

Iterative Byzantine Vector Consensus in Incomplete Graphs *

Nitin H. Vaidya

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
nhv@illinois.edu

July 8, 2013

Abstract

This work addresses *Byzantine vector consensus* (BVC), wherein the input at each process is a d -dimensional vector of reals, and each process is expected to decide on a *decision vector* that is in the *convex hull* of the input vectors at the fault-free processes [3, 8]. The input *vector* at each process may also be viewed as a *point* in the d -dimensional Euclidean space \mathbf{R}^d , where $d > 0$ is a finite integer. Recent work [3, 8] has addressed Byzantine vector consensus in systems that can be modeled by a *complete* graph. This paper considers Byzantine vector consensus in *incomplete* graphs. In particular, we address a particular class of *iterative* algorithms in incomplete graphs, and prove a necessary condition, and a sufficient condition, for the graphs to be able to solve the vector consensus problem iteratively. We present an iterative Byzantine vector consensus algorithm, and prove it correct under the sufficient condition. The necessary condition presented in this paper for vector consensus does not match with the sufficient condition for $d > 1$; thus, a weaker condition may potentially suffice for Byzantine vector consensus.

1 Introduction

This work addresses *Byzantine vector consensus* (BVC), wherein the input at each process is a d -dimensional vector of reals, and each process is expected to decide on a *decision vector* that is in the *convex hull* of the input vectors at the fault-free processes [3, 8]. The input *vector* at each process may also be viewed as a *point* in the d -dimensional Euclidean space \mathbf{R}^d , where $d > 0$ is a finite integer. Due to this correspondence, we use the terms *point* and *vector* interchangeably. Recent work [3, 8] has addressed Byzantine vector consensus in systems that can be modeled by a *complete* graph. The correctness conditions for Byzantine vector consensus (elaborated below) cannot be satisfied by independently performing consensus on each element of the input vectors; therefore, new algorithms are necessary. Here we consider Byzantine vector consensus in *incomplete* graphs. In particular, we address a particular class of *iterative* algorithms in incomplete graphs, and prove a necessary condition, and a sufficient condition, for the graphs to be able to solve the vector consensus problem iteratively. The paper extends our past work on *scalar* consensus in incomplete graphs in presence of Byzantine faults [9], which yielded an exact characterization of graphs in which the problem is solvable. We present an iterative Byzantine vector consensus algorithm, and

*This research is supported in part by National Science Foundation award CNS-1059540 and Army Research Office grant W-911-NF-0710287. 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.

prove it correct under the sufficient condition; the proof follows a structure previously used in our work to prove correctness of other consensus algorithms [7, 5].

The necessary condition presented in this paper for vector consensus does not match with the sufficient condition for $d > 1$; thus, it is possible that a weaker condition may also suffice for Byzantine vector consensus. We hope that this paper will motivate further work on identifying the tight sufficient condition.

In other related work [6], we present another generalization of the consensus problem considered in [3, 8]. In particular, [6] considers the problem of deciding on a *convex hull* (instead of just one point) that is contained in the convex hull of the inputs at the fault-free nodes.

The paper is organized as follows. Section 2 presents our system model. The iterative algorithm structure considered in our work is presented in Section 3. Section 4 presents a necessary condition, and Section 5 presents a sufficient condition. Section 5 also presents an iterative algorithm and proves its correctness under the sufficient condition. The paper concludes with a summary in Section 6.

2 System Model

The system is assumed to be *synchronous*.¹ The communication network is modeled as a simple *directed* graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ is the set of n processes, and \mathcal{E} is the set of directed edges between the processes in \mathcal{V} . Thus, $|\mathcal{V}| = n$. We assume that $n \geq 2$, since the consensus problem for $n = 1$ is trivial. Process i can reliably transmit messages to process j , $j \neq i$, if and only if the directed edge (i, j) is in \mathcal{E} . Each process can send messages to itself as well, however, for convenience of presentation, we exclude self-loops from set \mathcal{E} . That is, $(i, i) \notin \mathcal{E}$ for $i \in \mathcal{V}$. We will use the terms *edge* and *link* interchangeably.

For each process i , let N_i^- be the set of processes from which i has incoming edges. That is, $N_i^- = \{j \mid (j, i) \in \mathcal{E}\}$. Similarly, define N_i^+ as the set of processes to which process i has outgoing edges. That is, $N_i^+ = \{j \mid (i, j) \in \mathcal{E}\}$. Since we exclude self-loops from \mathcal{E} , $i \notin N_i^-$ and $i \notin N_i^+$. However, we note again that each process can indeed send messages to itself.

We consider the Byzantine failure model, with up to f processes becoming faulty. A faulty process may *misbehave* arbitrarily. The faulty processes may potentially collaborate with each other. Moreover, the faulty processes are assumed to have a complete knowledge of the execution of the algorithm, including the states of all the processes, contents of messages the other processes send to each other, the algorithm specification, and the network topology.

Notation: We use the notation $|X|$ to denote the size of a set or a multiset, and the notation $\|x\|$ to denote the absolute value of a real number x .

3 Byzantine Vector Consensus and Iterative Algorithms

Byzantine vector consensus: We are interested in **iterative** algorithms that satisfy the following conditions in presence of up to f Byzantine faulty processes:

¹Analogous results can be similarly derived for asynchronous systems, using the asynchronous algorithm structure presented in [9] for the case of $d = 1$.

- *Termination*: Each fault-free process must terminate after a finite number of iterations.
- *Validity*: The state of each fault-free process at the end of **each iteration** must be in the convex hull of the d -dimensional input vectors at the fault-free processes.
- ϵ -*Agreement*: When the algorithm terminates, the l -th elements of the decision vectors at any two fault-free processes, where $1 \leq l \leq d$, must be within ϵ of each other, where $\epsilon > 0$ is a pre-defined constant.

Any information carried over by a process from iteration t to iteration $t + 1$ is considered the state of process t at the end of iteration t . The above *validity* condition forces the algorithms to maintain “minimal” state, for instance, precluding the possibility of remembering messages received in several of the past iterations, or remembering the history of detected misbehavior of the neighbors. Therefore, we focus on algorithms with a simple iterative structure, described below.

Iterative structure: Each process i maintains a state variable \mathbf{v}_i , which is a d -dimensional vector. The initial state of process i is denoted as $\mathbf{v}_i[0]$, and it equals the *input* provided to process i . For $t \geq 1$, $\mathbf{v}_i[t]$ denotes the state of process i at the *end* of the t -th iteration of the algorithm. At the *start* of the t -th iteration ($t \geq 1$), the state of process i is $\mathbf{v}_i[t - 1]$. The iterative algorithms of interest will require each process i to perform the following three steps in the t -th iteration. Each “value” referred in the algorithm below is a d -dimensional vector (or, equivalently, a point in the d -dimensional Euclidean space).

1. *Transmit step*: Transmit current state, namely $\mathbf{v}_i[t - 1]$, on all outgoing edges to processes in N_i^+ .
2. *Receive step*: Receive values on all incoming edges from processes in N_i^- . Denote by $r_i[t]$ the multiset² of values received by process i from its neighbors. The size of multiset $r_i[t]$ is $|N_i^-|$.
3. *Update step*: Process i updates its state using a transition function T_i as follows. T_i is a part of the specification of the algorithm, and takes as input the multiset $r_i[t]$ and state $\mathbf{v}_i[t - 1]$.

$$\mathbf{v}_i[t] = T_i (r_i[t], \mathbf{v}_i[t - 1]) \quad (1)$$

The decision (or output) of each process equals its state when the algorithm terminates.

