Byzantine Convex Consensus: Preliminary Version*†

Lewis Tseng[1,3], and Nitin Vaidya[2,3]

[1] Department of Computer Science,
[2] Department of Electrical and Computer Engineering, and
[3] Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Email: {ltseng3, nhv}@illinois.edu

Technical Report

July 3rd, 2013

Abstract

Much of the past work on asynchronous approximate Byzantine consensus has assumed *scalar* inputs at the nodes [3, 7]. Recent work has yielded approximate Byzantine consensus algorithms for the case when the input at each node is a d-dimensional vector, and the nodes must reach consensus on a vector in the convex hull of the input vectors at the fault-free nodes [8, 12]. The d-dimensional vectors can be equivalently viewed as *points* in the d-dimensional Euclidean space. Thus, the algorithms in [8, 12] require the fault-free nodes to decide on a point in the d-dimensional space.

In this paper, we generalize the problem to allow the decision to be a *convex polytope* in the d-dimensional space, such that the decided polytope is within the convex hull of the input vectors at the fault-free nodes. We name this problem as *Byzantine convex consensus* (BCC), and present an asynchronous approximate BCC algorithm with optimal fault tolerance. Ideally, the goal here is to agree on a convex polytope that is as large as possible. While we do not claim that our algorithm satisfies this goal, we show a bound on the output convex polytope chosen by our algorithm.

*We present an optimal algorithm in our follow-up work [11].

†This research is supported in part by National Science Foundation award CNS 1059540. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies or the U.S. government.

1 Introduction

Much of the past work on asynchronous approximate Byzantine consensus has assumed *scalar* inputs at the nodes [3, 7]. Recent work has yielded approximate Byzantine consensus algorithms for the case when the input at each node is a d-dimensional vector, and the nodes must reach consensus on a vector in the convex hull of the input vectors at the fault-free nodes [8, 12]. The d-dimensional vectors can be equivalently viewed as *points* in the d-dimensional Euclidean space. Thus, the algorithms in [8, 12] require the fault-free nodes to decide on a point in the d-dimensional space. In this paper, we generalize the problem to allow the decision to be a *convex polytope* in the d-dimensional space, such that the decided polytope is within the convex hull of the input vectors at the fault-free nodes. We name this problem as *Byzantine convex consensus* (BCC), and present an asynchronous approximate BCC algorithm with optimal fault tolerance.

We consider Byzantine convex consensus (BCC) in an *asynchronous* system consisting of n nodes, of which at most f may be Byzantine faulty. The Byzantine faulty nodes may behave in an arbitrary fashion, and may collude with each other. Each node i has a d-dimensional vector of reals as its *input* x_i. All nodes can communicate with each other directly on reliable and FIFO (first-in first-out) channels. Thus, the underlying communication graph can be modeled as a *complete graph*, with the set of nodes being $V = \{1, 2, \cdots, n\}$. The impossibility of *exact* consensus in asynchronous systems [4] applies to BCC as well. Therefore, we consider the *Approximate BCC* problem with the following requirements:

- **Validity**: The *output* (or *decision*) at each fault-free node must be a convex polytope in the convex hull of the d-dimensional input vectors at the fault-free nodes. (In a degenerate case, the output polytope may simply be a single *point*.)
- **Approximate Agreement**: For any $\epsilon > 0$, the *Hausdorff distance* (defined below) between the output polytopes at any two fault-free nodes must be at most ϵ.

Ideally, the fault-free nodes should reach consensus on the largest possible convex polytope that satisfies the validity constraint. We present an optimal algorithm that agrees on a convex polytope that is as *large* as possible under adversarial conditions in our follow-up work [11]. The motivation behind reaching consensus on a convex polytope is that a solution to BCC is expected to also facilitate solutions to a large range of consensus problems (e.g., Byzantine vector consensus [8, 12], or convex function optimization over a convex hull of the inputs at fault-free nodes). Future work will explore these potential applications.

To simplify the presentation, we do not include a *termination* condition above. We instead prove that approximate agreement condition is *eventually* satisfied by our proposed algorithm (in addition to validity). However, we can augment the proposed algorithm using techniques similar to those in [1, 8, 12], to terminate within a finite number of rounds.

Definition 1 *For two convex polytopes h_1, h_2, the* Hausdorff distance *is defined as [6]*

$$\boldsymbol{d}_H(h_1, h_2) \quad = \quad \max \quad \{ \quad \max_{p_1 \in h_1} \ \min_{p_2 \in h_2} d(p_1, p_2), \qquad \max_{p_2 \in h_2} \ \min_{p_1 \in h_1} d(p_1, p_2) \quad \}$$

where $d(p, q)$ is the Euclidean distance between points p and q.

Lower Bound on n: As noted above, [8, 12] consider the problem of reaching approximate Byzantine consensus on a vector (or a point) in the convex hull of the d-dimensional input vectors at the fault-free nodes, and show that $n \geq (d+2)f+1$ is necessary. [9] generalizes the same lower bound to colorless tasks. The lower bound proof in [8, 12] also implies that $n \geq (d+2)f+1$ is necessary to ensure that BCC is solvable. We do not reproduce the lower bound proof here, but in the rest of the paper, we assume that $n \geq (d+2)f+1$, and also that $n \geq 2$ (because consensus is trivial when $n = 1$).

2 Preliminaries

Some notations introduced throughout the paper are summarized in Appendix A. In this section, we introduce operations $\mathcal{H}$, H_l, H, and a *reliable broadcast* primitive used later in the paper.

Definition 2 *Given a set of points X, $\mathcal{H}(X)$ is defined as the convex hull of the points in X.*

Definition 3 *Suppose that ν convex polytopes $h_1, h_2, \cdots, h_\nu$, and ν constants $c_1, c_2, \cdots, c_\nu$ are given such that (i) $0 \leq c_i \leq 1$ and $\sum_{i=1}^{\nu} c_i = 1$, and (ii) for $1 \leq i \leq \nu$, if $c_i \neq 0$, then $h_i \neq \emptyset$. Linear combination of these convex polytopes, $H_l(h_1, h_2, \cdots, h_\nu;\ c_1, c_2, \cdots, c_\nu)$, is defined as follows:*

- *Let $Q := \{i \mid c_i \neq 0,\ 1 \leq i \leq \nu\}$.*
- *$p \in H_l(h_1, h_2, \cdots, h_\nu;\ c_1, c_2, \cdots, c_\nu)$ if and only if*

$$\text{for each } i \in Q, \text{ there exists } p_i \in h_i, \quad \text{such that } \ p = \sum_{i \in Q} c_i p_i \tag{1}$$

Note that a convex polytope may possibly consist of a single point. Because h_i's above are all convex, $H_l(h_1, h_2, \cdots, h_\nu;\ c_1, c_2, \cdots, c_\nu)$ is also a convex polytope (proof included in Appendix B for completeness). The parameters for H_l consist of two lists, a list of polytopes $h_1, \cdots, h_\nu$, and a list of weights $c_1, \ldots, c_\nu$. With an abuse of notation, we will specify one or both of these lists as either a *row vector* or a *multiset*, with the understanding that the *row vector* or *multiset* here represents an ordered list of its elements.

Function H below is called in our algorithm with parameters $(\mathcal{V}, t)$ wherein t is a round index ($t \geq 0$) and $\mathcal{V}$ is a set of tuples of the form $(h, j, t-1)$, where j is a node identifier; when $t = 0$, h is a point in the d-dimensional Euclidean space, and when $t > 0$, h is a convex polytope.

Function $H(\mathcal{V}, t)$

1. Define multiset $X := \{h \mid (h, j, t-1) \in \mathcal{V}\}$. In our use of function H, each $h \in X$ is always non-empty.
2. If $t = 0$ then `temp` $:=$ $\cap_{C \subseteq X, |C| = |X| - f}\ \ \mathcal{H}(C)$.
   In our use of function H, when $t = 0$, each $h \in X$ is simply a point. The intersection above is over the convex hulls of the subsets of X of size $|X| - f$.
3. If $t > 0$ then `temp` $:=$ $H_l(X; \frac{1}{|X|}, \cdots, \frac{1}{|X|})$. Note that all the weights here are equal to $\frac{1}{|X|}$.
4. Return `temp`.

***Reliable Broadcast* Primitive:** We will use the *reliable broadcast* primitive from [1], which is also used in other related work [8, 12].

- As seen later, our algorithm proceeds in asynchronous rounds, and the nodes perform reliable broadcast of messages that each consist of a 3-tuple of the form (v, i, t): here i denotes the sender node's identifier, t is round index, and v is message value (the value v itself is often a tuple). The operation $\texttt{RBSend}(v, i, t)$ is used by node i to perform *reliable broadcast* of (v, i, t) in round t.

- When message (v, i, t) is *reliably received* by some node j, the event $\texttt{RBRecv}(v, i, t)$ is said to have occurred at node j (note that j may possibly be equal to i). The second element in a reliably received message 3-tuple, namely i above, is always identical to the identifier of the node that performed the corresponding reliable broadcast. When we say that node j reliably receives (v, i, t) we mean that event $\texttt{RBRecv}(v, i, t)$ occurs at node j.

Each fault-free node performs one reliable broadcast (RBSend) in each round of our algorithm. The reliable broadcast primitive has the properties listed below, as proved previously [1, 9].

- **Fault-Free Integrity:** If a fault-free node i *never* reliably broadcasts (v, i, t), then no fault-free node ever reliably receives (v, i, t).

- **Fault-Free Liveness:** If a fault-free node i performs reliable broadcast of (v, i, t), then each fault-free node eventually reliably receives (v, i, t).

