Iterative Approximate Byzantine Consensus in Arbitrary Directed Graphs *

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

[1] Department of Electrical and Computer Engineering,
[2] Department of Computer Science, and
[3] Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Email: {nhv, ltseng3, gliang2}@illinois.edu

Technical Report

January 19, 2012

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

1 Introduction

In this paper, we explore the problem of iterative approximate Byzantine consensus in arbitrary directed graphs. In particular, we prove a necessary and sufficient condition for the existence of iterative byzantine consensus algorithms. Additionally, we use our sufficient condition to examine whether such algorithms exist for some specific graphs.

Approximate Byzantine consensus [5] is a natural extension of original Byzantine Generals (or Byzantine consensus) problem [9]. The goal in approximate consensus is to allow the fault-free nodes to agree on values that are approximately equal to each other. There exist iterative algorithms for the approximate consensus problem that work correctly in fully connected graphs [5, 12] when the number of nodes n exceeds $3f$, where f is the upper bound on the number of failures. In [6], Fekete studies the convergence rate of approximate consensus algorithms. Ben-Or et al. develop an algorithm based on Gradcast to solve approximate consensus efficiently in a fully connected network [3].

There have been attempts at achieving approximate consensus iteratively in partially connected graphs. In [8], Kieckhafer and Azadmanesh examined the necessary conditions in order to achieve "local" convergence and performed a case study on global convergence in some special graphs. Later, they extended their work to asynchronous systems [2]. In [1], Azadmanesh et al. showed how to build a special network, called Partially Fully Connected Network, in which global convergence is achieved. Srinivasan and Azadmanesh studied the application of iterative approximate consensus in data aggregation, and developed an analytical approach using Markov chains [13, 14].

In [16], Sundaram and Hadjicostis explored Byzantine-fault tolerant distributed function calculation in an arbitrary network assuming a *broadcast model*. Under the *broadcast* model, every transmission of a node is received by all its neighbors. Hence, faulty nodes can send false data, but they have to send exactly the same piece of data to all their neighbors. They proved that distributed function calculation is possible if network connectivity is at least $2f + 1$. Their algorithm maintains more "history" (a sequence of previous states) than the iterative algorithms considered in this paper.

In [18], Zhang and Sundaram studied the sufficient conditions for iterative consensus algorithm under "f-local" fault model. They also provided a construction of graphs satisfying the sufficient conditions.

LeBlanc and Koutsoukos [10] address a continuous time version of the Byzantine consensus problem in *complete* graphs. Recently, for the *broadcast* model, LeBlanc et al. have independently developed necessary and sufficient conditions for f-fault tolerant approximate consensus in arbitrary graphs [17]; in [11] they have developed some *sufficient* conditions for correctness of a class of iterative consensus algorithms.

To the best of our knowledge, characterization of *tight necessary and sufficient* conditions for iterative approximate consensus in arbitrary directed graphs in the presence of Byzantine faults under point-to-point model is still an open problem. Iterative approximate consensus algorithms without any fault tolerance capability (i.e., $f = 0$) in arbitrary graphs have been explored extensively. The proof of convergence presented in this paper is inspired by the

prior work on non-fault-tolerant algorithms [4].

2 Preliminaries

2.1 Network Model

The 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 nodes in $\mathcal{V}$. We use the terms "edge" and "link" interchangeably. We assume that $n \geq 2$, since the consensus problem for $n = 1$ is trivial. If a directed edge $(i, j) \in \mathcal{E}$, then node i can reliably transmit to node j. For convenience, we exclude self-loops from $\mathcal{E}$, although every node is allowed to send messages to itself. We also assume that all edges are authenticated, such that when a node j receives a message from node i (on edge (i, j)), it can correctly determine that the message was sent by node i. 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, $N_i^+ = \{\, j \mid (i, j) \in \mathcal{E} \,\}$. By definition, $i \notin N_i^-$ and $i \notin N_i^+$. However, we emphasize that each node can indeed send messages to itself. The network is assumed to be synchronous.

2.2 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 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 state of the other nodes in the system and a complete knowledge of specification of the algorithm.

2.3 Iterative Approximate Byzantine Consensus

We consider iterative Byzantine consensus as follows:

- Up to f nodes in the network may be Byzantine faulty.
- Each node starts with an *input*, which is assumed to be a single real number.
- 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. $v_i[0]$ denotes the initial state of node i, which is set equal to its *input*. Note that, at the start of the t-th iteration ($t > 0$), the state of node i is $v_i[t-1]$.
- The goal of an approximate consensus algorithm is to allow each node to compute an *output* in *each iteration* with the following two properties:

  - **Validity:** The output of each node is within the convex hull of the inputs at the *fault-free* nodes.
  - **Convergence:** The outputs of the different fault-free nodes converge to an identical value as $t \to \infty$.

- **Output constraint:** For the family of iterative algorithms considered in this paper, output of node i at time t is **equal to** its state $v_i[t]$.

The iterative algorithms will be implemented as follows:

- At the start of t-th iteration, $t \geq 1$, each node i sends $v_i[t-1]$ on all its outgoing links (to nodes in N_i^+).
- Denote by $r_i[t]$ the vector of values received by node i from nodes in N_i^- at time t. The size of vector $r_i[t]$ is $|N_i^-|$.
- Node i updates its state using some transition function Z_i as follows, where Z_i is part of the specification of the algorithm:

$$v_i[t] = Z_i(r_i[t], v_i[t-1])$$

  Since the inputs are real numbers, and because we impose the above *output constraint*, the state of each node in each iteration is also viewed as a real number.

The function Z_i may be dependent on the network topology. However, as seen later, for convergence, it suffices for each node i to know N_i^-.

Observe that, given the state of the nodes at time $t-1$, their state at time t is independent of the prior history. The evolution of the state of the nodes may, therefore, be modeled by a Markov chain (although we will not use that approach in this paper).

We now introduce some notations.

- Let $\mathcal{F}$ denote the set of Byzantine faulty nodes, where $|\mathcal{F}| \leq f$. Thus, the set of fault-free nodes is $\mathcal{V} - \mathcal{F}$. [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 time t). Recall that, due to the *output constraint*, the state of node i at the end of iteration t (i.e., $v_i[t]$) is also its output in iteration t.
- $\mu[t] = \min_{i \in \mathcal{V}-\mathcal{F}} v_i[t]$. $\mu[t]$ is the smallest state among the fault-free nodes at time t (we will use the phrase "at time t" interchangeably with "at the end of t-th iteration").

With the above notation, we can restate the *validity* and *convergence* conditions as follows:

- **Validity:** $\forall t > 0$, $U[t] \leq U[0]$ and $\mu[t] \geq \mu[0]$

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

- **Convergence:** $\lim_{t\to\infty}\ U[t] - \mu[t] = 0$

The *output constraint* and the *validity* condition together imply that the iterative algorithms of interest do not maintain a "sense of time". In particular, the iterative computation by the algorithm, as captured in functions Z_i, cannot explicitly take the elapsed time (or t) into account.[2] Due to this, the validity condition for algorithms of interest here becomes:

Validity: $\forall t > 0,\ U[t] \leq U[t-1]$ and $\mu[t] \geq \mu[t-1]$ (1)

In the discussion below, when we refer to the validity condition, we mean (1).

For illustration, below we present Algorithm 1 that satisfies the *output constraint*. The algorithm has been proved to achieve *validity* and *convergence* in fully connected graphs with $n > 3f$ [5, 12]. We will later address correctness of this algorithm in arbitrary graphs. Here, we assume that each node $v \in \mathcal{V}$ has at least $2f$ incoming links. That is $|N_i^-| \geq 2f$. Later, we will show that there is no iterative Byzantine consensus if this condition does not hold.

