Asynchronous reference frame agreement in a quantum network

Islam, Tanvirul; Wehner, Stephanie

DOI
10.1088/1367-2630/18/3/033018

Publication date
2016

Document Version
Final published version

Published in
New Journal of Physics

PAPER • OPEN ACCESS

Asynchronous reference frame agreement in a quantum network

To cite this article: Tanvirul Islam and Stephanie Wehner 2016 *New J. Phys.* **18** 033018

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
22 May 2015

REVISED
25 January 2016

ACCEPTED FOR PUBLICATION
15 February 2016

PUBLISHED
10 March 2016

PAPER

Asynchronous reference frame agreement in a quantum network

Tanvirul Islam[1,2,3,4] **and Stephanie Wehner**[2,3]

[1] School of Computing, National University of Singapore, 13 Computing Drive, 117417 Singapore
[2] Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, 117543 Singapore
[3] Qutech, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft
[4] Author to whom any correspondence should be addressed.

E-mail: tanvir@locc.la **and** steph@locc.la

Keywords: quantum networks, asynchronous protocol, reference frame agreement

Abstract

An efficient implementation of many multiparty protocols for quantum networks requires that all the nodes in the network share a common reference frame. Establishing such a reference frame from scratch is especially challenging in an asynchronous network where network links might have arbitrary delays and the nodes do not share synchronised clocks. In this work, we study the problem of establishing a common reference frame in an asynchronous network of n nodes of which at most t are affected by arbitrary unknown error, and the identities of the faulty nodes are not known. We present a protocol that allows all the correctly functioning nodes to agree on a common reference frame as long as the network graph is complete and not more than $t < n/4$ nodes are faulty. As the protocol is asynchronous, it can be used with some assumptions to synchronise clocks over a network. Also, the protocol has the appealing property that it allows any existing two-node asynchronous protocol for reference frame agreement to be lifted to a robust protocol for an asynchronous quantum network.

1. Introduction

To use quantum cryptography on a global scale one must first have a functioning quantum internet [1]. Recently this necessity has inspired a lot of effort in the research and development of satellite [2–6], and ground based [7–9] quantum networks. The possible applications of such networks are not restricted to only cryptography. A fully general quantum network will allow us to perform general distributed quantum computing [10–12].

In this work, we study problems related to initialisation and construction of quantum networks. More specifically, we study how well n nodes in an asynchronous quantum network can agree on a reference frame in the presence of at most t arbitrarily faulty nodes among them. By asynchronous network we mean in this setting we do not require the nodes to share a clock to begin with, and the channel delays might vary arbitrarily in each use. In fact, an asynchronous protocol only assumes any message sent from a correct node to a correct node will eventually reach the destination, without imposing any bound on the channel delay. This assumption captures the most general reference frame agreement problem in a quantum network because during the initialisation of the network the pairwise channel delays might be unknown, clocks might not be synchronised and spatial reference frames might be unaligned.

In a quantum channel, the qubits are encoded in some physical degree of freedom. For example, the polarisation direction of a photon is often used to encode qubits. This requires the sender and receiver to agree on some set of orthonormal directions as their common spatial reference frame. Another example is the time-bin qubits, where both of the parties require synchronised clocks. That is, they must have a pre-agreed temporal reference frame.

So far these reference frame agreement problems are studied in a bipartite setting [13–19] with the exception of [20], where spatial direction are agreed on in a *synchronised* network of n nodes. More specifically in [20] it is assumed that the network is synchronous. That is, all the nodes of the network have a shared clock and all the link delays have known upper bound. The bipartite reference frame agreement problem have been studied

extensively (see [21] for a review). However, agreeing on a reference frame in an asynchronous network of n nodes remained open.

There are protocols that allow Bell inequality tests and quantum information exchange between nodes without a pre-shared reference frame (see, for example [22–24]). However, the ability to reliably share reference frames among multiple nodes gives significant technological advantages by simplifying the implementation of most protocols. Moreover, reference frame agreement protocols have important implications in fields that are not directly related to quantum information.

One advantage of having an asynchronous reference frame agreement protocol for a network with a certain number of faulty nodes is that once a spatial reference frame is established, then new robust protocols can potentially be built on top of it to perform network-wide clock synchronisation. This is a task important by itself with various applications in security, navigation and finance [25]. The primary difficulty of executing any protocol in an asynchronous network comes from the fact that in the presence of incorrect, that is, arbitrarily faulty nodes it is impossible to decide for a correct receiver whether a message is not arriving because the sender is faulty and not sending anything at all, or the sender is correct but the channel is taking a very long time to transfer the message. Therefore, it is nontrivial to decide how long to wait for a message before moving on to the next step of a protocol.

Another difficulty is that unlike in classical information theory where information can be represented in bits, a reference frame can only be transferred from scratch by exchanging systems which have an inherent sense of direction [26]. Examples of such systems are spin qubits and photon polarisation qubits. The receiver can extract direction information from these systems, for example, by performing tomography on them. While preparing the direction any node P_i will know the description of the direction as a vector v_i in its local frame. Once the quantum system carrying that direction arrives at a receiver P_j, the receiver constructs a representation of the direction in it's own local frame as v_j. Such an estimation procedure inevitably introduces some error even in correct transmissions. That is, depending on the precision of the instruments one can only expect to have $d(v_i, v_j) \leqslant \delta$ for some $\delta > 0$, where $d(v_i, v_j)$ is the Euclidian distance between v_i and v_j. However, this distance metric does not make sense as it is, because v_i and v_j are vector representations in two different local frames. So we must redefine our distance metric $d(.,.)$ where distance is computed by converting both vectors in the frame of the first argument. As a result $d(v_i, v_j)$ remains a valid distance measure even though P_i and P_j do not know each other's local frame. This computation of distance between two vectors of different reference frames is only done in the analysis of the protocol and not by the nodes while playing the protocol. Any distance computed by a node inside a protocol is only between vectors for which it has a representation in its local frame. This inherent imperfection of message transmission must be accounted for by any reference frame agreement protocol. We capture this in the definition as,

Definition 1. For $\eta > 0$, a protocol in an asynchronous network of n nodes is an *η-asynchronous reference frame agreement protocol* if it satisfies the following conditions.

Termination. Every correct node P_i eventually terminates and outputs a direction v_i.

Correctness. If correct node P_i outputs v_i and correct node P_j outputs v_j then $d(v_i, v_j) \leqslant \eta$.

However, we have to achieve these termination and correctness condition in the presence of incorrect or faulty nodes. As it is unknown which nodes are faulty this resembles the Byzantine fault tolerance model [27] studied in classical distributed computing. For quantum networks our assumptions are,

1. The pairwise channels are *public*. That is, the messages are not secret. As a result, an adversary can see the content of a message between two correct nodes and adapt its strategy accordingly.
2. The pairwise channels are authenticated. That is, if a correct node sends a message to another correct node the message cannot be altered by any adversary. However, there might be channel noises, which can be dealt with, as in [20].
3. The pairwise channel delays might be controlled by the faulty nodes. That is, the faulty nodes can control the channel delays, even the delays for message passing between any pair of correct nodes.
4. If a correct node sends a message to another correct node, then the message eventually reaches the receiver. That is, even though the delay is controlled by some adversaries they cannot put infinite delay on the message between two correct nodes. However, the delay can be arbitrarily large.
5. The faulty nodes might have correlated error. To create a protocol which tolerates the worst kind of faults, we also assume that the faulty nodes can cooperate with each other and have a global strategy to thwart the

protocol. This is a realistic assumption because some nodes in a region might show correlated error which affects a part of the network.

Under all these assumptions we give an η-*asynchronous reference frame agreement protocol* for a network of n nodes that can tolerate up to $t < n/4$ faulty nodes. We review some preliminaries before presenting the main results.