We assume that each element of the input vector at each fault-free process is lower bounded by a constant μ and upper bounded by a constant U . The iterative algorithm may terminate after a number of rounds that is a function of μ and U . μ and U are assumed to be known *a priori*. This assumption holds in many practical systems, because the input vector elements represent quantities that are constrained. For instance, if the input vectors are probability vectors, then $U = 1$ and $\mu = 0$. If the input vectors represent locations in 3-dimensional space occupied by mobile robots, then U and μ are determined by the boundary of the region in which the robots are allowed to operate.

In Section 4, we develop a necessary condition that the graph $G(\mathcal{V}, \mathcal{E})$ must satisfy in order for the Byzantine vector consensus algorithm to be solvable using the above iterative structure. In Section 5, we develop a sufficient condition, such that the Byzantine vector consensus algorithm is solvable using the above iterative structure in any graph that satisfies this condition. We present an iterative algorithm, and prove its correctness under the sufficient condition.

²The same value may occur multiple times in a multiset.

4 A Necessary Condition

Hereafter, when we refer to an iterative algorithm, we mean an algorithm with the iterative structure specified in the previous section. In this section, we state a necessary condition on graph $G(\mathcal{V}, \mathcal{E})$ to be able to achieve Byzantine vector consensus using an iterative algorithm. First we introduce some notations.

Definition 1

- Define \mathbf{e}_0 to be a d -dimensional vector with all its elements equal to 0. Thus, \mathbf{e}_0 corresponds to the origin in the d -dimensional Euclidean space.
- Define \mathbf{e}_i , $1 \leq i \leq d$, to be a d -dimensional vector with the i -th element equal to 2ϵ , and the remaining elements equal to 0. Recall that ϵ is the parameter of the ϵ -agreement condition.

Definition 2 For non-empty disjoint sets of processes A and B , and a non-negative integer c ,

- $A \xrightarrow{c} B$ if and only if there exists a process $v \in B$ that has at least $c+1$ incoming edges from processes in A , i.e., $|N_v^- \cap A| \geq c+1$.
- $A \not\xrightarrow{c} B$ iff $A \xrightarrow{c} B$ is not true.

Definition 3 $\mathcal{H}(X)$ denotes the convex hull of a multiset of points X .

Now we state the necessary condition.

Condition NC: For any partition $V_0, V_1, \dots, V_p, C, F$ of set \mathcal{V} , where $1 \leq p \leq d$, $V_k \neq \emptyset$ for $0 \leq k \leq p$, and $|F| \leq f$, there exist i, j ($0 \leq i, j \leq p$, $i \neq j$), such that

$$V_i \cup C \xrightarrow{f} V_j$$

That is, there are $f+1$ incoming links from processes in $V_i \cup C$ to some process in V_j .

Lemma 1 If the Byzantine vector consensus problem can be solved using an iterative algorithm in $G(\mathcal{V}, \mathcal{E})$, then $G(\mathcal{V}, \mathcal{E})$ satisfies Condition NC.

Proof: The proof is by contradiction. Suppose that Condition NC is not true. Then there exists a certain partition $V_0, V_1, \dots, V_p, C, F$ such that $V_k \neq \emptyset$ ($1 \leq k \leq p$), $|F| \leq f$, and for $0 \leq i, k \leq p$, $V_k \cup C \not\xrightarrow{f} V_i$.

Let the initial state of each process in V_i be \mathbf{e}_i ($0 \leq i \leq p$). Suppose that all the processes in set F are faulty. For each link (j, k) such that $j \in F$ and $k \in V_i$ ($0 \leq i \leq p$), the faulty process j sends value \mathbf{e}_i to process k in each iteration.

We now prove by induction that if the iterative algorithm satisfies the validity condition then the state of each fault-free process $k \in V_i$ at the start of iteration t equals \mathbf{e}_i , for all $t > 0$. The claim

is true for $t = 1$ by assumption on the inputs at the fault-free processes. Now suppose that the claim is true through iteration t , and prove it for iteration $t + 1$. Thus, the state of each fault-free process in V_i at the start of iteration t equals \mathbf{e}_i , $0 \leq i \leq p$.

Consider any fault-free process $j \in V_i$, where $0 \leq i \leq p$. In iteration t , process j will receive $\mathbf{v}_g[t-1]$ from each fault-free incoming neighbor g , and receive \mathbf{e}_i from each faulty incoming neighbor. These received values form the multiset $r_j[t]$. Since the condition in the lemma is assumed to be false, for any $k \neq i$, $0 \leq k \leq p$, we have

$$V_k \cup C \not\xrightarrow{f} V_i.$$

Thus, at most f incoming neighbors of j belong to $V_k \cup C$, and therefore, at most f values in $r_j[t]$ equal \mathbf{e}_k .

Since process j does not know which of its incoming neighbors, if any, are faulty, it must allow for the possibility that any of its f incoming neighbors are faulty. Let $A_k \subseteq V_k \cup C$, $k \neq i$, be the set containing all the incoming neighbors of process j in $V_k \cup C$. Since $V_k \cup C \not\xrightarrow{f} V_i$, $|A_k| \leq f$; therefore, all the processes in A_k are *potentially* faulty. Also, by assumption, the values received from all fault-free processes equal their input, and the values received from faulty processes in F equal \mathbf{e}_i . Thus, due to the validity condition, process j must choose as its new state a value that is in the convex hull of the set

$$S_k = \{\mathbf{e}_m \mid m \neq k, 0 \leq m \leq p\}.$$

where $k \neq i$. Since this observation is true for each $k \neq i$, it follows that the new state $\mathbf{v}_j[t]$ must be a point in the convex hull of

$$\bigcap_{1 \leq k \leq p, k \neq i} \mathcal{H}(S_k).$$

It is easy to verify that the above intersection only contains the point \mathbf{e}_i . Therefore, $\mathbf{v}_j[t] = \mathbf{e}_i$. Thus, the state of process j at the start of iteration $t + 1$ equals \mathbf{e}_i . This concludes the induction.

The above result implies that the state of each fault-free process remains unchanged through the iterations. Thus, the state of any two fault-free processes differs in at least one vector element by 2ϵ , precluding ϵ -agreement. \square

The above lemma demonstrates the necessity of Condition NC. Necessary condition NC implies a lower bound on the number of processes $n = |\mathcal{V}|$ in $G(\mathcal{V}, \mathcal{E})$, as stated in the next lemma.

Lemma 2 *Suppose that the Byzantine vector consensus problem can be solved using an iterative algorithm in $G(\mathcal{V}, \mathcal{E})$. Then, $n \geq (d + 2)f + 1$.*

Proof: Since the Byzantine vector consensus problem can be solved using an iterative algorithm in $G(\mathcal{V}, \mathcal{E})$, by Lemma 1, graph G must satisfy Condition NC. Suppose that $2 \leq |\mathcal{V}| = n \leq (d + 2)f$. Then there exists p , $1 \leq p \leq d$, such that we can partition \mathcal{V} into sets V_0, \dots, V_p, F such that for each V_i , $0 < |V_i| \leq f$, and $|F| \leq f$. Define $C = \emptyset$. Since $|C \cup V_i| \leq f$ for each i , it is clear that this partition of \mathcal{V} cannot satisfy Condition NC. This is a contradiction. \square

When $d = 1$, the input at each process is a scalar. For the $d = 1$ case, our prior work [9] yielded a tight necessary and sufficient condition for Byzantine consensus to be achievable in $G(\mathcal{V}, \mathcal{E})$ using iterative algorithms. For $d = 1$, the necessary condition stated in Lemma 1 is equivalent to the

necessary condition in [9]. We previously showed that, for $d = 1$, the same condition is also sufficient [9]. However, in general, for $d > 1$, Condition NC is not proved sufficient. Instead, we prove the sufficiency of another condition stated in the next section.