Algorithm 1

Steps that should be performed by each node $i \in \mathcal{V}$ in the t-th iteration are as follows. Note that the faulty nodes may deviate from this specification. Output of node i at time t is $v_i[t]$.

1. Transmit current state $v_i[t-1]$ on all outgoing edges.
2. Receive values on all incoming edges (these values form vector $r_i[t]$ of size $|N_i^-|$).
3. 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 $j \in N_i^*$. Then, $|N_i^*[t]| = |N_i^-| - 2f$. By definition, $i \notin N_i^*[t]$. Note that if $j \in \{i\} \cup N_i^*[t]$ is fault-free, then $w_j = v_j[t-1]$. Define

$$v_i[t] = Z_i(r_i[t], v_i[t-1]) = \sum_{j \in \{i\} \cup N_i^*[t]} a_i\, w_j \tag{2}$$

where

$$a_i = \frac{1}{|N_i^-| + 1 - 2f}$$

The "weight" of each term on the right side of (2) is a_i, and these weights add to 1. Also, $0 < a_i \leq 1$. For future reference, let us define α as:

$$\alpha = \min_{i \in \mathcal{V}}\ a_i \tag{3}$$

---

[2] In a practical implementation, the algorithm may keep track of time, for instance, to decide to terminate after a certain number of iterations.

3 Necessary Condition

For an iterative Byzantine approximate consensus algorithm satisfying the *output constraint*, the *validity* condition, and the *convergence* condition to exist, the underlying graph $G(\mathcal{V}, \mathcal{E})$ must satisfy a necessary condition proved in this section. We now define relations $\Rightarrow$ and $\not\Rightarrow$ that are used frequently in our proofs.

Definition 1 *For non-empty disjoint sets of nodes A and B, $A \Rightarrow B$ iff there exists a node $v \in B$ that has at least $f+1$ incoming links from nodes in A, i.e., $|N_v^- \cap A| > f$.*
$A \not\Rightarrow B$ iff $A \Rightarrow B$ is not *true.*

Theorem 1 *Let sets F, L, C, R form a partition[3] of $\mathcal{V}$, such that*

- $0 \leq |F| \leq f$,
- $0 < |L|$, *and*
- $0 < |R|$

Then, at least one of the two conditions below must be true.

- $C \cup R \Rightarrow L$
- $L \cup C \Rightarrow R$

Proof: The proof is by contradiction. Let us assume that a correct iterative consensus algorithm exists, and $C \cup R \not\Rightarrow L$ and $L \cup C \not\Rightarrow R$. Thus, for any $i \in L$, $|N_i^- \cap (C \cup R)| < f+1$, and $j \in R$, $|N_j^- \cap (L \cup C)| < f+1$, Figure 1 illustrates the sets used in this proof.

Also assume that the nodes in F (if F is non-empty) are all faulty, and the remaining nodes, in sets L, R, C, are fault-free. Note that the fault-free nodes are not necessarily aware of the identity of the faulty nodes.

Consider the case when (i) each node in L has input m, (ii) each node in R has input M, such that $M > m$, and (iii) each node in C, if C is non-empty, has an input in the range $[m, M]$.

At the start of iteration 1, suppose that the faulty nodes in F (if non-empty) send $m^- < m$ to nodes in L, send $M^+ > M$ to nodes in R, and send some arbitrary value in $[m, M]$ to the nodes in C (if C is non-empty). This behavior is possible since nodes in F are faulty. Note that $m^- < m < M < M^+$. Each fault-free node $k \in \mathcal{V} - \mathcal{F}$, sends to nodes in N_k^+ value $v_k[0]$ in iteration 1.

[3]Sets $X_1, X_2, X_3, ..., X_p$ are said to form a partition of set X provided that (i) $\cup_{1 \leq i \leq p} X_i = X$ and $X_i \cap X_j = \Phi$ when $i \neq j$.

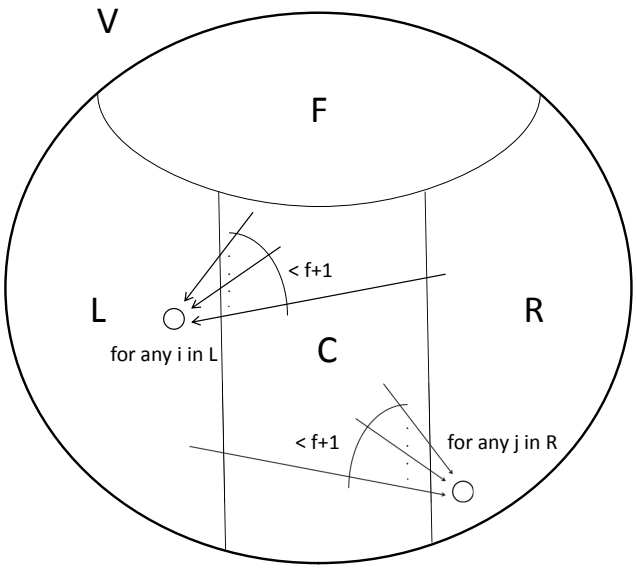

Figure 1: Illustration for the proof of Theorem 1. In this figure, $C \cup R \not\Rightarrow L$ and $L \cup C \not\Rightarrow R$.

Consider any node $i \in L$. Denote $N'(i) = N_i^- \cap (C \cup R)$. Since $C \cup R \not\Rightarrow L$, $|N'(i)| \leq f$. Node i will then receive m^- from the nodes in F, and values in $[m, M]$ from the nodes in $N'(i)$, and m from the nodes in $\{i\} \cup (L \cap N_i^-)$.

Consider four cases:

- F and $N'(i)$ are both empty: In this case, all the values that i receives are from nodes in $\{i\} \cup (L \cap N_i^-)$, and are identical to m. By validity condition (1), node i must set its new state, $v_i[1]$, to be m as well.

- F is empty and $N'(i)$ is non-empty: In this case, since $|N'(i)| \leq f$, from i's perspective, it is possible that all the nodes in $N'(i)$ are faulty, and the rest of the nodes are fault-free. In this situation, the values sent to node i by the fault-free nodes (which are all in $\{i\} \cup (L \cap N_i^-)$) are all m, and therefore, $v_i[1]$ must be set to m as per the validity condition (1).

- F is non-empty and $N'(i)$ is empty: In this case, since $|F| \leq f$, it is possible that all the nodes in F are faulty, and the rest of the nodes are fault-free. In this situation, the values sent to node i by the fault-free nodes (which are all in $\{i\} \cup (L \cap N_i^-)$) are all m, and therefore, $v_i[1]$ must be set to m as per the validity condition (1).

- Both F and $N'(i)$ are non-empty: From node i's perspective, consider two possible scenarios: (a) nodes in F are faulty, and the other nodes are fault-free, and (b) nodes in $N'(i)$ are faulty, and the other nodes are fault-free.

  In scenario (a), from node i's perspective, the non-faulty nodes have values in $[m, M]$ whereas the faulty nodes have value m^-. According to the validity condition (1),

$v_i[1] \geq m$. On the other hand, in scenario (b), the non-faulty nodes have values m^- and m, where $m^- < m$; so $v_i[1] \leq m$, according to the validity condition (1). Since node i does not know whether the correct scenario is (a) or (b), it must update its state to satisfy the validity condition in both cases. Thus, it follows that $v_i[1] = m$.

Observe that in each case above $v_i[1] = m$ for each node $i \in L$. Similarly, we can show that $v_j[1] = M$ for each node $j \in R$.

Now consider the nodes in set C, if C is non-empty. All the values received by the nodes in C are in $[m, M]$, therefore, their new state must also remain in $[m, M]$, as per the validity condition.