2. Preliminaries

The problem of reference frame agreement over an asynchronous quantum network is necessarily multidisciplinary in nature. That is, it combines various concepts from quantum physics, information theory, cryptography and distributed computing. In this section we introduce several concepts from these fields that will be useful throughout this work.

2.1. Reference frame

2.1.1. Spatial reference frame

A *spatial reference frame* defines a co-ordinate system in space. For example in a Cartesian coordinate system, once the Cartesian frame $(\vec{x}, \vec{y}, \vec{z})$ is specified any vector $v = \alpha\vec{x} + \beta\vec{y} + \gamma\vec{z}$ can be represented as (α, β, γ) where α, β and γ are scalers. For two distant parties, who only have the knowledge of their own local frame, it becomes necessary to establish a shared reference frame before they can successfully communicate spatial information (such as, location and orientation).

We use quantum communications to send a direction between a sender and a receiver. Any protocol that allows transmission of direction between two nodes with δ accuracy is called a 2-party δ-estimate direction protocol. As an example we refer to the protocol 1, 2ED, one of the simplest possible protocols as studied in [13]. Here a sender creates many identical qubits with their Bloch vector pointing to the intended direction and the receiver measures them with Pauli measurements. From the statistics of the measurement outcomes, the receiver then estimates the Bloch vector's direction within Euclidian distance δ with probability of success $q_{\text{succ}} \geqslant 1 - e^{-\Omega(\delta^2 m)}$ where m is the number of qubits exchanged. That is, the Protocol 2ED allows the sender to transmit a direction u which is received as the direction v at the receiver, such that the inequality $d(u, v) \leqslant \delta$ holds with probability $q_{\text{succ}} \geqslant 1 - e^{-\Omega(\delta^2 m)}$. We emphasise that, this work allows us to lift any two party δ-estimate direction protocol into a protocol for a quantum network of n nodes.

Protocol 1: 2ED

input : Sender, direction u
output : Receiver, direction v

Sender: 2ED-Send
- Prepare $3n$ qubits with direction u
- Send them to the receiver

Receiver: 2ED-Receive
- Receive $3n$ qubits from the sender
- Measure n qubits with σ_x and compute p_x, the frequency of getting outcome $+1$
- Similarly on the remaining qubits, compute p_y and p_z with measurements σ_y and σ_z on n qubits each
- Assign $x \leftarrow 2p_x - 1$, $y \leftarrow 2p_y - 1$, $z \leftarrow 2p_z - 1$; Assign $l \leftarrow \sqrt{x^2 + y^2 + z^2}$
- Output $v \leftarrow (x/l, y/l, z/l)$

2.1.2. Temporal reference frame

Similar to spatial reference frames multiple parties might need to synchronise their clock rates and time differences. Once they have established it, we say that they share a *temporal reference frame* and they are synchronised in time. Any multiparty protocol or computation performed by systems that do not share a temporal reference frame are respectively called *asynchronous protocol* or *asynchronous computation*.

2.2. Asynchronous communication

In an asynchronous network we assume that the nodes do not share any synchronised clock. And the communication channel between each pair is such that a message takes an arbitrary amount of time to propagate through it. Here the only guarantee is, if a message is transmitted from a correct node the message will eventually reach to the receiver. Also, a node might take an arbitrary amount of time to perform the next step in a protocol. In this setup, to analyse the time complexity of an asynchronous protocol we only count the maximum number

Table 1. Channel primitive: **A message**

Step	Classical	Quantum
1	begin	$\perp$
2	m_c	m_q
3	end	$\perp$

of steps executed by any node before the protocol completes, and call it the running time of the protocol. The performance, in terms of execution time, of an asynchronous agreement protocol is determined by its expected running time. The expectation is thereby taken over all possible random inputs of the nodes, random bits used by the nodes, as well as all possible random behaviour of the faulty nodes. The exact probability distributions may not be known, but the goal is to show that the expected running time is low for all possible distributions.

2.2.1. The asynchronous message

In the absence of a synchronised clock, each message must have a 'begin' and 'end' tag. Also, depending on the particular application, a message might carry a [type] tag. In our problem we don't have a shared reference frame. For this reason, we cannot use the quantum channel to carry these [type] tags. This requires us to have a parallel classical channel that uses some classical degree of freedom to carry bits.

We assume that each pair of nodes are connected by an asynchronous public authenticated CQ-channel (classical quantum channel), which can send a message using both classical and quantum degrees of freedom in the absence of a shared reference frame. An example of such combined message is shown in table 1 where each quantum message m_q is sandwiched between a classical 'begin' and an 'end' tag and also accompanied by a classical type tag m_c. The symbol $\perp$ denotes quantum signals that can be ignored.

The only assumption is the nodes can match the classical and quantum parts of the message.

2.2.2. Asynchronous interactive consistency

Our protocol uses the solution to the following interactive consistency problem which was first proposed by Pease, Shostak and Lamport [28].

Definition 2. (The Interactive Consistency Problem). Consider a complete network of n nodes in which communication lines are private. Among the n nodes up to t might be faulty. Let $P_1, P_2, \ldots, P_n$ denote the nodes. Suppose that each node P_i has some private value of information $V_i \in |V| \geqslant 2$. The question is whether it is possible to devise a protocol that, given n, $t \geqslant 0$, will allow each correct node to compute a vector of values with an element for each of the n processors, such that:

1. All the correct nodes compute exactly the same vector.
2. The element of this vector corresponding to a given correct node is the private value of that node.

For an asynchronous network, Ben-Or and El-Yaniv [29] gives a protocol Asynchronous-IC which solves this problem for $t < n/3$ in constant expected time. We use this protocol as a subroutine.

Not that the Asynchronous-IC requires private asynchronous classical channels. Whereas, we only require public authenticated classical and quantum channels between each pair of nodes in the network. The reason is, with authenticated public quantum channels each pair of nodes can play 2ED type protocol and establish a bipartite reference frame. Once the bipartite reference frame is established between each pair using the public authenticated classical and quantum channels they can perform QKD which gives them a private classical channel. So, they can play Asynchronous-IC at a later stage of the protocol. We emphasise that, even thought by playing pairwise 2ED each honest pair of nodes can share a reference frame between them the goal of this paper is to have a global shared reference frame which is non-trivial in the presence of faulty nodes.

3. Results

In this paper we give a protocol that can take any two-party reference frame agreement protocol and lift it up to a fault tolerant multiparty reference frame agreement protocol. More specifically, we present the first protocol A-Agree which allows n nodes in a fully connected asynchronous quantum network to agree on a reference frame in the presence of $t < n/4$ faulty nodes. The result can be summarised in the following theorem.

Theorem 1. *In a complete network of* n *nodes that are pairwise connected by public authenticated quantum and classical channels, if a bipartite δ-estimate direction protocol that uses* m *qubits to achieve success probability $q_{\text{succ}} \geqslant 1 - e^{-\Omega(m\delta^2)}$ is used, then protocol A-Agree is a 42δ-asynchronous reference frame agreement protocol with success probability at least $1 - e^{-\Omega(m\delta^2 - \log n)}$, that can tolerate up to $t < n/4$ faulty nodes.*

Note that, here we use the Ω notation. Therefore, the bounds on success probability asymptotically holds for large enough m. This is not a drawback because, for example, where photon polarisation is used to carry directional information, the pulses of polarised light created by the source would contain large number of photons and allow the protocol to achieve high success probability for a network of an arbitrary size.

The problem of both synchronous and asynchronous agreement on classical bits in the presence of arbitrarily faulty nodes is extensively studied in classical literature as Byzantine agreement problem [27]. However, we emphasise that a classical protocol cannot be used in our problem because firstly, unlike classical network, any communication of direction among correct nodes in a quantum network will have inherent noises. As a result any classical protocol would see all the correct nodes as faulty nodes and the protocol will fail. Secondly, one cannot use the classical protocol directly because one cannot represent a reference frame using only classical bits [26]. However, classical literature can still inform us on important questions such as, how to achieve constant expected time, how to handle asynchronicity. Some of the approaches of our protocol regarding these questions are influenced by [30]. We also use the interactive consistency protocol by Ben-Or *et al* [29] as a subroutine.