- **Global Uniqueness:** If two fault-free nodes i, j reliably receive (v, k, t) and (w, k, t), respectively, then $v = w$, even if node k is faulty.

- **Global Liveness:** For any two fault-free nodes i, j, if i reliably receives (v, k, t), then j will eventually reliably receive (v, k, t), even if node k is faulty.

3 Proposed Algorithm: Verified Averaging

The proposed algorithm (named *Verified Averaging*) proceeds in asynchronous rounds. The input at each node i is a d-dimensional vector of reals, denoted as x_i. In each round t ($t \geq 0$), each node i computes a state variable h_i, which represents a convex polytope in the d-dimensional Euclidean space. We will refer to the value of h_i at the *end* of the t-th round performed by node i as $h_i[t]$, $t \geq 0$. Thus, for $t \geq 1$, $h_i[t-1]$ is the value of h_i at the *start* of the t-th round at node i.

Motivated by previous work that uses a mechanism to simulate omission failures in presence of Byzantine faults [2], our algorithm uses a similar technique, named *verification*. Informally, the *verification* mechanism ensures that if a faulty node deviates from the algorithm specification (except possibly choosing an invalid input vector), then its incorrect messages will be ignored by the fault-free nodes. Thus, aside from choosing a bad input, a faulty node cannot cause any other damage to the execution. Before we present the algorithm, we introduce a convention for the brevity of presentation:

- When we say that $(*, i, t) \in \mathcal{V}$, we mean that *there exists* z *such that* $(z, i, t) \in \mathcal{V}$.

- When we say that $(*, i, t) \notin \mathcal{V}$, we mean that $\forall z,\ (z, i, t) \notin \mathcal{V}$.

The proposed *Verified Averaging* algorithm for node $i \in V$ is presented below. All references to line numbers in our discussion refer to numbers listed on the right side of the algorithm pseudo-code. Whenever a message is reliably received by any node, a handler is called to process that message. Handlers for multiple reliably received messages may execute *concurrently* at a given node. For correct behavior, lines 3-7 below are executed **atomically**, and similarly, lines 11-16 are executed **atomically**.

In the proposed algorithm, in round 0, each node i uses `RBSend` to reliably broadcast $(x_i, i, 0)$ where x_i is its input (line 1).

Lines 2-7 specify the event handler for event $\texttt{RBRecv}(x, j, 0)$ at node i. Whenever a new message of the form $(x, j, 0)$ is reliably received by node i (line 2), the set $\mathit{Verified}_i[0]$ is updated (line 3). Note that the message received by node i on line 2 may possibly have been reliably broadcast by node i itself. When size of set $\mathit{Verified}_i[0]$ becomes at least $n - f$ for the first time (line 4), node i computes $h_i[0]$ (line 6); the *Verified* set used for computing $h_i[0]$ is saved as $\mathit{Verified}_i^c[0]$ (line 5). Having computed $h_i[0]$, node i can proceed to round 1 (line 7). Note that new messages may still be added to $\mathit{Verified}_i[0]$ afterwards whenever event of the form $\texttt{RBRecv}(x_j, j, 0)$ occurs. Thus, $\mathit{Verified}_i[0]$ may continue to grow even after node i has proceeded to round 1; however, $\mathit{Verified}_i^c[0]$ is not modified again. Recall that lines 3-7 are performed atomically.

On entering round t, $t \geq 1$, each node i reliably broadcasts $((h_i[t-1], \mathit{Verified}_i^c[t-1]), i, t)$ (line 8).

Lines 9-16 specify the event handler for event $\texttt{RBRecv}((h, \mathcal{V}), j, t)$ at node i. Whenever a message of the form $((h, \mathcal{V}), j, t)$ is reliably received from node j (line 10), node i first waits until its own set $\mathit{Verified}_i[t-1]$ becomes large enough to contain $\mathcal{V}$. Note that $\mathit{Verified}_i[t-1]$ is initially computed in round $t-1$, but it may continue to grow even after node i proceeds to round t. If the condition $\mathcal{V} \subseteq \mathit{Verified}_i[t-1]$ never becomes true, then this message is not processed further. The message $((h, \mathcal{V}), j, t)$ is considered correct if all the following conditions are true: (i) $\mathcal{V} \subseteq \mathit{Verified}_i[t-1]$, (ii) $|\mathcal{V}| \geq n - f$, (iii) $(*, j, t-2) \in \mathcal{V}$, and (iv) $h = H(\mathcal{V}, t-1)$. Conditions (ii), (iii) and (iv) are tested in line 11, and node i does not reach line 11 until condition (i) becomes true at line 10. If the message $((h, \mathcal{V}), j, t)$ is considered correct, then $(h, j, t-1)$ it is added to $\mathit{Verified}_i[t]$ (line 12).

When both conditions $|\mathit{Verified}_i[t]| \geq n - f$ and $(h_i[t-1], i, t-1) \in \mathit{Verified}_i[t]$ are true for the first time (line 13), node i computes $h_i[t]$ (lines 14 and 15), and then proceeds to round $t+1$. Similar to round 0, the set used in computing $h_i[t]$ is saved as $\mathit{Verified}_i^c[t]$ (line 14). While $\mathit{Verified}^c[t]$ remains unchanged afterwards, $\mathit{Verified}_i[t]$ may continue to grow, even after node i proceeds to round $t+1$, if new round t messages are reliably received later. Recall that lines 11-16 are performed atomically.

Verified Averaging Algorithm: Steps at node i

Initialization: All sets used below are initialized to $\emptyset$.

Round 0:

- $\texttt{RBSend}(x_i, i, 0)$ 1
- **Event handler for event** $\texttt{RBRecv}(x, j, 0)$ **at node** i : 2

  Lines 3-7 are performed atomically.

  - $\mathit{Verified}_i[0] := \mathit{Verified}_i[0] \cup \{(x, j, -1)\}$ 3

Comment: The third element of the 3-tuple added to *Verified* above is set as -1 to facilitate consistent treatment in round 0 and rounds $t > 0$.

4 – When $|\mathit{Verified}_i[0]| \geq n - f$ & $(x_i, i, -1) \in \mathit{Verified}_i[0]$ <u>both true for the first time</u>,

5 $\mathit{Verified}_i^c[0] := \mathit{Verified}_i[0]$

6 $h_i[0] := H(\mathit{Verified}_i^c[0], 0)$

7 Proceed to round 1

Round t $(t \geq 1)$**:**

- 8 `RBSend`$((h_i[t-1], \mathit{Verified}_i^c[t-1]), i, t)$
- 9 **Event handler for event** `RBRecv`$((h, \mathcal{V}), j, t)$ **at node** i :
  - 10 Wait until $\mathcal{V} \subseteq \mathit{Verified}_i[t-1]$

    <u>*Lines 11-16 are performed atomically.*</u>

  - 11 If $|\mathcal{V}| \geq n - f$ and $(*, j, t-2) \in \mathcal{V}$ and $h = H(\mathcal{V}, t-1)$
    12 then $\mathit{Verified}_i[t] := \mathit{Verified}_i[t] \cup \{(h, j, t-1)\}$
  - 13 When $|\mathit{Verified}_i[t]| \geq n - f$ & $(h_i[t-1], i, t-1) \in \mathit{Verified}_i[t]$ <u>both true for first time</u>
    14 $\mathit{Verified}_i^c[t] := \mathit{Verified}_i[t]$
    15 $h_i[t] := H(\mathit{Verified}_i^c[t], t)$
    16 Proceed to round $t+1$

Definition 4 *A node k's execution of round r, $r \geq 0$, is said to be* <u>**verified by a fault-free node** i</u> *if, eventually node i reliably receives message of the form $((h, \mathcal{V}), k, r+1)$ from node k, and subsequently adds (h, k, r) to $\mathit{Verified}_i[r+1]$. Note that node k may possibly be faulty. Node k's execution of round r is said to be* <u>**verified**</u> *if it is verified by at least one fault-free node.*

We now introduce some more notations (which are also summarized in Appendix A):

- For a *given* execution of the proposed *Verified Averaging* algorithm, let F denote the *actual* set of faulty nodes in the execution. Let $|F| = \phi$. Thus, $0 \leq \phi \leq f$.
- For $r \geq 0$, let $F_v[r]$ denote the set of faulty nodes whose round r execution is verified by at least one fault-free node, as per Definition 4. Note that $F_v[r] \subseteq F$.
- Define $\overline{F_v}[r] = F - F_v[r]$, for $r \geq 0$.

For each faulty node $k \in F_v[r]$, by Definition 4, there must exist a fault-free node i that eventually reliably receives a message of the form $((h, \mathcal{V}), k, r+1)$ from node k, and adds (h, k, r) to $\mathit{Verified}_i[r+1]$. Given these h and $\mathcal{V}$, for future reference, let us define

$$h_k[r] = h \tag{2}$$

$$\mathit{Verified}_k^c[r] = \mathcal{V} \tag{3}$$

Node i verifies node k's round r execution after node i has entered its round $r+1$. Since round r execution of faulty node k above is verified by fault-free node i, due to the check performed by node i at line 11, the equality below holds for $h_k[r]$ and $\mathit{Verified}^c_k[r]$ defined in (2) and (3).

$$h_k[r] \quad = \quad H(\mathit{Verified}^c_k[r], r) \tag{4}$$

(The proof of Claim 5 in Appendix E elaborates on the above equality.) While the algorithm requires each node k to maintain variables $h_k[r]$ and $\mathit{Verified}^c_k[r]$, we cannot assume correct behavior on the part of the faulty nodes. However, from the perspective of each fault-free node that verifies the round r execution of faulty node $k \in F_v[r]$, node k behaves "as if" these local variable take the values specified in (2) and (3) that satisfy (4). Note that if the round r execution of a faulty node k is verified by more than one fault-free node, due to the *Global Uniqueness* of reliable broadcast, all these fault-free nodes must have reliably received identical round $r+1$ messages from node k.