The above discussion implies that, at the end of the first iteration, the following conditions hold true: (i) state of each node in L is m, (ii) state of each node in R is M, and (iii) state of each node in C is in $[m, M]$. These conditions are identical to the initial conditions listed previously. Then, by induction, it follows that for any $t \geq 0$, $v_i[t] = m, \forall i \in L$, and $v_j[t] = M, \forall j \in R$. Since L and R contain fault-free nodes, the convergence requirement is not satisfied. This is a contradiction to the assumption that a correct iterative algorithm exists. □

Corollary 1 *Let $\{F, L, R\}$ be a partition of $\mathcal{V}$, such that $0 \leq |F| \leq f$, and L and R are non-empty. Then, either $L \Rightarrow R$ or $R \Rightarrow L$.*

Proof: The proof follows by setting $C = \Phi$ in Theorem 1. □

While the two corollaries below are also proved in prior literature [7], we derive it again using the necessary condition above.

Corollary 2 *The number of nodes n must exceed $3f$ for the existence of a correct iterative consensus algorithm tolerating f failures.*

Proof: The proof is by contradiction. Suppose that $2 \leq n \leq 3f$, and consider the following two cases:

- $2 \leq n \leq 2f$: Suppose that L, R, F is a partition of $\mathcal{V}$ such that $|L| = \lceil n/2 \rceil \leq f$, $|R| = \lfloor n/2 \rfloor \leq f$ and $F = \Phi$. Note that L and R are non-empty, and $|L| + |R| = n$.
- $2f < n \leq 3f$: Suppose that L, R, F is a partition of $\mathcal{V}$, such that $|L| = |R| = f$ and $|F| = n - 2f$. Note that $0 < |F| \leq f$.

In both cases above, Corollary 1 is applicable. Thus, either $L \Rightarrow R$ or $R \Rightarrow L$. For $L \Rightarrow R$ to be true, L must contain at least $f + 1$ nodes. Similarly, for $R \Rightarrow L$ to be true, R must contain at least $f + 1$ nodes. Therefore, at least one of the sets L and R must contain more than f nodes. This contradicts our choice of L and R above (in both cases, size of L and R is $\leq f$). Therefore, n must be larger than $3f$. □

Corollary 3 *When $f > 0$, for each node $i \in \mathcal{V}$, $|N_i^-| \geq 2f+1$, i.e., each node i has at least $2f+1$ incoming links.*

Proof: The proof is by contradiction. Suppose that for some node i, $|N_i^-| \leq 2f$. Define set $L = \{i\}$. Partition N_i^- into two sets F and H such that $|H| = \lfloor |N_i^-|/2 \rfloor \leq f$ and $|F| = \lceil |N_i^-|/2 \rceil \leq f$. Define $R = V - F - L = V - F - \{i\}$. Thus, $N_i^- \cap R = H$, and $|N_i^- \cap R| \leq f$. Therefore, since $L = \{i\}$, $R \not\Rightarrow L$. Also, since $|L| = 1 < f+1$, $L \not\Rightarrow R$.

This violates Corollary 1. □

4 Useful Lemmas

Definition 2 *For disjoint sets A, B, $in(A \Rightarrow B)$ denotes the set of all the nodes in B that each have at least $f+1$ incoming links from nodes in A. More formally,*

$$in(A \Rightarrow B) = \{\ v\ |v \in B \text{ and } \ f+1 \leq |N_v^- \cap A|\ \}$$

With a slight abuse of notation, when $A \not\Rightarrow B$, define $in(A \Rightarrow B) = \Phi$.

Definition 3 *For* non-empty disjoint *sets A and B, set A is said to* **propagate to** *set B in l steps, where $l > 0$, if there exist sequences of sets $A_0, A_1, A_2, \cdots, A_l$ and $B_0, B_1, B_2, \cdots, B_l$ (propagating sequences) such that*

- *$A_0 = A$, $B_0 = B$, $B_l = \Phi$, and, for $\tau < l$, $B_\tau \neq \Phi$.*
- *for $0 \leq \tau \leq l-1$,*
  - *$A_\tau \Rightarrow B_\tau$,*
  - *$A_{\tau+1} = A_\tau \cup in(A_\tau \Rightarrow B_\tau)$, and*
  - *$B_{\tau+1} = B_\tau - in(A_\tau \Rightarrow B_\tau)$*

Observe that A_τ and B_τ form a partition of $A \cup B$, and for $\tau < l$, $in(A_\tau \Rightarrow B_\tau) \neq \Phi$. Also, when set A propagates to set B, length l above is necessarily finite. In particular, l is upper bounded by $n - f - 1$, since set A must be of size at least $f+1$ for it to propagate to B.

Lemma 1 *Assume that $G(\mathcal{V}, \mathcal{E})$ satisfies Theorem 1. Consider a partition A, B, F of $\mathcal{V}$ such that A and B are non-empty, and $|F| \leq f$. If $B \not\Rightarrow A$, then set A propagates to set B.*

Proof: Since A, B are non-empty, and $B \not\Rightarrow A$, by Corollary 1, we have $A \Rightarrow B$.

The proof is by induction. Define $A_0 = A$ and $B_0 = B$. Thus $A_0 \Rightarrow B_0$ and $B_0 \not\Rightarrow A_0$. Note that A_0 and B_0 are non-empty.

Induction basis: For some $\tau \geq 0$,

- for $0 \leq k < \tau$, $A_k \Rightarrow B_k$, and $B_k \neq \Phi$,
- either $B_\tau = \Phi$ or $A_\tau \Rightarrow B_\tau$,
- for $0 \leq k < \tau$, $A_{k+1} = A_k \cup in(A_k \Rightarrow B_k)$, and $B_{k+1} = B_k - in(A_k \Rightarrow B_k)$

Since $A_0 \Rightarrow B_0$, the induction basis holds true for $\tau = 0$.

Induction: If $B_\tau = \Phi$, then the proof is complete, since all the conditions specified in Definition 3 are satisfied by the sequences of sets $A_0, A_1, \cdots, A_\tau$ and $B_0, B_1, \cdots, B_\tau$.

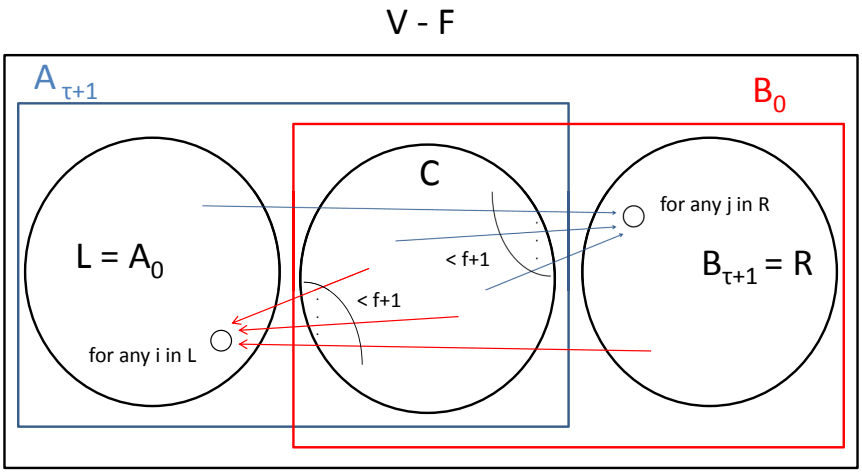

Figure 2: Illustration for the proof Lemma 1. In this figure, $B_0 \not\Rightarrow A_0$ and $A_{\tau+1} \not\Rightarrow B_{\tau+1}$

.