Before giving the protocols we first need to define some notation.

$w_i[j]$ represents a vector received by node P_i from node P_j using the bipartite direction estimation protocol. This vector is represented with respects to P_i's local reference frame.

In our protocol sending (type, v) to some node means the sender uses a *δ-estimate direction protocol* to send the direction v to the receiver. The sender also sends the classical tag [type] associated to this direction. The receiver will receive an approximation of the sent direction as v' where $d(v, v') \leqslant \delta$. Our protocol uses four different tags as types. They are, init, echo, ready_1 and ready_2.

Next, we fix a notation for a cluster of vectors of certain types where the cluster has a certain cluster centre, which is the average of the vectors, and a cluster parameter. We write it as $C_i^{\delta}([\text{types}], w_c)$. This means the cluster with cluster centre w_c is computed and stored by node P_i, has a cluster parameter δ and contains only the vectors with associated tags in [types]. Here, [types] is a comma separated list of [type]s. The cluster parameter δ denotes that for all $u, v \in C_i^{\delta}([\text{types}], w_c)$ the distance $d(u, v) \leqslant \delta$.

For example, $C_i^{\delta}([\text{ready}_1, \text{ready}_2], v_c)$ denotes a cluster in which each vector has tags ready_1 or ready_2 with cluster centre v_c such that $\forall u, v \in C_i^{\delta}([\text{ready}_1, \text{ready}_2], v_c)$, and $d(u, v) \leqslant \delta$.

$P(C_i^{\delta}([\text{type}], w_c))$ is the set of all the nodes P_j such that, $w_i[j] \in C_i^{\delta}([\text{type}], w_c)$. That is, it is the set of node id's from which P_i have received the vectors in the cluster $C_i^{\delta}([\text{type}], w_c)$.

Now we give our protocol in two steps. First, we give a protocol for asynchronous broadcast, which allows any sender to securely send a direction to all the other nodes. However, if the sender is faulty the protocol might never terminate. Using this as a primitive we later give our asynchronous agreement protocol.

3.1. Asynchronous broadcast

As the name suggests using this protocol a sender node can send some message to all the other nodes in an asynchronous network. At first sight a naive protocol of just sending the message to all other nodes one by one seems to be a valid protocol. However, this naive protocol does not work if the sender intentionally sends different message to different nodes, which can easily happen in networks with faulty nodes. To guard from it, all the other nodes must communicate between each other to make sure they are receiving the same message, or a close approximation to it. However, as we have at most t faulty nodes, this verification also becomes tricky. The whole thing becomes more challenging because the network is not synchronous. As a result a receiver who is waiting for a message, cannot be certain whether to keep waiting (because the message might be taking a long time in the channel) or move on (the sending node might be faulty and not sending the message at all). Our protocol takes care of all these challenges.

Formally the protocol is defined as,

Definition 3. For $\eta > 0, \zeta > 0$, a protocol which is initiated by a sender node P_s, in an asynchronous network of n nodes, is called a (η, ζ)-*asynchronous reference frame broadcast protocol* if it satisfies the following conditions.

Termination.

1. If the sender is correct then every correct node eventually completes the protocol.
2. If any correct node completes the protocol, then all the correct nodes eventually complete the protocol.

Consistency. If one correct node P_k outputs a direction v_k then all pairs of correct nodes P_i and P_j eventually output directions v_i, v_j where $d(v_i, v_j) \leqslant \eta$.

Correctness. If P_s is correct and broadcasts a direction u and if a correct node P_i outputs v_i then $d(u, v_i) \leqslant \zeta$.

We emphasize that the termination condition of *asynchronous reference frame* ***broadcast*** is much weaker than the termination condition of *asynchronous reference frame* ***agreement*** because in the broadcast protocol we do not require that the correct nodes complete the protocol if the sender is faulty. Also, in an agreement protocol there is no designated sender node, whereas the broadcast protocol has a sender node.

We achieve asynchronous broadcast by our protocol AR-Cast. The following theorem summarises its properties.

Theorem 2. *In a complete network of n nodes that are pairwise connected by public authenticated classical and quantum channels, if a bipartite δ-estimate direction protocol that uses* m *qubits to achieve success probability* $q_{\text{succ}} \geqslant 1 - e^{-\Omega(m\delta^2)}$ *is used, then protocol AR-Cast is a* $(42\delta, 14\delta)$asynchronous reference frame broadcast protocol, *with success probability at least* $1 - e^{-\Omega(m\delta^2 - \log n)}$ *that can tolerate up to* $t < n/4$ *faulty nodes.*

Protocol 2: AR-Cast

input : Sender inputs direction u
output : $\forall\, i$ P_i outputs direction v_i

Epoch 0: (**Only Sender**)
  Send-to-all (init, u)

Epoch 1: (**Player** P_i)
  Listen to init, echo, ready$_1$ and ready$_2$ type messages.
  Wait until Either received one (init, u_i) **Then**
    Send-to-all (echo, u_i).
    Goto Epoch 2.
  Or until received a cluster of directions $C_i^{4\delta}$([echo], w_c) of size at least $(n - 2t)$ **And** a cluster of directions $C_i^{10\delta}$([ready$_1$, ready$_2$], v_c) of size at least $(t + 1)$, so that, $d(w_c, v_c) \leqslant 10\delta$ **Then**
    Send-to-all (ready$_2$, w_c).
    Goto Epoch 3.

Epoch 2: (**Player** P_i)
  Listen to echo, ready$_1$ and ready$_2$ type messages.
  Wait until Either there exists a cluster of directions $C_i^{4\delta}$([echo], w_c) of size at least $(n - t)$ **Then**
    Send-to-all (ready$_1$, w_c).
    Goto Epoch 3.
  Or until there exists a cluster of directions $C_i^{4\delta}$([echo], w_c) of size at least $(n - 2t)$ **And** a cluster of directions $C_i^{10\delta}$([ready$_1$, ready$_2$], v_c) of size at least $(t + 1)$, so that, $d(w_c, v_c) \leqslant 10\delta$, **Then**
    Send-to-all (ready$_2$, w_c).
    Goto Epoch 3.

Epoch 3: (**Player** P_i)
  Wait until there exists a cluster of directions $C_i^{20\delta}$([ready$_1$, ready$_2$], v_c) of size at least $(n - t)$ **Then**
    Output v_c.
    Halt

The protocol 2: AR-Cast works roughly as follows. In Epoch 0 the sender sends its intended direction to all as a [init] type message. In Epoch 1 all the nodes wait until they receive an [init] from sender or sufficient number of confirmations from other nodes that they have received some directions and proceed to the next epoch. This way, even if some correct node never receives an [init] message, if the other correct nodes are advancing through the protocol, then this node in Epoch 1 will not stay behind waiting. In Epoch 2 the correct nodes, which have decided upon a direction, notify the other nodes about their decision by sending ready$_1$ or ready$_2$ type messages

to all. All these previous epochs make sure that all the correct nodes eventually arrive at Epoch 3 and outputs a direction which satisfies theorem 2. The formal proofs are given in the appendix.

3.2. Asynchronous agreement

Now we give our main protocol A-Agree which uses AR-Cast as a subroutine and allows the correct nodes in an asynchronous network to agree on a reference frame.