5 A Sufficient Condition

We now present Condition SC that is later proved to be sufficient for achieving Byzantine vector consensus in graph $G(\mathcal{V}, \mathcal{E})$ using an iterative algorithm.

Condition SC: For any partition F, L, C, R of set \mathcal{V} , such that L and R are both non-empty, and $|F| \leq f$, at least one of these conditions is true: $R \cup C \xrightarrow{df} L$, or $L \cup C \xrightarrow{df} R$.

Later in the paper we will present a Byzantine vector consensus algorithm named `Byz-Iter` that is proved correct in all graphs that satisfy Condition SC. The proof will make use of Lemmas 3 and 4 presented below.

Lemma 3 For $f > 0$, if graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition SC, then in-degree of each process in \mathcal{V} must be at least $(d + 1)f + 1$. That is, for each $i \in \mathcal{V}$, $|N_i^-| \geq (d + 1)f + 1$.

Lemma 3 is proved in Appendix A.

Definition 4 Reduced Graph: For a given graph $G(\mathcal{V}, \mathcal{E})$ and $\mathcal{F} \subset \mathcal{V}$ such that $|\mathcal{F}| \leq f$, a graph $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$ is said to be a reduced graph, if: (i) $\mathcal{V}_{\mathcal{F}} = \mathcal{V} - \mathcal{F}$, and (ii) $\mathcal{E}_{\mathcal{F}}$ is obtained by first removing from \mathcal{E} all the links incident on the processes in \mathcal{F} , and then removing up to df additional incoming links at each process in $\mathcal{V}_{\mathcal{F}}$.

Note that for a given $G(\mathcal{V}, \mathcal{E})$ and a given \mathcal{F} , multiple reduced graphs may exist (depending on the choice of the links removed at each process).

Lemma 4 Suppose that graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition SC, and $\mathcal{F} \subset \mathcal{V}$. Then, in any reduced graph $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$, there exists a process that has a directed path to all the remaining processes in $\mathcal{V}_{\mathcal{F}}$.

Lemma 4 is proved in Appendix B.

5.1 Algorithm Byz-Iter

We will prove that, if graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition SC, then Algorithm `Byz-Iter` presented below achieves Byzantine vector consensus. Algorithm `Byz-Iter` has the three-step structure described in Section 3.

The proposed algorithm is based on the following result by Tverberg [4].

Theorem 1 (Tverberg's Theorem [4]) For any integer $f \geq 0$, and for every multiset Y containing at least $(d + 1)f + 1$ points in \mathbf{R}^d , there exists a partition Y_1, \dots, Y_{f+1} of Y into $f + 1$ non-empty multisets such that $\bigcap_{i=1}^{f+1} \mathcal{H}(Y_i) \neq \emptyset$.

The points in Y above need not be distinct [4]; thus, the same point may occur multiple times in Y , and also in each of its subsets (Y_l 's) above. The partition in Theorem 1 is called a *Tverberg partition*, and the points in $\cap_{l=1}^{f+1} \mathcal{H}(Y_l)$ in Theorem 1 are called **Tverberg points**.

Algorithm Byz-Iter

Each iteration consists of three steps: *Transmit*, *Receive*, and *Update*:

1. *Transmit step*: Transmit current state $\mathbf{v}_i[t-1]$ on all outgoing edges.
2. *Receive step*: Receive values on all incoming edges. These values form multiset $r_i[t]$ of size $|N_i^-|$. (If a message is not received from some incoming neighbor, then that neighbor must be faulty. In this case, the missing message value is assumed to be \mathbf{e}_0 by default. Recall that we assume a *synchronous* system.)
3. *Update step*: Form a multiset $Z_i[t]$ using the steps below:
 - Initialize $Z_i[t]$ as empty.
 - Add to $Z_i[t]$, any one *Tverberg point* corresponding to *each* multiset $C \subseteq r_i[t]$ such that $|C| = (d+1)f+1$. Since $|C| = (d+1)f+1$, by Theorem 1, such a Tverberg point exists.

$Z_i[t]$ is a multiset; thus a single point may appear in $Z_i[t]$ more than once. Note that $|Z_i[t]| = \binom{|r_i[t]|}{(d+1)f+1} \leq \binom{n}{(d+1)f+1}$. Compute new state $\mathbf{v}_i[t]$ as:

$$\mathbf{v}_i[t] = \frac{\mathbf{v}_i[t-1] + \sum_{\mathbf{z} \in Z_i[t]} \mathbf{z}}{1 + |Z_i[t]|} \quad (2)$$

Termination: Each fault-free process terminates after completing t_{end} iterations, where t_{end} is a constant defined later in (9). The value of t_{end} depends on graph $G(\mathcal{V}, \mathcal{E})$, constants U and μ defined earlier, and parameter ϵ of ϵ -agreement.

The proof of correctness of Algorithm **Byz-Iter** makes use of a matrix representation of the algorithm's behavior. Before presenting the matrix representation, we introduce some notations and definitions related to matrices.

5.2 Matrix Preliminaries

We use boldface 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 $a \leq b$,

$$\Pi_{\tau=a}^b \mathbf{A}[\tau] = \mathbf{A}[b] \mathbf{A}[b-1] \cdots \mathbf{A}[a] \quad (3)$$

For a row stochastic matrix \mathbf{A} , coefficients of ergodicity $\delta(\mathbf{A})$ and $\lambda(\mathbf{A})$ are defined as follows [10]:

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

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

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

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

Claim 2 If all the elements in any one column of matrix \mathbf{A} are lower bounded by a constant γ , then $\lambda(\mathbf{A}) \leq 1 - \gamma$. That is, if $\exists g$, such that $\mathbf{A}_{ig} \geq \gamma, \forall i$, then $\lambda(\mathbf{A}) \leq 1 - \gamma$.

5.3 Correctness of Algorithm Byz-Iter

This section presents a key lemma, Lemma 5, that helps us in proving the correctness of Algorithm Byz-Iter. In particular, Lemma 5 allows us to use results for non-homogeneous Markov chains to prove the correctness of Algorithm Byz-Iter.

Let \mathcal{F} denote the actual set of faulty processes in a given execution of Algorithm Byz-Iter. Let $|\mathcal{F}| = \psi$. Thus, $0 \leq \psi \leq f$. Without loss of generality, suppose that processes 1 through $(n - \psi)$ are fault-free, and if $\psi > 0$, processes $(n - \psi + 1)$ through n are faulty.

In the analysis below, it is convenient to view the state of each process as a point in the d -dimensional Euclidean space. Denote by $\mathbf{v}[0]$ the column vector consisting of the initial states of the $(n - \psi)$ fault-free processes. The i -th element of $\mathbf{v}[0]$ is $\mathbf{v}_i[0]$, the initial state of process i . Thus, $\mathbf{v}[0]$ is a vector consisting of $(n - \psi)$ points in the d -dimensional Euclidean space. Denote by $\mathbf{v}[t]$, for $t \geq 1$, the column vector consisting of the states of the $(n - \psi)$ fault-free processes at the end of the t -th iteration. The i -th element of vector $\mathbf{v}[t]$ is state $\mathbf{v}_i[t]$.

