Iterative Approximate Consensus in the presence of Byzantine Link Failures *

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

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

Technical Report

Abstract. This paper explores the problem of reaching approximate consensus in synchronous point-to-point networks, where each directed link of the underlying communication graph represents a communication channel between a pair of nodes. We adopt the *transient Byzantine link* failure model [15, 16], where an omniscient adversary controls a subset of the *directed* communication links, but the nodes are assumed to be *fault-free*.

Recent work has addressed the problem of reaching approximate consensus in incomplete graphs with Byzantine *nodes* using a *restricted class* of iterative algorithms that maintain only a small amount of memory across iterations [22, 21, 23, 12]. However, to the best of our knowledge, we are the first to consider approximate consensus in the presence of Byzantine *links*. We extend our past work that provided exact characterization of graphs in which the iterative approximate consensus problem in the presence of Byzantine *node* failures is solvable [22, 21]. In particular, we prove a *tight* necessary and sufficient condition on the underlying communication graph for the existence of iterative approximate consensus algorithms under *transient Byzantine link* model. The condition answers (part of) the open problem stated in [16].

1 Introduction

Approximate consensus can be related to many distributed computations in networked systems, such as data aggregation [10], decentralized estimation [17], and flocking [9]. Extensive work has addressed the problem in the presence of *Byzantine nodes* [11] in either complete networks [6, 1] or arbitrary directed networks

* This research is supported in part by National Science Foundation award CNS 1329681. 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.

[22, 12, 21]. As observed in [2, 18], link failures become more and more prevalent. Thus, it is of interest to consider the problem of approximate consensus in the presence of Byzantine *link* failures.

This paper explores such problem in synchronous point-to-point networks, where each directed link of the underlying communication graph represents a communication channel between a pair of nodes. The link failures are modeled using a *transient Byzantine link* failure model (formal definition in Section 2) [15, 16], in which different sets of link failures may occur at different time. We consider the problem in arbitrary directed graphs using a *restricted class* of iterative algorithms that maintain only a small amount of memory across iterations, e.g., the algorithms do not require the knowledge of the network topology. Such iterative algorithms are of interest in networked systems, since they have low complexity and do not rely on global knowledge [12]. In particular, the iterative algorithms have the following properties, which we will state more formally later:

- **Initial state** of each node is equal to a real-valued *input* provided to that node.
- **Termination**: The algorithm terminates in finite number of iterations.
- **Validity**: After each iteration of the algorithm, the state of each node must stay in the *convex hull* of the states of all the nodes at the end of the *previous* iteration.
- **ϵ-agreement**: For any $\epsilon > 0$, when the algorithm terminates, the difference between any pair of nodes is guaranteed to be within ϵ.

Main Contribution This paper extends our recent work on approximate consensus under node failures [22, 21]. The main contribution is identifying a *tight* necessary and sufficient condition for the graphs to be able to reach approximate consensus under *transient Byzantine link* failure models [15, 16] using restricted iterative algorithms; our proof of correctness follows a structure previously used in our work to prove correctness of other consensus algorithms in incomplete networks [21, 23]. The use of matrix analysis is inspired by the prior work on non-fault-tolerant consensus (e.g., [9, 3]).

Related Work Approximate consensus has been studied extensively in synchronous as well as asynchronous systems. Bertsekas and Tsitsiklis explored reaching approximate consensus without failures in synchronous dynamic network, where the underlying communication graph is time-varying [3]. Dolev et al. considered approximate consensus in the presence of *Byzantine nodes* in both synchronous and asynchronous systems [6], where the network is assumed to be a clique, i.e., a complete network. Subsequently, for complete graphs, Abraham et al. proposed an algorithm to achieve approximate consensus with *Byzantine nodes* in asynchronous systems using optimal number of nodes [1].

Recent work has addressed approximate consensus in incomplete graphs with faulty *nodes* [22, 12, 21]. [22, 21] and [12] showed exact characterizations of graphs in which the approximate consensus problem is solvable in the presence of Byzantine nodes and malicious nodes, respectively. Malicious node is a restricted type

of Byzantine node in which every node is forced to send the identical message to all of its neighbors.

Much effort has also been devoted to the problem of achieving consensus in the presence of link failures [4, 2, 18, 15, 16]. Charron-Bost and Schiper proposed a HO (Heard-Of) model that captures both the link and node failures at the same time [4]. However, the failures are assumed to be benign in the sense that no corrupted message will ever be received in the network. Santoro and Widmayer proposed the *transient* Byzantine link failure model: a different set of links can be faulty at different time [15, 16]. They characterized a necessary condition and a sufficient condition for undirected networks to achieve consensus in the transient link failure model; however, the conditions are *not* tight (i.e., do not match): necessary and sufficient conditions are specified in terms of node degree and edge-connectivity,[1] respectively. Subsequently, Biely et al. proposed another link failure model that imposes an upper bound on the number of faulty links incident to each node [2]. As a result, it is possible to tolerate $O(n^2)$ link failures with n nodes in the new model. Under this model, Schmid et al. proved lower bounds on number of nodes, and number of rounds for achieving consensus [18]. However, incomplete graphs were not considered in [2, 18].