Protocol 3: A-Agree

input : $\forall\, i, P_i$ inputs direction u_i
output : $\forall\, i, P_i$ outputs direction v_i

Epoch 0: (**Player** P_i)
- Create a direction array w_i of size n.
- $\forall j$, initialize $w_i[j] \leftarrow \perp$.
- Run AR-Cast(u_i).
  // everyone broadcasts their local input
- Store received direction from P_j in $w_i[j]$.
- After receiving $(3t+1)$ such directions **Goto** Epoch 1. However, still continue the incomplete AR-Casts in parallel.

Epoch 1: (**Player** P_i)
- Create a bit string a_i of size n.
- **for** $j \leftarrow 1$ **to** n **do**
  - **if** $w_i[j] \neq \perp$ **Then**
    - Assign $a_i[j] \leftarrow 1$.
  - **else**
    - Assign $a_i[j] \leftarrow 0$.

// a_i records which A-Casts are completed so far by P_i

- Run Asynchronous-IC(a_i).
  // This step reports to all which A-Casts are successfully received by P_i
- Store the output of Asynchronous-IC in vector b_i such that, element $b_i[j]$ is received from P_j.
  // After this step every correct nodes know which A-Casts are reported to be complete by which node
- **Wait until** Asynchronous-IC completes **Then**
  - **Goto** Epoch 2

Epoch 2: (Player P_i)
- Let k_i be the index of a column which has at least $(t+1)$ 1s in it. So that, for any other index l of column with (t+1) 1s $k < l$. // After completion of Asynchronous-IC each row of b_i is a bit string of length n. That is b_i is essentially an $n \times n$ bit matrix.
- **Wait until** the A-Cast initiated by P_{k_i} completes **Then**
  - Assign $v \leftarrow w_i[k_i]$.
  - Abort all incomplete A-Casts that are running since Epoch 0.
  - Output v.

In Epoch 0 of protocol 3: A-Agree each of the nodes P_i proposes a direction u_i, which represents their local frame. They broadcast this direction using AR-Cast. All the correct nodes wait for at least $(3t+1)$ such broadcasts to be complete. Then they enter Epoch 1. Since, there are $(3t+1)$ correct nodes they will eventually arrive at Epoch 1. In this step all the correct nodes create a bit string of length n where j'th bit represents if the j'th AR-Cast has been completed successfully in Epoch 0. Then all the nodes send this bit string to all by playing Asynchronous-IC. After this they enter Epoch 2. In this Epoch every node has the same set of bit strings. They now look for the lowest inter k such that at least $(t+1)$ bit strings have a 1 in the k'th index of the string. If they have completed that k'th AR-Cast they output their direction received from that broadcast. If the k'th AR-Cast is not complete for a node, it waits until it completes and then output. The election of k ensures that at least one correct node has completed the k'th AR-Cast so by Consistency of asynchronous reference frame broadcast all the correct nodes will eventually complete the k'th AR-Cast. This ensures that the A-Agree eventually completes.

There is no conditional loop in this protocol and all the subroutines run in constant expected time. So, the A-Agree is also a constant expected time protocol. The formal proofs are given in the appendix.

4. Conclusion

In this work we have presented the first asynchronous reference frame agreement protocol. The synchronous protocol for spatial reference frame agreement presented in [20] can tolerate up to $t < n/3$ faulty nodes. Whereas, the asynchronous protocol we have presented tolerates only $t < n/4$ faulty nodes. Even though we pay this extra price in fault tolerance, an asynchronous protocol is a fully general reference frame agreement protocol. If message delays are fixed, our protocol can also be used to synchronise clocks [31], which is an important problem in its own right. There are classical protocols for asynchronous agreement on bits which achieve $t < n/3$ in constant expected time, it remains open to see if this bound can be achieved by reference frame agreement protocols for a quantum network.

Acknowledgments

We thank Loïck Magnin and Michael Ben-Or (via Loick Magnin) for helpful pointers, and David Elkouss for comments on an earlier version of this article. This work was supported by NRF CRP Grant 'Space based QKD' and STW, QuTech. Stephanie Wehner is also supported by NWO VIDI Grant.

Appendix

A.1. Asynchronous reference frame broadcast

To prove correctness of our AR-Cast we have to prove theorem 2 as repeated here.

Theorem 2. *In a complete network of n nodes that are pairwise connected by public authenticated quantum and classical channels, if a bipartite δ-estimate direction protocol that uses m qubits to achieve success probability $q_{\text{succ}} \geqslant 1 - e^{-\Omega(m\delta^2)}$ is used, then protocol AR-Cast is a* $(42\delta,\ 14\delta)$-asynchronous reference frame broadcast protocol, *with success probability at least $1 - e^{-\Omega(m\delta^2 - \log n)}$ that can tolerate up to $t < n/4$ faulty nodes.*

For this we observe several properties of protocol 2 in the following lemmas. The first observation is that if two different correct nodes send $[\text{ready}_1]$-type messages then the direction they send are close to each other with high probability.

Lemma 1. *For $t < n/4$, $\delta > 0$, $q_{\text{succ}} > 0$, if two correct nodes P_i and P_j send $(\lfloor \text{ready}_1 \rfloor, u)$ and $(\lfloor \text{ready}_1 \rfloor, v)$ respectively, then $d(u, v) \leqslant 10\delta$ with probability at least $q_{\text{succ}}^{n+n^2}$.*

Proof. In step 4 of Epoch 2 when a $[\text{ready}_1]$ message is generated there are at most n init messages originated from the sender and at most n^2 echo messages generated by the other nodes. So, with probability at least $q_{\text{succ}}^{n+n^2}$ all the transmissions which are among correct nodes are successful. Conditioning on this, we prove,

$$d(u, v) \leqslant 10\delta. \tag{1}$$

We show this in two steps. First, we show that there exists a common correct node P_k in $P(C_i^{4\delta}([\text{echo}],\ u))$ and $P(C_j^{4\delta}([\text{echo}],\ v))$, where $C_i^{4\delta}([\text{echo}],\ u)$ and $C_j^{4\delta}([\text{echo}],\ v)$ are the cluster of echo type directions received by P_i and P_j, respectively . Then using the triangle inequality with the fact that the echo vector from P_k must be close to both of the cluster centers u and v, we derive inequality (1).

Now, for the first step, let us denote A_i and A_j to be the set of nodes from which the vectors respectively in $C_i^{4\delta}([\text{echo}],\ u)$ and $C_j^{4\delta}([\text{echo}],\ v)$ have originated. And B_i and B_j to be the correct nodes in A_i and A_j respectively. Formally,

$$A_i = P(C_i^{4\delta}([\text{echo}],\ u)), \tag{2}$$

$$A_j = P(C_j^{4\delta}([\text{echo}],\ v)), \tag{3}$$

$$B_i = \{P_l : P_l \in A_i \text{ and } P_l \text{ is correct.}\}, \tag{4}$$

$$B_j = \{P_l : P_l \in A_j \text{ and } P_l \text{ is correct.}\}. \tag{5}$$

Note that at this step $|A_i| \geqslant n - t$ and $|A_j| \geqslant n - t$. We want to show that,

$$B_i \bigcap B_j \neq \varnothing. \tag{6}$$

We do this by contradiction: let us assume that,

$$B_i \bigcap B_j = \varnothing. \tag{7}$$

Note that,

$$|A_i| \geqslant n - t \tag{8}$$
$$\Rightarrow |A_i - B_i| + |B_i| \geqslant n - t, \tag{9}$$
$$\Rightarrow t + |B_i| \geqslant n - t, \tag{10}$$
$$\Rightarrow |B_i| \geqslant n - 2t, \tag{11}$$
$$\Rightarrow |B_i| > n - 2(n/4) = n/2. \tag{12}$$

Here, inequality (10) holds because at most t of the nodes are faulty. And inequality (12) holds because $t < n/4$.