Lemma 5 Suppose that graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition SC. Then the state updates performed by the fault-free processes in the t -th iteration ($t \geq 1$) of Algorithm Byz-Iter can be expressed as

$$\mathbf{v}[t] = \mathbf{M}[t] \mathbf{v}[t-1] \tag{4}$$

where $\mathbf{M}[t]$ is a $(n - \psi) \times (n - \psi)$ row stochastic matrix with the following property: there exists a reduced graph $H[t]$, and a constant β ($0 < \beta \leq 1$) that depends only on graph $G(\mathcal{V}, \mathcal{E})$, such that

$$\mathbf{M}_{ij}[t] \geq \beta$$

if $j = i$ or edge (j, i) is in $H[t]$.

Proof: The proof is presented in Appendix C. □

Matrix $\mathbf{M}[t]$ above is said to be a transition matrix. As the lemma states, $\mathbf{M}[t]$ is a row stochastic matrix. The proof of Lemma 5 shows how to identify a suitable row stochastic matrix

$\mathbf{M}[t]$ for each iteration t . The matrix $\mathbf{M}[t]$ depends on t , as well as the behavior of the faulty processes. $\mathbf{M}_i[t]$ is the i -th row of transition matrix $\mathbf{M}[t]$. Thus, (4) implies that

$$\mathbf{v}_i[t] = \mathbf{M}_i[t] \mathbf{v}[t-1]$$

That is, the state of any fault-free process i at the end of iteration t can be expressed as a convex combination of the state of just the fault-free processes at the end of iteration $t-1$. Recall that vector \mathbf{v} only includes the state of fault-free processes.

Theorem 2 *Algorithm Byz-Iter satisfies the termination, validity and ϵ -agreement conditions.*

Proof: Sections 5.4, 5.5 and 5.6 provide the proof that Algorithm Byz-Iter satisfies the three conditions for Byzantine vector consensus. This proof follows a structure used to prove correctness of other consensus algorithms in our prior work [7, 5]. \square

5.4 Algorithm Byz-Iter Satisfies the Validity Condition

Observe that $\mathbf{M}[t+1](\mathbf{M}[t]\mathbf{v}[t-1]) = (\mathbf{M}[t+1]\mathbf{M}[t])\mathbf{v}[t-1]$. Therefore, by repeated application of (4), we obtain for $t \geq 1$,

$$\mathbf{v}[t] = \left(\prod_{\tau=1}^t \mathbf{M}[\tau] \right) \mathbf{v}[0] \tag{5}$$

Since each $\mathbf{M}[\tau]$ is row stochastic, the matrix product $\prod_{\tau=1}^t \mathbf{M}[\tau]$ is also a row stochastic matrix. Recall that vector \mathbf{v} only includes the state of fault-free processes. Thus, (5) implies that the state of each fault-free process i at the end of iteration t can be expressed as a convex combination of the initial state of the fault-free processes. Therefore, the validity condition is satisfied.

5.5 Algorithm Byz-Iter Satisfies the Termination Condition

Algorithm Byz-Iter stops after a finite number (t_{end}) of iterations, where t_{end} is a constant that depends only on $G(\mathcal{V}, \mathcal{E})$, U , μ and ϵ . Therefore, trivially, the algorithm satisfies the termination condition. Later, using (9) we define a suitable value for t_{end} .

5.6 Algorithm Byz-Iter Satisfies the ϵ -Agreement Condition

The proof structure below is derived from our previous work wherein we proved the correctness of an iterative algorithm for *scalar* Byzantine consensus (i.e., the case of $d = 1$) [7] and its generalization to a broader class of fault sets [5].

Let R_F denote the set of all the reduced graph of $G(\mathcal{V}, \mathcal{E})$ corresponding to fault set F . Thus, $R_{\mathcal{F}}$ is the set of all the reduced graph of $G(\mathcal{V}, \mathcal{E})$ corresponding to actual fault set \mathcal{F} . Let

$$r = \max_{|F| \leq f} |R_F|.$$

r depends only on $G(\mathcal{V}, \mathcal{E})$ and f , and it is finite. Note that $|R_{\mathcal{F}}| \leq r$.

For each reduced graph $H \in R_{\mathcal{F}}$, define connectivity matrix \mathbf{H} as follows, where $1 \leq i, j \leq n - \psi$:

- $\mathbf{H}_{ij} = 1$ if either $j = i$, or edge (j, i) exists in reduced graph H .

- $\mathbf{H}_{ij} = 0$, otherwise.

Thus, the non-zero elements of row \mathbf{H}_i correspond to the incoming links at process i in the reduced graph H , and the self-loop at process i . Observe that \mathbf{H} has a non-zero diagonal.

Lemma 6 *For any $H \in R_{\mathcal{F}}$, and any $k \geq n - \psi$, matrix product \mathbf{H}^k has at least one non-zero column (i.e., a column with all elements non-zero).*

Proof: Each reduced graph contains $n - \psi$ processes because the fault set \mathcal{F} contain ψ processes. By Lemma 4, at least one process in the reduced graph, say process p , has directed paths to all the processes in the reduced graph H . Element \mathbf{H}_{jp}^k of matrix product \mathbf{H}^k is 1 if and only if process p has a directed path to process j containing at most k edges; each of these directed paths must contain less than $n - \psi$ edges, because the number of processes in the reduced graph is $n - \psi$. Since p has directed paths to all the processes, it follows that, when $k \geq n - \psi$, all the elements in the p -th column of \mathbf{H}^k must be non-zero. \square

For matrices \mathbf{A} and \mathbf{B} of identical dimensions, we say that $\mathbf{A} \leq \mathbf{B}$ if and only if $\mathbf{A}_{ij} \leq \mathbf{B}_{ij}$, $\forall i, j$. Lemma 7 relates the transition matrices with the connectivity matrices. Constant β used in the lemma below was introduced in Lemma 5.

Lemma 7 *For any $t \geq 1$, there exists a reduced graph $H[t] \in R_{\mathcal{F}}$ such that $\beta \mathbf{H}[t] \leq \mathbf{M}[t]$, where $\mathbf{H}[t]$ is the connectivity matrix for $H[t]$.*

Proof: Appendix D presents the proof. \square

Lemma 8 *At least one column in the matrix product $\prod_{t=u}^{u+r(n-\psi)-1} \mathbf{H}[t]$ is non-zero.*

Proof: Since $\prod_{t=u}^{u+r(n-\psi)-1} \mathbf{H}[t]$ is a product of $r(n - \psi)$ connectivity matrices corresponding to the reduced graphs in $R_{\mathcal{F}}$, and $|R_{\mathcal{F}}| \leq r$, connectivity matrix corresponding to at least one reduced graph in $R_{\mathcal{F}}$, say matrix \mathbf{H}_* , will appear in the above product at least $n - \psi$ times.

By Lemma 6, $\mathbf{H}_*^{n-\psi}$ contains a non-zero column; say the p -th column of \mathbf{H}_* is non-zero. Also, by definition, all the connectivity matrices ($\mathbf{H}[t]$) have a non-zero diagonal. These two observations together imply that the p -th column in the product $\prod_{t=u}^{u+r(n-\psi)-1} \mathbf{H}[t]$ is non-zero.³ \square

Let us now define a sequence of matrices $\mathbf{Q}(i)$, $i \geq 1$, such that each of these matrices is a product of $r(n - \psi)$ of the $\mathbf{M}[t]$ matrices. Specifically,

$$\mathbf{Q}(i) = \prod_{t=(i-1)r(n-\psi)+1}^{ir(n-\psi)} \mathbf{M}[t] \quad (6)$$

From (5) and (6) observe that

$$\mathbf{v}[kr(n - \psi)] = \left(\prod_{i=1}^k \mathbf{Q}(i) \right) \mathbf{v}[0] \quad (7)$$

Lemma 9 *For $i \geq 1$, $\mathbf{Q}(i)$ is a row stochastic matrix, and*

$$\lambda(\mathbf{Q}(i)) \leq 1 - \beta^{r(n-\psi)}.$$

³The product $\prod_{t=z}^{z+r(n-\psi)-1} \mathbf{H}[t]$ can be viewed as the product of $(n - \psi)$ instances of \mathbf{H}_* “interspersed” with matrices with non-zero diagonals.

