Matrix Representation of Iterative Approximate Byzantine Consensus in Directed Graphs*

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

March 8, 2012

Abstract

This paper presents a proof of correctness of an iterative approximate Byzantine consensus (IABC) algorithm for directed graphs. The iterative algorithm allows fault-free nodes to reach approximate conensus despite the presence of up to f Byzantine faults. Necessary conditions on the underlying network graph for the existence of a correct IABC algorithm were shown in our recent work [15, 16]. [15] also analyzed a specific IABC algorithm and showed that it performs correctly in any network graph that satisfies the necessary condition, proving that the necessary condition is also sufficient. In this paper, we present an alternate proof of correctness of the IABC algorithm, using a familiar technique based on transition matrices [9, 3, 17, 19].

The key contribution of this paper is to exploit the following observation: for a *given* evolution of the state vector corresponding to the state of the fault-free nodes, many alternate state transition matrices may be chosen to model that evolution correctly. For a given state evolution, we identify one approach to suitably "design" the transition matrices so that the standard tools for proving convergence can be applied to the Byzantine fault-tolerant algorithm as well. In particular, the transition matrix for each iteration is designed such that each row of the matrix contains a large enough number of elements that are bounded away from 0.

*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.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
08 MAR 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Matrix Representation of Iterative Approximate Byzantine Consensus in Directed Graphs	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Illinois at Urbana-Champaign,Department of Electrical and Computer Engineering,Urbana,IL,91801**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper presents a proof of correctness of an iterative approximate Byzantine consensus (IABC) algorithm for directed graphs. The iterative algorithm allows fault- free nodes to reach approximate conensus despite the presence of up to f Byzantine faults. Necessary conditions on the underlying network graph for the existence of a correct IABC algorithm were shown in our recent work [15, 16]. [15] also analyzed a speci c IABC algorithm and showed that it performs correctly in any network graph that satis es the necessary condition, proving that the necessary condition is also su - cient. In this paper, we present an alternate proof of correctness of the IABC algorithm using a familiar technique based on transition matrices [9, 3, 17, 19]. The key contribution of this paper is to exploit the following observation: for a given evolution of the state vector corresponding to the state of the fault-free nodes many alternate state transition matrices may be chosen to model that evolution cor- rectly. For a given state evolution, we identify one approach to suitably design" the transition matrices so that the standard tools for proving convergence can be applied to the Byzantine fault-tolerant algorithm as well. In particular, the transition matrix for each iteration is designed such that each row of the matrix contains a large enough number of elements that are bounded away from 0.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**17**	

1 Introduction

Dolev et al. [5] introduced the notion of *approximate Byzantine consensus* by relaxing the requirement of *exact* consensus [14]. The goal in approximate consensus is to allow the fault-free nodes to agree on values that are approximately equal to each other (and *not necessarily* exactly identical). In presence of Byzantine faults, while *exact* consensus is impossible in *asynchronous* systems [7], approximate consensus is achievable [5]. The notion of approximate consensus is of interest in *synchronous* systems as well, since approximate consensus can be achieved using simple distributed algorithms that do *not* require complete knowledge of the network topology [4].

In this paper, we are interested in iterative algorithms for achieving approximate Byzantine consensus in synchronous point-to-point networks that are modeled by arbitrary *directed* graphs. The *iterative approximate Byzantine consensus* (IABC) algorithms of interest have the following properties, which we will soon state more formally:

- *Initial state* of each node is equal to a real-valued *input* provided to that node.
- *Validity* condition: After each iteration of an IABC algorithm, the state of each fault-free node must remain in the *convex hull* of the states of the fault-free nodes at the end of the *previous* iteration.
- *Convergence* condition: For any $\epsilon > 0$, after a sufficiently large number of iterations, the states of the fault-free nodes are guaranteed to be within ϵ of each other.

Certain IABC algorithms have been shown to satisfy the above properties in *fully connected* graphs [5, 14], and in *arbitrary directed* graphs satisfying a tight necessary condition [15, 16]. Please refer to [15, 16] for a summary of the related work.

The main contribution of this paper is to develop an alternate proof of correctness for a IABC algorithm, which was proved correct in arbitrary graphs that satisfy a necessary condition developed in our prior work [15]. The alternate proof is based on transition matrices that capture the behavior of the IABC algorithm executed by the fault-free nodes. This work is inspired by, and borrows some matrix analysis tools from, other work that also uses transition matrices in related contexts [9, 3, 17, 19]. This paper exploits the following observation: for a *given* evolution of the state vector corresponding to the state of the fault-free nodes, many alternate state transition matrices may potentially be chosen to emulate that evolution correctly. For a given state evolution, we identify one approach to suitably "design" the transition matrices so that the standard tools can be applied to prove convergence of the Byzantine fault-tolerant algorithm in *all networks* that satisfy a necessary condition (proved in [16]) on the network communication graph. In particular, the transition matrix for each iteration is designed such that each row of the matrix contains a large enough number of elements that are bounded away from 0.

2 Network and Failure Models

Network Model: The system is assumed to be *synchronous*. The communication network is modeled as a simple *directed* graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of n nodes, and $\mathcal{E}$ is the set of directed edges between the nodes in $\mathcal{V}$. Node i can reliably transmit messages to node j if and only if the directed edge (i, j) is in $\mathcal{E}$. Each node can send messages to itself as well, however, for convenience, we *exclude self-loops* from set $\mathcal{E}$. That is, $(i, i) \notin \mathcal{E}$ for $i \in \mathcal{V}$. With a slight abuse of terminology, we will use the terms *edge* and *link* interchangeably in our presentation.

For each node i, let N_i^- be the set of nodes 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 nodes to which node 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 node can indeed send messages to itself. A necessary condition for correctness of an IABC algorithm for $f > 0$ is that $|N_i^-| > 2f$ [15].

Node j is said to be an *incoming neighbor* of node i, if $j \in N_i^-$. Similarly, j is said to be an *outgoing neighbor* of node i, if $j \in N_i^+$.