Now,

$$|A_i \cup A_j| = |(A_i - B_i) \cup (A_j - B_j) \cup B_i \cup B_j|,$$
$$\geqslant |(A_j - B_j)| + |B_i| + |B_j|, \tag{13}$$
$$= |A_j| + |B_i|, \tag{14}$$
$$> (n - t) + n/2, \tag{15}$$
$$> n - n/4 + n/2 = 5n/4 \tag{16}$$

Here, inequality (13) uses inequality (7), inequality (15) follows from the definition from the size of A_j and inequality (12). And inequality (16) follows because, $t < n/4$. However, this is a contradiction, because there are only n nodes in the network. Therefore, we have,

$$B_i \bigcap B_j \neq \varnothing. \tag{17}$$

So, there exists a common correct node $P_k \in B_i \cap B_j$ in $P(C_i^{4\delta}([\text{echo}],\ u))$ and $P(C_j^{4\delta}([\text{echo}],\ v))$. Since P_k is correct, it must have sent the same echo type message to both P_i and P_j. So, using the triangle inequality we have,

$$d(w_i[k],\ w_j[k]) \leqslant d(w_i[k],\ u_k) + d(u_k,\ w_j[k]), \tag{18}$$
$$\leqslant \delta + \delta = 2\delta. \tag{19}$$

Now inequality (1) follows because,

$$d(u,\ v) \leqslant d(u,\ w_i[k]) + d(w_i[k],\ w_j[k]) + d(w_j[k],\ v), \tag{20}$$
$$\leqslant 4\delta + d(w_i[k],\ w_j[k]) + 4\delta, \tag{21}$$
$$\leqslant 4\delta + 2\delta + 4\delta = 10\delta. \tag{22}$$

Here, inequality (21) follows from the definitions of $C_i^{4\delta}([\text{echo}],\ u)$ and $C_j^{4\delta}([\text{echo}],\ v)$ and inequality (22) follows from inequality (19). □

In lemma 1 we have shown the relation between two $[\text{ready}_1]$ type directions from two different correct nodes. Now we show that if a correct node sends a $[\text{ready}_1]$ and another correct node sends a $[\text{ready}_2]$ type message then the directions they send are close with high probability. Both of these proofs use similar techniques.

Lemma 2. *For $t < n/4$, $\delta > 0$, $q_{\text{succ}} > 0$, if two correct nodes P_i and P_j send ($\lceil\text{ready}_1\rfloor$,u) and ($\lceil\text{ready}_2\rfloor$,v) accordingly, then $d(u,\ v) \leqslant 10\delta$ with probability at least $q_{\text{succ}}^{n+2n^2}$.*

Proof. When a $[\text{ready}_2]$ message is generated there are at most n init, n^2 echo and in total n^2 $[\text{ready}_1]$ or $[\text{ready}_2]$ messages generated in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all the transmissions which are among correct nodes are successful. Conditioning on this, we show that,

$$d(u,\ v) \leqslant 10\delta. \tag{23}$$

We do this in two steps, first we show that there is a common correct node P_k in $P(C_i^{4\delta}([\text{echo}],\ u))$ and $P(C_j^{4\delta}([\text{echo}],\ v))$. Then using the triangle inequality with the fact that both of the cluster centers u and v must be close to the echo direction sent from P_k we prove the inequality (23).

Now, for the first step, let us denote A_i and A_j to be the set of nodes from which the vectors respectively in $(C_i^{4\delta}([\text{echo}],\ u)$ and $C_j^{4\delta}([\text{echo}],\ v)$ have originated. And B_i and B_j to be the correct nodes in A_i and A_j respectively. Formally,

$$A_i = P(C_i^{4\delta}([\text{echo}],\ u)), \tag{24}$$

$$A_j = P(C_j^{4\delta}([\text{echo}],\ v)), \tag{25}$$

$$B_i = \{P_l : P_l \in A_i \text{ and } P_l \text{ is correct.}\}, \tag{26}$$

$$B_j = \{P_l : P_l \in A_j \text{ and } P_l \text{ is correct.}\}. \tag{27}$$

Note that here $|A_i| \geqslant n - t$ and $|A_j| \geqslant n - 2t$. We want to show that,

$$B_i \bigcap B_j \neq \varnothing. \tag{28}$$

We do this by contradiction: let us assume that,

$$B_i \bigcap B_j = \varnothing. \tag{29}$$

Note that,

$$|A_i| \geqslant n - t \tag{30}$$

$$\Rightarrow |A_i - B_i| + |B_i| \geqslant n - t, \tag{31}$$

$$\Rightarrow t + |B_i| \geqslant n - t, \tag{32}$$

$$\Rightarrow |B_i| \geqslant n - 2t, \tag{33}$$

$$\Rightarrow |B_i| > n - 2(n/4) = n/2. \tag{34}$$

Here, inequality (32) holds because at most t of the nodes are faulty. And inequality (34) holds because $t < n/4$.

Now,

$$\begin{aligned} |A_i \cup A_j| &= |(A_i - B_i) \cup (A_j - B_j) \cup B_i \cup B_j|, \\ &\geqslant |(A_j - B_j)| + |B_i| + |B_j|, \end{aligned} \tag{35}$$

$$= |A_j| + |B_i|, \tag{36}$$

$$> (n - 2t) + n/2, \tag{37}$$

$$> n - n/2 + n/2 = n \tag{38}$$

Here, inequality (37) follows from the definition of A_j and inequality (34). And inequality (38) follows because, $t < n/4$. However, this is a contradiction, because there are only n nodes in the network. Therefore, we have,

$$B_i \bigcap B_j \neq \varnothing. \tag{39}$$

So, there exists a common correct node P_k in $P(C_i^{4\delta}([\text{echo}],\ u))$ and $P(C_j^{4\delta}([\text{echo}],\ v))$. As P_k is correct, it must have sent the same echo type message to both P_i and P_j. So, using the triangle inequality we have,

$$d(w_i[k],\ w_j[k]) \leqslant d(w_i[k],\ u_k) + d(u_k,\ w_j[k]), \tag{40}$$

$$\leqslant \delta + \delta = 2\delta. \tag{41}$$

Now inequality (23) follows because,

$$d(u,\ v) \leqslant d(u,\ w_i[k]) + d(w_i[k],\ w_j[k]) + d(w_j[k],\ v), \tag{42}$$

$$\leqslant 4\delta + d(w_i[k],\ w_j[k]) + 4\delta, \tag{43}$$

$$\leqslant 4\delta + 2\delta + 4\delta = 10\delta. \tag{44}$$

Here, inequality (43) follows from the definitions of $C_i^{4\delta}([\text{echo}],\ u)$ and $C_j^{4\delta}([\text{echo}],\ v)$ and inequality (44) follows from inequality (41). □

Now we show that all the correct nodes cannot send only $[\text{ready}_2]$ type messages. That is, if there exists a $[\text{ready}_2]$ message sent from a correct node, then there must pre-exist a $[\text{ready}_1]$ message sent from another correct node.

Lemma 3. *For $t < n/4$, $\delta > 0$, $q_{\text{succ}} > 0$, if a correct node* P_j *sends* $(\lceil\text{ready}_2\rfloor,\text{v})$, *then with probability at least* $q_{\text{succ}}^{n+2n^2}$, *there exists a correct node* P_i *which has sent* $(\lceil\text{ready}_1\rfloor,\text{u})$.

Proof. When a [ready$_2$] message is generated there are at most n [init], n^2 [echo] and in total n^2 [ready$_1$] or [ready$_2$] messages generated in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all the transmissions which are among correct nodes are successful. In this case, just before making the decision to send a ([ready$_2$],v) message node P_j must have received at least (t+1) [ready$_1$] or [ready$_2$] messages from nodes in $P(C_i^{10\delta}([\text{ready}_1,\ \text{ready}_2]v_c))$. Of these, at least one node—let's call it P_k—is correct. If P_k has also sent a [ready$_2$] type message, we can find another correct node in its $P(C_k^{10\delta}([\text{ready}_1,\ \text{ready}_2]v_c))$ and so on. This way, eventually we will find a correct node who has sent a [ready$_1$] type message.

