Parameter-independent Iterative Approximate Byzantine Consensus [*]

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

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

Technical Report

August 23rd, 2012

Abstract

In this work, we explore iterative approximate Byzantine consensus algorithms that do not make explicit use of the global parameter of the graph, i.e., the upper-bound on the number of faults, f.

[*] 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.

1 Introduction

We consider "iterative" algorithms for achieving approximate Byzantine consensus in synchronous point-to-point communication networks that are modeled by arbitrary *directed* graphs. The *iterative approximate Byzantine consensus* (IABC) algorithms of interest have the following properties:

- *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.[1]
- *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.

In this paper, we are interested in *parameter-independent* algorithms that do not require explicit knowledge of the upper bound on the number of faults to be tolerated. In particular, we introduce a specific parameter-independent IABC algorithm, named *Middle* Algorithm. We derive a necessary condition on the underlying communication graph under which the *Middle* algorithm can tolerate up to f Byzantine faults. For graphs that satisfy this necessary condition, we show the correctness of *Middle* Algorithm, proving that our necessary condition is tight.

For a more thorough discussion on related work, please refer to our previous work [3].

2 System Model

Communication 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 $\mathcal{V}$. With a slight abuse of terminology, we will use the terms *edge* and *link* interchangeably. We assume that $n \geq 2$, since the consensus problem for $n = 1$ is trivial. 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 $i \in \mathcal{V}$.

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, $N_i^+ = \{\, j \mid (i, j) \in \mathcal{E}\,\}$. Nodes in N_i^- and N_i^+ are, respectively, said to be incoming and outgoing neighbors of node i. 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.

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.

[1] See Section 6 for a variation on the validity condition.

3 Middle Algorithm

The *Middle* algorithm is an iterative approximate Byzantine consensus (IABC) algorithm, and its structure is similar to other algorithms studied in prior work [1, 2, 3]. 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 ($t \geq 0$). 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 $v_i[t-1]$. The *Middle* algorithm requires each node i to perform the following three steps in iteration t, where $t > 0$. Note that the faulty nodes may deviate from this specification.

Middle Algorithm

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 $|N_i^-|$. When a fault-free node expects to receive a message from a neighbor but does not receive the message, the message value is assumed to be equal to some *default value*.
3. *Update step:*

   - Sort the values in $r_i[t]$ in an increasing order with ties being broken arbitrarily, and use the sorted order of values to form a partition of nodes in N_i^- into sets B, M, T as follows: (i) set B contains nodes from whom the smallest $\lfloor |N_i^-|/3 \rfloor$ values in the sorted $r_i[t]$ are received, (ii) set T contains nodes from whom the largest $\lfloor |N_i^-|/3 \rfloor$ values in the sorted $r_i[t]$ are received, and (iii) set M contains the remaining nodes from whom the values in the "middle" of sorted $r_i[t]$ are received. That is, $M = N_i^- - B - T$. [2] Thus, $|M| = |N_i^-| - 2\lfloor |N_i^-|/3 \rfloor$.
   - Let w_j denote the value received from node $j \in M$. For convenience, define $w_i = v_i[t-1]$ to be the value node i "receives" from itself. Observe that if $j \in \{i\} \cup M$ is fault-free, then $w_j = v_j[t-1]$.
   - Define

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

     where

     $$a_i \;=\; \frac{1}{|M|+1} \;=\; \frac{1}{|N_i^-| - 2\lfloor |N_i^-|/3 \rfloor + 1}$$

     The "weight" of each term on the right-hand side of (1) is a_i, and these weights add to 1. Also, $0 < a_i \leq 1$.

     For future reference, let us define α as:

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

We now define $U[t]$ and $\mu[t]$, assuming that $\mathcal{F}$ is the set of Byzantine faulty nodes, with the nodes in $\mathcal{V} - \mathcal{F}$ being fault-free.

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

- $U[t] = \max_{i \in \mathcal{V}-\mathcal{F}} v_i[t]$. $U[t]$ 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 fault-free nodes.