Now consider the case when $B_\tau \neq \Phi$. By assumption, $A_k \Rightarrow B_k$, for $0 \leq k \leq \tau$. Define $A_{\tau+1} = A_\tau \cup in(A_\tau \Rightarrow B_\tau)$ and $B_{\tau+1} = B_\tau - in(A_\tau \Rightarrow B_\tau)$. Our goal is to prove that either $B_{\tau+1} = \Phi$ or $A_{\tau+1} \Rightarrow B_{\tau+1}$. If $B_{\tau+1} = \Phi$, then the induction is complete. Therefore, now let us assume that $B_{\tau+1} \neq \Phi$ and prove that $A_{\tau+1} \Rightarrow B_{\tau+1}$. We will prove this by contradiction.

Suppose that $A_{\tau+1} \not\Rightarrow B_{\tau+1}$. Define subsets L, C, R as follows: $L = A_0$, $C = A_{\tau+1} - A_0$ and $R = B_{\tau+1}$. Figure 2 illustrates the sets used in this proof. Due to the manner in which A_k's and B_k's are defined, we also have $C = B_0 - B_{\tau+1}$. Observe that L, C, R, F form a partition of $\mathcal{V}$, where L, R are non-empty, and the following relationships hold:

- $C \cup R = B_0$, and
- $L \cup C = A_{\tau+1}$

Rewriting $B_0 \not\Rightarrow A_0$ and $A_{\tau+1} \not\Rightarrow B_{\tau+1}$, using the above relationships, we have, respectively,

$$C \cup R \not\Rightarrow L,$$

and

$$L \cup C \not\Rightarrow R$$

This violates the necessary condition in Theorem 1. This is a contradiction, completing the induction.

Thus, we have proved that, either (i) $B_{\tau+1} = \Phi$, or (ii) $A_{\tau+1} \Rightarrow B_{\tau+1}$. Eventually, for large enough t, B_t will become Φ, resulting in the propagating sequences $A_0, A_1, \cdots, A_t$ and $B_0, B_1, \cdots, B_t$, satisfying the conditions in Definition 3. Therefore, A propagates to B.

□

Lemma 2 *Assume that $G(\mathcal{V}, \mathcal{E})$ satisfies Theorem 1. For any partition A, B, F of $\mathcal{V}$, where A, B are both non-empty, and $|F| \leq f$, at least one of the following conditions must be true:*

- *A propagates to B, or*
- *B propagates to A*

Proof: Consider two cases:

- $A \not\Rightarrow B$: Then by Lemma 1, B propagates to A, completing the proof.
- $A \Rightarrow B$: In this case, consider two sub-cases:
  - A propagates to B: The proof in this case is complete.
  - A does not propagate to B: Thus, propagating sequences defined in Definition 3 do not exist in this case. More precisely, there must exist $k > 0$, and sets $A_0, A_1, \cdots, A_k$ and $B_0, B_1, \cdots, B_k$, such that:
    - $A_0 = A$ and $B_0 = B$, and
    - for $0 \leq i \leq k-1$,
      - $A_i \Rightarrow B_i$,
      - $A_{i+1} = A_i \cup in(A_i \Rightarrow B_i)$, and
      - $B_{i+1} = B_i - in(A_i \Rightarrow B_i)$.
    - $B_k \neq \Phi$ and $A_k \not\Rightarrow B_k$.

The last condition above violates the requirements for A to propagate to B.

Now $A_k \neq \Phi$, $B_k \neq \Phi$, and A_k, B_k, F form a partition of $\mathcal{V}$. Since $A_k \not\Rightarrow B_k$, by Lemma 1, B_k propagates to A_k.

Since $B_k \subseteq B_0 = B$, $A \subseteq A_k$, and B_k propagates to A_k, it should be easy to see that B propagates to A.

□

---

5 Sufficiency

We prove that the necessary condition in Theorem 1 is sufficient. In particular, we will prove that Algorithm 1 satisfies *validity* and *convergence* conditions when the necessary condition is satisfied.

In the discussion below, assume that graph $G(\mathcal{V}, \mathcal{E})$ satisfies Theorem 1, and that $\mathcal{F}$ is the set of faulty nodes in the network. Thus, the nodes in $\mathcal{V} - \mathcal{F}$ are fault-free. Since Theorem 1 holds for $G(\mathcal{V}, \mathcal{E})$, all the subsequently developed corollaries and lemmas in Sections 3 and 4 also hold for $G(\mathcal{V}, \mathcal{E})$.

Theorem 2 *Suppose that $G(\mathcal{V}, \mathcal{E})$ satisfies Theorem 1. Then Algorithm 1 satisfies the* validity *condition (1).*

Proof: Consider the t-th iteration, and any fault-free node $i \in \mathcal{V} - \mathcal{F}$. Consider two cases:

- $f = 0$: In (2), note that $v_i[t]$ is computed using states from the previous iteration at node i and other nodes. By definition of $\mu[t-1]$ and $U[t-1]$, $v_j[t-1] \in [\mu[t-1], U[t-1]]$ for all fault-free nodes $j \in \mathcal{V} - \mathcal{F}$. Thus, in this case, all the values used in computing $v_i[t]$ are in the range $[\mu[t-1], U[t-1]]$. Since $v_i[t]$ is computed as a weighted average of these values, $v_i[t]$ is also within $[\mu[t-1], U[t-1]]$.

- $f > 0$: By Corollary 3, $|N_i^-| \geq 2f+1$, and therefore, $|r_i[t]| \geq 2f+1$. When computing set $N_i^*[t]$, the largest f and smallest f values from $r_i[t]$ are eliminated. Since at most f nodes are faulty, it follows that, either (i) the values received from the faulty nodes are all eliminated, or (ii) the values from the faulty nodes that still remain are between values received from two fault-free nodes. Thus, the remaining values in $r_i[t]$ ($v_j[t-1]$, $\forall j \in N_i^*[t]$) are all in the range $[\mu[t-1], U[t-1]]$. Also, $v_i[t-1]$ is in $[\mu[t-1], U[t-1]]$, as per the definition of $\mu[t-1]$ and $U[t-1]$. Thus $v_i[t]$ is computed as a weighted average of values in $[\mu[t-1], U[t-1]]$, and, therefore, it will also be in $[\mu[t-1], U[t-1]]$.

Since $\forall i \in \mathcal{V} - \mathcal{F}$, $v_i[t] \in [\mu[t-1], U[t-1]]$, the validity condition (1) is satisfied. □

Lemma 3 *Consider node $i \in \mathcal{V} - \mathcal{F}$. Let $\psi \leq \mu[t-1]$. Then, for $j \in \{i\} \cup N_i^*[t]$,*

$$v_i[t] - \psi \geq a_i \ (w_j - \psi)$$

Specifically, for fault-free $j \in \{i\} \cup N_i^*[t]$,

$$v_i[t] - \psi \geq a_i \ (v_j[t-1] - \psi)$$

Proof: In (2), for each $j \in N_i^*[t]$, consider two cases:

- Either $j = i$ or $j \in N_i^*[t] \cap (\mathcal{V} - \mathcal{F})$: Thus, j is fault-free. In this case, $w_j = v_j[t-1]$. Therefore, $\mu[t-1] \leq w_j \leq U[t-1]$.
- j is faulty: In this case, f must be non-zero (otherwise, all nodes are fault-free). From Corollary 3, $|N_i^-| \geq 2f+1$. Then it follows that, in step 2 of Algorithm 1, the smallest f values in $r_i[t]$ contain the state of at least one fault-free node, say k. This implies that $v_k[t-1] \leq w_j$. This, in turn, implies that $\mu[t-1] \leq w_j$.