Proofs of Lemmas 1, 2 and 3 below are presented in Appendices D, F, and H, respectively. These lemmas are used to prove the correctness of the *Verified Averaging* algorithm.

Lemma 1 *If all the fault-free nodes progress to the start of round t, $t \geq 0$, then all the fault-free nodes will eventually progress to the start of round $t+1$.*

Lemma 2 *For each node $i \in V - \overline{F_v}[0]$, the polytope $h_i[0]$ is non-empty.*

Lemma 3 *For $r \geq 0$, if $b \in \overline{F_v}[r]$, then for all $\tau \geq r$,*

- *$b \in \overline{F_v}[\tau]$, and*
- *for all $i \in V - \overline{F_v}[\tau+1]$, $(*, b, \tau) \notin \mathit{Verified}^c_i[\tau+1]$.*

4 Correctness

We first introduce some terminology and definitions related to matrices. Then, we develop a *transition matrix* representation of the proposed algorithm, and use that to prove its correctness.

4.1 Matrix Preliminaries

We use boldface upper case letters to denote matrices, rows of matrices, and their elements. For instance, $\mathbf{A}$ denotes a matrix, $\mathbf{A}_i$ denotes the i-th row of matrix $\mathbf{A}$, and $\mathbf{A}_{ij}$ denotes the element at the intersection of the i-th row and the j-th column of matrix $\mathbf{A}$.

Definition 5 *A vector is said to be stochastic if all its elements are non-negative, and the elements add up to 1. A matrix is said to be row stochastic if each row of the matrix is a stochastic vector.*

For matrix products, we adopt the "backward" product convention below, where $a \leq b$,

$$\Pi_{\tau=a}^{b} \mathbf{A}[\tau] = \mathbf{A}[b]\mathbf{A}[b-1]\cdots\mathbf{A}[a] \tag{5}$$

For a row stochastic matrix $\mathbf{A}$, coefficients of ergodicity $\delta(\mathbf{A})$ and $\lambda(\mathbf{A})$ are defined as follows [13]:

$$\begin{aligned} \delta(\mathbf{A}) &= \max_j \max_{i_1,i_2} \|\mathbf{A}_{i_1 j} - \mathbf{A}_{i_2 j}\| \\ \lambda(\mathbf{A}) &= 1 - \min_{i_1,i_2} \sum_j \min(\mathbf{A}_{i_1 j}, \mathbf{A}_{i_2 j}) \end{aligned}$$

Claim 1 *For any p square row stochastic matrices $\mathbf{A}(1), \mathbf{A}(2), \ldots, \mathbf{A}(p)$,*

$$\delta(\Pi_{\tau=1}^{p} \mathbf{A}(\tau)) \ \leq \ \Pi_{\tau=1}^{p} \ \lambda(\mathbf{A}(\tau)).$$

Claim 1 is proved in [5]. Claim 2 below follows directly from the definition of $\lambda(\cdot)$.

Claim 2 *If there exists a constant γ, where $0 < \gamma \leq 1$, such that, for any pair of rows i, j of matrix $\mathbf{A}$, there exists a column g (that may depend on i, j) such that, $\min(\mathbf{A}_{ig}, \mathbf{A}_{jg}) \geq \gamma$, then $\lambda(\mathbf{A}) \leq 1 - \gamma < 1$.*

Let $\mathbf{v}$ be a column vector with n elements, such that the i-th element of vector $\mathbf{v}$, namely $\mathbf{v}_i$, is a convex polytope in the d-dimensional Euclidean space. Let $\mathbf{A}$ be a $n \times n$ row stochastic square matrix. Then multiplication of matrix $\mathbf{A}$ and vector $\mathbf{v}$ is performed by multiplying each row of $\mathbf{A}$ with column vector $\mathbf{v}$ of polytopes. Formally,

$$\mathbf{A}\mathbf{v} = [H_l(\mathbf{v}^T; \mathbf{A}_1) \qquad H_l(\mathbf{v}^T; \mathbf{A}_2) \qquad \ldots \qquad H_l(\mathbf{v}^T; \mathbf{A}_n)]^T \tag{6}$$

where ^{T} denotes the transpose operation (thus, $\mathbf{v}^T$ is the transpose of $\mathbf{v}$). H_l is defined in Definition 3. Thus, the result of the multiplication $\mathbf{A}\mathbf{v}$ is a column vector consisting of n convex polytopes. Similarly, product of row vector $\mathbf{A}_i$ and above vector $\mathbf{v}$ is obtained as follows, and it is a polytope.

$$\mathbf{A}_i\mathbf{v} \ = \ H_l(\mathbf{v}^T \ ; \ \mathbf{A}_i) \tag{7}$$

4.2 Transition Matrix Representation of *Verified Averaging*

Let $\mathbf{v}[t]$, $t \geq 0$, denote a column vector of length $|V| = n$. In the remaining discussion, we will refer to $\mathbf{v}[t]$ be the state of the system at the end of round t. In particular, $\mathbf{v}_i[t]$ for $i \in V$ is viewed as the state of node i at the end of round t. We define $\mathbf{v}[0]$ as follows:

(I1) For each fault-free node $i \in V - F$, $\mathbf{v}_i[0] := h_i[0]$.

(I2) For each faulty node $k \in F_v[0]$, $\mathbf{v}_k[0] := h_k[0]$, where $h_k[0]$ is defined in (2).

(I3) For each faulty node $k \in \overline{F_v}[0]$, $\mathbf{v}_k[0]$ is *arbitrarily* defined as the origin, or the all-0 vector. We will justify this arbitrary choice later.

We will show that the state evolution can be represented in a matrix form as in (8), for a suitably chosen $n \times n$ matrix $\mathbf{M}[t]$. $\mathbf{M}[t]$ is said to be the *transition matrix* for round t.

$$\mathbf{v}[t] = \mathbf{M}[t] \ \mathbf{v}[t-1], \qquad t \geq 1 \tag{8}$$

For all $t \geq 0$, Theorem 1 below proves that, for each $i \in V - \overline{F_v}[t]$, $h_i[t] = \mathbf{v}_i[t]$.

Given a particular execution of the algorithm, we construct the transition matrix $\mathbf{M}[t]$ for round $t \geq 1$ using the following procedure.

Construction of the Transition Matrix for Round t $(t \geq 1)$

- For each node $i \in V - \overline{F_v}[t]$, and each $k \in V$:

If $(h_k[t-1], k, t-1) \in \mathit{Verified}_i^c[t]$, then

$$\mathbf{M}_{ik}[t] = \frac{1}{|\,\mathit{Verified}_i^c[t]|} \tag{9}$$

Otherwise,

$$\mathbf{M}_{ik}[t] = 0 \tag{10}$$

Comment: For a faulty node $i \in F_v[t]$, $h_i[t]$ and $\mathit{Verified}_i^c[t]$ are defined in (2) and (3).

- For each node $j \in \overline{F_v}[t]$, and each $k \in V$,

$$\mathbf{M}_{jk}[t] \;\; = \;\; \frac{1}{n} \tag{11}$$

Theorem 1 *For $r \geq 0$, with state evolution specified as $\mathbf{v}[r+1] = \mathbf{M}[r+1]\mathbf{v}[r]$ using $\mathbf{M}[r+1]$ constructed above, for all $i \in V - \overline{F_v}[r]$, (i) $h_i[r]$ is non-empty, and (ii) $h_i[r] = \mathbf{v}_i[r]$.*

Proof:

The proof of the theorem is by induction. The theorem holds for $r = 0$ due to Lemma 2, and the choice of the elements of $\mathbf{v}[0]$, as specified in (I1), (I2) and (I3) above.

Now, suppose that the theorem holds for $r = t-1$ where $t-1 \geq 0$, and prove it for $r = t$. Thus, by induction hypothesis, for all $i \in V - \overline{F_v}[t-1]$, $h_i[t-1] = \mathbf{v}_i[t-1] \neq \emptyset$. Now, $\mathbf{v}[t] = \mathbf{M}[t]\mathbf{v}[t-1]$.

- In round $t \geq 1$, each fault-free node $i \in V - F$ computes its new state $h_i[t]$ at line 15 using function $H(\mathit{Verified}_i^c[t], t)$. The function $H(\mathit{Verified}_i^c[t], t)$ for $t \geq 1$ then computes a linear combination of $|\,\mathit{Verified}_i^c[t]|$ convex hulls, with all the weights being equal to $\frac{1}{|\,\mathit{Verified}_i^c[t]|}$. Also, by Definition 4 and the definition of $\overline{F_v}[t-1]$, if $(h, j, t-1) \in \mathit{Verified}_i^c[t]$, then $j \notin \overline{F_v}[t-1]$ (i.e., $j \in V - \overline{F_v}[t-1]$). Therefore, if $(h, j, t-1) \in \mathit{Verified}_i^c[t]$, then either j is fault-free, or it is faulty and its round $t-1$ execution is verified: thus, $h = h_j[t-1]$.

  Also, by induction hypothesis, $h = h_j[t-1] \neq \emptyset$. This implies that $h_i[t] = H(\mathit{Verified}_i^c[t], t)$ is non-empty.

  Then observe that, by defining $\mathbf{M}_{ik}[t]$ elements as in (9) and (10), we ensure that $\mathbf{M}_i[t]\mathbf{v}[t-1]$ equals $H(\mathit{Verified}_i^c[t], t)$, and hence equals $h_i[t]$.