To see this, let us define a directed graph $G(V,\ E)$ with vertex set $V = \{P_i : P_i \text{ is correct}\}$, and

$$E = \{(P_k,\ P_i) : P_k \text{ has sent ready}_2 \text{ after receiving ready}_1 \text{ or ready}_2 \text{ from } P_i\}. \tag{45}$$

One can convince oneself that G is a directed acyclic graph because any cycle in the graph would violate the cause and effect relation of the edge directions. Now if we look at the connected component of this graph containing P_j there must exist a node P_i in this component with no outgoing edges. Because V only contains correct nodes. This implies P_i is a correct node which has sent a [ready$_1$] type message ([ready$_1$],u). This completes the proof. □

Now the only thing that remains is to show that two [ready$_2$] type directions sent from two correct nodes are close with high probability.

Lemma 4. *For $t < n/4$, $\delta > 0$, $q_{\text{succ}} > 0$, if two nodes* P_i *and* P_j *sends* ($\lceil$ready$_2\rfloor$,u) *and* ($\lceil$ready$_2\rfloor$,v) *respectively, then $d(u,\ v) \leqslant 20\delta$ with probability at least $q_{\text{succ}}^{n+2n^2}$.*

Proof. When a [ready$_2$] message is generated there are at most n [init], n^2 [echo] and in total n^2 [ready$_1$] or [ready$_2$] messages generated in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions which are between correct nodes are successful. Conditioning on this, we show that, if correct P_i sends ([ready$_2$],u) then from lemma 3 there exists a correct node P_k which has sent ([ready$_1$],w). From lemma 2,

$$d(u,\ w) \leqslant 10\delta, \tag{46}$$

and

$$d(v,\ w) \leqslant 10\delta. \tag{47}$$

Using the triangle inequality with these we get,

$$d(u,\ v) \leqslant d(u,\ w) + d(w,\ v) \leqslant 10\delta + 10\delta = 20\delta. \tag{48}$$

□

Now we are ready to prove that our protocol 2 satisfies the first termination condition of definition 3.

Lemma 5 (Termination 1). *For $t < n/4$, $\delta > 0$, $q_{\text{succ}} > 0$, if the sender* P_k *is correct then the protocol 2* AR-Cast *eventually terminates with probability at least $q_{\text{succ}}^{n+n^2}$.*

Proof. There are at most n [init] messages, n^2 [echo] messages and n^2 [ready$_1$] or [ready$_2$] type messages exchanged in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions which are between correct nodes are successful. In this case, if the sender is correct all the correct nodes eventually receive [init] messages that are at most 2δ apart from each other and send an echo message. So, all the received [echo] messages are at most 3δ apart from the received direction in the [init] message of any correct node. Any node that has sent a [ready$_1$] type message will go to epoch 3. The faulty nodes cannot stop the [init] and [echo] messages from correct nodes but they can manipulate the delays, so that some of the correct nodes send [ready$_2$] type messages. After sending the [ready$_2$] these correct nodes will eventually arrive at Epoch 3. From lemmas 1 and 2 we can see that for any correct P_i all the received [ready$_1$] and [ready$_2$] directions will be in $C_i^{16\delta}([\text{ready}_1,\ \text{ready}_2],\ v_c)$. And because there are $(n - t)$ of them originating from the correct nodes the protocol 2 AR-Cast will eventually terminate. Note that, if the sender is faulty, the definition of $(\eta,\ \zeta)$-reference frame broadcast protocol (derinition 3) do not require any termination. □

Now we show that if one correct node outputs a direction, then all the correct nodes eventually output directions that are close to each other.

Lemma 6 (Consistency). *For* $t < n/4$, $\delta > 0$, $q_{\text{succ}} > 0$, *in protocol* AR-cast, *if a correct node* $\mathrm{P_k}$ *outputs* $\mathrm{v_k}$ *then all pair of correct nodes* P_i, P_j *eventually output* v_i, v_j *respectively such that,* $d(v_i, v_j) \leqslant 42\delta$ *with probability at least* $q_{\text{succ}}^{n+n^2}$.

Proof. When a $[\text{ready}_2]$ message is generated there are at most n init, n^2 echo and in total n^2 $[\text{ready}_1]$ or $[\text{ready}_2]$ messages generated in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions which are between correct nodes are successful. In this case, we prove,

$$d(v_i, v_j) \leqslant 42\delta, \tag{49}$$

by showing that the successful completion of P_k implies there are enough echo, $[\text{ready}_1]$ and $[\text{ready}_2]$ type messages generated by correct nodes so that all the other correct nodes eventually receive them and successfully terminate and each pair of their outputs satisfies inequality (49).

Now, if a correct node P_k outputs v_k then this implies it has received at least $(n - t)$ $[\text{ready}_1]$ or $[\text{ready}_2]$ messages from nodes in $P(C_k^{20\delta}([\text{ready}_1, \text{ready}_2], v_k))$, of which at least $(n - 2t)$ are correct. Messages from these correct nodes eventually reach all the other correct nodes. Also, from lemma 3 there exists a correct node which has sent a $[\text{ready}_1]$ message which implies all the correct nodes eventually receive at least $(n - 2t)$ echo messages. That is, all the correct nodes waiting in Epoch 1 or Epoch 2 will satisfy the condition of sending a $[\text{ready}_2]$ message and go to Epoch 3. Any correct node P_i, P_j waiting in Epoch 3 will eventually receive all the $[\text{ready}_1]$ or $[\text{ready}_2]$ messages sent from correct nodes in $P(C_i^{20\delta}([\text{ready}_1, \text{ready}_2], v_i))$ and $P(C_j^{20\delta}([\text{ready}_1, \text{ready}_2], v_j))$ accordingly, and output v_i, v_j accordingly.

Now we show that $P(C_i^{20\delta}([\text{ready}_1, \text{ready}_2], v_i))$ and $P(C_j^{20\delta}([\text{ready}_1, \text{ready}_2], v_j))$ have at least one common correct node, which implies the cluster centers are close.

To see this note that each of these clusters have at least $(n - 2t) > n - 2(n/4) = n/2$ correct nodes. That is more than n correct nodes in total. However there are total n nodes in the networks. This implies at least some of the correct nodes are common in both clusters. Let P_l be such a node.

Now using triangular inequality we have,

$$\begin{aligned} d(v_i, v_j) &\leqslant d(v_i, v_i[l]) + d(v_i[l], v_l) \\ &\quad + d(v_l, v_j[l]) + d(v_j[l], v_j), \end{aligned} \tag{50}$$

$$\leqslant 20\delta + \delta + \delta + 20\delta = 42\delta. \tag{51}$$

Here inequality (51) follows using lemma 4. □

Now the second termination condition.

Lemma 7 (Termination 2). *For* $t < n/4$, $\delta > 0$, $q_{\text{succ}} > 0$, *if a correct node* $\mathrm{P_i}$ *completes the protocol then all the correct nodes complete the protocol with probability at least* $q_{\text{succ}}^{n+2n^2}$.

Proof. This lemma is a corollary of lemma 6. Because lemma 6 ensures completion with probability at least $q_{\text{succ}}^{n+2n^2}$. □

Now we are ready to prove that our protocol satisfies the correctness condition of definition 3.

Lemma 8 (*Correctness*). *For* $t < n/4$, $\delta > 0$, $q_{\text{succ}} > 0$, *if a correct sender* $\mathrm{P_s}$ *sends* (*init*,*u*) *and a correct node* $\mathrm{P_i}$ *outputs* $\mathrm{v_i}$ *then* $d(u, v_i) \leqslant 14\delta$ *with probability at least* $q_{\text{succ}}^{n+2n^2}$.