Failure Model: We consider the Byzantine failure model, with up to f nodes becoming faulty. A faulty node may *misbehave* arbitrarily. Possible misbehavior includes sending incorrect and mismatching (or inconsistent) messages to different neighbors. The faulty nodes may potentially collaborate with each other. Moreover, the faulty nodes are assumed to have a complete knowledge of the execution of the algorithm, including the states of all the nodes, contents of messages the other nodes send to each other, the algorithm specification, and the network topology.

3 Iterative Approximate Byzantine Consensus (IABC)

Each node i maintains state v_i, with $v_i[t]$ denoting the state of node i at the *end* of the t-th iteration of the algorithm. Initial state of node i, $v_i[0]$, is equal to the initial *input* provided to node i. At the *start* of the t-th iteration ($t > 0$), the state of node i is $v_i[t-1]$.

Let $\mathcal{F}$ denote the set of faulty nodes. Thus, the nodes in $\mathcal{V} - \mathcal{F}$ are non-faulty.[1]

- $U[t] = \max_{i \in \mathcal{V}-\mathcal{F}} v_i[t]$. $U[t]$ is the largest state among the fault-free nodes at the end of the t-th iteration. Since the initial state of each node is equal to its input, $U[0]$ is equal to the maximum value of the initial input at the fault-free nodes.
- $\mu[t] = \min_{i \in \mathcal{V}-\mathcal{F}} v_i[t]$. $\mu[t]$ is the smallest state among the fault-free nodes at the end of the t-th iteration. $\mu[0]$ is equal to the minimum value of the initial input at the

[1] For sets X and Y, $X - Y$ contains elements that are in X but not in Y. That is, $X - Y = \{i \mid i \in X,\ i \notin Y\}$.

fault-free nodes.

The following conditions must be satisfied by an IABC algorithm in presence of up to f Byzantine faulty nodes:

- *Validity:* $\forall t > 0, \quad \mu[t] \geq \mu[t-1] \quad \text{and} \quad U[t] \leq U[t-1]$
- *Convergence:* $\lim_{t\to\infty} U[t] - \mu[t] = 0$. Equivalently, $\lim_{t\to\infty} v_i[t] - v_j[t] = 0$, for $i, j \in \mathcal{V} - \mathcal{F}$.

An iterative algorithm is said to be *correct* if it satisfies the *validity* and *convergence* conditions. We will prove the correctness of Algorithm 1 below in all graphs that satisfy the necessary condition in Theorem 2 of [16]. The algorithm should be performed by each node i in the t-th iteration, $t \geq 1$. The faulty nodes may deviate from the algorithm specification. If a fault-free node does not receive an expected message from an incoming neighbor (in the *Receive step* below), then that message is assumed to have some default value.

Algorithm 1

Steps to be performed by node i in the t-th iteration:

1. *Transmit step:* Transmit current state $v_i[t-1]$ on all outgoing edges.
2. *Receive step:* Receive values on all incoming edges. These values form vector $r_i[t]$ of size $|N_i^-|$.
3. *Update step:* Sort the values in $r_i[t]$ in an increasing order, and eliminate the smallest f values, and the largest f values (breaking ties arbitrarily). Let $N_i^*[t]$ denote the identifiers of nodes from whom the remaining $|N_i^-| - 2f$ values were received, and let w_j denote the value received from node $j \in N_i^*[t]$.

   For convenience, define $w_i = v_i[t-1]$.

   Observe that if $j \in \{i\} \cup N_i^*[t]$ is fault-free, then $w_j = v_j[t-1]$.

   Define

$$v_i[t] \;=\; \sum_{j \in \{i\} \cup N_i^*[t]} a_i \, w_j \tag{1}$$

   where

$$a_i \;=\; \frac{1}{|N_i^-| - 2f + 1} \;=\; \frac{1}{|N_i^*[t]| + 1}$$

   Recall that $i \notin N_i^*[t]$ because $(i, i) \notin \mathcal{E}$. The "weight" of each term on the right-hand side of (1) is a_i, and these weights add to 1.

   Observe that $0 < a_i \leq 1$.

For future reference, let us define α as:

$$\alpha = \min_{i \in \mathcal{V}} \ a_i \tag{2}$$

Note that $0 < \alpha \leq 1$. Specifically, α is a positive constant that is dependent only on f and the graph $G(\mathcal{V}, \mathcal{E})$.

Similar algorithms have been proven to work correctly in *fully connected* graphs [5, 15] and *arbitrary directed* graphs satisfying the necessary condition stated in [15]. In this paper, we provide an alternate proof of correctness in such arbitrary graphs, using an alternate form of the necessary condition [16].

4 Matrix Preliminaries

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

Definition 1 *A vector is said to be* stochastic *if all the elements of the vector 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 a row stochastic matrix $\mathbf{A}$, coefficients of ergodicity $\delta(\mathbf{A})$ and $\lambda(\mathbf{A})$ are defined as [18]:

$$\delta(\mathbf{A}) := \max_{j} \ \max_{i_1, i_2} \ |\mathbf{A}_{i_1 j} - \mathbf{A}_{i_2 j}|, \tag{3}$$

$$\lambda(\mathbf{A}) := 1 - \min_{i_1, i_2} \sum_{j} \min(\mathbf{A}_{i_1 j} \ , \mathbf{A}_{i_2 j}). \tag{4}$$

It is easy to see that $0 \leq \delta(\mathbf{A}) \leq 1$ and $0 \leq \lambda(\mathbf{A}) \leq 1$, and that the rows are all identical if and only if $\delta(\mathbf{A}) = 0$. Additionally, $\lambda(\mathbf{A}) = 0$ if and only if $\delta(\mathbf{A}) = 0$.

The next result from [8] establishes a relation between the coefficient of ergodicity $\delta(\cdot)$ of a product of row stochastic matrices, and the coefficients of ergodicity $\lambda(\cdot)$ of the individual matrices defining the product.