- For $i \in F_v[t]$ as well, as shown in (4), $h_i[t] = H(\mathit{Verified}_i^c[t], t)$, where $h_i[t]$ and $\mathit{Verified}_i^c[t]$ are as defined in (2) and (3). The function $H(\mathit{Verified}_i^c[t], t)$ for $t \geq 1$ then computes a linear combination of $|\,\mathit{Verified}_i^c[t]|$ convex hulls, with all the weights being equal to $\frac{1}{|\,\mathit{Verified}_i^c[t]|}$. Consider an element $(h, j, t-1)$ in $\mathit{Verified}_i^c[t]$. We argue that $j \in V - \overline{F_v}[t-1]$. Suppose this is not true, i.e., $j \in \overline{F_v}[t-1]$. By Definition 4, node i's round t execution is verified by some fault-free node k, which implies that eventually, $\mathit{Verified}_i^c[t] \subseteq \mathit{Verified}_k[t]$. However, since k is fault-free, and $(h, j, t-1) \notin \mathit{Verified}_k[t]$, a contradiction. Hence, if $(h, j, t-1) \in \mathit{Verified}_i^c[t]$, then $j \in V - \overline{F_v}[t-1]$. That is, if $(h, j, t-1) \in \mathit{Verified}_i^c[t]$, then either j is fault-free, or it is faulty and its round $t-1$ execution is verified: thus, $h = h_j[t-1]$.

  Also, by induction hypothesis, $h = h_j[t-1] \neq \emptyset$. This implies that $h_i[t] = H(\mathit{Verified}_i^c[t], t)$ is non-empty.

Then observe that, by defining $\mathbf{M}_{ik}[t]$ elements as in (9) and (10), we ensure that $\mathbf{M}_i[t]\mathbf{v}[t-1]$ equals $H(\mathit{Verified}_i^c[t], t)$, and hence equals $h_i[t]$.

□

Now, we argue that for $t \geq 0$, the state $\mathbf{v}_j[t]$ for each node $j \in \overline{F_v}[t]$ does not affect the state of the nodes $V - \overline{F_v}[\tau]$, for $\tau \geq t+1$. From the discussion in the above proof, we see that for $j \in \overline{F_v}[t]$, $(*, j, t) \notin \mathit{Verified}_i^c[t+1]$ for $i \in V - \overline{F_v}[t+1]$. Thus, the sate $\mathbf{v}_j[t]$ does not affect the state $h_i[t+1]$. Then, by Lemma 3, if $j \in \overline{F_v}[t]$, then $j \in \overline{F_v}[\tau]$, for $\tau \geq t+1$. Thus, by the same argument, the sate $\mathbf{v}_j[\tau]$ does not affect the state $h_i[t+1]$. This justifies the somewhat arbitrary choice of $\mathbf{v}_j[0]$ for $j \in \overline{F_v}[0]$, and $\mathbf{M}_{jk}[t]$ in (11) for $j \in \overline{F_v}[t],\ t \geq 1$. This choice does simplify the remaining proof somewhat.

The above discussion shows that, for $t \geq 1$, the evolution of $\mathbf{v}[t]$ can be written as in (8), that is, $\mathbf{v}[t] = \mathbf{M}[t]\mathbf{v}[t-1]$. Given the matrix product definition in (6), it is easy to verify that

$$\mathbf{M}[\tau+1]\ \left(\mathbf{M}[\tau]\mathbf{v}[\tau-1]\right) \ = \ \left(\mathbf{M}[\tau+1]\mathbf{M}[\tau]\right)\ \mathbf{v}[\tau-1] \ \text{ for } \tau \geq 1.$$

Therefore, by repeated application of (8), we obtain:

$$\mathbf{v}[t] \ = \ \left(\Pi_{\tau=1}^{t}\mathbf{M}[\tau]\right)\mathbf{v}[0], \qquad t \geq 1 \tag{12}$$

Recall that we adopt the "backward" matrix product convention presented in (5).

Lemma 4 *For $t \geq 1$, transition matrix $\mathbf{M}[t]$ constructed using the above procedure satisfies the following conditions.*

- *For $i, j \in V$, there exists a fault-free node $g(i,j)$ such that $\mathbf{M}_{ig(i,j)}[t] \geq \frac{1}{n}$.*
- *$\mathbf{M}[t]$ is a row stochastic matrix, and $\lambda(\mathbf{M}[t]) \leq 1 - \frac{1}{n}$.*

The proof of Lemma 4 is presented in Appendix J.

4.3 Correctness of *Verified Averaging*

Definition 6 *A convex polytope h is said to be* valid *if every point in h is in the convex hull of the inputs at the fault-free nodes.*

Lemmas 5 and 6 below are proved in Appendices K and L, respectively.

Lemma 5 *$h_i[0]$ for each node $i \in V - \overline{F_v}[0]$ is valid.*

Lemma 6 *Suppose non-empty convex polytopes $h_1, h_2, \cdots, h_k$ are all valid. Consider k constants $c_1, c_2, \cdots, c_k$ such that $0 \leq c_i \leq 1$ and $\sum_{i=1}^{k} c_i = 1$. Then the linear combination of these convex polytopes, $H_l(h_1, h_2, \cdots, h_k; c_1, c_2, \cdots, c_k)$, is valid.*

Theorem 2 *Verified Averaging satisfies the* validity *and* approximate agreement *properties after a large enough number of asynchronous rounds.*

Proof: Repeated applications of Lemma 1 ensures that the fault-free nodes will progress from round 0 through round r, for any $r \geq 0$, allowing us to use (12). Consider round $t \geq 1$. Let

$$\mathbf{M}^* \;=\; \Pi_{\tau=1}^{t}\mathbf{M}[\tau]. \tag{13}$$

(To simplify the presentation, we do not include the round index $[t]$ in the notation $\mathbf{M}^*$ above.) Then $\mathbf{v}[t] = \mathbf{M}^*\mathbf{v}[0]$. By Lemma 4, each $\mathbf{M}[t]$ is a *row stochastic* matrix, therefore, $\mathbf{M}^*$ is also row stochastic. By Lemma 5, $h_i[0] = \mathbf{v}_i[0]$ for each $i \in V - \overline{F_v}[0]$ is valid. Therefore, by Lemma 6, $\mathbf{M}^*_i\mathbf{v}[0]$ for each $i \in V - F$ is valid. Also, by Theorem 1 and (12), $h_i[t] = \mathbf{M}^*_i\mathbf{v}[0]$ for $i \in V - F$. Thus, $h_i[t]$ is valid, and *Verified Averaging* satisfies the validity condition for all $t \geq 0$.

Let us define $\alpha = 1 - \frac{1}{n}$. By Lemma 4, $\lambda(\mathbf{M}[t]) \leq 1 - \frac{1}{n} = \alpha$. Then by Claim 1,

$$\delta(\mathbf{M}^*) = \delta(\Pi_{\tau=1}^{t}\mathbf{M}[\tau]) \;\leq\; \lim_{t\to\infty}\Pi_{\tau=1}^{t}\lambda(\mathbf{M}[\tau]) \;\leq\; \left(1 - \frac{1}{n}\right)^t \;=\; \alpha^t \tag{14}$$

Consider any two fault-free nodes $i, j \in V - F$. By (14), $\delta(\mathbf{M}^*) \leq \alpha^t$. Therefore, by the definition of $\delta(\cdot)$, for $1 \leq k \leq n$,

$$\|\mathbf{M}^*_{ik} - \mathbf{M}^*_{jk}\| \leq \alpha^t \tag{15}$$

By Lemma 3, and construction of the transition matrices, it should be easy to see that $\mathbf{M}^*_{ib} = 0$ for $b \in \overline{F_v}[0]$. Then, for any point p^*_i in $h_i[t] = \mathbf{M}^*_i\mathbf{v}[0]$, there must exist, for all $k \in V - \overline{F_v}[0]$, $p_k \in h_k[0]$, such that

$$p^*_i = \sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{ik}p_k = \left(\sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{ik}p_k(1), \quad \sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{ik}p_k(2), \cdots, \quad \sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{ik}p_k(d)\right) \tag{16}$$

where $p_k(l)$ denotes the value of p_k's l-th coordinate. Now choose point p^*_j in $h_j[t]$ defined as follows.

$$p^*_j = \sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{jk}p_k = \left(\sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{jk}p_k(1), \quad \sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{jk}p_k(2), \cdots, \quad \sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{jk}p_k(d)\right) \tag{17}$$

Then the Euclidean distance between p^*_i and p^*_j is $d(p^*_i, p^*_j)$. The following derivation is obtained by simple algebraic manipulation, using (15), (16) and (17). The omitted steps in the algebraic manipulation are shown in Appendix M.

$$d(p^*_i, p^*_j) = \sqrt{\sum_{l=1}^{d}(p^*_i(l) - p^*_j(l))^2} = \sqrt{\sum_{l=1}^{d}\left(\sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{ik}p_k(l) - \sum_{k\in V-\overline{F_v}[0]} \mathbf{M}^*_{jk}p_k(l)\right)^2}$$
$$\leq \alpha^t\sqrt{\sum_{l=1}^{d}\left(\sum_{k\in V-\overline{F_v}[0]} \|p_k(l)\|\right)^2} \;\leq\; \alpha^t\Omega \tag{18}$$

where $\Omega = \max_{p_k\in h_k[0], k\in V-\overline{F_v}[0]} \sqrt{\sum_{l=1}^{d}(\sum_{k\in V-\overline{F_v}[0]}\|p_k(l)\|)^2}$. Because the $h_k[0]$'s in the definition of Ω are all valid (by Lemma 5), Ω can itself be upper bounded by a function of the input vectors at the fault-free nodes. Since $\alpha = 1 - \frac{1}{n} < 1$, for large enough t, $\alpha^t\Omega < \epsilon$, for any given ϵ. Therefore, for fault-free i, j, for large enough t, for each point $p^*_i \in h_i[t]$ there exists a point $p^*_j[t] \in h_j[t]$ such that $d(p^*_i, p^*_j) < \epsilon$ (and, similarly, vice-versa). Thus, by Definition 1, eventually Hausdorff distance $\mathbf{d}_H(h_i[t], h_j[t]) < \epsilon$. Since this holds true for any pair of fault-free nodes i, j, approximate agreement property is eventually satisfied. □

