Vol.2020-QS-1 No.7
2020/10/16

Distributed Quantum Proofs for Replicated Data

Pierre Fraigniaud[1] François Le Gall[2] Harumichi Nishimura[3,a)] Ami Paz[4]

概要：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.

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 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* [25], and it extends the seminal concept of *proof-labeling schemes* [36]. 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

[1] IRIF, CNRS and Université de Paris, France
[2] Graduate School of Mathematics, Nagoya University, Japan
[3] Graduate School of Informatics, Nagoya University, Japan
[4] Faculty of Computer Science, Universität Wien, Austria
[a)] hnishimura@i.nagoya-u.ac.jp

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], [27] 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* [23]. 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], [22], [40]. 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], [24], [28], [29], [38], [39]), 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 $x_1, \ldots, x_t$ are stored at t terminal nodes $u_1, \ldots, u_t$ in a network $G = (V, E)$, where node u_i stores x_i. We denote EQ_n^t the problem of checking the equality $x_1 = \ldots = x_t$ 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 4):

Main Results. 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., [9], [12], [48]), checking whether a given state (or a reduced state of it) is pure or entangled with the environment system (e.g., [1], [26], [33], [34]), 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*.

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 5). 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], [27]). Proof-labeling schemes were introduced in [36], and variants have been studied in [21], [25]. Randomized proof-labeling schemes have been studied in [23]. 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* [35]. 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. We refer to [15], [22], [40] 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], [24], [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 [29], [38], the CONGEST-CLIQUE model [28] and the LOCAL model [39]. 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 [10], [31], [32], [44]. 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 [32], [44].

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 $f(x_1,\ldots,x_t) = 1$ or not, where $x_1,\ldots,x_t$ are assigned to t nodes of a graph. Specifically, an instance of the problem f is a t-tuple $(x_1,\ldots,x_t) \in \{0,1\}^n \times \ldots \times \{0,1\}^n$, a connected graph $G = (V,E)$, and an ordered sequence $v_1,\ldots,v_t$ of distinct nodes of G. The node v_i is given x_i as input, for $i = 1,\ldots,t$. All the other nodes receive no inputs. We consider distributed Merlin-Arthur (dMA) protocols for deciding whether $f(x_1,\ldots,x_t) = 1$, 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 $(x_1,\ldots,x_t) \in \{0,1\}^n \times \ldots \times \{0,1\}^n$, every connected graph G, and every ordered sequence $v_1,\ldots,v_t$ of distinct nodes in G:

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

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

The completeness condition guarantees that, when the system is in a "legal" state (specified by $f(x_1,\ldots,x_t) = 1$), with probability at least a all nodes accept. The soundness condition guarantees that, when the system is in an "illegal" state (specified by $f(x_1,\ldots,x_t) = 0$), 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 $\mathsf{EQ}_n^t(x_1,\ldots,x_t) = 1 \iff x_1 = \cdots = x_t$.

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 $v_0,\ldots,v_r$, $r \geq 1$, 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 $v_1,\ldots,v_{r-1}$ 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

[*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 5, where we show classical lower bounds that hold even for $\mu > 1$.

setting is very much related to communication complexity.

Classical two-party communication complexity.

We refer to [37] 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., [37].

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. [41], for instance).

Quantum two-party communication complexity, first introduced by Yao [47], 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. Quantum Distributed Proofs on Paths

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 the known result (cf. Section 2) about one-way communication complexity of EQ_n, the following result is a direct application of Theorem 1.

Corollary 1. There exists a one-sided quantum Merlin-

[*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* [41] whose length is at most twice the one of the mixed state.

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]

4. Certifying Equality in General Graphs

We now extend our protocol for checking equality between n-bit strings $x_1, \ldots, x_t$ stored at $t \geq 2$ distinct nodes $u_1, \ldots, u_t$ 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.

4.1 Reduction to Trees

Let $G = (V, E)$ be a connected simple graph, and let $u_1, \ldots, u_t$ 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 $\max_{i=1,\ldots,t} \mathsf{dist}_G(u_1, u_i) = \min_{j=1,\ldots,t} \max_{i=1,\ldots,t} \mathsf{dist}_G(u_j, u_i)$. Let $r = \max_{i=1,\ldots,t} \mathsf{dist}_G(u_1, u_i)$ be the *radius* of the t terminals $u_1, \ldots, u_t$. We construct a tree T rooted at u_1, that has all terminals as leaves, maximum degree t and depth at most $r+1$. 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 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., [43]) and can be proved using standard techniques (see, e.g., [36]). We refer to the tree T in the construction described above.

Lemma 1. 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.

4.2 Certifying Equality in Trees

Based on our tree construction from a graph and Lemma 1, 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 2. 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.

5. 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], [42]).

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 3. 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 $1 - 2p$.

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

Reference to the full version.

See Ref. [20] for the full version of this unrefereed manuscript.

参考文献

[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.

[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.

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

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

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

[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.

[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.

[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.

[9] Hugue Blier and Alain Tapp. A quantum characterization of NP. *Computational Complexity*, 21(3):499–510, 2012.

[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.

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

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

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[20] Pierre Fraigniaud, François Le Gall, Harumichi Nishimura, and Ami Paz. Distributed quantum proofs for replicated data. arXiv:2002.10018, 2020.

[21] 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.

[22] 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.

[23] Pierre Fraigniaud, Boaz Patt-Shamir, and Mor Perry. Randomized proof-labeling schemes. *Distributed Computing*, 32(3):217–234, 2019.

[24] 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.

[25] Mika Göös and Jukka Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12:19:1–19:33, 2016.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

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

[31] Hartmut Klauck. On Arthur Merlin games in communication complexity. In *26th Annual IEEE Conference on Computational Complexity (CCC)*, pages 189–199, 2011.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] Amos Korman, Shay Kutten, and David Peleg. Proof labeling schemes. *Distributed Computing*, 22(4):215–233, 2010.

[37] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[38] 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.

[39] 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.

[40] 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.

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

[42] 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.

[43] David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[44] Ran Raz and Amir Shpilka. On the power of quantum proofs. In *19th IEEE Conference on Computational Complexity (CCC)*, pages 260–274, 2004.

[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.

[46] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

[47] Andrew Chi-Chih Yao. Quantum circuit complexity. In *34th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 352–361, 1993.

[48] Andrew Chi-Chih Yao. On the power of quantum fingerprinting. In *35th ACM Symposium on Theory of Computing (STOC)*, pages 77–81, 2003.