The *Middle* algorithm is correct if it satisfies the following conditions in the presence of up to f Byzantine faulty nodes:

- *Validity:* $\forall t > 0, \;\; \mu[t] \geq \mu[t-1] \;\; \text{and} \quad U[t] \leq U[t-1]$
- *Convergence:* $\lim_{t \to \infty} \; U[t] - \mu[t] = 0$

The objective in this paper is to identify the necessary and sufficient conditions for Middle algorithm to satisfy the above validity and convergence conditions for a given $G(\mathcal{V}, \mathcal{E})$.

4 Necessary Condition

For the *Middle* algorithm to be correct, the network graph $G(\mathcal{V}, \mathcal{E})$ must satisfy the necessary condition proved in this section. We first define relations $\Rightarrow$ and $\not\Rightarrow$ that are used frequently in our discussion.

Definition 1 *For non-empty disjoint sets of nodes A and B,*

- *$A \Rightarrow B$ iff there exists a node $v \in B$ such that*

$$\frac{|N_v^- \cap A|}{|N_v^-|} > \frac{1}{3} \tag{3}$$

- *$A \not\Rightarrow B$ iff $A \Rightarrow B$ is* not *true.*

Theorem 1 *Suppose that Middle Algorithm is correct in graph $G(\mathcal{V}, \mathcal{E})$ in the presence of up to f Byzantine faults. Then, both the following conditions must be true:*

- *For every node $v \in \mathcal{V}$, $|N_v^-| \geq 3f$.*
- *Let sets F, L, C, R form a partition[3] of $\mathcal{V}$, such that L and R are both non-empty, and $|F| \leq f$. Then, either $C \cup R \Rightarrow L$, or $L \cup C \Rightarrow R$.*

Proof:

[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 (ii) $X_i \cap X_j = \Phi$ if $i \neq j$.

Proof of first condition: The first condition is trivially true when $f = 0$. Thus, let us now assume that $f \geq 1$. Suppose by way of contradiction that there exists a node i such that $|N_i^-| < 3f$. Consider two cases in iteration 1:

- $|N_i^-| = 0$: Suppose that node i has initial input of X, and all the remaining nodes have input x, where $x < X$. Since node i has no incoming edges, clearly, $v_i[1] = X$.

  Consider two cases:

  - There exists a node $j \neq i$ such that $(i, j) \in \mathcal{E}$, and the in-degree of node j is such that the value X is *not* eliminated in the *Update* step, i.e., $|N_j^-| \leq 2$: In this case, $v_j[1] > x$ since $X > x$. However, in the event that node i is actually faulty, $v_j[1]$ will not satisfy the validity condition, since the initial inputs at all the fault-free nodes are all x (if node i were to be faulty).
  - For each node $j \neq i$, either $(i, j) \notin \mathcal{E}$, or $(i, j) \in \mathcal{E}$ but the value received from node i is dropped at node j during the *Update* step: In this case, all the values that affect the new state of node j are x, and $v_j[1] = x$. It is easy to see that the same scenario will repeat in each iteration, violating convergence condition when all the nodes (including i) are fault-free (v_i remains at X, and for each node $j \neq i, v_j$ remains at x).

- $|N_i^-| \geq 1$: Assume that $min(f, |N_i^-|)$ incoming neighbors of node i are faulty, and that all the remaining nodes are fault-free. Let F denote the set of faulty nodes. Note that $|F| \geq 1$.

  Let $R = \mathcal{V} - \{i\} - F$. Consider the case when (i) each node in R has input x, and (ii) node i has input $X > x$. In the *Transmit* step of iteration 1, suppose that the faulty nodes in F send a sufficiently large value Y (elaborated below) on outgoing links to node i, and send value x on outgoing links to nodes in R. This behavior is possible since nodes in F are faulty. Each fault-free node $k \in \mathcal{V} - F$ sends $v_k[0]$ (its input) on all its outgoing links.