5 Convex Polytope Obtained by *Verified Averaging*

Recall that $|F| = \phi \leq f$. Let $G = \cup_{i \in \{V-F\}} x_i$ be the set of the inputs at all fault-free nodes. Thus, $|G| = n - \phi \geq n - f$. Define a convex polytope I as follows.

$$I = \cap_{D \subset G, |D| = n-2f-\phi} \mathcal{H}(D) \tag{19}$$

Lemma 7 *For all $i \in V - F$ and $t \geq 0$, $I \subseteq h_i[t]$.*

The lemma is proved in Appendix N. The lemma establishes a "lower bound" on the convex polytope that the fault-free nodes decide on. Due to Theorem 1, $h_i[t]$ is always non-empty for $i \in V - F$. However, I may possibly be empty, depending on the inputs at the fault-free nodes, and the total number of nodes. We believe that it may be possible to improve the above "lower bound" by using a somewhat more complex algorithm (using the *stable-vectors* primitive from [9] in round 0). This improvement is left as a topic for future work. In the follow-up work [11], we present an optimal algorithm that agrees on a convex polytope that is as *large* as possible under adversarial conditions

6 Summary

This paper addresses *Byzantine Convex Consensus* (BCC), wheresin each node has a d-dimensional vector as its *input*, and each fault-free node has decide on a polytope that is in the *convex hull* of the input vectors at the fault-free nodes. We present an *asynchronous approximate* BCC algorithm with optimal fault tolerance, and establish a lower bound on the convex polytope agreed upon.

References

[1] I. Abraham, Y. Amit, and D. Dolev. Optimal resilience asynchronous approximate agreement. In *OPODIS*, pages 229–239, 2004.

[2] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*. Wiley Series on Parallel and Distributed Computing, 2004.

[3] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl. Reaching approximate agreement in the presence of faults. *J. ACM*, 33:499–516, May 1986.

[4] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32:374–382, April 1985.

[5] J. Hajnal. Weak ergodicity in non-homogeneous markov chains. In *Proceedings of the Cambridge Philosophical Society*, volume 54, pages 233–246, 1958.

[6] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[7] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[8] H. Mendes and M. Herlihy. Multidimensional approximate agreement in byzantine asynchronous systems. In *ACM Symposium on Theory of Computing (STOC)*, 2013.

[9] H. Mendes, C. Tasson, and M. Herlihy. The topology of asynchronous byzantine colorless tasks. *CoRR*, abs/1302.6224, 2013.

[10] M. A. Perles and M. Sigorn. A generalization of Tverberg's theorem. *CoRR*, abs/0710.4668, 2007.

[11] L. Tseng and N. H. Vaidya. Byzantine Convex Consensus: An Optimal Algorithm. Tech. Report, CSL UIUC, 2013. http://www.crhc.illinois.edu/wireless/papers/BCC_optimal_tech.pdf.

[12] N. H. Vaidya and V. K. Garg. Byzantine vector consensus in complete graphs. *CoRR*, abs/1302.2543, 2013. To appear at ACM PODC 2013.

[13] J. Wolfowitz. Products of indecomposable, aperiodic, stochastic matrices. In *Proceedings of the American Mathematical Society*, volume 14, pages 733–737, 1963.

A Notations

This appendix summarizes some of the notations and terminology introduced throughout the paper.

- n = number of nodes. We assume that $n \geq 2$.
- f = maximum number of Byzantine nodes.
- $V = \{1, 2, \cdots, n\}$ is the set of all nodes.
- d = dimension of the input vector at each node.
- $d(p, q)$ = the function returns the Euclidean distance between points p and q.
- $d_H(h_1, h_2)$ = the Hausdorff distance between convex polytopes h_1, h_2.
- $\mathcal{H}(C)$ = the convex hull of a multiset C.
- $H_l(h_1, h_2, \cdots, h_k;\ c_1, c_2, \cdots, c_k)$, defined in Section 2, is a linear combination of convex polytopes $h_1, h_2, ..., h_k$ with weights $c_1, c_2, \cdots, c_k$.
- $H(\mathcal{V}, t)$ is a function defined in Section 2.
- $|X|$ = the size of a *multiset* or *set* X.
- $\|a\|$ = the absolute value of a real number a.
- F denotes the *actual* set of faulty nodes in an execution of the algorithm.
- $\phi = |F|$. Thus, $0 \leq \phi \leq f$.
- $F_v[t]$, $t \geq 0$, denotes the set of faulty nodes whose round t execution is verified by at least one fault-free node, as per Definition 4.
- $\overline{F_v}[t] = F - F_v[t]$, $t \geq 0$.
- $\alpha = 1 - \frac{1}{n}$.
- We use boldface upper case letters to denote matrices, rows of matrices, and their elements. For instance, $\mathbf{A}$ denotes a matrix, $\mathbf{A}_i$ denotes the i-th row of matrix $\mathbf{A}$, and $\mathbf{A}_{ij}$ denotes the element at the intersection of the i-th row and the j-th column of matrix $\mathbf{A}$.

B $H_l(h_1, h_2, \cdots, h_\nu;\ c_1, c_2, \cdots, c_\nu)$ is Convex

Claim 3 *$H_l(h_1, h_2, \cdots, h_\nu;\ c_1, c_2, \cdots, c_\nu)$ defined in Definition 3 is convex.*

Proof:

The proof is straightforward.

Let

$$h_L := H_l(h_1, h_2, \cdots, h_\nu;\ c_1, c_2, \cdots, c_\nu)$$

and

$$Q := \{i \mid c_i \neq 0,\ 1 \leq i \leq \nu\}.$$

Given any two points x, y in h_L, by Definition 3, we have

$$x = \sum_{i \in Q} c_i p_{(i,x)} \quad \text{for some } p_{(i,x)} \in h_i, \quad i \in Q \tag{20}$$

and

$$y = \sum_{i \in Q} c_i p_{(i,y)} \quad \text{for some } p_{(i,y)} \in h_i, \quad i \in Q \tag{21}$$

Now, we show that any convex combination of x and y is also in h_L. Consider a point z such that

$$z = \theta x + (1-\theta) y \quad \text{where } 0 \le \theta \le 1 \tag{22}$$

Substituting (20) and (21) into (22), we have

$$\begin{aligned} z &= \theta \sum_{i \in Q} c_i p_{(i,x)} + (1-\theta) \sum_{i \in Q} c_i p_{(i,y)} \\ &= \sum_{i \in Q} c_i \left(\theta p_{(i,x)} + (1-\theta) p_{(i,y)} \right) \end{aligned} \tag{23}$$

Define $p_{(i,z)} = \theta p_{(i,x)} + (1-\theta) p_{(i,y)}$ for all $i \in Q$. Since h_i is convex, and $p_{(i,z)}$ is a convex combination of $p_{(i,x)}$ and $p_{(i,y)}$, $p_{(i,z)}$ is also in h_i. Substituting the definition of $p_{(i,z)}$ in (23), we have

$$z = \sum_{i \in Q} c_i \; p_{(i,z)} \quad \text{where } p_{(i,z)} \in h_i, \quad i \in Q$$

Hence, by Definition 3, z is also in h_L. Therefore, h_L is convex.

□

C Claim 4

Claim 4 *Consider fault-free nodes $i, j \in V - F$. For $t \ge 0$, if $(h, k, t-1) \in Verified_i[t]$ at some point of time, then eventually $(h, k, t-1) \in Verified_j[t]$.*

Proof: The proof is by induction.

Induction basis: For round $t = 0$, node i adds $(h, k, -1)$ to $Verified_i[0]$ whenever it reliably receives message $(h, k, 0)$. (For round 0 messages, h is just a single point.) Then by *Global Liveness* property of reliable broadcast, node j will eventually reliably receive the same message, and add $(h, k, -1)$ to $Verified_j[0]$.

Induction: Consider round $t \ge 1$. Assume that the statement of the lemma holds true through rounds $t-1$. Therefore, if $(h, k, t-2) \in Verified_i[t-1]$ at some point of time, then eventually $(h, k, t-2) \in Verified_j[t-1]$.