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

$$\delta(\mathbf{Q}(1)\mathbf{Q}(2) \cdots \mathbf{Q}(p)) \ \leq \ \Pi_{i=1}^{p} \ \lambda(\mathbf{Q}(i)). \tag{5}$$

Claim 1 is proved in [8]. It implies that if, for all i, $\lambda(\mathbf{Q}(i)) \leq 1-\gamma$ for some $\gamma > 0$, then $\delta(\mathbf{Q}(1), \mathbf{Q}(2) \cdots \mathbf{Q}(p))$ will approach zero as p approaches ∞.

Definition 2 *A row stochastic matrix* $\mathbf{H}$ *is said to be a* scrambling *matrix, if* $\lambda(\mathbf{H}) < 1$ [8, 18].

In a scrambling matrix $\mathbf{H}$, since $\lambda(\mathbf{H}) < 1$, for each pair of rows i_1 and i_2, there exists a column j (which may depend on i_1 and i_2) such that $\mathbf{H}_{i_1 j} > 0$ and $\mathbf{H}_{i_2 j} > 0$, and vice-versa [8, 18]. As a special case, if any one column of a row stochastic matrix $\mathbf{H}$ contains only non-zero elements that are lower bounded by some constant $\gamma > 0$, then $\mathbf{H}$ must be scrambling, and $\lambda(\mathbf{H}) \leq 1-\gamma$.

5 Matrix Representation of Algorithm 1

Recall that $\mathcal{F}$ is the set of faulty nodes. Let $|\mathcal{F}| = \phi$. Without loss of generality, suppose that nodes 1 through $(n-\phi)$ are fault-free, and if $\phi > 0$, nodes $(n-\phi+1)$ through n are faulty.

Denote by $\mathbf{v}[0]$ the column vector consisting of the initial states of all the *fault-free* nodes. Denote by $\mathbf{v}[t]$, where $t \geq 1$, the column vector consistsing of the states of all the *fault-free* nodes at the end of the t-th iteration, $t \geq 1$. The i-th element of vector $\mathbf{v}[t]$ is state $v_i[t]$. The size of the column vector $\mathbf{v}[t]$ is $(n-\phi)$.

Claim 2 *We can express the iterative update of the state of a fault-free node* i $(1 \leq i \leq n-\phi)$ *performed in (1) using the matrix form in (6) below, where* $\mathbf{M}_i[t]$ *satisfies the following four conditions.*

$$v_i[t] \;=\; \mathbf{M}_i[t]\ \mathbf{v}[t-1] \tag{6}$$

In addition to t*, the row vector* $\mathbf{M}_i[t]$ *may depend on the state vector* $\mathbf{v}[t-1]$ *as well as the behavior of the faulty nodes in* $\mathcal{F}$*. For simplicity, the notation* $\mathbf{M}_i[t]$ *does not explicitly represent this dependence.*

1. $\mathbf{M}_i[t]$ *is a* stochastic *row vector of size* $(n-\phi)$*. Thus,* $\mathbf{M}_{ij}[t] \geq 0$*, for* $1 \leq j \leq n-\phi$*, and*
$$\sum_{1 \leq j \leq n-\phi} \mathbf{M}_{ij}[t] \;=\; 1$$
2. $\mathbf{M}_{ii}[t]$ *equals* a_i *defined in Algorithm 1. Recall that* $a_i \geq \alpha$*.*
3. $\mathbf{M}_{ij}[t]$ *is non-zero* **only if** $(j,i) \in \mathcal{E}$ *or* $j = i$*.*
4. *At least* $|N_i^- \cap (\mathcal{V}-\mathcal{F})| - f + 1$ *elements in* $\mathbf{M}_i[t]$ *are lower bounded by some constant* $\beta > 0$*, to be defined later (*β *is independent of* i*). Note that* $N_i^- \cap (\mathcal{V}-\mathcal{F})$ *is the set of fault-free incoming neighbors of node* i*.*

Proof: The proof of this claim is presented in Section 5.1 below. The last condition above plays an important role in the proof, and the main contribution of this paper is to "design" $\mathbf{M}_i[t]$ to make this condition true. □

By "stacking" (6) for different i, $1 \leq i \leq n-\phi$, we can represent the state update for all the fault-free nodes together using (7) below, where $\mathbf{M}[t]$ is a $(n-\phi) \times (n-\phi)$ matrix, with its i-th row being equal to $\mathbf{M}_i[t]$ in (6).

$$\mathbf{v}[t] \quad = \quad \mathbf{M}[t]\ \mathbf{v}[t-1] \tag{7}$$

The four properties of $\mathbf{M}_i[t]$ imply that $\mathbf{M}[t]$ is a row stochastic matrix with a non-zero diagonal. Also, the i-th row of $\mathbf{M}[t]$ contains $|N_i^- \cap (\mathcal{V}-\mathcal{F})| - f + 1$ elements lower bounded by β (β will be defined later). This property of $\mathbf{M}[t]$ turns out to be important in proving convergence of Algorithm 1.

$\mathbf{M}[t]$ is said to be a *transition matrix*.

By repeated application of (7), we obtain:

$$\mathbf{v}[t] \quad = \quad \left(\Pi_{i=1}^{t}\mathbf{M}[i]\right)\ \mathbf{v}[0]$$

5.1 Correctness of Claim 2

Figure 1 illustrates the various sets used here. Some of the sets in this figure are not yet defined, and will be defined later in the paper.

We prove the correctness of Claim 2 by constructing $\mathbf{M}_i[t]$ for $1 \leq i \leq n-\phi$ that satisfies the conditions in Claim 2. Recall that nodes 1 through $n-\phi$ are fault-free, and the remaining ϕ nodes ($\phi \leq f$) are faulty.

Consider a fault-free node i performing the *Update step* in Algorithm 1. Recall that the largest f and the smallest f values are eliminated from $r_i[t]$. Let us denote by L and S, respectively, the set of nodes[2] from whom the largest f values and the smallest f values were received by node i in iteration t. Thus, $|L| = |S| = f$, $N_i^*[t] = N_i^- - (L \cup S)$, and $|N_i^*[t]| = |N_i^- - (L \cup S)| = |N_i^-| - 2f$.