Proof: $\mathbf{Q}(i)$ is a product of row stochastic matrices ($\mathbf{M}[t]$); therefore, $\mathbf{Q}(i)$ is row stochastic. From Lemma 7, for each $t \geq 1$,

$$\beta \mathbf{H}[t] \leq \mathbf{M}[t]$$

Therefore,

$$\beta^{r(n-\psi)} \prod_{t=(i-1)r(n-\psi)+1}^{ir(n-\psi)} \mathbf{H}[t] \leq \prod_{t=(i-1)r(n-\psi)+1}^{ir(n-\psi)} \mathbf{M}[t] = \mathbf{Q}(i)$$

By using $u = (i-1)r(n-\psi) + 1$ in Lemma 8, we conclude that the matrix product on the left side of the above inequality contains a non-zero column. Therefore, since $\beta > 0$, $\mathbf{Q}(i)$ on the right side of the inequality also contains a non-zero column.

Observe that $r(n-\psi)$ is finite, and hence, $\beta^{r(n-\psi)}$ is non-zero. Since the non-zero terms in $\mathbf{H}[t]$ matrices are all 1, the non-zero elements in $\prod_{t=(i-1)r(n-\psi)+1}^{ir(n-\psi)} \mathbf{H}[t]$ must each be ≥ 1 . Therefore, there exists a non-zero column in $\mathbf{Q}(i)$ with all the elements in the column being $\geq \beta^{r(n-\psi)}$. Therefore, by Claim 2, $\lambda(\mathbf{Q}(i)) \leq 1 - \beta^{r(n-\psi)}$. \square

Let us now continue with the proof of ϵ -agreement. Consider the coefficient of ergodicity $\delta(\prod_{i=1}^t \mathbf{M}[i])$.

$$\begin{aligned} \delta(\prod_{i=1}^t \mathbf{M}[i]) &= \delta \left(\left(\prod_{i=\lfloor \frac{t}{r(n-\psi)} \rfloor r(n-\psi)+1}^t \mathbf{M}[i] \right) \left(\prod_{i=1}^{\lfloor \frac{t}{r(n-\psi)} \rfloor} \mathbf{Q}(i) \right) \right) \quad \text{by definition of } Q(i) \\ &\leq \lambda \left(\prod_{i=\lfloor \frac{t}{r(n-\psi)} \rfloor r(n-\psi)+1}^t \mathbf{M}[i] \right) \prod_{i=1}^{\lfloor \frac{t}{r(n-\psi)} \rfloor} \lambda(\mathbf{Q}(i)) \quad \text{by Claim 1} \\ &\leq \prod_{i=1}^{\lfloor \frac{t}{r(n-\psi)} \rfloor} \lambda(\mathbf{Q}(i)) \quad \text{because } \lambda(\cdot) \leq 1 \\ &\leq \left(1 - \beta^{r(n-\psi)} \right)^{\lfloor \frac{t}{r(n-\psi)} \rfloor} \quad \text{by Lemma 9} \\ &\leq (1 - \beta^{rn})^{\lfloor \frac{t}{rn} \rfloor} \quad \text{because } 0 < \beta \leq 1 \text{ and } 0 \leq \psi < n. \end{aligned} \quad (8)$$

Observe that the upper bound on right side of (8) depends only on graph $G(\mathcal{V}, \mathcal{E})$ and t , and is independent of the input vectors, the fault set \mathcal{F} , and the behavior of the faulty processes. Also, the upper bound on the right side of (8) is a non-increasing function of t . Define t_{end} as the smallest positive integer t for which the right hand side of (8) is smaller than $\frac{\epsilon}{n \max(\|U\|, \|\mu\|)}$, where $\|x\|$ denotes the absolute value of real number x . Thus,

$$\delta(\prod_{i=1}^{t_{end}} \mathbf{M}[i]) \leq (1 - \beta^{rn})^{\lfloor \frac{t_{end}}{rn} \rfloor} < \frac{\epsilon}{n \max(\|U\|, \|\mu\|)} \quad (9)$$

Recall that β and r depend only on $G(\mathcal{V}, \mathcal{E})$. Thus, t_{end} depends only on graph $G(\mathcal{V}, \mathcal{E})$, and constants U , μ and ϵ .

Recall that $\prod_{i=1}^t \mathbf{M}[i]$ is a $(n-\psi) \times (n-\psi)$ row stochastic matrix. Let $\mathbf{M}^* = \prod_{i=1}^t \mathbf{M}[i]$. From (5) we know that state $\mathbf{v}_j[t]$ of any fault-free process j is obtained as the product of the j -th row of $\prod_{i=1}^t \mathbf{M}[i]$ and $\mathbf{v}[0]$. That is, $\mathbf{v}_j[t] = \mathbf{M}_j^* \mathbf{v}[0]$.

Recall that $\mathbf{v}_j[t]$ is a d -dimensional vector. Let us denote the l -th element of $\mathbf{v}_j[t]$ as $\mathbf{v}_j[t](l)$, $1 \leq l \leq d$. Also, by $\mathbf{v}[0](l)$, let us denote a vector consisting of the l -th elements of $\mathbf{v}_i[0]$, $\forall i$. Then by the definitions of $\delta(\cdot)$, U and μ , for any two fault-free processes j and k , we have

$$\|\mathbf{v}_j[t](l) - \mathbf{v}_k[t](l)\| = \|\mathbf{M}_j^* \mathbf{v}[0](l) - \mathbf{M}_k^* \mathbf{v}[0](l)\| \quad (10)$$

$$= \left\| \sum_{i=1}^{n-\psi} \mathbf{M}_{ji}^* \mathbf{v}_i[0](l) - \sum_{i=1}^{n-\psi} \mathbf{M}_{ki}^* \mathbf{v}_i[0](l) \right\| \quad (11)$$

$$= \left\| \sum_{i=1}^{n-\psi} (\mathbf{M}_{ji}^* - \mathbf{M}_{ki}^*) \mathbf{v}_i[0](l) \right\| \quad (12)$$

$$\leq \sum_{i=1}^{n-\psi} \|\mathbf{M}_{ji}^* - \mathbf{M}_{ki}^*\| \|\mathbf{v}_i[0](l)\| \quad (13)$$

$$\leq \sum_{i=1}^{n-\psi} \delta(\mathbf{M}^*) \|\mathbf{v}_i[0](l)\| \quad (14)$$

$$\leq (n - \psi) \delta(\mathbf{M}^*) \max(\|U\|, \|\mu\|) \quad (15)$$

$$\leq (n - \psi) \max(\|U\|, \|\mu\|) \delta(\Pi_{i=1}^t \mathbf{M}[i]) \quad (16)$$

$$\leq n \max(\|U\|, \|\mu\|) \delta(\Pi_{i=1}^t \mathbf{M}[i]) \quad \text{because } 0 \leq \psi < n$$

Therefore, by (9) and (16),

$$\|\mathbf{v}_i[t_{end}](l) - \mathbf{v}_j[t_{end}](l)\| < \epsilon, \quad 1 \leq l \leq d. \quad (17)$$

The output of a fault-free process equals its state at termination (after t_{end} iterations). Thus, (17) implies that Algorithm `Byz-Iter` satisfies the ϵ -agreement condition.