Proof. There are at most n init messages, n^2 echo messages and n^2 $[\text{ready}_1]$ or $[\text{ready}_2]$ type messages exchanged in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions which are between correct nodes are successful.

In this case we prove the lemma in three steps. First, we show that all the $[\text{ready}_1]$ type directions sent from correct nodes are close to u. Secondly, we show that all the $[\text{ready}_2]$ type directions sent from the correct nodes are close to u. And finally, from these we conclude the proof.

For the first step, let us assume that correct node P_i has sent a $([\text{ready}_1], v_i)$ message in Epoch 2. So, it has received at least $(n - t)$ echo type messages, of which at least $(n - 2t)$ are from correct nodes. Let's assume for some correct node P_j $w_i[j] \in C_i^{4\delta}(v_i)$. Since P_j is correct, using the triangle inequality, we have,

$$d(u, w_i[j]) \leqslant d(u, u_j) + d(u_j, w_i[j]), \tag{52}$$

$$\leqslant \delta + \delta = 2\delta. \tag{53}$$

The diameter of the cluster $C_i^{4\delta}(v_i)$ is 4δ. So, we have, $d(v_i, w_i[j]) \leqslant 2\delta$. Using this and (53) with the triangle inequality, we have,

$$d(u, v_i) \leqslant d(u, w_i[j]) + d(w_i[j], v_i), \tag{54}$$

$$\leqslant 2\delta + 2\delta = 4\delta. \tag{55}$$

Now, for the second step, let us assume that a correct node P_l has sent a $([\text{ready}_2], v_l)$ message from Epoch 1 or Epoch 2. So, v_l is a cluster center of at least $(n - 2t)$ echo type messages. Of which at least $(n - 3t)$ are correct. So, a similar reasoning to the previous step shows,

$$d(u, v_l) \leqslant 4\delta. \tag{56}$$

Finally, since the sender is correct from lemma 5 we know, all the correct nodes eventually enter Epoch 3 and successfully complete the epoch.

Let us assume a correct node P_i has received a cluster of $[\text{ready}_1]$ or $[\text{ready}_2]$ type directions $C_i^{20\delta}([\text{ready}_1, \text{ready}_2], v_c)$ of size at least $(n - t)$. So, there is a correct node P_k for which $v_i[k] \in C_i^{20\delta}([\text{ready}_1, \text{ready}_2], v_c)$. Here, $C_i^{20\delta}([\text{ready}_1, \text{ready}_2], v_c)$ is a cluster of diameter 20δ. So, we have $d(v_i[k], v_c) \leqslant 10\delta$. Using the triangle inequality with this, and (55) and (56), we have,

$$d(u, v_c) \leqslant d(u, w_i[k]) + d(w_i[k], v_c), \tag{57}$$

$$\leqslant 4\delta + 10\delta = 14\delta. \tag{58}$$

This concludes the proof. □

Now we give an auxiliary lemma that shows how the probability of success scales with the number of nodes and the success probability of the δ-estimate direction protocol.

Lemma 9. *If a two-node direction estimation protocol is used that transmits* m *qubits to* δ *approximate a direction which succeeds with probability* $q_{\text{succ}} \geqslant (1 - e^{-\Omega(m\delta)})$ *then with probability at least* $q_{\text{succ}}^{n+2n^2} \geqslant 1 - e^{-\Omega(m\delta^2 - \log n)}$, *all the direction transmissions of init, echo,* $[\text{ready}_1]$ *and* $[\text{ready}_2]$ *type messages are successful.*

Proof. There are at most n init messages, n^2 echo messages and n^2 $[\text{ready}_1]$ or $[\text{ready}_2]$ type messages exchanged in the protocol. With probability at least $q_{\text{succ}}^{n+2n^2}$ all of these transmissions which are between correct nodes are successful. Now,

$$q_{\text{succ}}^{n+2n^2} \geqslant (1 - e^{-\Omega(m\delta^2)})^{n+2n^2}, \tag{59}$$

$$\geqslant 1 - (n + 2n^2)e^{-\Omega(m\delta^2)}, \tag{60}$$

$$\geqslant 1 - e^{-\Omega(m\delta^2 - \log n)} \tag{61}$$

Here inequality (60) follows using Bernoulli's inequality, which is, $(1 + x)^r \geqslant 1 + rx$ for all real $x \geqslant -1$ and integer $r \geqslant 2$. □

We see that, theorem 2 follows from lemma 5–9.

A.2. Asynchronous Agreement

So far we have presented an asynchronous broadcast protocol where a designated sender initiates the protocol with a direction. One major weakness of the protocol is that, if the sender is faulty then the protocol might never terminate, because in this case the correct nodes cannot decide whether the sender is faulty and not sending the [init] message, or correct but very slow. On the other hand, in an asynchronous reference frame agreement protocol the main goal is to allow the correct nodes to agree on some direction despite the presence of—up to a certain number of—unidentified faulty nodes in the network. This requires extra caution to make sure that the protocol eventually terminates. We show that our protocol 3 A-Agree successfully solves this problem by proving theorem 1. We repeat the theorem here.

Theorem 1. *In a complete network of* n *nodes that are pairwise connected by public authenticated classical and quantum channels, if a bipartite* δ*-estimate direction protocol that uses* m *qubits to achieve success probability* $q_{\text{succ}} \geqslant 1 - e^{-\Omega(m\delta^2)}$ *is used, then protocol A-Agree is a* 42δ*-asynchronous reference frame agreement protocol with success probability at least* $1 - e^{-\Omega(m\delta^2 - \log n)}$, *that can tolerate up to* $t < n/4$ *faulty nodes.*

There are three epochs in protocol 3. Any correct node that successfully terminates must start at Epoch 0 and terminate at Epoch 3. At each Epoch the nodes inside it, and all the messages transmitted and received by the node while in that Epoch satisfies some invariance properties. We describe and prove these properties in the following lemmas. We first show that a correct node will eventually enter Epoch 1.

Lemma 10. *For $t < n/4$, all the correct nodes eventually enter Epoch 1 of A-Agreement with probability at least $q_{\text{succ}}^{n^2+2n^3} \geqslant 1 - e^{-\Omega(m\delta^2 - \log n)}$.*

Proof. Each of the n nodes has initiated an AR-Cast in Epoch 0. Each of the AR-Casts has a success probability at least $q_{\text{succ}}^{n+2n^2}$. So, with probability at least $q_{\text{succ}}^{n^2+2n^3}$ all the AR-Casts from correct senders are successful. From lemma 9 this is at least $1 - e^{-\Omega(m\delta^2 - \log n)}$.

As $t < n/4$, there are at least $(3t + 1)$ correct nodes who initiates AR-Cast as sender. According to theorem 2 these $(3t + 1)$AR-Casts will eventually terminate. So, every correct receiver will eventually receive at least $(3t + 1)$ directions and go to Epoch 1 with probability at least $q_{\text{succ}}^{n^2+2n^3}$. □

Each of the correct nodes stores the output of the Asynchronous-IC protocol in an array b_i. Here b_i can be seen as an $n \times n$ matrix of bits where row j is received from node j. We can observe the following property of this matrix.

Lemma 11. *For $t < n/4$ and correct node $\mathrm{P_i}$, after instruction 9 of Epoch 1 of A-Agreement, there exists a column in $\mathrm{b_i}$ with at least $(t + 1)$ 1 s in it.*

Proof. We show this by a counting argument. Note that a correct node arrives at Epoch 1 only after it have received at least $(3t + 1)$ directions from other players. As a result after step 7 of Epoch 1 a_i contains at least $(3t + 1)$ 1's. These a_i's become the rows of b_i after step 9. There are at most t faulty nodes. So, at least $(3t + 1)$ rows of b_i are originated from correct nodes. Each of these rows must contain at least $(3t + 1)$ 1's. So b_i has at least $(3t + 1)^2$ 1 s.