Thus, for all $j \in \{i\} \cup N_i^*[t]$, we have $\mu[t-1] \leq w_j$. Therefore,

$$w_j - \psi \geq 0 \text{ for all } j \in \{i\} \cup N_i^*[t] \tag{4}$$

Since weights in Equation 2 add to 1, we can re-write that equation as,

$$\begin{aligned} v_i[t] - \psi &= \sum_{j \in \{i\} \cup N_i^*[t]} a_i \, (w_j - \psi) \qquad (5) \\ &\geq a_i \, (w_j - \psi), \quad \forall j \in \{i\} \cup N_i^*[t] \qquad \text{from (4)} \end{aligned}$$

For non-faulty $j \in \{i\} \cup N_i^*[t]$, $w_j = v_j[t-1]$, therefore,

$$v_i[t] - \psi \geq a_i \, (v_j[t-1] - \psi) \tag{6}$$

□

Lemma 4 *Consider node $i \in \mathcal{V} - \mathcal{F}$. Let $\Psi \geq U[t-1]$. Then, for $j \in \{i\} \cup N_i^*[t]$,*

$$\Psi - v_i[t] \geq a_i \ (\Psi - w_j)$$

Specifically, for fault-free $j \in \{i\} \cup N_i^*[t]$,

$$\Psi - v_i[t] \geq a_i \ (\Psi - v_j[t-1])$$

The proof of Lemma 4 is similar to that of Lemma 3. The proof is presented in Appendix C.

The lemma below uses parameter α defined in (3).

Lemma 5 *At time s (i.e., at the end of the s-th iteration), suppose that the fault-free nodes in $\mathcal{V} - \mathcal{F}$ can be partitioned into non-empty sets R, L such that (i) R propagates to L in l steps, and (ii) the states of nodes in R are confined to an interval of length $\leq \frac{U[s]-\mu[s]}{2}$. Then,*

$$U[s+l] - \mu[s+l] \leq \left(1 - \frac{\alpha^l}{2}\right) (U[s] - \mu[s]) \tag{7}$$

Proof: Since R propagates to L, as per Definition 3, there exist sequences of sets $R_0, R_1, \cdots, R_l$ and $L_0, L_1, \cdots, L_l$, where

- $R_0 = R$, $L_0 = L$, $L_l = \Phi$, for $0 \leq \tau < l$, $L_\tau \neq \Phi$, and
- for $0 \leq \tau \leq l-1$,
  - $R_\tau \Rightarrow L_\tau$,
  - $R_{\tau+1} = R_\tau \cup in(R_\tau \Rightarrow L_\tau)$, and
  - $L_{\tau+1} = L_\tau - in(R_\tau \Rightarrow L_\tau)$

Let us define the following bounds on the states of the nodes in R at the end of the s-th iteration:

$$M = max_{j \in R}\ v_j[s] \tag{8}$$
$$m = min_{j \in R}\ v_j[s] \tag{9}$$

By the assumption in the statement of Lemma 5,

$$M - m \leq \frac{U[s] - \mu[s]}{2} \tag{10}$$

Also, $M \leq U[s]$ and $m \geq \mu[s]$. Therefore, $U[s] - M \geq 0$ and $m - \mu[s] \geq 0$.

The remaining proof of Lemma 5 relies on derivation of the three intermediate claims below.

Claim 1 *For $0 \leq \tau \leq l$, for each node $i \in R_\tau$,*

$$v_i[s+\tau] - \mu[s] \geq \alpha^\tau (m - \mu[s]) \tag{11}$$

Proof of Claim 1: The proof is by induction.

Induction basis: For some τ, $0 \leq \tau < l$, for each node $i \in R_\tau$, (11) holds. By definition of m, the induction basis holds true for $\tau = 0$.

Induction: Assume that the induction basis holds true for some τ, $0 \leq \tau < l$. Consider $R_{\tau+1}$. Observe that R_τ and $R_{\tau+1} - R_\tau$ form a partition of $R_{\tau+1}$; let us consider each of these sets separately.

- Set R_τ: By assumption, for each $i \in R_\tau$, (11) holds true. By validity of Algorithm 1, $\mu[s] \leq \mu[s+\tau]$. Therefore, setting $\psi = \mu[s]$ in Lemma 3, we get,

$$\begin{aligned} v_i[s+\tau+1] - \mu[s] &\geq a_i\ (v_i[s+\tau] - \mu[s]) \\ &\geq a_i\ \alpha^\tau (m - \mu[s]) \qquad \text{due to (11)} \\ &\geq \alpha^{\tau+1} (m - \mu[s]) \qquad \text{due to (3)} \end{aligned}$$

- Set $R_{\tau+1} - R_\tau$: Consider a node $i \in R_{\tau+1} - R_\tau$. By definition of $R_{\tau+1}$, we have that $i \in in(R_\tau \Rightarrow L_\tau)$. Thus,

$$|N_i^- \cap R_\tau| \geq f + 1$$

In Algorithm 1, $2f$ values (f smallest and f largest) received by node i are eliminated before $v_i[s + \tau + 1]$ is computed at the end of $(s + \tau + 1)$-th iteration. Consider two possibilities:

  - Value received from one of the nodes in $N_i^- \cap R_\tau$ is **not** eliminated. Suppose that this value is received from fault-free node $p \in N_i^- \cap R_\tau$. Then, by an argument similar to the previous case, we can set $\psi = \mu[s]$ in Lemma 3, to obtain,

$$\begin{aligned} v_i[s+\tau+1] - \mu[s] &\geq a_i\ (v_p[s+\tau] - \mu[s]) \\ &\geq a_i\ \alpha^\tau (m - \mu[s]) \qquad \text{due to (11)} \\ &\geq \alpha^{\tau+1}(m - \mu[s]) \qquad \text{due to (3)} \end{aligned}$$

  - Values received from **all** (there are at least $f+1$) nodes in $N_i^- \cap R_\tau$ are eliminated. Note that in this case f must be non-zero (for $f = 0$, no value is eliminated, as already considered in the previous case). By Corollary 3, we know that each node must have at least $2f + 1$ incoming edges. Since at least $f + 1$ values from nodes in $N_i^- \cap R_\tau$ are eliminated, and there are at least $2f + 1$ values to choose from, it follows that the values that are **not** eliminated[4] are within the interval to which the values from $N_i^- \cap R_\tau$ belong. Thus, there exists a node k (possibly faulty) from whom node i receives some value w_k – which is not eliminated – and a fault-free node $p \in N_i^- \cap R_\tau$ such that

$$v_p[s + \tau] \leq w_k \tag{12}$$

Then by setting $\psi = \mu[s]$ in Lemma 3 we have

$$\begin{aligned} v_i[s+\tau+1] - \mu[s] &\geq a_i\ (w_k - \mu[s]) \\ &\geq a_i\ (v_p[s+\tau] - \mu[s]) \qquad \text{due to (12)} \\ &\geq a_i\ \alpha^\tau (m - \mu[s]) \qquad \text{due to (11)} \\ &\geq \alpha^{\tau+1}(m - \mu[s]) \qquad \text{due to (3)} \end{aligned}$$

Thus, we have shown that for all nodes in $R_{\tau+1}$,

$$v_i[s + \tau + 1] - \mu[s] \geq \alpha^{\tau+1}(m - \mu[s])$$

This completes the proof of Claim 1.

[4] At least one value received from the nodes in N_i^- is not eliminated, since there are $2f + 1$ incoming edges, and only $2f$ values are eliminated.

Claim 2 *For each node $i \in \mathcal{V} - \mathcal{F}$,*

$$v_i[s+l] - \mu[s] \geq \alpha^l (m - \mu[s]) \tag{13}$$

Proof of Claim 2:

Notice that by definition, $R_l = \mathcal{V} - \mathcal{F}$. Then the proof follows by setting $\tau = l$ in the above Claim 1.

By a procedure similar to the derivation of Claim 2 above, we can also prove the claim below. The proof of Claim 3 is presented in the Appendix for completeness.