For any set of nodes X here, let δ_X and g_X respectively denote the number of faulty nodes, and the number of fault-free nodes, in set X. For instance, δ_L and g_L denote, respectively, the number of faulty and fault-free nodes in set L. Thus,

$$\delta_L + g_L \ = \ \delta_S + g_S \ = \ f$$

Let

$$\delta = |N_i^- \cap \mathcal{F}|$$

[2]Although L and S may be different for each t, for simplicity, we do not explicitly represent this dependence on t in the notations L and S.

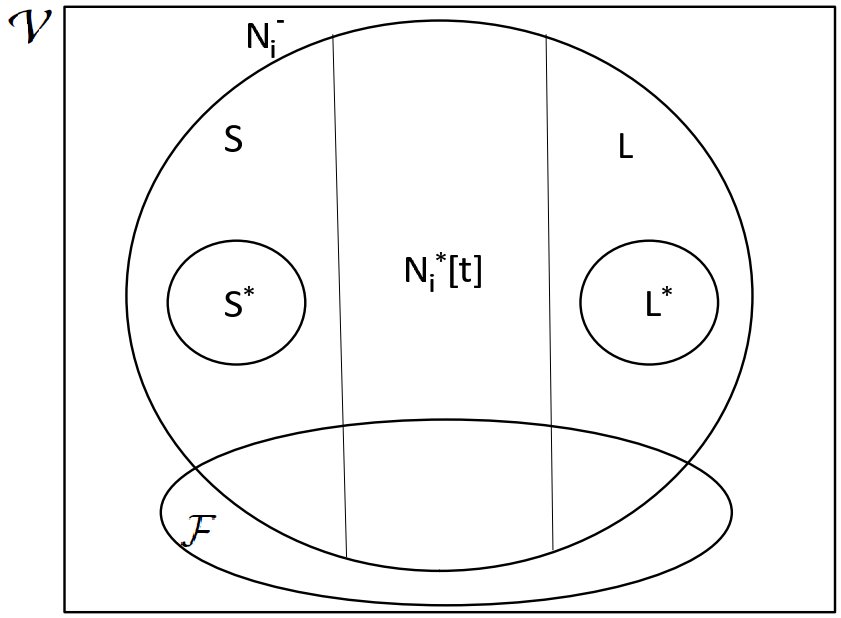

Figure 1: Illustration of sets $\mathcal{V}$, $\mathcal{F}$, N_i^-, $N_i^*[t]$, L^* and S^*

That is, the number of faulty incoming neighbors of node i is denoted as δ. Therefore, $\delta \leq \phi \leq f$, and

$$\delta = \delta_L + \delta_S + \delta_{N_i^*[t]}$$

Then, it follows that

$$g_L = f - \delta_L = \delta_S + \delta_{N_i^*[t]} + (f - \delta), \text{ and} \quad (8)$$

$$g_S = f - \delta_S = \delta_L + \delta_{N_i^*[t]} + (f - \delta) \quad (9)$$

For fault-free node i, we now define the elements of row $\mathbf{M}_i[t]$. We consider two cases separately: (i) $f - \delta + \delta_{N_i^*[t]} = 0$, and (ii) $f - \delta + \delta_{N_i^*[t]} > 0$.

5.1.1 $f - \delta + \delta_{N_i^*[t]} = 0$

We know that $(f - \delta) \geq 0$ and $\delta_{N_i^*[t]} \geq 0$. Therefore, $f - \delta + \delta_{N_i^*[t]} = 0$ implies that $f = \delta$ and $\delta_{N_i^*[t]} = 0$. Thus, in this case, all the nodes in $N_i^*[t]$ are fault-free.

- For each $j \in \{i\} \cup N_i^*[t]$, define $\mathbf{M}_{ij}[t] = a_i$. Element $\mathbf{M}_{ij}[t]$ corresponds to the term $a_i w_j$ in (1).

  Recall that $a_i \geq \alpha$, and that each node in $\{i\} \cup N_i^*[t]$ in this case is fault-free.

- For each j such that $j \in \mathcal{V} - \mathcal{F}$ and $j \notin \{i\} \cup N_i^*[t]$, define $\mathbf{M}_{ij}[t] = 0$.

Observe that with the above definition of elements of $\mathbf{M}_i[t]$,

$$\mathbf{M}_i[t]\mathbf{v}[t-1] = \sum_{k \in \{i\} \cup N_i^*[t]} a_i w_k$$

In the above procedure, we have set $|N_i^*[t]| + 1$ elements of $\mathbf{M}_i[t]$ equal to a_i (recall that $a_i \geq \alpha$).

Now, because $\delta = f$ and $|N_i^*[t]| = |N_i^-| - 2f$, we have $|N_i^- \cap (\mathcal{V} - \mathcal{F})| - f + 1 = |N_i^-| - \delta - f + 1 = |N_i^-| - 2f + 1 = |N_i^*[t]| + 1$. Also, in this case $a_i = 1/(|N_i^*[t]| + 1)$. Thus, it should be easy to see that the conditions in Claim 2 are satisfied by defining $\beta = \alpha$.

5.1.2 $f - \delta + \delta_{N_i^*[t]} > 0$

Since $0 \leq \delta_{N_i^*[t]} \leq \delta \leq f$, $f - \delta + \delta_{N_i^*[t]} > 0$ implies that $f > 0$. When $f > 0$, the necessary condition in [15] implies that $|N_i^-| \geq 2f + 1$. Therefore, the set $N_i^*[t]$ is non-empty. As per (1), each node $k \in N_i^*[t]$ contributes $a_i\, w_k$ to the new state $v_i[t]$ of node i. We will define elements of $\mathbf{M}_i[t]$ to account for the contribution of each node $k \in N_i^*[t]$.

Define subsets L^* and S^* such that $L^* \subseteq L$, $S^* \subseteq S$, $L^* \cap \mathcal{F} = S^* \cap \mathcal{F} = \Phi$, and $|L^*| = |S^*| = f - \delta + \delta_{N_i^*[t]}$. That is, sets L^* and S^* are subsets of L and S, respectively, each of size $f - \delta + \delta_{N_i^*[t]}$, and containing only fault-free nodes. Expressions (8) and (9) for g_L and g_S imply that such subsets exist.