Now we will prove that the lemma holds for round t. Suppose that at some time μ, $(h, k, t-1) \in Verified_i[t]$. Thus, node i must have reliably received (in line 9 of round t) a message of the form $((h, \mathcal{V}), k, t)$ such that the following conditions are true at some real time time μ:

- Condition 1: $\mathcal{V} \subseteq \mathit{Verified}_i[t-1]$ (due to line 10, and the fact that $\mathit{Verified}_i[t-1]$ can only grow with time)
- Condition 2: $|\mathcal{V}| \geq n - f$, $(*, k, t-2) \in \mathcal{V}$ and $h = H(\mathcal{V}, t-1)$ (due to line 11)

The correctness of the lemma through round $t-1$ implies that eventually each element of $\mathit{Verified}_i[t-1]$ will be included in $\mathit{Verified}_j[t-1]$. Thus, because $\mathcal{V} \subseteq \mathit{Verified}_i[t-1]$ at time μ, eventually $\mathcal{V} \subseteq \mathit{Verified}_j[t-1]$. Also, the *Global Liveness* property implies that eventually node j will reliably receive the same message $((h, \mathcal{V}), k, t)$ that was reliably received by node i; therefore, as in Condition 2 above, node j will also find that $|\mathcal{V}| \geq n-f$, $(*, k, t-2) \in \mathcal{V}$ and $h = H(\mathcal{V}, t-1)$. Therefore, by lines 10-12, it follows that eventually $(h, k, t-1) \in \mathit{Verified}_j[t]$. □

D Proof of Lemma 1

Lemma 1: *If all the fault-free nodes progress to the start of round t, $t \geq 0$, then all the fault-free nodes will eventually progress to the start of round $t+1$.*

Proof: The proof is by induction. By assumption, all nodes begin round 0 eventually, and perform reliable broadcast of their input (line 1). By *Fault-Free Liveness* property of *reliable broadcast*, each fault-free node i will eventually reliably receive messages from all the $n-f$ fault-free nodes. All the messages reliably received in round 0 result in addition of an element to the set $\mathit{Verified}_i[0]$ at fault-free node i (line 3); therefore, $\mathit{Verified}_i[0]$ will eventually be of size at least $n-f$. It follows that each fault-free node will eventually complete round 0, and proceed to round 1 (lines 4-7).

Now we assume that all the fault-free nodes have progressed to the start of round t, where $t \geq 1$, and prove that all the fault-free nodes will eventually progress to the start of round $t+1$.

Consider fault-free nodes $i, j \in V - F$. In line 8 of round t, fault-free node j performs reliable broadcast of $((h_j[t-1], \mathit{Verified}^c_j[t-1]), j, t)$. By *Fault-free Liveness* of reliable broadcast, fault-free node i will eventually reliably receive message $((h_j[t-1], \mathit{Verified}^c_j[t-1]), j, t)$ from fault-free node j. By Claim 4, eventually $\mathit{Verified}^c_j[t-1] \subseteq \mathit{Verified}_i[t-1]$; therefore, node i will progress past line 10 in the handler for message $((h_j[t-1], \mathit{Verified}^c_j[t-1]), j, t)$. Moreover, since node j is fault-free, it follows the algorithm specification correctly. Therefore, the checks on line 11 in the handler at node i for message $((h_j[t-1], \mathit{Verified}^c_j[t-1]), j, t)$ will all be correct. Therefore, by lines 11-12, node i will eventually include $(h_j[t-1], j, t-1)$ in $\mathit{Verified}_i[t]$. Since the above argument holds for all fault-free nodes i, j, it implies that each fault-free node i eventually adds $(h_j[t-1], j, t-1)$ to $\mathit{Verified}_i[t]$, for each the fault-free node j (including $j = i$). Therefore, at each fault-free node i, eventually, $|\mathit{Verified}_i[t]| \geq n-f$, and $(h_i[t-1], i, t-1) \in \mathit{Verified}_i[t]$ (because the previous statement holds for $j = i$ too), thus satisfying both the conditions at line 13. Thus, each fault-free node i will eventually proceed to round $t+1$ (lines 13-16).

□

E Claims 5 and 6

Claim 5 *If faulty node i's round t execution is verified by a fault-free node j, then the following statements hold:*

(i) For $t \geq 0$, $\mathit{Verified}^c_i[t] \geq n-f$ and $h_i[t] = H(\mathit{Verified}^c_i[t], t)$,

(ii) For $t \geq 0$, eventually $\mathit{Verified}^c_i[t] \subseteq \mathit{Verified}_j[t]$, and

(iii) For $t \geq 1$, node i's round $t-1$ execution is also verified by node j.

Proof: Let $t \geq 0$. Suppose that node i's round t execution is verified by a fault-free node j. In this case, we can use definitions (2) and (3) of $h_i[t]$ and $Verified_i^c[t]$. Definition 4 implies that node j eventually reliably receives message $((h_i[t], Verified_i^c[t]), i, t+1)$ from node i, and subsequently adds (at line 12 of its round $t+1$) $(h_i[t], i, t)$ to $Verified_j[t+1]$. This implies that this message satisfies the checks done by node j at line 11: Specifially, (a) $|Verified_i^c[t]| \geq n-f$ and $h_i[t] = H(Verified_i^c[t], t)$, and (b) $(*, i, t-1) \in Verified_i^c[t]$. Also, by the time node j adds $(h_i[t], i, t+1)$ to $Verified_j[t+1]$, the condition on line 10 also holds: specifically, $Verified_i^c[t] \subseteq Verified_j[t]$, proving claim (ii) stated above. Also, (a) above proves claim (i).

$(*, i, t-1) \in Verified_i^c[t]$ and $Verified_i^c[t] \subseteq Verified_j[t]$ together imply that eventually $(*, i, t-1) \in Verified_j[t]$.

Suppose that $t \geq 1$. Then the above observation that eventually $(*, i, t-1) \in Verified_j[t]$, and Definition 4, imply that round $t-1$ execution of node i is verified by node j. This proves (iii). □

Claim 6 *If faulty node i's round t execution is verified by a fault-free node j, $t \geq 0$, then for all r such that $0 \leq r \leq t$, node i's round r execution is verified by node j.*

Proof: The claim is trivially true for $t = 0$. The proof of the claim for $t > 0$ follows by repeated application of Claim 5(iii) above. □

F Proof of Lemma 2

The proof of Lemma 2 uses the following theorem by Tverberg [10]:

Theorem 3 *(Tverberg's Theorem [10]) For any integer $f \geq 1$, for every multiset Y containing at least $(d+1)f+1$ points in a d-dimensional space, there exists a partition $Y_1, .., Y_{f+1}$ of Y into $f+1$ non-empty multisets such that $\cap_{l=1}^{f+1} \mathcal{H}(Y_l) \neq \emptyset$.*

Now we prove Lemma 2.

Lemma 2: *For each node $i \in V - \overline{F_v}[0]$, the polytope $h_i[0]$ is non-empty.*

Proof: Note that $V - \overline{F_v}[0] = (V - F) \cup F_v[0]$.

- For a fault-free node $i \in V - F$, since it behaves correctly, $|Verified_i^c[0]| \geq n - f$ and $h_i[0] = H(Verified_i^c[0], 0)$, due to lines 4-7.
- For faulty node $i \in F_v[0]$ as well, by Claim 5(i) in Appendix E, $|Verified_i^c[0]| \geq n - f$ and $h_i[0] = H(Verified_i^c[0], 0)$.

Thus, for each $i \in V - \overline{F_v}[0]$, $|Verified_i^c[0]| \geq n - f$ and $h_i[0] = H(Verified_i^c[0], 0)$.

Consider any $i \in V - \overline{F_v}[0]$. Consider the computation of polytope $h_i[0]$ as $H(Verified_i^c[0], 0)$. By step 1 of function H in Section 2, $|X| = |Verified_i^c[0]| \geq n - f$. Recall that, due to the lower bound on n discussed in Section 1, we assume $n \geq (d+2)f+1$. Thus, in function H, $|X| \geq n-f \geq (d+1)f+1$. By Theorem 3, there exists a partition $X_1, X_2, \cdots, X_{f+1}$ of X into multisets X_j such that $\cap_{j=1}^{f+1} \mathcal{H}(X_j) \neq \emptyset$. Let us define

$$J = \cap_{i=1}^{f+1} \mathcal{H}(X_j) \tag{24}$$

Thus, J is non-empty. In step 2 of function H (for $t=0$), because $|X| \geq n-f$, each multiset C used in the computation of function H is of size at least $n-2f$. Thus, each C excludes only f elements of X, whereas there are $f+1$ multisets in the above partition of X. Therefore, each set C in step 2 of function H will fully contain at least one multiset X_j from the partition. Therefore, $\mathcal{H}(C)$ will contain J. Since this holds true for all C's, J is contained in the convex polytope computed by function H. Since J is non-empty, $h_i[0] = H(\mathit{Verified}_i^c[0], 0)$ is non-empty.

□

G Claim 7

Claim 7 *For $t \geq 0$, if $b \in \overline{F_v}[t]$, then for all $i \in V - \overline{F_v}[t+1]$, $(*, b, t) \notin \mathit{Verified}_i^c[t+1]$.*

Proof: Consider faulty node $b \in \overline{F_v}[t]$. Note that $V - \overline{F_v}[t+1] = (V-F) \cup F_v[t+1]$.

- Consider a fault-free node $i \in V-F$. Since $b \in \overline{F_v}[t]$, node b's round t execution is *not* verified by *any* fault-free node. Therefore, by Definition 4, for fault-free node $i \in V-F$, **at all times**, $(*, b, t) \notin \mathit{Verified}_i[t+1]$. Therefore, by line 14, $(*, b, t) \notin \mathit{Verified}_i^c[t+1]$.
- Consider a faulty node $i \in F_v[t+1]$. In this case, the proof is by contradiction. In particular, for some h, assume that $(h, b, t) \in \mathit{Verified}_i^c[t+1]$. Since $i \in F_v[t+1]$, there exists a fault-free node j that verifies the round $t+1$ execution of node i. Therefore, by Claim 5(ii) in Appendix E, eventually $\mathit{Verified}_i^c[t+1] \subseteq \mathit{Verified}_j[t+1]$. This observation, along with the above assumption that $(h, b, t) \in \mathit{Verified}_i^c[t+1]$, implies that eventually $(h, b, t) \in \mathit{Verified}_j[t+1]$. Since node j is fault-free, Definition 4 implies that execution of node b in round t is verified, and hence $b \in F_v[t]$. This is a contradiction. Therefore, $(*, b, t) \notin \mathit{Verified}_i^c[t+1]$.