Claim 3 *For each node $i \in \mathcal{V} - \mathcal{F}$,*

$$U[s] - v_i[s+l] \geq \alpha^l (U[s] - M) \tag{14}$$

Now let us resume the proof of the Lemma 5. Note that $R_l = \mathcal{V} - \mathcal{F}$. Thus,

$$\begin{aligned} U[s+l] &= \max_{i \in \mathcal{V}-\mathcal{F}} v_i[s+l] \\ &\leq U[s] - \alpha^l (U[s] - M) \qquad \text{by (14)} \end{aligned} \tag{15}$$

and

$$\begin{aligned} \mu[s+l] &= \min_{i \in \mathcal{V}-\mathcal{F}} v_i[s+l] \\ &\geq \mu[s] + \alpha^l (m - \mu[s]) \qquad \text{by (13)} \end{aligned} \tag{16}$$

Subtracting (16) from (15),

$$\begin{aligned} U[s+l] - \mu[s+l] &\leq U[s] - \alpha^l (U[s] - M) - \mu[s] - \alpha^l (m - \mu[s]) \\ &= (1 - \alpha^l)(U[s] - \mu[s]) + \alpha^l (M - m) && (17) \\ &\leq (1 - \alpha^l)(U[s] - \mu[s]) + \alpha^l \, \frac{U[s] - \mu[s]}{2} \qquad \text{by (10)} && (18) \\ &\leq (1 - \frac{\alpha^l}{2})(U[s] - \mu[s]) && (19) \end{aligned}$$

This concludes the proof of Lemma 5. □

Theorem 3 *Suppose that $G(\mathcal{V}, \mathcal{E})$ satisfies Theorem 1. Then Algorithm 1 satisfies the* convergence *condition.*

Proof: Our goal is to prove that, given any $\epsilon > 0$, there exists τ such that

$$U[t] - \mu[t] \leq \epsilon \quad \forall t \geq \tau \tag{20}$$

Consider s-th iteration, for some $s \geq 0$. If $U[s] - \mu[s] = 0$, then the algorithm has already converged, and the proof is complete, with $\tau = s$.

Now consider the case when $U[s] - \mu[s] > 0$. Partition $\mathcal{V} - \mathcal{F}$ into two subsets, A and B, such that, for each node $i \in A$, $v_i[s] \in \left[\mu[s], \frac{U[s]+\mu[s]}{2}\right)$, and for each node $j \in B$, $v_j[s] \in \left[\frac{U[s]+\mu[s]}{2}, U[s]\right]$. By definition of $\mu[s]$ and $U[s]$, there exist fault-free nodes i and j such that $v_i[s] = \mu[s]$ and $v_j[s] = U[s]$. Thus, sets A and B are both non-empty. By Lemma 2, one of the following two conditions must be true:

- Set A propagates to set B. Then, define $L = B$ and $R = A$. The states of all the nodes in $R = A$ are confined within an interval of length $< \frac{U[s]+\mu[s]}{2} - \mu[s] \leq \frac{U[s]-\mu[s]}{2}$.
- Set B propagates to set A. Then, define $L = A$ and $R = B$. In this case, states of all the nodes in $R = B$ are confined within an interval of length $\leq U[s] - \frac{U[s]+\mu[s]}{2} \leq \frac{U[s]-\mu[s]}{2}$.

In both cases above, we have found non-empty sets L and R such that (i) L, R is a partition of $\mathcal{V} - \mathcal{F}$, (ii) R propagates to L, and (iii) the states in R are confined to an interval of length $\leq \frac{U[s]-\mu[s]}{2}$. Suppose that R propagates to L in $l(s)$ steps, where $l(s) \geq 1$. By Lemma 5,

$$U[s + l(s)] - \mu[s + l(s)] \leq \left(1 - \frac{\alpha^{l(s)}}{2}\right)(U[s] - \mu[s]) \tag{21}$$

Since $n - f - 1 \geq l(s) \geq 1$ and $0 < \alpha \leq 1$, $0 \leq \left(1 - \frac{\alpha^{l(s)}}{2}\right) < 1$.

Let us define the following sequence of iteration indices:

- $\tau_0 = 0$,
- for $i > 0$, $\tau_i = \tau_{i-1} + l(\tau_{i-1})$, where $l(s)$ for any given s was defined above.

By repeated application of the argument leading to (21), we can prove that, for $i \geq 0$,

$$U[\tau_i] - \mu[\tau_i] \leq \left(\Pi_{j=1}^{i}\left(1 - \frac{\alpha^{\tau_i - \tau_{i-1}}}{2}\right)\right)(U[0] - \mu[0]) \tag{22}$$

For a given ϵ, by choosing a large enough i, we can obtain

$$\left(\Pi_{j=1}^{i}\left(1 - \frac{\alpha^{\tau_i - \tau_{i-1}}}{2}\right)\right)(U[0] - \mu[0]) \leq \epsilon$$

and, therefore,

$$U[\tau_i] - \mu[\tau_i] \leq \epsilon \tag{23}$$

For $t \geq \tau_i$, by validity of Algorithm 1, it follows that

$$U[t] - \mu[t] \leq U[\tau_i] - \mu[\tau_i] \leq \epsilon$$

This concludes the proof. □

6 Applications

In this section, we use the results in the previous sections to examine whether iterative approximate Byzantine consensus algorithm exists in some specific networks.

6.1 Core Network

Graph $G(\mathcal{V}, \mathcal{E})$ is said to be undirected iff $(i, j) \in \mathcal{E}$ implies that $(j, i) \in \mathcal{E}$. We now define a class of undirected graphs, named *core network*.

Definition 4 Core Network: *A graph $G(\mathcal{V}, \mathcal{E})$ consisting of $n > 3f$ nodes is said to be a* core network *if the following properties are satisfied: (i) it includes a clique formed by nodes in $K \subseteq \mathcal{V}$, such that $|K| = 2f + 1$, as a subgraph and, (ii) each node $i \notin K$ has links to all the nodes in K. That is, (i) $\forall\ i, j \in K, (i, j) \in \mathcal{E}$ and $(j, i) \in \mathcal{E}$, and (ii) $\forall\ v \in \mathcal{V} - K$, and $\forall\ u \in K$, $(v, u) \in \mathcal{E}$ and $(u, v) \in \mathcal{E}$.*

It is easy to show that a core network satisfies the necessary condition in Theorem 1. Therefore, Algorithm 1 achieves approximate consensus in such network. We conjecture that a core network with $n = 3f + 1$ has the smallest number of edges possible in any undirected network of $3f + 1$ nodes for which an iterative approximate consensus algorithm exists.

6.2 Hypercube

If the conensus algorithms are **not** required to satisfy the consatraints imposed on iterative algorithms in this paper, then it is known that conensus can be achieved in undirected graphs with connectivity $> 2f$ [12]. However, connectivity of $2f + 1$ by itself is not sufficient for iterative algorithms of interest in this paper. For example, a d-dimensional binary hypercube is an undirected graph consisting of 2^d nodes and has connectivity d. However, a cut of this graph that removes edges along any one dimension fails to satisfy the necessary condition in Theorem 1, since each node has exactly one edge that belongs to the cut. Thus, each node in one part of the partition is neighbor to fewer than $f + 1$ nodes in the other part, for any $f \geq 1$. Figure 3 illustrates such a partition for a 3-dimensional binary cube. Each undirected link (i, j) in the figure represents two directed edges, namely, (i, j) and (j, i).

6.3 Chord Network

A *chord* network is a directed graph defined as follows. This network is similar but not identical to the network in [15].