Let

$$L^* = \{l_j \mid 1 \leq j \leq f - \delta + \delta_{N_i^*[t]}\}$$

and

$$S^* = \{s_j \mid 1 \leq j \leq f - \delta + \delta_{N_i^*[t]}\}.$$

Consider any node $k \in N_i^*[t]$. For each j, $1 \leq j \leq f - \delta + \delta_{N_i^*[t]}$,

$$v_{s_j}[t-1] \leq w_k \leq v_{l_j}[t-1]$$

Therefore, we can find weights $\lambda_{k,j} \geq 0$ and $\psi_{k,j} \geq 0$ such that

$$\lambda_{k,j} + \psi_{k,j} \;=\; 1$$

and

$$w_k \;=\; \lambda_{k,j}\, v_{l_j}[t-1] \;+\; \psi_{k,j}\, v_{s_j}[t-1]$$

Clearly, at least one of the weights $\lambda_{k,j}$ and $\psi_{k,j}$ must be $\geq 1/2$. Now, observe that

$$a_i\, w_k \;=\; \frac{a_i}{f - \delta + \delta_{N_i^*[t]}} \sum_{1 \leq j \leq f - \delta + \delta_{N_i^*[t]}} \left(\lambda_{k,j}\, v_{l_j}[t-1] + \psi_{k,j}\, v_{s_j}[t-1]\right) \qquad (10)$$

The above equality is true independent of whether k is fault-free or faulty. We will later use the above equality for the case when k is a faulty node. When k is fault-free,

$$w_k = v_k[t-1],$$

and we can similarly obtain the equality below.

$$a_i w_k \;=\; \frac{a_i}{2} v_k[t-1] \;+\; \frac{a_i}{2(f-\delta+\delta_{N_i^*[t]})} \sum_{1 \le j \le f-\delta+\delta_{N_i^*[t]}} \left(\lambda_{k,j}\, v_{l_j}[t-1] + \psi_{k,j}\, v_{s_j}[t-1]\right) \tag{11}$$

We now use (1), (10) and (11) to define elements of $\mathbf{M}_i[t]$ in the following four cases:

- **Case 1: Node i**
  Define $\mathbf{M}_{ii}[t] = a_i$. This is obtained by observing in (1) that the contribution of node i to the new state $v_i[t]$ is $a_i w_i = a_i v_i[t-1]$.

- **Case 2: Fault-free nodes in $N_i^*[t]$**
  For each $k \in N_i^*[t] \cap (\mathcal{V}-\mathcal{F})$, define $\mathbf{M}_{ik}[t] = \frac{a_i}{2}$. This choice is motivated by (11) wherein the contribution of node k to $a_i w_k$ is $\frac{a_i}{2} v_k[t-1]$. In Case 2, $|N_i^*[t] \cap (\mathcal{V}-\mathcal{F})| = |N_i^-| - \delta$ elements of $\mathbf{M}_i[t]$ are defined.

- **Case 3: Nodes in L^* and S^***
  For $1 \le j \le f-\delta+\delta_{N_i^*[t]}$, consider $l_j \in L^*$. In this case,

  $$\mathbf{M}_{il_j}[t] \;=\; \sum_{k \in N_i^*[t] \cap \mathcal{F}} \frac{a_i}{f-\delta+\delta_{N_i^*[t]}} \lambda_{k,j} \;+\; \sum_{k \in N_i^*[t] \cap (\mathcal{V}-\mathcal{F})} \frac{a_i}{2(f-\delta+\delta_{N_i^*[t]})} \lambda_{k,j}$$

  Similarly, for $1 \le j \le f-\delta+\delta_{N_i^*[t]}$, consider $s_j \in S^*$. In this case,

  $$\mathbf{M}_{is_j}[t] \;=\; \sum_{k \in N_i^*[t] \cap \mathcal{F}} \frac{a_i}{f-\delta+\delta_{N_i^*[t]}} \psi_{k,j} \;+\; \sum_{k \in N_i^*[t] \cap (\mathcal{V}-\mathcal{F})} \frac{a_i}{2(f-\delta+\delta_{N_i^*[t]})} \psi_{k,j}$$

  These expressions are obtained by summing (10) and (11), respectively, over the faulty and fault-free nodes in $N_i^*[t]$, and then identifying the contribution of each node in L^* and S^* to this sum. Recall the earlier observation that at least one of $\lambda_{k,j}$ and $\psi_{k,j}$ must be $\ge 1/2$ for each pair k, j where $k \in N_i^*[t]$ and $1 \le j \le f-\delta+\delta_{N_i^*[t]}$. Therefore, it follows that at least $f-\delta+\delta_{N_i^*[t]}$ elements of $\mathbf{M}_i[t]$ defined in Case 3 must be $\ge \frac{a_i}{4(f-\delta+\delta_{N_i^*[t]})}$.

- **Case 4: Nodes in $(\mathcal{V}-\mathcal{F}) - (\{i\} \cup N_i^*[t] \cup L^* \cup S^*)$**
  These fault-free nodes have not yet been considered in Cases 1, 2 and 3. For each node $k \in (\mathcal{V}-\mathcal{F}) - (\{i\} \cup N_i^*[t] \cup L^* \cup S^*)$, we assign $\mathbf{M}_{ik}[t] = 0$.

Observe that above the definition of the elements of $\mathbf{M}_i[t]$ ensures that

$$\sum_{j \in \{i\} \cup N_i^*[t]} a_i w_j \ = \ \mathbf{M}_i[t]\mathbf{v}[t-1]$$

However, the contribution by the faulty nodes in $N_i^*[t]$ in (1) is now replaced by an equivalent contribution by the nodes in L^* and S^*.

Now let us verify that the four conditions in Claim 2 hold for the above assignments to the elements of $\mathbf{M}_i[t]$.