□

H Proof of Lemma 3

Lemma 3: *For $r \geq 0$, if $b \in \overline{F_v}[r]$, then for all $\tau \geq r$,*

- *$b \in \overline{F_v}[\tau]$, and*
- *for all $i \in V - \overline{F_v}[\tau+1]$, $(*, b, \tau) \notin \mathit{Verified}_i^c[\tau+1]$.*

Proof:

Recall that $F_v[r] \subseteq F$, and $\overline{F_v}[r] = F - F_v[r]$.

For $r \geq 0$, consider a faulty node $b \in \overline{F_v}[r]$. Thus, $b \in F$.

We first prove that $b \in \overline{F_v}[\tau]$, for $\tau \geq r$. This is trivially true for $\tau = r$. So we only need to prove this for $\tau > r$. The proof is by contradiction.

Suppose that there exists $\tau > r$ such that $b \notin \overline{F_v}[\tau]$. Thus, $b \in F_v[\tau]$. The definition of $\overline{F_v}[\tau]$ implies that node b's round τ execution is verified by some fault-free node j. Then Claim 6 implies that node b's round r execution is verified by node j. Hence by the definition of $F_v[r]$, $b \in F_v[r]$. This is a contradiction. This proves that $b \in \overline{F_v}[\tau]$.

Now, since $b \in \overline{F_v}[\tau]$, by Claim 7, for all $i \in V - \overline{F_v}[\tau+1]$, $(*, b, \tau) \notin \mathit{Verified}_i^c[\tau+1]$.

□

I Claims 8, 9 and 10

Claim 8 *For $t \geq 0$, a fault-free node i adds at most one message from node j to $\mathit{Verified}_i[t]$, even if j is faulty.*

Proof: As stated in the properties of the primitive in Section 2, each fault-free node i will reliably receive at most one message of the form $(*, j, t)$ from node j. Since $\mathit{Verified}_i[t]$ only contains tuples corresponding to reliably received messages, the claim follows. □

Claim 9 *For $t \geq 1$, consider nodes $i, j \in V - \overline{F_v}[t]$. If $(h, k, t) \in \mathit{Verified}_i^c[t]$ and $(h', k, t) \in \mathit{Verified}_j^c[t]$, then $h = h'$.*

Proof: We consider four cases:

- $i, j \in V - F$: In this case, due to *Global Uniqueness* property of reliable broadcast, nodes i and j cannot reliably receive different round t messages from the same node. Hence the claim follows.
- $i \in V - F$ and $j \in F_v[t]$: Suppose that fault-free node p verifies round t execution of node j. Then by Claim 5(ii), eventually $\mathit{Verified}_j^c[t] \subseteq \mathit{Verified}_p[t]$. Since nodes i and p are both fault-free. Therefore, similar to the previous case, due to the *Global Uniqueness* property, nodes i and p cannot reliably receive distinct round t messages. Thus, if $(h, k, t) \in \mathit{Verified}_i^c[t]$ and $(h', k, t) \in \mathit{Verified}_j^c[t] \subseteq \mathit{Verified}_p[t]$, then $h = h'$.
- $j \in V - F$ and $i \in F_v[t]$: This case is similar to the previous case.
- $i, j \in F_v[t]$: In this case, there exist fault-free nodes k_i and k_j that verify round t execution of nodes i and j, respectively. Thus, by Claim 5(ii), eventually $(h, i, t) \in \mathit{Verified}_i^c[t] \subseteq \mathit{Verified}_{k_i}[t]$ and $(h', i, t) \in \mathit{Verified}_j^c[t] \subseteq \mathit{Verified}_{k_j}[t]$. Since k_i, k_j are fault-free, *Global Uniqueness* implies that $h = h'$.

□

Claim 10 *For $t \geq 1$, consider nodes $i, j \in V - \overline{F_v}[t]$. There exists a fault-free node $g \in V - F$ such that $(h_g[t-1], g, t-1) \in \mathit{Verified}_i^c[t] \cap \mathit{Verified}_j^c[t]$.*

Proof: For any fault-free node, say p, due to the conditions checked on line 13, $|\mathit{Verified}_p^c[t]| \geq n - f$. For a node $k \in F_v[t]$, recall that $h_k[t]$ and $\mathit{Verified}_k^c[t]$ are defined in (2) and (3). Thus, by Definition 4, there exists some fault-free node, say q, that reliably receives message $((h_k[t], \mathit{Verified}_k^c[t]), k, t+1)$ from node k in round $t+1$, and after performing checks on line 13, adds $(h_k[t], k, t)$ to $\mathit{Verified}_q^c[t+1]$. The checks on line 13, performed by fault-free node q, ensure that $|\mathit{Verified}_k^c[t]| \geq n - f$.

Above argument implies that for the nodes $i, j \in V - \overline{F_v}[t]$, $Verified_i^c[t]$ and $Verified_j^c[t]$ both contain at least $n-f$ messages. Therefore, by Claims 8 and 9, there will be at least $n-2f \geq df+1 \geq f+1$ elements in $Verified_i^c[t] \cap Verified_j^c[t]$. Since f is the upper bound on the number of faulty nodes, at least one element in $Verified_i^c[t] \cap Verified_j^c[t]$ corresponds to a fault-free node, say node $g \in V - F$. That is, there exists $g \in V - F$ such that $(h_g[t-1], g, t-1) \in Verified_i^c[t] \cap Verified_j^c[t]$.

□

J Proof of Lemma 4

Lemma 4: *For $t \geq 1$, transition matrix $\mathbf{M}[t]$ constructed using the above procedure satisfies the following conditions.*

- *For $i, j \in V$, there exists a fault-free node $g(i,j)$ such that $\mathbf{M}_{ig(i,j)}[t] \geq \frac{1}{n}$.*
- *$\mathbf{M}[t]$ is a row stochastic matrix, and $\lambda(\mathbf{M}[t]) \leq 1 - \frac{1}{n}$.*

Proof:

- To prove the first claim in the lemmas, we consider four cases for node pairs i, j.

  (i) $i, j \in V - \overline{F_v}[t]$:

  By Claim 10, there exists a node $g(i,j)$ such that $(h_{g(i,j)}[t-1], g(i,j), t-1) \in Verified_i^c[t] \cap Verified_j^c[t]$. By (9) in the procedure to construct $\mathbf{M}[t]$, $\mathbf{M}_{ig(i,j)}[t] = \frac{1}{|Verified_i^c[t]|} \geq \frac{1}{n}$ and $\mathbf{M}_{jg(i,j)}[t] = \frac{1}{|Verified_j^c[t]|} \geq \frac{1}{n}$.

  (ii) $i \in \overline{F_v}[t]$ and $j \in V - \overline{F_v}[t]$:

  $|Verified_j^c[t]| \geq n - f$ elements of $\mathbf{M}_j[t]$ are equal to $\frac{1}{|Verified_j^c[t]|} \geq \frac{1}{n}$. Since $n - f \geq (d+1)f + 1 \geq 2f + 1$, there exists a fault-free node $g(i,j)$ such that $\mathbf{M}_{jg(i,j)} \geq \frac{1}{n}$. By (11), all elements of $\mathbf{M}_i[t]$, including $\mathbf{M}_{ig(i,j)}[t] = \frac{1}{n}$.

  (iii) $j \in \overline{F_v}[t]$ and $i \in V - \overline{F_v}[t]$: Similar to case (ii).

  (iv) $i, j \in \overline{F_v}[t]$:

  By (11) in the procedure to construct $\mathbf{M}[t]$, all n elements in $\mathbf{M}_i[t]$ and $\mathbf{M}_j[t]$ both equal $\frac{1}{n}$. Choose a fault-free node as node $g(i,j)$. Then $\mathbf{M}_{ig(i,j)}[t] = \mathbf{M}_{ig(i,j)}[t] = \frac{1}{n}$.

- Observe that, by construction, for each $i \in V$, the row vector $\mathbf{M}_i[t]$ is stochastic. Thus, $\mathbf{M}[t]$ is row stochastic. Also, due to the claim proved in the previous item, and Claim 2, $\lambda(\mathbf{M}[t]) \leq 1 - \frac{1}{n} < 1$.

□

K Proof of Lemma 5

Lemma 5: $h_i[0]$ *for each node* $i \in V - \overline{F_v}[0]$ *is valid.*

Proof: Consider two cases:

- $i \in V - F$: Due to Claim 8 in Appendix I and the fact that i is fault-free $\mathit{Verified}_i^c[0]$ contains at most f elements corresponding to faulty nodes. Recall that $h_i[0]$ is obtained using function $H(\mathit{Verified}_i^c[0], 0)$. Then, due to the *Fault-Free Integrity* property of reliable broadcast, and the definition of function H, at least one set C used in item 2 of function H will contain only the inputs of fault-free nodes. Therefore, $h_i[0]$ is in the convex hull of the inputs at fault-free nodes. That is, $h_i[0]$ is valid.

- $i \in F_v[0]$: Suppose that round 0 execution of node i is verified by fault-free node j. By Claim 5 in Appendix E, $h_i[0] = H(\mathit{Verified}_i^c[0], 0)$, and eventually $\mathit{Verified}_i^c[0] \subseteq \mathit{Verified}_j[0]$. Thus, eventually, $H(\mathit{Verified}_i^c[0], 0) \subseteq H(\mathit{Verified}_j[0], 0)$. Since j is fault-free, and $|\mathit{Verified}_j[0]| \geq n - f$, by an argument similar to the previous item, $H(\mathit{Verified}_j[0], 0)$ is valid. This implies that $h_i[0] = H(\mathit{Verified}_i^c[0], 0)$ is also valid.