Definition 5 Chord network: *A graph $G(\mathcal{V}, \mathcal{E})$ consisting of $n > 3f$ nodes is said to be a chord network if (i) $\mathcal{V} = \{0, 1, \cdots, n-1\}$, (ii) $\forall i \in \mathcal{V}$, $(i, j) \in \mathcal{E}$ iff $j = i + k \mod n$, where $1 \leq k \leq 2f+1$. That is, for each node $i \in \mathcal{V}$, $(i, (i+1) \mod n), (i, (i+2) \mod n), ..., (i, (i+2f+1) \mod n) \in E$.*

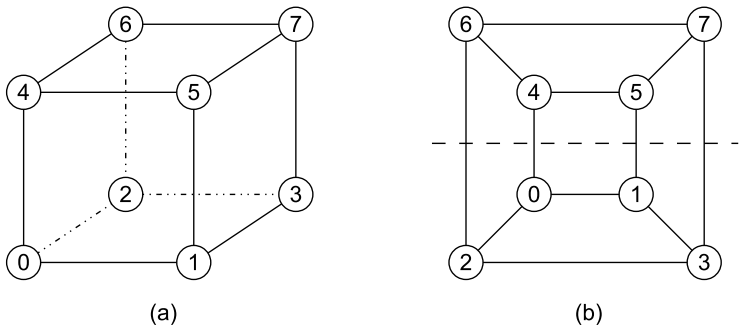

Figure 3: (a) 3-dimensional cube. (b) 3-dimensional cube redrawn to illustrate the partitions {0,1,2,3} and {4,5,6,7}.

The case when $f = 1$ and $n = 4$ results in a fully connected graph, which trivially satisfies Theorem 1. The following results can be shown for two other specific chord networks:

- When $f = 2$ and $n = 7$, the chord network does not satisfy Theorem 1.

  Let $\mathcal{V} = \{0, 1, ..., 6\}$. Then the counter example is as follows:

  Let node $5, 6$ be faulty. Then consider $L = \{0, 2\}$ and $R = \{1, 3, 4\}$. This partition fails Theorem 1. Obviously, $L \not\Rightarrow R$, since $|L| < f + 1 = 3$. However, $R \not\Rightarrow L$, since $N_0^- \cap R = \{3, 4\}$ and $N_2^- \cap R = \{1, 4\}$, which have size less than 3. Notice that this example also illustrate that connectivity of $2f + 1$ by itself is not sufficient in an directed and symmetric network.

- The Chord network with $f = 1$ and $n = 5$ satisfies Theorem 1.

7 Conclusion

This paper proves a necessary and sufficient condition for the existence of iterative approximate consensus algorithm in arbitrary directed graphs. As a special case, our results can also be applied to undirected graphs. We also use the necessary and sufficient condition to determine whether such iterative algorithms exist for certain specific graphs.

In our ongoing research, we are exploring extensions of the above results by relaxing some of the assumptions made in this work.

References

[1] A.H. 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.M. 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] Michael Ben-Or, Danny Dolev, and Ezra N. Hoch. Simple gradecast based algorithms. *CoRR*, abs/1007.1049, 2010.

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

[5] Danny Dolev, Nancy A. Lynch, Shlomit S. Pinter, Eugene W. Stark, and William 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] Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. In *Proceedings of the fourth annual ACM symposium on Principles of distributed computing*, PODC '85, pages 59–70, New York, NY, USA, 1985. ACM.

[8] 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.

[9] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Trans. on Programming Languages and Systems*, 1982.

[10] Heath LeBlanc and Xenofon Koutsoukos. "Consensus in Networked Multi-Agent Systems with Adversaries," 14th international conference on Hybrid systems: computation and control (HSCC), 2011.

[11] (via personal communication with H. LeBlanc, January 19, 2012) Heath J. LeBlanc and Xenofon Koutsoukos. "Low Complexity Resilient Consensus in Networked Multi-Agent Systems with Adversaries," to appear at 15th international conference on Hybrid systems: computation and control (HSCC), 2012.

[12] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[13] Satish M. Srinivasan and Azad H. Azadmanesh. Data aggregation in partially connected networks. *Comput. Commun.*, 32:594–601, March 2009.

[14] Satish M. Srinivasana and Azad H. Azadmanesh. Exploiting markov chains to reach approximate agreement in partially connected networks. In *Symposium on Performance Evaluation of Computer and Telecommunication Systems*, 2007.

[15] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *Networking, IEEE/ACM Transactions on*, 11(1):17 – 32, Feb. 2003.

[16] S. Sundaram and C.N. Hadjicostis. Distributed function calculation via linear iterative strategies in the presence of malicious agents. *Automatic Control, IEEE Transactions on*, 56(7):1495 –1508, July 2011.

[17] (via personal communication with S. Sundaram, January 18, 2012) Heath LeBlanc, Haotian Zhang, Shreyas Sundaram, and Xenofon Koutsoukos. "Consensus of Multi-Agent Networks in the Presence of Adversaries Using Only Local Information," submitted to HiCoNs 2012.

[18] Haotian Zhang and Shreyas Sundaram. Robustness of Information Diffusion Algorithms to Locally Bounded Adversaries. In *CoRR*, volume abs/1110.3843, 2011. http://arxiv.org/abs/1110.3843

A Proof of Claim 3

In this section, we will prove the claim 3 in Section 5:

For each node $i \in \mathcal{V} - \mathcal{F}$,

$$U[s] - v_i[s+l] \geq \alpha^l (U[s] - M)$$

Proof: Similar to the proof of claim 2, we will first prove the following claim:

Claim 4 *For $0 \leq \tau \leq l$, for each node $i \in R_\tau$,*

$$U[s] - v_i[s+\tau] \geq \alpha^\tau (U[s] - M) \tag{24}$$

Proof of Claim 4: The proof is by induction.

Induction basis: For some τ, $0 \leq \tau < l$, for each node $i \in R_\tau$, (24) holds. By definition of M, the induction basis holds true for $\tau = 0$.

Induction: Assume that the induction basis holds true for some τ, $0 \leq \tau < l$. Consider $R_{\tau+1}$. Observe that R_τ and $R_{\tau+1} - R_\tau$ form a partition of $R_{\tau+1}$; let us consider each of these sets separately.

- Set R_τ: By assumption, for each $i \in R_\tau$, (24) holds true. By validity of Algorithm 1, $U[s] \geq U[s+\tau]$. Therefore, setting $\Psi = U[s]$ in Lemma 4, we get,

$$\begin{aligned} U[s] - v_i[s+\tau+1] &\geq a_i \, (U[s] - v_i[s+\tau]) \\ &\geq a_i \, \alpha^\tau (U[s] - M) \qquad \text{due to (24)} \\ &\geq \alpha^{\tau+1} (U[s] - M) \qquad \text{due to (3)} \end{aligned}$$

- Set $R_{\tau+1} - R_\tau$: Consider a node $i \in R_{\tau+1} - R_\tau$. By definition of $R_{\tau+1}$, we have that $i \in in(R_\tau \Rightarrow L_\tau)$. Thus,

$$|N_i^- \cap R_\tau| \geq f + 1$$

In Algorithm 1, $2f$ values (f smallest and f largest) received by node i are eliminated before $v_i[s+\tau+1]$ is computed at the end of $(s+\tau+1)$-th iteration. Consider two possibilities:

  - Value received from one of the nodes in $N_i^- \cap R_\tau$ is **not** eliminated. Suppose that this value is received from fault-free node $p \in N_i^- \cap R_\tau$. Then, by an argument similar to the previous case, we can set $\Psi = U[s]$ in Lemma 4, to obtain,

$$\begin{aligned} U[s] - v_i[s+\tau+1] &\geq a_i\ (U[s] - v_p[s+\tau]) \\ &\geq a_i\ \alpha^\tau (U[s] - M) \qquad \text{due to (24)} \\ &\geq \alpha^{\tau+1}(U[s] - M) \qquad \text{due to (3)} \end{aligned}$$