However, if no column had at least $(t + 1)$ 1 s, then there would be at most $(4t + 1) \times t$ 1 s in b_i. This contradicts the fact that b_i has at least $(3t + 1)^2$ 1 s. So, there must exist a column with at least $(t + 1)$ 1 s in it. □

We show that all the correct nodes select the same column which has at least $t + 1$ 1s in it.

Lemma 12. *After instruction 2 of Epoch 2 of A-Agreement, if correct node P_i has k_i and correct node $\mathrm{P_j}$ has $\mathrm{k_j}$, then $k_i = k_j$.*

Proof. After completion of protocol Asynchronous-IC in Epoch 1, all the correct nodes compute the same output vector. That is, $b_i = b_j$ for all correct P_i and P_j. Also, from lemma 11 we know there exists a column in b_i with at least $(t + 1)$ 1 s. So, in Epoch 2 step 2 when correct node P_i and P_j selects k_i and k_j to be the chronologically smallest column index that has at least $(t + 1)$ 1 s. They select the same column. i.e., $k_i = k_j$. □

Now that every correct node P_i agrees on a column k_i of b_i, we observe that.

Lemma 13. *If a correct node $\mathrm{P_i}$ selects $\mathrm{k_i}$ in instruction 2 of Epoch 2, then the AR-Cast initiated by P_{k_i} in Epoch 0 eventually completes successfully.*

Proof. We show this by showing that at least one correct node has completed the AR-Cast initiated by P_{k_i}. Then the lemma follows from the termination condition of AR-Cast.

Each row $b_i[j]$ represents P_i's knowledge of which AR-Casts are successfully received by P_j. For example, if $b_i[j][l] = 1$, then it means node P_j has reported to P_i that it has completed the AR-Cast initiated by node P_l in Epoch 0. If there are at least $(t + 1)$ 1 s in the k_ith column of b_i, it means that there are $(t + 1)$ nodes who report that they have received the AR-Cast initiated by node P_{k_i} in Epoch 0. At least one of these reports is from a correct node. So, from the termination condition of AR-Cast (lemma 6) all the correct nodes eventually successfully complete the AR-Cast by P_k. □

Now we are ready to prove **theorem** 1.

Proof. There are at most n AR-Casts initiated in Epoch 0 of which $(n - t)$ are by correct nodes. From lemma 9 each of these succeeds with probability $q_{\text{succ}}^{n+2n^2} \geqslant 1 - e^{-\Omega(m\delta^2 - \log n)}$. So all the correct AR-Casts succeed with,

$$q_{\text{succ}}^{n^2+2n^3} \geqslant (1 - e^{-\Omega(m\delta^2 - \log n)})^n, \tag{62}$$

$$\geqslant 1 - e^{-\Omega(m\delta^2 - \log n)}. \tag{63}$$

Here inequality (63) follows from Bernoulli's inequality. Conditioned on this we show,

Correctness. To prove consistency we show that if a correct node P_i outputs v_i and a correct node P_j outputs v_j then $d(v_i, v_j) \leqslant 42\delta$. From step 4 of Epoch 2 of A-Agree we see that,

$$v_i = w_i[k_i], \tag{64}$$

$$v_j = w_j[k_j]. \tag{65}$$

From lemma 6 we know that for $t < n/4$,

$$d(w_i[k_i], w_j[k_j]) \leqslant 42\delta. \tag{66}$$

This with (64) and (65) gives,

$$d(v_i, v_j) \leqslant 42\delta. \tag{67}$$

Termination. To prove termination we have to show that every correct node P_i terminates with an output direction v_i.

To prove this we show that P_i eventually completes all the Epochs of A-Agree. From lemma 10 we see that P_i must enter Epoch 1 from Epoch 0. All the steps in Epoch 1 are of constant expected time. So, a correct node will eventually complete them and go to Epoch 2. Only in step 3 of Epoch 2 P_i waits for completion of AR-Cast from P_{k_i}. However, from lemma 13 we know that this AR-Cast eventually successfully completes. All the other incomplete AR-Casts are then aborted at step 5 and the protocol terminates with output v_i. □

References

[1] Kimble H J 2008 *Nature* **453** 1023
[2] Aspelmeyer M, Jennewein T, Pfennigbauer M, Leeb W and Zeilinger A 2003 *IEEE J. Sel. Topics Quantum Electron* **9** 1541
[3] Bonato C, Tomaello A, Deppo V D, Naletto G and Villoresi P 2009 *New J. Phys.* **11** 045017
[4] Peng C-Z *et al* 2005 *Phys. Rev. Lett.* **94** 150501
[5] Bonato C, Aspelmeyer M, Jennewein T, Pernechele C, Villoresi P and Zeilinger A 2006 *Opt. Express* **14** 10050
[6] Armengol J M P *et al* 2008 *Acta Astronaut.* **63** 165
[7] Sasaki M *et al* 2011 *Opt. Express* **19** 10387
[8] Cirac J I, Zoller P, Kimble H J and Mabuchi H 1997 *Phys. Rev. Lett.* **78** 3221
[9] Elliott C 2002 *New J. Phys.* **4** 46
[10] Beals R, Brierley S, Gray O, Harrow A W, Kutin S, Linden N, Shepherd D and Stather M 2013 *Proc. R. Soc.* A **469** 20120686
[11] Li Y and Benjamin S C 2012 *New J. Phys.* **14** 093008
[12] Barz S, Kashefi E, Broadbent A, Fitzsimons J F, Zeilinger A and Walther P 2012 *Science* **335** 303
[13] Massar S and Popescu S 1995 *Phys. Rev. Lett.* **74** 1259
[14] Peres A and Scudo P F 2001a *Phys. Rev. Lett.* **87** 167901
[15] Bagan E, Baig M, Muñoz-Tapia R and Rodriguez A 2004 *Phys. Rev.* A **69** 010304
[16] Chiribella G and D'Ariano G M 2004 *J. Math. Phys.* **45** 4435
[17] Bagan E and Muñoz-Tapia R 2006 *Int. J. Quantum Inf.* **4** 5
[18] Giovannetti V, Lloyd S and Maccone L 2006 *Phys. Rev. Lett.* **96** 010401
[19] Skotiniotis M and Gour G 2012 *New J. Phys.* **14** 073022
[20] Islam T, Magnin L, Sorg B and Wehner S 2014 *New J. Phys.* **16** 063040
[21] Bartlett S D, Rudolph T and Spekkens R W 2007 *Rev. Mod. Phys.* **79** 555
[22] Shadbolt P, Vértesi T, Liang Y-C, Branciard C, Brunner N and O'Brien J L 2012 *Sci. Rep.* **2** 470
[23] Brask J B, Chaves R and Brunner N 2013 *Physical Review* A **88** 012111
[24] D'Ambrosio V, Nagali E, Walborn S P, Aolita L, Slussarenko S, Marrucci L and Sciarrino F 2012 *Nat. Commun.* **3** 961
[25] Komar P, Kessler E M, Bishof M, Jiang L, Sorensen A S, Ye J and Lukin M D 2014 *Nat. Phys.* **10** 582
[26] Peres A and Scudo P F 2001b *Phys. Rev. Lett.* **86** 4160
[27] Lamport L, Shostak R and Pease M 1982 *ACM T. Prog. Lang. Sys.* **4** 382
[28] Pease M, Shostak R and Lamport L 1980 *J. ACM* **27** 228
[29] Ben-Or M and El-Yaniv R 2003 *DISTRIB COMPUT* **16** 249
[30] Canetti R and Rabin T 1993 *Proc. ACM STOC'93* (ACM) pp 42–51
[31] Chuang I L 2000 *Phys. Rev. Lett.* **85** 2006