012. doi:10.1145/2141938.2141939.

46 Andrew Chi-Chih Yao. Quantum circuit complexity. In *34th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 352–361, 1993. doi:10.1109/SFCS.1993.366852.

47 Andrew Chi-Chih Yao. On the power of quantum fingerprinting. In *35th ACM Symposium on Theory of Computing (STOC)*, pages 77–81, 2003. doi:10.1145/780542.780554.