1. Observe that all the elements of $\mathbf{M}_i[t]$ are non-negative. Case 1 specifies just $\mathbf{M}_{ii}[t] = a_i$. The elements of $\mathbf{M}_i[t]$ specified in Case 2 add up to

   $$\frac{a_i}{2} \ |N_i^*[t] \cap (\mathcal{V} - \mathcal{F})|$$

   Recall that for each j, $1 \le j \le (f - \delta + \delta_{N_i^*[t]})$, $\lambda_{k,j} + \psi_{k,j} = 1$ for $k \in N_i^*[t]$. Therefore, when added over all $k \in N_i^*[t]$ and $1 \le j \le (f - \delta + \delta_{N_i^*[t]})$, the elements of $\mathbf{M}_i[t]$ specified in Case 3 add up to

   $$a_i \ |N_i^*[t] \cap \mathcal{F}| \ + \ \frac{a_i}{2} \ |N_i^*[t] \cap (\mathcal{V} - \mathcal{F})|$$

   Therefore, when all the elements of $\mathbf{M}_i[t]$ defined in Cases 1, 2 and 3 are added together, we get

   $$a_i \ + \ a_i \, |N_i^*[t] \cap \mathcal{F}| \ + \ a_i |N_i^*[t] \cap (\mathcal{V} - \mathcal{F})| \ = \ a_i(|N_i^*[t]| + 1) \ = \ 1$$

   because $a_i = 1/(|N_i^*[t]| + 1)$. Now observe that the elements specified in Cases 1, 2 and 3 are clearly ≤ 1. In the expression for $\mathbf{M}_{il_j}[t]$ in Case 3, observe that the two summations on the right side together contain $|N_i^*[t]|$ terms, and in these terms, observe that $\lambda_{k,j} \le 1$, $f - \delta + \delta_{N_i^*[t]} \ge 1$ and $a_i = \frac{1}{|N_i^*[t]|+1}$. Therefore, $\mathbf{M}_{il_j}[t] < 1$. Similarly, we can show that $\mathbf{M}_{is_j}[t] < 1$ as well.

   Thus, we have shown that $\mathbf{M}_i[t]$ is a stochastic vector.

2. $\mathbf{M}_{ii}[t] = a_i$ as specified in Case 1.

3. Since $\mathbf{M}_{ij}[t]$ is defined to be non-zero only in Cases 1, 2 and 3, which consider the nodes only in $\{i\} \cup N_i^-$, it follows that $\mathbf{M}_{ij}[t]$ is non-zero *only if* $(j, i) \in \mathcal{E}$ or $j = i$.

4. Cases 1 and 2 together set $1 + |N_i^*[t] \cap (\mathcal{V} - \mathcal{F})| = 1 + |N_i^*[t]| - \delta_{N_i^*[t]}$ elements of $\mathbf{M}_i[t]$ to be $\ge a_i/2$. We observed earlier that Case 3 results in at least $f - \delta + \delta_{N_i^*[t]}$ elements of $\mathbf{M}_i[t]$ being $\ge \frac{a_i}{4(f-\delta+\delta_{N_i^*[t]})}$. Also, observe that the elements of $\mathbf{M}_i[t]$ specified in Cases 1 and 2 are distinct from those specified in Case 3, and that $\frac{a_i}{2} \ge \frac{a_i}{4(f-\delta+\delta_{N_i^*[t]})}$. Thus, overall, at least

   $$(1 + |N_i^*[t]| - \delta_{N_i^*[t]}) \ + \ f - \delta + \delta_{N_i^*[t]} \ = \ |N_i^*[t]| + f - \delta + 1 \ = \ |N_i^-| - f - \delta + 1$$
   $$= \ |N_i^- \cap (\mathcal{V} - \mathcal{F})| - f - 1$$

elements of $\mathbf{M}_i[t]$ are set $\geq \frac{a_i}{4(f-\delta+\delta_{N_i^*[t]})}$. Derivation of the above equation uses the facts that $|N_i^*[t]| = |N_i^-| - 2f$ and $|N_i^- \cap (\mathcal{V} - \mathcal{F})| = |N_i^-| - \delta$. Then by defining β as below, condition 4 in Claim 2 holds true.

$$\beta = \frac{\alpha}{4(f - \delta + \delta_{N_i^*[t]})}$$

Therefore, Claim 2 is proved correct.

5.2 Correspondence Between Sufficiency Condition and M[t]

Let us define set $R_{\mathcal{F}}$ of subgraphs of $G(\mathcal{V}, \mathcal{E})$ as follows.

$$\begin{aligned} R_{\mathcal{F}} \quad = \quad & \{H \mid H \text{ is obtained by removing all the faulty} \\ & \text{nodes from } \mathcal{V} \text{ along with their edges, and then} \\ & \text{removing any additional } f \text{ incoming edges} \\ & \text{at each fault-free node}\} \end{aligned} \tag{12}$$

Thus, $\mathcal{V} - \mathcal{F}$ is the set of nodes in each graph in $R_{\mathcal{F}}$.

Let τ denote $|R_{\mathcal{F}}|$. τ depends on $\mathcal{F}$ and the underlying network, and it is finite.

Claim 3 *Suppose that graph $G(\mathcal{V}, \mathcal{E})$ satisfies the necessary condition in Theorem 2 in [16]. Then it follows that in each $H \in R_{\mathcal{F}}$, there exists at least one node that has directed paths to all the nodes in H (consisting of the edges in H).*

Proof: The proof follows from Theorem 2 of [16]. □

In this discussion, let us denote a graph by an italic upper case letter, and the corresponding *connectivity matrix* using the same letter in boldface upper case. Thus, $\mathbf{H}$ will denote the connectivity matrix for graph $H \in R_{\mathcal{F}}$; $\mathbf{H}$ is defined as follows: (i) for $1 \leq i, j \leq n - \phi$, if there is a directed link from node j to node i in graph H then $\mathbf{H}_{ij} = 1$, and (ii) $\mathbf{H}_{ii} = 1$ for $1 \leq i \leq n - \phi$. Note that in our notation, the i-th row of $\mathbf{H}$ (that is, $\mathbf{H}_i$) corresponds to the incoming links at node i, and the self-loop at node i. The connectivity matrix $\mathbf{H}$ for any $H \in R_{\mathcal{F}}$ has a non-zero diagonal.