6 Summary

This paper addresses *Byzantine vector consensus* (BVC), wherein the input at each process is a d -dimensional vector of reals, and each process is expected to decide on a *decision vector* that is in the *convex hull* of the input vectors at the fault-free processes [3, 8]. We address a particular class of *iterative* algorithms in *incomplete* graphs, and prove a necessary condition (NC), and a sufficient condition (SC), for the graphs to be able to solve the vector consensus problem iteratively. This paper extends our past work on *scalar* consensus (i.e., $d = 1$) in incomplete graphs in presence of Byzantine faults [9, 7], which yielded an exact characterization of graphs in which the problem is solvable for $d = 1$. However, the necessary condition NC presented in the paper for *vector* consensus does not match with the sufficient condition SC. We hope that this paper will motivate further work on identifying the tight sufficient condition.

References

- [1] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill Higher Education, 2006.
- [2] J. Hajnal. Weak ergodicity in non-homogeneous markov chains. In *Proceedings of the Cambridge Philosophical Society*, volume 54, pages 233–246, 1958.
- [3] H. Mendes and M. Herlihy. Multidimensional approximate agreement in byzantine asynchronous systems. In *45th ACM Symposium on the Theory of Computing (STOC)*, June 2013.

- [4] M. A. Perles and M. Sigron. A generalization of Tverberg’s theorem, 2007. CoRR, <http://arxiv.org/abs/0710.4668>.
- [5] L. Tseng and N. H. Vaidya. Iterative approximate byzantine consensus under a generalized fault model. In *International Conference on Distributed Computing and Networking (ICDCN)*, January 2013.
- [6] L. Tseng and N. H. Vaidya, Byzantine Convex Consensus: An Optimal Algorithm, 2013. CoRR, <http://arxiv.org/abs/1307.1332>.
- [7] N. H. Vaidya. Matrix representation of iterative approximate byzantine consensus in directed graphs. *CoRR* <http://arxiv.org/abs/1203.1888>, March 2012.
- [8] N. H. Vaidya and V. K. Garg. Byzantine vector consensus in complete graphs. In *ACM Symposium on Principles of Distributed Computing (PODC)*, July 2013.
- [9] N. H. Vaidya, L. Tseng, and G. Liang. Iterative approximate byzantine consensus in arbitrary directed graphs. In *ACM Symposium on Principles of Distributed Computing (PODC)*, July 2012.
- [10] J. Wolfowitz. Products of indecomposable, aperiodic, stochastic matrices. In *Proceedings of the American Mathematical Society*, pages 733–737, 1963.

A Proof of Lemma 3

Lemma 3 For $f > 0$, if graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition SC, then in-degree of each process in \mathcal{V} must be at least $(d + 1)f + 1$. That is, for each $i \in \mathcal{V}$, $|N_i^-| \geq (d + 1)f + 1$.

Proof: The proof is by contradiction. As per the assumption in the lemma, $f > 0$, and graph $G(\mathcal{V}, \mathcal{E})$ satisfies condition SC.

Suppose that some process i has in-degree at most $(d+1)f$. Define $L = \{i\}$, and $C = \emptyset$. Partition the processes in $\mathcal{V} - \{i\}$ into sets R and F such that $|F| \leq f$, $|F \cap N_i^-| \leq f$ and $|R \cap N_i^-| \leq df$. Such sets R and F exist because in-degree of process i is at most $(d + 1)f$. L, R, C, F thus defined form a partition of \mathcal{V} .

Now, $f > 0$ and $d \geq 1$, and $|L \cup C| = 1$. Thus, there can be at most 1 link from $L \cup C$ to any process in R , and $1 \leq df$. Therefore, $L \cup C \not\xrightarrow{df} R$. Also, because $C = \emptyset$, $|(R \cup C) \cap N_i^-| = |R \cap N_i^-| \leq df$. Thus, there can be at most df links from $R \cup C$ to process i , which is the only process in $L = \{i\}$. Therefore, $R \cup C \not\xrightarrow{df} L$. Thus, the above partition of \mathcal{V} does not satisfy Condition SC. This is a contradiction. \square

B Proof of Lemma 4

Before presenting the proof of Lemma 4, we introduce some terminology.

Definition 6 Graph decomposition: Let H be a directed graph. Partition graph H into strongly connected components, H_1, H_2, \dots, H_h , where h is a non-zero integer dependent on graph H , such that

- every pair of processes **within** the same strongly connected component has directed paths in H to each other, and
- for each pair of processes, say i and j , that belong to two **different** strongly connected components, either i does not have a directed path to j in H , or j does not have a directed path to i in H .

Construct a graph H^* wherein each strongly connected component H_k above is represented by vertex c_k , and there is an edge from vertex c_k to vertex c_l only if the processes in H_k have directed paths in H to the processes in H_l .

It is known that the decomposition graph H^* is a directed *acyclic* graph [1].

Definition 7 Source component: Let H be a directed graph, and let H^* be its decomposition as per Definition 6. Strongly connected component H_k of H is said to be a source component if the corresponding vertex c_k in H^* is not reachable from any other vertex in H^* .

Lemma 4 Suppose that graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition SC, and $\mathcal{F} \subset \mathcal{V}$. Then, in any reduced graph $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$, there exists a process that has a directed path to all the remaining processes in $\mathcal{V}_{\mathcal{F}}$.

Proof: Suppose that graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition SC. We first prove that the reduced graph $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$ contains exactly one *source component*.

Since $|\mathcal{F}| < |\mathcal{V}|$, reduced graph $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$ contains at least one process; therefore, at least one source component must exist in the reduced graph H . (If H consists of a single strongly connected component, then that component is trivially a source component.)

So it remains to prove that $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$ cannot contain more than one source component. The proof is by contradiction.

Suppose that the decomposition of $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$ contains at least two source components. Let the sets of processes in two such source components of the reduced graph H be denoted as L and R , respectively. Let $C = \mathcal{V}_{\mathcal{F}} - L - R = \mathcal{V} - \mathcal{F} - L - R$. Observe that \mathcal{F}, L, C, R form a partition of the processes in \mathcal{V} . Since L is a source component in the reduced graph $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$, there are no directed links in $\mathcal{E}_{\mathcal{F}}$ from any process in $C \cup R$ to the processes in L . Similarly, since R is a source component in the reduced graph H , there are no directed links in $\mathcal{E}_{\mathcal{F}}$ from any process in $L \cup C$ to the processes in R . These observations, together with the manner in which $\mathcal{E}_{\mathcal{F}}$ is defined, imply that (i) there are at most df links in \mathcal{E} from the processes in $C \cup R$ to each process in L , and (ii) there are at most df links in \mathcal{E} from the processes in $L \cup C$ to each process in R . Therefore, in graph $G(\mathcal{V}, \mathcal{E})$, $C \cup R \not\stackrel{df}{\rightarrow} L$ and $L \cup C \not\stackrel{df}{\rightarrow} R$. This violates Condition SC, resulting in a contradiction. Thus, we have proved that $H(\mathcal{V}_{\mathcal{F}}, \mathcal{E}_{\mathcal{F}})$ must contain exactly one source component.

Consider any process in the unique source component, say process s . By definition of a strongly connected component, process s has directed paths to all the processes in the source component using the edges in $\mathcal{E}_{\mathcal{F}}$. Also, by the uniqueness of the source component, all other strongly connected components in H (if any exist) are **not** source components, and hence reachable from the source component the edges in $\mathcal{E}_{\mathcal{F}}$. Therefore, process s also has paths to all the processes in $\mathcal{V}_{\mathcal{F}}$ that are outside the source component as well. Therefore, process s has paths to all the process in $\mathcal{V}_{\mathcal{F}}$. This proves the lemma. \square

The above proof shows that, if Condition SC is true, then each reduced graph contains exactly one source component. It is also possible to show that, if each reduced graph H contains exactly one source component, then Condition SC is satisfied.