  Since $|N_i^-| < 3f$, set M at node i in iteration 1 contains at least one value received from a faulty incoming neighbor. Then it is easy to see that the faulty nodes can choose Y such that $v_i[1] > X$. Since i is fault-free, and $v_i[1]$ exceeds the initial input at all the fault-free nodes, the validity condition is violated.

In all cases above, either validity or convergence is violated, contradicting the assumption that the *Middle* algorithm is correct in the given graph.

Proof of second condition: Since the first condition is already proved to be necessary, we assume that the graph satisfies that condition. The proof for the second condition is also by contradiction. Suppose that the second condition is violated, i.e., in G, there exists some partition F, L, C, R such that $|C \cup R| \not\Rightarrow L$ and $|L \cup C| \not\Rightarrow R$. Thus, for any $i \in L$, $\frac{|N_i^- \cap (C \cup R)|}{|N_i^-|} \leq \frac{1}{3}$, and for any $j \in R$, $\frac{|N_j^- \cap (L \cup C)|}{|N_j^-|} \leq \frac{1}{3}$.

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

Consider the case when (i) each node in L has initial input x, (ii) each node in R has initial input X, such that $X > x$, and (iii) each node in C (if non-empty) has an input in the interval (x, X).

In the *Transmit* step of iteration 1, suppose that each faulty node in F (if non-empty) sends $x^- < x$ on outgoing links to nodes in L, sends $X^+ > X$ on outgoing links to nodes in R, and sends some arbitrary value in interval $[x, X]$ on outgoing links to nodes in C (if non-empty). This behavior is possible since nodes in F are faulty. Note that $x^- < x < X < X^+$. Each fault-free node $k \in \mathcal{V} - F$ sends $v_k[0]$ to nodes in N_k^+ in iteration 1.

Consider a node $i \in L$. In iteration 1, node i receives x^- from the nodes in $N_i^- \cap F$, x from the nodes in $\{i\} \cup (N_i^- \cap L)$, and values in $(x, X]$ from the nodes in $N_i^- \cap (C \cup R)$. Then in the *Update* step, $|B| \geq f \geq |F|$ due to the first condition, i.e., $|N_i^-| \geq 3f$. Furthermore, set T (calculated in the *Update* step at node i) contains all the values from $N_i^- \cap (C \cup R)$, since $|C \cup R| \not\Rightarrow L$, i.e., $\frac{|N_i^- \cap (C \cup R)|}{|N_i^-|} \leq \frac{1}{3}$, and the values received from the nodes in $C \cup R$ are the largest values in vector $r_i[1]$. Recall that in the *Update* step, node i would eliminate sets B and T, and the remaining values, i.e., values in $\{i\} \cup M$, are all x, and therefore, $v_i[1]$ will be set to x as per (1).

Thus, $v_i[1] = x$ for each node $i \in L$. Similarly, we can show that $v_j[1] = X$ for each node $j \in R$. Now consider the nodes in set C (if non-empty). The initial state of nodes in C is in (x, X), and all the values received from the neighbors are in $[x, X]$, therefore, their new state of the nodes in C will remain in (x, X) when using the *Middle* algorithm (since the node's own state is assigned a non-zero weight in (1)).

The above discussion implies that, at the end of iteration 1, the following conditions hold true: (i) state of each node in L is x, (ii) state of each node in R is X, and (iii) state of each node in C is in the interval (x, X). These conditions are identical to the initial conditions listed previously. Then, by a repeated application of the above argument (proof by induction), it follows that for any $t \geq 0$, $v_i[t] = x$ for all $i \in L$, $v_j[t] = X$ for all $j \in R$ and $v_k[t] \in (x, X)$ for all $k \in C$.

Since L and R both contain fault-free nodes, the convergence requirement is not satisfied. This is a contradiction to the assumption that a correct iterative algorithm exists.

□

5 Sufficient Condition

In Theorems 2 and 3 in this section, we prove that Middle Algorithm satisfies *validity* and *convergence* conditions, respectively, provided that $G(\mathcal{V}, \mathcal{E})$ satisfies the condition below, which matches the necessary condition stated in Theorem 1.