□

L Proof of Lemma 6

The proof is straightforward, but included here for completeness.

Lemma 6: *Suppose non-empty convex polytopes $h_1, h_2, \cdots, h_k$ are all valid. Consider k constants $c_1, c_2, \cdots, c_k$ such that $0 \leq c_i \leq 1$ and $\sum_{i=1}^k c_i = 1$. Then the linear combination of these convex polytopes, $H_l(h_1, h_2, \cdots, h_k; c_1, c_2, \cdots, c_k)$, is valid.*

Proof:

Observe that the points in $H_l(h_1, \cdots, h_k; c_1, \cdots, c_k)$ are convex combinations of the points in $h_1, \cdots, h_k$, because $\sum_{i=1}^k c_i = 1$ and $0 \leq c_i \leq 1$, for $1 \leq i \leq k$. Let G be the set of input vectors at the fault-free nodes in $V - F$. Then, $\mathcal{H}(G)$ is the convex hull of the inputs at the fault-free nodes. Since h_i, $1 \leq i \leq k$, is valid, each point $p \in h_i$ is in $\mathcal{H}(G)$. Since $\mathcal{H}(G)$ is a convex polytope, it follows that any convex combination of the points in $h_1, \cdots, h_k$ is also in $\mathcal{H}(G)$.

□

M Algebraic Manipulation in the Proof of Theorem 2

$$
\begin{aligned}
d(p_i^*, p_j^*) &= \sqrt{\sum_{l=1}^{d} (p_i^*(l) - p_j^*(l))^2} \\
&= \sqrt{\sum_{l=1}^{d} \left(\sum_{k \in V - \overline{F_v}[0]} \mathbf{M}_{ik}^* p_k(l) - \sum_{k \in V - \overline{F_v}[0]} \mathbf{M}_{jk}^* p_k(l) \right)^2} \qquad \text{by (16) and (17)} \\
&= \sqrt{\sum_{l=1}^{d} \left(\sum_{k \in V - \overline{F_v}[0]} (\mathbf{M}_{ik}^* - \mathbf{M}_{jk}^*) p_k(l) \right)^2} \\
&\le \sqrt{\sum_{l=1}^{d} \left[\alpha^{2t} \left(\sum_{k \in V - \overline{F_v}[0]} \|p_k(l)\| \right)^2 \right]} \qquad \text{by (15)} \\
&= \alpha^t \sqrt{\sum_{l=1}^{d} \left(\sum_{k \in V - \overline{F_v}[0]} \|p_k(l)\| \right)^2} \qquad (25)
\end{aligned}
$$

N Proof of Lemma 7

We first prove a claim that will be used in the proof of Lemma 7.

Claim 11 *For $t \ge 1$, define $\mathbf{M}'[t] = \Pi_{\tau=1}^{t} \mathbf{M}[\tau]$. Then, for all nodes $j \in V - \overline{F_v}[t]$, and $k \in \overline{F_v}[0]$, $\mathbf{M}'_{jk}[t] = 0$.*

Proof: The proof is by induction on t.

Induction Basis: Consider the case when $t = 1$. Recall that $V - \overline{F_v}[1] = (V - F) \cup F_v[1]$. Consider any $j \in V - \overline{F_v}[1]$, and $k \in \overline{F_v}[0]$. Then by Lemma 3, $(*, k, 0) \notin \mathit{Verified}_j^c[1]$. Then, due to (10), $\mathbf{M}_{jk}[1] = 0$, and hence $\mathbf{M}'_{jk}[1] = \mathbf{M}_{jk}[1] = 0$.

Induction: Consider $t \ge 2$. Assume that the claim holds true through $t-1$. Then, $\mathbf{M}'_{jk}[t-1] = 0$ for all $j \in V - \overline{F_v}[t-1]$ and $k \in \overline{F_v}[0]$. Recall that $\mathbf{M}'[t-1] = \Pi_{\tau=1}^{t-1} \mathbf{M}[\tau]$.

Now, we will prove that the claim holds true for t. Consider $j \in V - \overline{F_v}[t]$ and $k \in \overline{F_v}[0]$. Note that $\mathbf{M}'[t] = \Pi_{\tau=1}^{t} \mathbf{M}[\tau] = \mathbf{M}[t] \Pi_{\tau=1}^{t-1} \mathbf{M}[\tau] = \mathbf{M}[t] \mathbf{M}'[t-1]$. Thus, $\mathbf{M}'_{jk}[t]$ can be non-zero only if there exists a $q \in V$ such that $\mathbf{M}_{jq}[t]$ and $\mathbf{M}'_{qk}[t-1]$ are both non-zero.

For any $q \in \overline{F_v}[t-1]$, by Lemma (3), $(*, q, t-1) \notin \mathit{Verified}_j^c[t]$. Then, due to (10), $\mathbf{M}_{jq}[t] = 0$ for all $q \in \overline{F_v}[t-1]$. Additionally, by the induction hypothesis, for all $q \in V - \overline{F_v}[t-1]$ and $k \in \overline{F_v}[0]$, $\mathbf{M}'_{qk}[t-1] = 0$. Thus, these two observations together imply that there does not exist any $q \in V$ such that $\mathbf{M}_{jq}[t]$ and $\mathbf{M}'_{qk}[t-1]$ are both non-zero. Hence, $\mathbf{M}'_{jk}[t] = 0$. □

Recall from (19) that

$$I = \cap_{D \subset G, |D| = n-2f-\phi} \mathcal{H}(D)$$

where $\phi = |F|$, and G is the set of inputs at the $n - \phi$ fault-free nodes.

Lemma 7:*For all $i \in V - F$ and $t \geq 0$, $I \subseteq h_i[t]$.*

Proof: We first prove that the convex polytope I is contained in $h_i[0]$ for all $i \in V - \overline{F_v}[0]$. Note that $V - \overline{F_v}[0] = (V - F) \cup F_v[0]$.

Consider two cases:

- $i \in V - F$:

  Consider the computation of $h_i[0]$ at fault-free node i using function $H(\mathit{Verified}_i^c[0], 0)$ on line 6. Observe from the definition of function H in Section 2 that

  $$X := \{h \mid (h, j, -1) \in \mathit{Verified}_i^c[0],\ j \in V\}$$

  and

  $$h_i[0] \ := \ \cap_{C \subset X, |C| = |X| - f} \ \ \mathcal{H}(C).$$

  Since $|X| = |\mathit{Verified}_i^c[0]| \geq n - f$ and $\mathit{Verified}_i^c[0]$ contains tuples corresponding to at most $|F| = \phi$ nodes, $|X \cap G| \geq n - f - \phi$. Therefore, every multiset C in the computation of $h_i[0]$ contains inputs of at least $n - 2f - \phi$ fault-free nodes. Thus, $h_i[0]$ contains I.

- $i \in F_v[0]$:

  Suppose that round 0 execution of node i is verified by a fault-free node j. By Claim 5, eventually $\mathit{Verified}_i^c[0] \subseteq \mathit{Verified}_j[0]$, and $h_i[0] = H(\mathit{Verified}_i^c[0], 0)$. Since node j is fault-free, at most ϕ elements in $\mathit{Verified}_j[t]$ correspond to faulty nodes. Therefore, at most ϕ elements in $\mathit{Verified}_i^c[t]$ correspond to faulty nodes, and at least $|\mathit{Verified}_i^c[t]| - \phi \geq n - f - \phi$ correspond to fault-free nodes. Thus, similar to the previous case, $|X \cap G| \geq n - f - \phi$. Therefore, every multiset C in the computation of $h_i[0]$ contains inputs of at least $n - 2f - \phi$ fault-free nodes. Thus, $h_i[0]$ contains I.

Now we make several observations for each fault-free node $i \in V - F$:

- As shown above, $I \in h_j[0]$ for all $j \in V - \overline{F_v}[0]$.
- From (13), for $t \geq 1$,

  $$\mathbf{v}[t] = \mathbf{M}^* \mathbf{v}[0]$$

  where $\mathbf{v}_j[0] = h_j[0]$ for $j \in V - \overline{F_v}[0]$.
- By Theorem 1, $\mathbf{v}_i[t] = h_i[t]$.
- Observe that $\mathbf{M}^*$ equals $\mathbf{M}'[t]$ defined in Claim 11. Thus, due to Claim 11, $\mathbf{M}^*_{ik} = 0$ for $k \in \overline{F_v}[0]$ (i.e,. $k \notin V - \overline{F_v}[0]$).
- $\mathbf{M}^*$ is the product of row stochastic matrices; therefore, $\mathbf{M}^*$ itself is also row stochastic. Thus, for fault-free node i, $\mathbf{v}_i[t] = h_i[t]$ is obtained as the product of the i-th row of $\mathbf{M}^*$, namely $\mathbf{M}^*_i$, and $\mathbf{v}[0]$: this product yields a linear combination of the elements of $\mathbf{v}[0]$, where the weights are non-negative and add to 1 (because $\mathbf{M}^*_i$ is a stochastic row vector).

- From (7), recall that $\mathbf{M}_i^*\mathbf{v}[0] = H_l(\mathbf{v}[0]^T \; ; \; \mathbf{M}_i^*)$. Function H_l ignores the input polytopes for which the corresponding weight is 0. Finally, from the previous observations, we have that when the weight in $\mathbf{M}^*[i]$ is non-zero, the corresponding polytope in $\mathbf{v}[0]^T$ contains I. Therefore, the linear combination also contains I.

Thus, I is contained in $h_i[t] = \mathbf{v}_i[t] = \mathbf{M}_i^*\mathbf{v}[0]$.

□