Lemma 1 *For any $H \in R_{\mathcal{F}}$, $\mathbf{H}^{n-\phi}$ has at least one non-zero column.*

Proof: By Claim 3, in graph H there exists at least one node, say node k, that has a directed path in H to all the remaining nodes in H. Since the length of the path from k to

any other node in H can contain at most $n-\phi-1$ directed edges, the k-th column of matrix $\mathbf{H}^{n-\phi}$ will be non-zero.[3] □

Definition 3 *We will say that an element of a matrix is "non-trivial" if it is lower bounded by β.*

Definition 4 *For matrices $\mathbf{A}$ and $\mathbf{B}$ of identical size, and a scalar γ, $\mathbf{A} \leq \gamma\,\mathbf{B}$ provided that $\mathbf{A}_{ij} \leq \gamma\,\mathbf{B}_{ij}$ for all i, j.*

Lemma 2 *For any $t \geq 1$, there exists a graph $H[t] \in R_{\mathcal{F}}$ such that $\beta\,\mathbf{H}[t] \leq \mathbf{M}[t]$.*

Proof: Observe that the i-th row of the transition matrix $\mathbf{M}[t]$ corresponds to the state update performed at fault-free node i. Recall from Claim 2 that the $\mathbf{M}_{ij}$ is non-zero **only if** link $(j, i) \in \mathcal{E}$. Also, by Claim 2, $\mathbf{M}_i[t]$ (i.e., the i-th row of $\mathbf{M}[t]$) contains at least $|N_i^- \cap (\mathcal{V}-\mathcal{F})| - f + 1$ *non-trivial* elements corresponding to **fault-free** incoming neighbors of node i and itself (i.e., the diagonal element).

Now observe that, for any subgraph $H \in R_{\mathcal{F}}$, i-th row of $\mathbf{H}$ contains exactly $|N_i^- \cap (\mathcal{V}-\mathcal{F})| - f + 1$ non-zero elements, including the diagonal element.

Considering the above two observations, and the definition of set $R_{\mathcal{F}}$, the lemma follows. □

6 Correctness of Algorithm 1

The proof below uses techniques also applied in prior work (e.g., [9, 3, 17, 19]), with some similarities to the arguments used in [17, 19].

Lemma 3 *In the product below of $\mathbf{H}[t]$ matrices for consecutive $\tau(n-\phi)$ iterations, at least one column is non-zero.*

$$\Pi_{t=z}^{z+\tau(n-\phi)-1}\,\mathbf{H}[t]$$

[3] That is, all the elements of the column will be non-zero (more precisely, positive, since the elements of matrix $\mathbf{H}$ are non-negative). Also, such a non-zero column will exist in $\mathbf{H}^{n-\phi-1}$ too. We use the loose bound of $n-\phi$ to simplify the presentation.

Proof: Since the above product consists of $\tau(n-\phi)$ matrices in $R_{\mathcal{F}}$, at least one of the τ distinct connectivity matrices in $R_{\mathcal{F}}$, say matrix $\mathbf{H}_*$, will appear in the above product at least $n-\phi$ times.

Now observe that: (i) By Lemma 1, $\mathbf{H}_*^{n-\phi}$ contains a non-zero column, say the k-th column is non-zero, and (ii) all the $\mathbf{H}[t]$ matrices in the product contain a non-zero diagonal. These two observations together imply that the k-th column in the above product is non-zero. □

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

$$\mathbf{Q}(i) \ = \ \Pi_{t=(i-1)\tau(n-\phi)+1}^{i\tau(n-\phi)} \mathbf{M}[t]$$

Observe that

$$\mathbf{v}[k\tau(n-\phi)] \ = \ \left(\Pi_{i=1}^{k} \mathbf{Q}(i)\right) \mathbf{v}[0] \tag{13}$$

Lemma 4 *For $i \geq 1$, $\mathbf{Q}(i)$ is a scrambling row stochastic matrix, and $\lambda(\mathbf{Q}(i))$ is bounded from above by a constant smaller than 1.*

Proof: $\mathbf{Q}(i)$ is a product of row stochastic matrices ($\mathbf{M}[t]$), therefore, $\mathbf{Q}(i)$ is row stochastic.

From Lemma 2, for each t,

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

Therefore,

$$\beta^{\tau(n-\phi)} \ \Pi_{t=(i-1)\tau(n-\phi)+1}^{i\tau(n-\phi)} \mathbf{H}[t] \ \leq \ \mathbf{Q}(i)$$

By using $z = (i-1)(n-\phi)+1$ in Lemma 3, we conclude that the matrix product on the left side of the above inequality contains a non-zero column. Therefore, $\mathbf{Q}(i)$ contains a non-zero column as well. Therefore, $\mathbf{Q}(i)$ is a scrambling matrix.

Observe that $\tau(n-\phi)$ is finite, therefore, $\beta^{\tau(n-\phi)}$ is non-zero. Since the non-zero terms in $\mathbf{H}[t]$ matrices are all 1, the non-zero elements in $\Pi_{t=(i-1)\tau(n-\phi)+1}^{i\tau(n-\phi)}\mathbf{H}[t]$ must each be ≥ 1. Therefore, there exists a non-zero column in $\mathbf{Q}(i)$ with all the elements in the column being $\geq \beta^{\tau(n-\phi)}$. Therefore $\lambda(\mathbf{Q}(i)) \leq 1-\beta^{\tau(n-\phi)}$. □

Theorem 1 *Algorithm 1 satisfies the validity and the convergence conditions.*

Proof: Since $\mathbf{v}[t] = \mathbf{M}[t]\, v[t-1]$, and $\mathbf{M}[t]$ is a row stochastic matrix, it follows that Algorithm 1 satisfies the validity condition.