C Proof of Lemma 5

Recall that \mathcal{F} is actual set of faults in a given execution of the proposed algorithm, and $|\mathcal{F}| = \psi$. As noted before, without loss of generality, we assume that processes 1 through $n - \psi$ are fault-free, and rest are faulty. To simplify the terminology, the definition below assumes a certain iteration index $t \geq 1$.

Definition 8 χ -dependence: For a constant χ , $0 \leq \chi \leq 1$, a point \mathbf{r} in the convex hull of $\{\mathbf{v}_i[t - 1] \mid 1 \leq i \leq n - \psi\}$ is said to be χ -dependent on process k if there exist constants α_i , $1 \leq i \leq n - \psi$, such that $0 \leq \alpha_i \leq 1$, $\sum_{1 \leq k \leq n - \psi} \alpha_i = 1$, and

$$\alpha_k \geq \chi$$

such that

$$\mathbf{r} = \sum_{1 \leq i \leq n - \psi} \alpha_i \mathbf{v}_i[t - 1]$$

α_i is said to be the **weight** of $\mathbf{v}_i[t - 1]$ in the above convex combination.

Lemma 10 Let $P \subseteq \mathcal{V} - \mathcal{F}$ be a non-empty subset of fault-free processes. Any point \mathbf{r} in the convex hull of $\{\mathbf{v}_j[t - 1] \mid j \in P\}$ is $\frac{1}{n}$ -dependent on at least one fault-free process in P .

Proof: Recall that we assume processes 1 through $n - \psi$ to be fault-free, and the remaining processes to be faulty. Any point \mathbf{r} in the convex hull of the state of fault-free processes in P can be written as their convex combination. Since there are at most n fault-free processes in P , and their weights in the convex combination add to 1, at least one of the weights must be $\geq \frac{1}{n}$, proving the lemma. \square

Definition 9 Points in multiset R are said to be collectively χ -dependent on processes in set P , if for each $p \in P$, there exists $\mathbf{r} \in R$ such that \mathbf{r} is χ -dependent on p .

Lemma 5 Suppose that graph $G(\mathcal{V}, \mathcal{E})$ satisfies Condition SC. Then the state updates performed by the fault-free processes in the t -th iteration ($t \geq 1$) of Algorithm **Byz-Iter** can be expressed as

$$\mathbf{v}[t] = \mathbf{M}[t] \mathbf{v}[t - 1] \tag{18}$$

where $\mathbf{M}[t]$ is a $(n - \psi) \times (n - \psi)$ row stochastic matrix with the following property: there exists a reduced graph $H[t]$, and a constant β ($0 < \beta \leq 1$) that depends only on graph $G(\mathcal{V}, \mathcal{E})$, such that

$$\mathbf{M}_{ij}[t] \geq \beta$$

if $j = i$ or edge (j, i) is in $H[t]$.

Proof: We consider the case of $f = 0$ separately from $f > 0$.

- $f = 0$: When $f = 0$, all the processes are fault-free (i.e., $\mathcal{F} = \emptyset$), and $(d+1)f + 1 = 1$. In this case, there is only one reduced graph, which is identical to $G(\mathcal{V}, \mathcal{E})$. Because $(d+1)f + 1 = 1$, each multiset C used in the *Update step* of Algorithm **Byz-Iter** to compute multiset $Z_i[t]$ contains value received from exactly one incoming neighbor. (When $f = 0$, and Condition SC holds true, it is possible that exactly one process in the graph has no incoming neighbors. If some process j has no incoming neighbors, then $Z_j[t] = \emptyset$.)

For $C = \{\mathbf{x}\}$, that is, C containing a single point \mathbf{x} , the Tverberg point for $f = 0$ is \mathbf{x} as well. Thus, $|Z_i[t]| = |N_i^-|$, and $\mathbf{v}_i[t]$ is simply the average of $\mathbf{v}_i[t-1]$ and the values received from all the incoming neighbors of i , which are necessarily fault-free (because $f = 0$). Thus, $\mathbf{v}_i[t]$ is a convex combination of the elements of $\mathbf{v}[t-1]$, wherein the weight assigned to each j such that $j = i$ or $(j, i) \in \mathcal{E}$ is $\frac{1}{1+|N_i^-|}$. Since $1 + |N_i^-| \leq n$, by defining $\beta = \frac{1}{n}$, the statement of the lemma follows.

- $f > 0$: Consider a fault-free process i . Suppose that the number of faulty incoming neighbors of process i is $f_i \leq f$. When Condition SC holds, and $f > 0$, as shown in Lemma 3, each process has an in-degree of at least $(d+1)f + 1$. Therefore, for some integer $\kappa \geq 1$, let

$$|r_i[t]| = |N_i^-| = (d+1)f + \kappa = df + (f - f_i + \kappa) + f_i.$$

Recall that the *Update step* of Algorithm **Byz-Iter** enumerates suitable subsets C of multiset $r_i[t]$, and picks one *Tverberg point* corresponding to each such C . By an inductive argument we will identify κ such subsets $C_1, C_2, \dots, C_\kappa$, such that the Tverberg points added to $Z_i[t]$ corresponding to those κ subsets are collectively dependent on at least $(f+1) - f_i = f - f_i + \kappa$ fault-free incoming neighbors of process i . Let the Tverberg point added corresponding to C_j be denoted as \mathbf{z}_j .

- Consider a subset C_1 of $r_i[t]$ such that $|C_1| = (d+1)f + 1$. A Tverberg point \mathbf{z}_1 for C_1 is added to $Z_i[t]$ in the *Update step*. By the definition of a Tverberg point, there exists a partition V_1, V_2, \dots, V_{f+1} of multiset C_1 , wherein each V_j is non-empty, such that

$$\mathbf{z}_1 \in \bigcap_{1 \leq j \leq f+1} \mathcal{H}(V_j)$$

Since process i has at most f_i faulty incoming neighbors, at most f_i values in C_1 are received from faulty neighbors. Thus, at least $(f+1) - f_i = f - f_i + 1$ of the subsets in the above partition contain values received from only fault-free neighbors of process i . For each such fault-free V_k , $\mathbf{z}_1 \in \mathcal{H}(V_k)$, and by Lemma 10, \mathbf{z}_1 must be $\frac{1}{n}$ -dependent on at least one fault-free neighbor of i whose value is included in V_k . Since the V_j 's form a partition, this implies that there are $f - f_i + 1$ distinct fault-free incoming neighbors of i on which \mathbf{z}_1 is $\frac{1}{n}$ -dependent. Let $\{p_1, p_2, \dots, p_{f-f_i+1}\}$ denote $f - f_i + 1$ distinct incoming fault-free neighbors of i on which \mathbf{z}_1 is $\frac{1}{n}$ -dependent. Note that $\{p_1, p_2, \dots, p_{f-f_i+1}\}$ is a subset of the processes whose values are included in C_1 .

If $\kappa = 1$, then we have already identified the subsets C_1, \dots, C_κ as desired. If $\kappa > 1$, then we inductively identify the remaining C_i 's below.

- Suppose that $\kappa > 1$, and that we have identified subsets C_1, \dots, C_ν , where $1 \leq \nu < \kappa$ such that $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_\nu\}$ are collectively $\frac{1}{n}$ -dependent on $f - f_i + \nu$ distinct incoming fault-free neighbors of process i that form the set $\{p_1, p_2, \dots, p_{f-f_i+\nu}\}$. (The previous item proved the correctness of this assumption for $\nu = 1$.)