  - Values received from **all** (there are at least $f+1$) nodes in $N_i^- \cap R_\tau$ are eliminated. Note that in this case f must be non-zero (for $f = 0$, no value is eliminated, as already considered in the previous case). By Corollary 3, we know that each node must have at least $2f+1$ incoming edges. Since at least $f+1$ values from nodes in $N_i^- \cap R_\tau$ are eliminated, and there are at least $2f+1$ values to choose from, it follows that the values that are **not** eliminated are within the interval to which the values from $N_i^- \cap R_\tau$ belong. Thus, there exists a node k (possibly faulty) from whom node i receives some value w_k – which is not eliminated – and a fault-free node $p \in N_i^- \cap R_\tau$ such that

$$v_p[s+\tau] \geq w_k \tag{25}$$

Then by setting $\Psi = U[s]$ in Lemma 4 we have

$$\begin{aligned} U[s] - v_i[s+\tau+1] &\geq a_i\ (U[s] - w_k) \\ &\geq a_i\ (U[s] - v_p[s+\tau]) \qquad \text{due to (25)} \\ &\geq a_i\ \alpha^\tau (U[s] - M) \qquad \text{due to (24)} \\ &\geq \alpha^{\tau+1}(U[s] - M) \qquad \text{due to (3)} \end{aligned}$$

Thus, we have shown that for all nodes in $R_{\tau+1}$,

$$U[s] - v_i[s+\tau] \geq \alpha^{\tau+1}(U[s] - M)$$

This completes the proof of Claim 4.

Now, we are able to prove Claim 3.

Proof of Claim 3:

Notice that by definition, $R_l = \mathcal{V} - \mathcal{F}$. Then the proof follows by setting $\tau = l$ in the above Claim 4.

□

B Completing the proof of Lemma 2

The last line in the proof of Lemma 2 claims that:

"Since $B_k \subseteq B_0 = B$, $A \subseteq A_k$, and B_k propagates to A_k, it should be easy to see that B propagates to A."

We now prove the correctness of this claim.

Proof: Recall that A_i and B_i form a partition of $\mathcal{V} - F$.

Let us define $P = P_0 = B_k$ and $Q = Q_0 = A_k$. Thus, P propagates to Q. Suppose that $P_0, P_1, ...P_m$ and $Q_0, Q_1, \cdots, Q_m$ are the propagating sequences in this case, with P_i and Q_i forming a partition of $P \cup Q = A_k \cup B_k = \mathcal{V} - F$.

Let us define $R = R_0 = B$ and $S = S_0 = A$. Note that R, S form a partition of $A \cup B = \mathcal{V} - F$. Now, $P_0 = B_k \subseteq B = R_0$ and $S_0 = A \subseteq A_k = Q_0$. Also, $R_0 - P_0$ and S_0 form a partition of Q_0. Figure 4 illustrates some of the sets used in this proof.

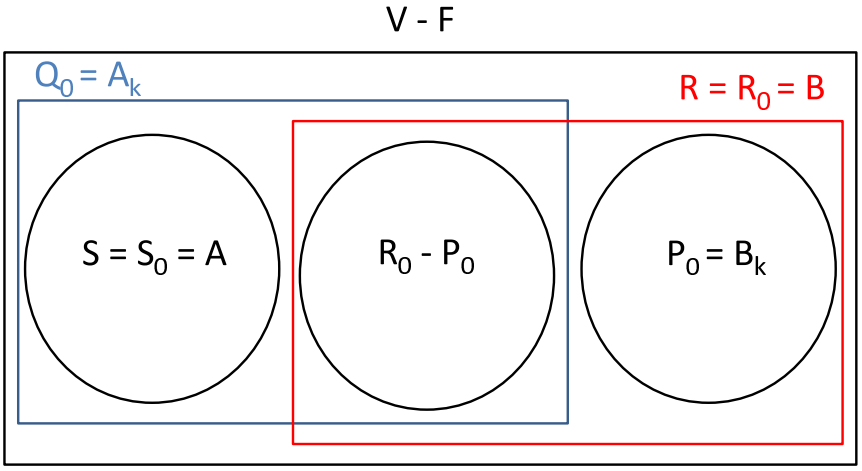

Figure 4: Illustration for the proof of the last line in Lemma 2. In this figure, $R_0 = P_0 \cup (R_0 - P_0)$ and $Q_0 = S_0 \cup (R_0 - P_0)$.

- Define $P_1 = P_0 \cup (in(P_0 \Rightarrow Q_0))$. Also, $R_1 = R_0 \cup (in(R_0 \Rightarrow S_0))$.

  Since $R_0 - P_0$ and S_0 are a partition of Q_0, the nodes in $in(P_0 \Rightarrow Q_0)$ belong to one of these two sets. Note that $R_0 - P_0 \subseteq R_0$. Also, $S_0 \cap in(P_0 \Rightarrow Q_0) \subseteq in(R_0 \Rightarrow S_0)$. Therefore, it follows that $P_1 = P_0 \cup (in(P_0 \Rightarrow Q_0)) \subseteq R_0 \cup (in(R_0 \Rightarrow S_0)) = R_1$.

  Thus, we have shown that, $P_1 \subseteq R_1$. Then it follows that $S_1 \subseteq Q_1$.

- For $0 \leq i < m$, let us define $R_{i+1} = R_i \cup in(R_i \Rightarrow S_i)$ and $S_{i+1} = S_i - in(R_i \Rightarrow S_i)$. Then following an argument similar to the above case, we can inductively show that, $P_i \subseteq R_i$ and $S_i \subseteq Q_i$. Due to the assumption on the length of the propagating sequence above, $P_m = P \cup Q = \mathcal{V} - F$. Thus, there must exist $r \leq m$, such that $R_r = \mathcal{V} - F$ and, for $i < r$, $R_i \neq \mathcal{V} - F$.

  The sequences $R_0, R_1, \cdots, R_r$ and $S_0, S_1, \cdots, S_r$ form propagating sequences, proving that $R = B$ propagates to $S = A$.

□

C Proof of Lemma 4

Proof: In (2), for each $j \in N_i^*[t]$, consider two cases:

- Either $j = i$ or $j \in N_i^*[t] \cap (\mathcal{V} - \mathcal{F})$: Thus, j is fault-free. In this case, $w_j = v_j[t-1]$. Therefore, $\mu[t-1] \leq w_j \leq U[t-1]$.

- j is faulty: In this case, f must be non-zero (otherwise, all nodes are fault-free). From Corollary 3, $|N_i^-| \geq 2f + 1$. Then it follows that, in step 2 of Algorithm 1, the largest f values in $r_i[t]$ contain the state of at least one fault-free node, say k. This implies that $v_k[t-1] \geq w_j$. This, in turn, implies that $U[t-1] \geq w_j$.

Thus, for all $j \in \{i\} \cup N_i^*[t]$, we have $U[t-1] \geq w_j$. Therefore,

$$\Psi - w_j \geq 0 \text{ for all } j \in \{i\} \cup N_i^*[t] \tag{26}$$

Since weights in Equation 2 add to 1, we can re-write that equation as,

$$\begin{aligned} \Psi - v_i[t] &= \sum_{j \in \{i\} \cup N_i^*[t]} a_i \left(\Psi - w_j\right) \\ &\geq a_i \left(\Psi - w_j\right), \quad \forall j \in \{i\} \cup N_i^*[t] \qquad \text{from (26)} \end{aligned} \tag{27}$$

For non-faulty $j \in \{i\} \cup N_i^*[t]$, $w_j = v_j[t-1]$, therefore,

$$\Psi - v_i[t] \geq a_i \left(\Psi - v_j[t-1]\right) \tag{28}$$

□