By Claim 1,

$$\begin{aligned}
\lim_{t\to\infty} \delta(\Pi_{i=1}^{t}\mathbf{M}[t]) \ &\leq \ \lim_{t\to\infty} \Pi_{i=1}^{t}\lambda(\mathbf{M}[t]) && (14)\\
&\leq \ \lim_{i\to\infty} \Pi_{i=1}^{\lfloor \frac{t}{\tau(n-\phi)} \rfloor}\lambda(\mathbf{Q}(i)) && (15)\\
&= \ 0 && (16)
\end{aligned}$$

The above argument makes use of the facts that $\lambda(\mathbf{M}[t]) \leq 1$ and $\lambda(\mathbf{Q}(i)) \leq (1-\beta^{\tau(n-\phi)}) < 1$. Thus, the rows of $\Pi_{i=1}^{t}\mathbf{M}[t]$ become identical in the limit. This observation, and the fact that $\mathbf{v}[t] = (\Pi_{i=1}^{t}\mathbf{M}[i])\mathbf{v}[t-1]$ together imply that the state of the fault-free nodes satisfies the convergence condition.

Now, the validity and convergence conditions together imply that there exists a positive scalar c such that

$$\lim_{t\to\infty} \mathbf{v}[t] \;=\; \lim_{t\to\infty} \left(\Pi_{i=1}^{t}\mathbf{M}[i]\right) \mathbf{v}[0] \;=\; c\,\mathbf{1}$$

where $\mathbf{1}$ denotes a column with all its elements being 1.

□

7 Extension of Above Results

In this paper, we analyzed IABC Algorithm 1 designed for synchronous systems. Similar analysis also applies for IABC Algorithm 2 presented in [16] for asynchronous systems.

The analysis will also naturally extend to an IABC algorithm for the *partially synchronous algorithmic* model presented in [4], which assumes a bounded delay in propagation of state between neighbors, and a bounded delay between consecutive state updates at each node in the network. The generalization of Algorithm 1 to the *partially synchronous algorithmic* model will allow a node i, if performing state update in iteration t, to form vector $r_i[t]$ using the most recent known states of its incoming neighbors; these states of the neighbors may correspond to any of the prior B iterations, for some bounded B. A similar IABC algorithm can also be used in time-varying network topologies (i.e., networks wherein the set of links available in iteration t varies with t); the above analysis will then extend to time-varying topologies as well, with the algorithm performing correctly so long as the connectivity matrices for the graphs at different t jointly satisfy some reasonable properties, as in [9, 3, 17].

8 Summary

We presented a proof of validity and convergence of Algorithm 1 by expressing the algorithm in the matrix form. The main contribution of the paper is to express the algorithm in matrix form that allows us to prove its convergence under certain necessary conditions on the underlying communication graph. Thus, the proof implies that the necessary conditions are also sufficient. The key to the proof is to "design" the transition matrix for each iteration such that each row of the matrix contains a large enough number of elements that are bounded away from 0.

References

[1] A. Azadmanesh and H. Bajwa. Global convergence in partially fully connected networks (pfcn) with limited relays. In *Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE*, volume 3, pages 2022 –2025 vol.3, 2001.

[2] M. H. Azadmanesh and R. Kieckhafer. Asynchronous approximate agreement in partially connected networks. *International Journal of Parallel and Distributed Systems and Networks*, 5(1):26–34, 2002. http://ahvaz.unomaha.edu/azad/pubs/ijpdsn.asyncpart.pdf

[3] F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, Weighted gossip: Distributed averaging using non-doubly stochastic matrices, in Proc. of IEEE International Symposium on Information Theory, June 2010, pp. 1753–1757.

[4] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Optimization and Neural Computation Series. Athena Scientific, 1997.

[5] 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.

[6] A. D. Fekete. Asymptotically optimal algorithms for approximate agreement. In *Proceedings of the fifth annual ACM symposium on Principles of distributed computing*, PODC '86, pages 73–87, New York, NY, USA, 1986. ACM.

[7] 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.

[8] J. Hajnal, "Weak ergodicity in non-homogeneous Markov chains, Proceedings of the Cambridge Philosophical Society, vol. 54, pp. pp. 233–246, 1958.

[9] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001, june 2003.

[10] R. M. Kieckhafer and M. H. Azadmanesh. Low cost approximate agreement in partially connected networks. *Journal of Computing and Information*, 3(1):53–85, 1993.

[11] H. LeBlanc, H. Zhang, S. Sundaram, and X. Koutsoukos. Consensus of multi-agent networks in the presence of adversaries using only local information. *HiCoNs*, 2012.

[12] H. LeBlanc and X. Koutsoukos. Consensus in networked multi-agent systems with adversaries. *14th International conference on Hybrid Systems: Computation and Control (HSCC)*, 2011.

[13] H. LeBlanc and X. Koutsoukos. Low complexity resilient consensus in networked multi-agent systems with adversaries. *15th International conference on Hybrid Systems: Computation and Control (HSCC)*, 2012.

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

[15] N. H. Vaidya, L. Tseng, and G. Liang. Iterative approximate Byzantine consensus in arbitrary directed graphs. *CoRR*, abs/1201.4183v1 (January 2012), abs/1201.4183v2 (Februar 2012). Available from `http://arxiv.org`.

[16] N. H. Vaidya, L. Tseng, and G. Liang. Iterative approximate Byzantine consensus in arbitrary directed graphs – Part II: Synchronous and asynchronous systems. Technical report, University of Illinois at Urbana-Champaign, February 2012. `http://www.crhc.illinois.edu/wireless/papers/approx_consensus_II.pdf`

[17] N. H. Vaidya, C. N. Hadjicostis, A. D. Dominguez-Garcia. Distributed Algorithms for Consensus and Coordination in the Presence of Packet-Dropping Communication Links - Part II: Coefficients of Ergodicity Analysis Approach. September 2011. Available from `http://arxiv.org/abs/1109.6392`.

[18] J. Wolfowitz, Products of indecomposable, aperiodic, stochastic matrices, Proceedings of the American Mathematical Society, vol. 14, no. 5, pp. pp. 733–737, 1963.

[19] H. Zhang and S. Sundaram. Robustness of Information Diffusion Algorithms to Locally Bounded Adversaries. In CoRR, volume abs/1110.3843, 2011. `http://arxiv.org/abs/1110.3843`