Pick a subset

$$C_{\nu+1} \subseteq r_i[t] - \bigcup_{j=1}^{\nu} \{\mathbf{v}_{p_j}[t-1]\},$$

such that $|C_{\nu+1}| = (d+1)f + 1$. In other words, $C_{\nu+1}$ does not contain values received from the ν neighbors in $\{p_1, p_2, \dots, p_\nu\}$ (these neighbors are fault-free by definition, and hence correctly send their state). Such a set $C_{\nu+1}$ must exist because $1 \leq \nu < \kappa$, and $|N_i^-| = (d+1)f + \kappa \geq (d+1)f + 1 + \nu$.

Note that $r_i[t]$ is a multiset, and $r_i[t] - \cup_{j=1}^\nu \{\mathbf{v}_{p_j}[t-1]\}$ is a multiset as well. As an example, if a value appears in $r_i[t]$ three times, and appears only once in $\cup_{j=1}^\nu \{\mathbf{v}_{p_j}[t-1]\}$, then that value will appear twice in $r_i[t] - \cup_{j=1}^\nu \{\mathbf{v}_{p_j}[t-1]\}$.

By an argument similar to the previous item, we can show that the Tverberg point $\mathbf{z}_{\nu+1}$ corresponding to $C_{\nu+1}$ must be $\frac{1}{n}$ -dependent on at least $f - f_i + 1$ faulty-free processes from whom the values in $C_{\nu+1}$ are received. By definition of $C_{\nu+1}$, processes p_1, \dots, p_ν are not among these $f - f_i + 1$ processes. Thus, among these $f - f_i + 1$ fault-free processes, there exists at least one fault-free incoming neighbor of i that is not included in $\{p_1, p_2, \dots, p_{f-f_i+\nu}\}$. Let us denote one such neighbor as $p_{f-f_i+\nu+1}$. Thus, we have identified set $\{p_1, p_2, \dots, p_{f-f_i+\nu+1}\}$ consisting of $f - f_i + \nu + 1$ fault-free incoming neighbors of process i such that the points in $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{\nu+1}\}$ are collectively $\frac{1}{n}$ -dependent on $\{p_1, p_2, \dots, p_{f-f_i+\nu+1}\}$.

Note that $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_\kappa\} \subseteq Z_i[t]$. The above argument inductively proves that there exist $f - f_i + \kappa$ incoming fault-free neighbors of process i , forming set $\{p_1, p_2, \dots, p_{f-f_i+\kappa}\}$ such that the points in $Z_i[t]$ are collectively $\frac{1}{n}$ -dependent on them. Now observe the following:

1. \mathbf{z}_1 is $\frac{1}{n}$ -dependent on each fault-free process in $\{p_1, p_2, \dots, p_{f-f_i+1}\}$. Then, for each j , $1 \leq j \leq f - f_i + 1$, there exists a convex combination representation of \mathbf{z}_1 in terms of elements of $\mathbf{v}[t-1]$, in which the weight of process p_j is at least $\frac{1}{n}$. By ‘‘averaging’’ over these $f - f_i + 1$ convex combination representations of \mathbf{z}_1 , we can obtain another convex combination representation of \mathbf{z}_1 in terms of the elements of $\mathbf{v}[t-1]$ in which weight of *each* process in $\{p_1, p_2, \dots, p_{f-f_i+1}\}$ is at least $\frac{1}{n(f-f_i+1)} \geq \frac{1}{n^2}$.
2. When $\kappa \geq 2$, for $2 \leq \nu \leq \kappa$, \mathbf{z}_ν is $\frac{1}{n}$ -dependent on fault-free process $p_{f-f_i+\nu}$. Thus, there exists a convex combination representation of \mathbf{z}_ν in terms of elements of $\mathbf{v}[t-1]$, in which the weight of process $p_{f-f_i+\nu}$ is at least $\frac{1}{n} \geq \frac{1}{n^2}$.

Recall that $\mathbf{v}_i[t]$ is computed as average of the points in $Z_i[t]$, where $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_\kappa\} \subseteq Z_i[t]$, and $|Z_i[t]| \leq \binom{n}{(d+1)f+1}$. Thus, the two observations above imply that there exists a convex combination representation of $\mathbf{v}_i[t]$ in terms of elements of $\mathbf{v}[t-1]$, in which the weight of each process in $\{p_1, p_2, \dots, p_{f-f_i+\kappa}\}$ is at least $\frac{1}{n^2(1+|Z_i[t]|)} \geq \frac{1}{n^2(1+\binom{n}{(d+1)f+1})}$.

$$\beta = \frac{1}{n^2 \left(1 + \binom{n}{(d+1)f+1}\right)} \quad (19)$$

and define set

$$P_i[t] = \{p_1, p_2, \dots, p_{f-f_i+\kappa}\}.$$

Note that $|N_i^- \cap (\mathcal{V} - \mathcal{F})| = (d+1)f + \kappa - f_i$. Thus,

$$|P_i[t]| = f - f_i + \kappa = |N_i^- \cap (\mathcal{V} - \mathcal{F})| - df \quad (20)$$

Recall that we chose i to be any fault-free process in $\mathcal{V} - \mathcal{F}$. Thus, for each fault-free process i , such a set $P_i[t]$ exists, where $|P_i[t]| = |N_i^- \cap (\mathcal{V} - \mathcal{F})| - df$. Therefore, for each fault-free

process i , there exists a convex combination representation of $\mathbf{v}_i[t]$ in terms of elements of $\mathbf{v}[t-1]$, in which the weight of each process in $\{i\} \cup P_i[t]$ is at least β . In particular, there exist weights α_j 's such that $\sum_{j \in \{i\} \cup N_i^-} \alpha_j = 1$, $0 \leq \alpha_j \leq 1$ for all $j \in \{i\} \cup N_i^-$, and

$$\mathbf{v}_i[t] = \sum_{j \in \{i\} \cup N_i^-} \alpha_j \mathbf{v}_j[t-1] \quad \text{and}$$

$$\alpha_j \geq \beta \text{ for } j \in \{i\} \cup P_i[t]. \quad (21)$$

Let us now define i -th row of matrix $\mathbf{M}[t]$ as follows:

- $\mathbf{M}_{ij}[t] = \alpha_j$, for $j \in \{i\} \cup N_i^-$, and
- $\mathbf{M}_{ij}[t] = 0$, otherwise.

Due to (20), the subgraph consisting of only the fault-free processes in $\mathcal{V} - \mathcal{F}$, such that each process $i \in \mathcal{V} - \mathcal{F}$ only has incoming links from the processes in $P_i[t]$ is a reduced graph. Then, defining this subgraph as $H[t]$, the lemma follows from (21). □

D Proof of Lemma 7

Lemma 7: *For any $t \geq 1$, there exists a reduced graph $H[t] \in R_{\mathcal{F}}$ such that $\beta \mathbf{H}[t] \leq \mathbf{M}[t]$, where $\mathbf{H}[t]$ is the connectivity matrix for $H[t]$.*

Proof: By Lemma 5, there exists a reduced graph $H[t]$ such that $\mathbf{M}_{ij}[t] \geq \beta$, if $j = i$ or edge (j, i) is in the reduced graph $H[t]$.

Let $\mathbf{H}[t]$ denote the connectivity matrix for reduced graph $H[t]$. Then $\mathbf{H}_{ij}[t]$ denotes the element in i -row and j -th column of $\mathbf{H}[t]$. By definition of the connectivity matrix, we know that $H_{ij}[t] = 1$ if $j = i$ or edge (j, i) is in the reduced graph; otherwise, $H_{ij}[t] = 0$.

The claim in Lemma 7 then follows from the above two observations. □