Sufficient condition:

- For every node $v \in \mathcal{V}$, $|N_v^-| \geq 3f$, and
- Let sets F, L, C, R form a partition of $\mathcal{V}$, such that L and R are both non-empty, and $|F| \leq f$. Then, either $C \cup R \Rightarrow L$, or $L \cup C \Rightarrow R$.

The claim below follows immediately from the second condition above by setting $C = \Phi$.

Claim 1 *Suppose that $G(\mathcal{V}, \mathcal{E})$ satisfies the* Sufficient *condition stated above. Let $\{F, L, R\}$ be a partition of $\mathcal{V}$, such that L and R are both non-empty and $|F| \leq f$. Then, either $L \Rightarrow R$ or $R \Rightarrow L$.*

Theorem 2 *Suppose that $\mathcal{F}$ is the set of Byzantine faulty nodes, and that $G(\mathcal{V}, \mathcal{E})$ satisfies the* sufficient *condition stated above. Then Middle Algorithm satisfies the* validity *condition.*

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

- $f = 0$: In this case, all nodes must be fault-free, and $\mathcal{F} = \Phi$. In (1) in Middle Algorithm, 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 $j \in \mathcal{V} - \mathcal{F} = \mathcal{V}$. Thus, in this case, all the values used in computing $v_i[t]$ are in the interval $[\mu[t-1], U[t-1]]$. Since $v_i[t]$ is computed as a weighted average of these values, $v_i[t]$ is also within $[\mu[t-1], U[t-1]]$.

- $f > 0$: Since $|N_i^-| \geq 3f$, $|r_i[t]| \geq 3f$. Thus set T in the *Update* step contains at least the largest f values from $r_i[t]$, and set B contains at least the smallest f values from $r_i[t]$. 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 $r_i[t]$ – that is, values received from nodes in set M – are all in the interval $[\mu[t-1], U[t-1]]$. 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 is satisfied. □

Definition 2 *For disjoint sets A, B, $in(A \Rightarrow B)$ denotes the set of all the nodes in B that have at least $1/3$ of the incoming edges from nodes in A. More formally,*

$$in(A \Rightarrow B) = \left\{ v \mid v \in B \text{ and } \frac{|N_v^- \cap A|}{|N_v^-|} > \frac{1}{3} \right\}$$

With an 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*

- $A_0 = A, \quad B_0 = B, \quad A_l = A \cup B, \quad B_l = \Phi, \quad B_\tau \neq \Phi$ *for* $\tau < l$, *and*
- *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, the number of steps l in the above definition is upper bounded by $n-1$.

Lemma 1 *Assume that $G(\mathcal{V}, \mathcal{E})$ satisfies the* sufficient *condition stated above. For any partition A, B, F of $\mathcal{V}$, where A, B are both non-empty, and $|F| \leq f$, either A propagates to B, or B propagates to A.*

The proof of Lemma 1 is similar to the proof in our prior work [3] – the proof is included in Appendix A.

The lemma below states that the interval to which the states at all the fault-free nodes are confined shrinks after a finite number of iterations of Middle Algorithm. Recall that $U[t]$ and $\mu[t]$ (defined in Section 3) are the maximum and minimum over the states at the fault-free nodes at the end of the t-th iteration.

Lemma 2 *Suppose that $G(\mathcal{V},\mathcal{E})$ satisfies the* sufficient *condition stated above, and $\mathcal{F}$ is the set of Byzantine faulty nodes. Moreover, at the end of the s-th iteration of Middle Algorithm, suppose that the fault-free nodes in $\mathcal{V}-\mathcal{F}$ can be partitioned into non-empty sets R and 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, with the* Middle *algorithm,*

$$U[s+l]-\mu[s+l] \;\leq\; \left(1-\frac{\alpha^l}{2}\right)(U[s]-\mu[s]) \qquad (4)$$

where α is as defined in (2).

The proof of the above lemma is presented in Appendix B.

Theorem 3 *Suppose that $\mathcal{F}$ is the set of Byzantine faulty nodes, and that $G(\mathcal{V},\mathcal{E})$ satisfies the* sufficient *condition stated above. Then the* Middle *algorithm satisfies the* convergence *condition.*

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

$$U[t]-\mu[t] \leq \epsilon \quad \forall t \geq \tau \qquad (5)$$

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 $\tau = s$ (recall that we have already proved that the algorithm satisfies the validity condition).

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 1, 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 strictly less than $\frac{U[s]+\mu[s]}{2} - \mu[s] \leq \frac{U[s]-\mu[s]}{2}$.
- 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 less than or equal to $U[s] - \frac{U[s]+\mu[s]}{2} \leq \frac{U[s]-\mu[s]}{2}$.

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 less than or equal to $\frac{U[s]-\mu[s]}{2}$. Suppose that R propagates to L in $l(s)$ steps, where $l(s) \geq 1$. Then by Lemma 2,

$$U[s+l(s)]-\mu[s+l(s)] \leq \left(1-\frac{\alpha^{l(s)}}{2}\right)(U[s]-\mu[s]) \qquad (6)$$

In the *Middle* algorithm, observe that $a_i > 0$ for all i. Therefore, α defined in (2) is > 0. Then, $n-1 \geq l(s) \geq 1$ and $0 < \alpha \leq 1$; hence, $0 \leq \left(1 - \frac{\alpha^{l(s)}}{2}\right) < 1$.

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.

If for some i, $U[\tau_i] - \mu[\tau_i] = 0$, then since the algorithm is already proved to satisfy the validity condition, we will have $U[t] - \mu[t] = 0$ for all $t \geq \tau_i$, and the proof of convergence is complete.

Now, suppose that $U[\tau_i] - \mu[\tau_i] \neq 0$ for the values of i in the analysis below. By repeated application of the argument leading to (6), we can prove that, for $i \geq 0$,

$$U[\tau_i] - \mu[\tau_i] \leq \left(\Pi_{j=1}^{i}\left(1 - \frac{\alpha^{\tau_j - \tau_{j-1}}}{2}\right)\right)(U[0] - \mu[0]) \tag{7}$$

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

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

and, therefore,

$$U[\tau_i] - \mu[\tau_i] \leq \epsilon \tag{8}$$

For $t \geq \tau_i$, by validity of the *Middle* algorithm, it follows that

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

This concludes the proof. □

6 Discussion

The results in this report can be easily extended to the following version of the validity condition:

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

This validity condition is weaker than the condition satisfied by the *Middle* algorithm, therefore, the algorithm satisfies this validity condition as well. Also, it should be easy to see that our necessary condition also holds under the above validity condition (the proof remains essentially unchanged).

In our analysis here, we assumed that the system is synchronous, and messages sent in each iteration are delivered in the same iteration. That is, the state update in the t-th iteration uses neighbors' states at the end of the $(t-1)$-th iteration. The results in this paper can be extended to the case when messages may be delayed such that the latest state available from a neighbor may be from iteration $(t-B)$, for some finite $B > 0$. In this case, our original validity condition will need to be modified to require that the state of the fault-free nodes at the end of any iteration remains in the convex hull of the fault-free nodes B iterations ago.

We now state a result without proof. Further details will be presented elsewhere. Consider an Erdös-Rényi random graphs $G_{n,p}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ contains n vertices, and edge $(i, j) \in \mathcal{E}$ with probability p independently for each (i, j). For large n, this random graph satisfies the condition in Theorem 1 with high probability if and only if $p = \Omega(t)$ where t is a threshold dependent on n and f.

7 Summary

This paper introduces a parameter-independent iterative algorithm, the *Middle* algorithm, that solves the approximate Byzantine consensus problem. The *Middle* algorithm does not explicitly use the global parameter of the graph, i.e., the upper-bound on the number of faults, f. We prove *tight* necessary and sufficient conditions for the correctness of the *Middle* algorithm that tolerates up to f Byzantine faults in directed graphs.

References

[1] 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.

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

[3] N. H. Vaidya, L. Tseng, and G. Liang. Iterative approximate byzantine consensus in arbitrary directed graphs. In *Proceedings of the thirty-first annual ACM symposium on Principles of distributed computing*, PODC '12. ACM, 2012.

A Proof of Lemma 1

To prove Lemma 1, we first prove the following Lemma.

Lemma 3 *Assume that $G(\mathcal{V}, \mathcal{E})$ satisfies the* Sufficient condition*. 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 $B \not\Rightarrow A$, by Claim 1, $A \Rightarrow B$.

Define $A_0 = A$ and $B_0 = B$. Now, for a suitable $l > 0$, we will build propagating sequences $A_0, A_1, \cdots A_l$ and $B_0, B_1, \cdots B_l$ inductively.

- Recall that $A = A_0$ and $B = B_0 \neq \Phi$. Since $A \Rightarrow B$, $in(A_0 \Rightarrow B_0) \neq \Phi$. Define $A_1 = A_0 \cup in(A_0 \Rightarrow B_0)$ and $B_1 = B_0 - in(A_0 \Rightarrow B_0)$.

  If $B_1 = \Phi$, then $l = 1$, and we have found the propagating sequence already.

  If $B_1 \neq \Phi$, then define $L = A = A_0$, $R = B_1$ and $C = A_1 - A = B - B_1$. Note that $B = R \cup C$, $A_1 = L \cup C$, and L, C, R, F form a partition of the set of nodes. Since $B \not\Rightarrow A$, $R \cup C \not\Rightarrow L$. Therefore, by the *Sufficient condition*, $L \cup C \Rightarrow R$. That is, $A_1 \Rightarrow B_1$.

- For increasing values of $i \geq 0$, given A_i and B_i, where $B_i \neq \Phi$, by following steps similar to the previous item, we can obtain $A_{i+1} = A_0 \cup in(A_i \Rightarrow B_i)$ and $B_{i+1} = B_i - in(A_i \Rightarrow B_i)$, such that either $B_{i+1} = \Phi$ or $A_{i+1} \Rightarrow B_{i+1}$.

In the above construction, l is the smallest index such that $B_l = \Phi$. □

Proof of Lemma 1

Proof: Consider two cases:

- $A \not\Rightarrow B$: Then by Lemma 3 above, 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*: Recall that $A \Rightarrow B$. Since A does not propagate to B, 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 $0 \leq i \leq k-1$,
      - o $A_i \Rightarrow B_i$,
      - o $A_{i+1} = A_i \cup in(A_i \Rightarrow B_i)$, and
      - o $B_{i+1} = B_i - in(A_i \Rightarrow B_i)$.
    - $B_k \neq \Phi$ $\underline{\text{and}}$ $A_k \not\Rightarrow B_k$.

    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 3 above, B_k propagates to A_k.

    Given that $B_k \subseteq B_0 = B$, $A = A_0 \subseteq A_k$, and B_k propagates to A_k, now we prove that B propagates to A.

    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.

    - Define $P_1 = P_0 \cup (in(P_0 \Rightarrow Q_0))$, and $Q_1 = \mathcal{V} - F - P_1 = Q_0 - (in(P_0 \Rightarrow Q_0))$. Also, $R_1 = R_0 \cup (in(R_0 \Rightarrow S_0))$, and $S_1 = \mathcal{V} - F - R_1 = S_0 - (in(R_0 \Rightarrow S_0))$.

      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 $0 \le i < m$, 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, $P_m = P \cup Q = \mathcal{V} - F$ and $Q_m = \Phi$. Thus, there must exist $r \le m$, such that for $i < r$, $R_i \neq \mathcal{V} - F$, and $R_r = \mathcal{V} - F$ and $S_r = \Phi$.
  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 $S = A$.

□

B Proof of Lemma 2

We first present two additional lemmas (using the notation in Middle Algorithm).

Lemma 4 *Suppose that $\mathcal{F}$ is the set of faulty nodes, and that $G(\mathcal{V}, \mathcal{E})$ satisfies the "sufficient condition" stated in Section 5. Consider node $i \in \mathcal{V} - \mathcal{F}$. Let $\psi \le \mu[t-1]$. Then, for $j \in \{i\} \cup M$,*

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

where w_j is the value received by node i from node j in the t-th iteration. Specifically, for fault-free $j \in \{i\} \cup M$,

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

Proof: In (1) in Middle Algorithm, for each $j \in \{i\} \cup M$, consider two cases:

- j is faulty-free: Then, either $j = i$ or $j \in M \cap (\mathcal{V} - \mathcal{F})$. In this case, $w_j = v_j[t-1]$. Therefore, $\mu[t-1] \le w_j \le U[t-1]$.
- j is faulty: In this case, f must be non-zero (otherwise, all nodes are fault-free). By Theorem 1, $|N_i^-| \ge 3f$. Then it follows that, in step 2 of the *Middle* algorithm, $|B| \ge f$, and set B contains the state of at least one fault-free node, say k. This implies that $v_k[t-1] \le w_j$. This, in turn, implies that $\mu[t-1] \le w_j$.

Thus, for all $j \in \{i\} \cup M$, we have $\mu[t-1] \le w_j$. Therefore,

$$w_j - \psi \ge 0 \text{ for all } j \in \{i\} \cup M \tag{9}$$

Since weights in (1) in Middle Algorithm add to 1, we can re-write that equation as,

$$\begin{aligned} v_i[t] - \psi &= \sum_{j \in \{i\} \cup M} a_i\,(w_j - \psi) \qquad (10) \\ &\ge a_i\,(w_j - \psi), \ \ \forall j \in \{i\} \cup M \quad \text{from (9)} \end{aligned}$$

For fault-free $j \in \{i\} \cup M$, $w_j = v_j[t-1]$, therefore,

$$v_i[t] - \psi \ \ \ge \ \ a_i\,(v_j[t-1] - \psi) \tag{11}$$

□

Lemma 5 *Suppose that $\mathcal{F}$ is the set of faulty nodes, and that $G(\mathcal{V}, \mathcal{E})$ satisfies the "sufficient condition" stated in Section 5. Consider fault-free node $i \in \mathcal{V} - \mathcal{F}$. Let $\Psi \geq U[t-1]$. Then, for $j \in \{i\} \cup M$,*

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

where w_j is the value received by node i from node j in the t-th iteration. Specifically, for fault-free $j \in \{i\} \cup M$,

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

Proof: The proof is similar to Lemma 4 proof. □

Proof of Lemma 2

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

- $R_0 = R, \quad L_0 = L, \quad R_l = R \cup L, \quad L_l = \Phi$, 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:

$$X = max_{j \in R}\, v_j[s] \quad (12)$$

$$x = min_{j \in R}\, v_j[s] \quad (13)$$

By the assumption in the statement of Lemma 2,

$$X - x \leq \frac{U[s] - \mu[s]}{2} \quad (14)$$

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

The remaining proof of Lemma 2 relies on derivation of the three intermediate claims below.

Claim 2 *For $0 \leq \tau \leq l$, for each node $i \in R_\tau$,*

$$v_i[s+\tau] - \mu[s] \;\geq\; \alpha^\tau (x - \mu[s]) \quad (15)$$

Proof of Claim 2: The proof is by induction.

Induction basis: By definition of x, (15) holds true for $\tau = 0$.

Induction: Assume that (15) 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 $i \in R_\tau$, (15) holds true. By validity of Middle Algorithm (proved in Theorem 2), $\mu[s] \leq \mu[s+\tau]$. Therefore, setting $\psi = \mu[s]$ and $t = s+\tau+1$ in Lemma 4, we get,

$$\begin{aligned} v_i[s+\tau+1] - \mu[s] &\geq a_i\ (v_i[s+\tau] - \mu[s]) \\ &\geq a_i\ \alpha^\tau(x - \mu[s]) \quad \text{due to (15)} \\ &\geq \alpha^{\tau+1}(x - \mu[s]) \quad \text{due to (2)} \\ &\quad \text{and because} \quad x - \mu[s] \geq 0 \end{aligned}$$

- 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,

$$\frac{|N_i^- \cap R_\tau|}{|N_i^-|} > \frac{1}{3}$$

  In Middle Algorithm, values in sets B and T 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, $p \in M$, and by an argument similar to the previous case, we can set $\psi = \mu[s]$ in Lemma 4, to obtain,

$$\begin{aligned} v_i[s+\tau+1] - \mu[s] &\geq a_i\ (v_p[s+\tau] - \mu[s]) \\ &\geq a_i\ \alpha^\tau(x - \mu[s]) \quad \text{due to (15)} \\ &\geq \alpha^{\tau+1}(x - \mu[s]) \quad \text{due to (2)} \\ &\quad \text{and because} \quad x - \mu[s] \geq 0 \end{aligned}$$

  - Values received from *all* nodes in $N_i^- \cap R_\tau$ are eliminated. Thus, $(N_i^- \cap R_\tau) \subseteq T \cup B$. Recall that $|N_i^- \cap R_\tau| > |N_i^-|/3 \geq |B| = |T|$. Thus, T and B both must contain at least one node from $N_i^- \cap R_\tau$. Therefore, the values that are *not* eliminated – that is, values received from nodes in M – are within the interval to which the values received from the nodes in $N_i^- \cap R_\tau$ belong. Thus, there exists a node k (possibly faulty) in M 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

$$v_p[s+\tau] \quad \leq \quad w_k \tag{16}$$

    Then by setting $\psi = \mu[s]$ and $t = s+\tau+1$ in Lemma 4, we have

$$\begin{aligned} v_i[s+\tau+1] - \mu[s] &\geq a_i\ (w_k - \mu[s]) \\ &\geq a_i\ (v_p[s+\tau] - \mu[s]) \quad \text{by (16)} \\ &\geq a_i\ \alpha^\tau(x - \mu[s]) \quad \text{due to (15)} \\ &\geq \alpha^{\tau+1}(x - \mu[s]) \quad \text{due to (2)} \\ &\quad \text{and because} \quad x - \mu[s] \geq 0 \end{aligned}$$

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

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

This completes the proof of Claim 2.

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

$$v_i[s+l] - \mu[s] \;\geq\; \alpha^l(x - \mu[s]) \tag{17}$$

Proof of Claim 3: Note that by definition, $R_l = \mathcal{V} - \mathcal{F}$. Then the proof follows by setting $\tau = l$ in the above Claim 2.

Claim 4 *For each node $i \in \mathcal{V} - \mathcal{F}$,*

$$U[s] - v_i[s+l] \geq \alpha^l(U[s] - X) \tag{18}$$

The proof of Claim 4 is similar to the proof of Claim 3.

Now let us resume the proof of the Lemma 2. Thus,

$$\begin{aligned} U[s+l] &= \max_{i \in \mathcal{V}-\mathcal{F}} v_i[s+l] \\ &\leq U[s] - \alpha^l(U[s] - X) \qquad \text{by (18)} \end{aligned} \tag{19}$$

and

$$\begin{aligned} \mu[s+l] &= \min_{i \in \mathcal{V}-\mathcal{F}} v_i[s+l] \\ &\geq \mu[s] + \alpha^l(x - \mu[s]) \qquad \text{by (17)} \end{aligned} \tag{20}$$

Subtracting (20) from (19),

$$\begin{aligned} & U[s+l] - \mu[s+l] \\ \leq\; & U[s] - \alpha^l(U[s] - X) - \mu[s] - \alpha^l(x - \mu[s]) \\ =\; & (1-\alpha^l)(U[s] - \mu[s]) + \alpha^l(X - x) \\ \leq\; & (1-\alpha^l)(U[s] - \mu[s]) + \alpha^l\,\frac{U[s] - \mu[s]}{2} \qquad \text{by (14)} \\ \leq\; & (1 - \frac{\alpha^l}{2})(U[s] - \mu[s]) \end{aligned}$$

This concludes the proof of Lemma 2. □