For consensus problem, it has been shown in [7] and [16], respectively, that an undirected graph of $2f + 1$ node-connectivity[2] and edge-connectivity is able to tolerate f Byzantine nodes and f Byzantine links. Independently, researchers showed that $2f + 1$ node-connectivity is both necessary and sufficient for the problem of information dissemination in the presence of either f faulty nodes [20] or f *fixed* faulty links [19].[3] However, both node-connectivity and edge-connectivity are not adequate for our problem as illustrated in Section 3.

Link failures have also been addressed under other contexts, such as distributed method for wireless control network [14], reliable transmission over packet network [13], or estimation over noisy links [17].

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 in our presentation. In simple graph, there is at most one directed edge from any node i to some other node j (But our results can be extended to multi-graph). 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

[1] A graph $G = (\mathcal{V}, \mathcal{E})$ is said to be k-edge connected, if $G' = (\mathcal{V}, \mathcal{E} - X)$ is connected for all $X \subseteq \mathcal{E}$ such that $|X| < k$.

[2] A graph $G = (\mathcal{V}, \mathcal{E})$ is said to be k-node connected, if $G' = (\mathcal{V} - X, \mathcal{E})$ is connected for all $X \subseteq \mathcal{V}$ such that $|X| < k$.

[3] Unlike the "transient" failures in our model, the faulty links are assumed to be fixed throughout the execution of the algorithm in [19].

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, $N_i^+ = \{ j \mid (i, j) \in \mathcal{E} \}$. 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. Similarly, let E_i^- be the set of incoming links incident to node i. That is, E_i^- contains all the links from nodes in N_i^- to node i, i.e., $E_i^- = \{(j, i) \mid j \in N_i^-\}$.

Failure Model: We consider the transient Byzantine *link* failure model [15, 16] for iterative algorithms in directed network. All nodes are assumed to be *fault-free*, and only send a single message once in each iteration. A link (i, j) is said to be faulty if the message sent by node i is different from the message received by node j in some iteration. Note that in our model, it is possible that link (i, j) is faulty while link (j, i) is fault-free.[4] In every iteration, up to f links may be faulty, at most f links may deliver incorrect message or drop message. Note that different sets of link failures may occur in different iterations.

A faulty link may tamper or drop messages. Also, the faulty links may be controlled by a single omniscient adversary. That is, the adversary is 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 IABC Algorithms and Example Network

In this section, we describe the structure of the *Iterative Approximate Byzantine Consensus* (IABC) algorithms of interest, and state conditions that they must satisfy. The IABC structure is identical to the one in our prior work on node failures [22, 21, 23].

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 $v_i[t-1]$. We assume that the input at each node is lower bounded by a constant μ and upper bounded by a constant U. The iterative algorithm may terminate after a number of iterations that is a function of and U. μ and U are assumed to be known a priori.

The IABC algorithms of interest will require each node i to perform the following three steps in iteration t, where $t > 0$. Note that the message sent via faulty links may deviate from this specification.

[4] For example, the described case is possible in wireless network, if node i's transmitter is broken while node i's receiver and node j's transmitter and receiver all function correctly.

1. *Transmit step:* Transmit current state, namely $v_i[t-1]$, on all outgoing edges (to nodes in N_i^+).
2. *Receive step:* Receive values on all incoming edges (from nodes in N_i^-). Denote by $r_i[t]$ the vector of values received by node i from its neighbors. The size of vector $r_i[t]$ is $|N_i^-|$. The values sent in iteration t are received in the same iteration (unless dropped by the faulty links).
3. *Update step:* Node i updates its state using a transition function T_i as follows. T_i is a part of the specification of the algorithm, and takes as input the vector $r_i[t]$ and state $v_i[t-1]$.

$$v_i[t] = T_i \ (\ r_i[t] \, , \, v_i[t-1] \ ) \tag{1}$$

The following properties must be satisfied by an IABC algorithm in the presence of up to f Byzantine faulty links:

- **Termination**: the algorithm terminates in finite number of iterations.
- **Validity:** $\forall t > 0, \quad \min_{i \in \mathcal{V}} v_i[t] \geq \min_{i \in \mathcal{V}} v_i[t-1]$ and $\max_{i \in \mathcal{V}} v_i[t] \geq \max_{i \in \mathcal{V}} v_i[t-1]$.
- **ϵ-agreement:** If the algorithm terminates after t_{end} iterations, then $\forall i, j \in \mathcal{V}, |v_i[t_{end}] - v_j[t_{end}]| < \epsilon$.

The objective in this paper is to identify the necessary and sufficient conditions for the existence of a *correct* IABC algorithm (i.e., an algorithm satisfying the above properties) for a given $G(\mathcal{V}, \mathcal{E})$.

Example Network We give an example showing that node- and edge-connectivity are not adequate for specifying the *tight* condition in directed graphs. Consider the case when $f = 1$ in the network in Figure 1. In the network, nodes A, B, C, D form a clique, while node E has only incoming edges from nodes B, C, D. It is obvious that the node- and edge-connectivity of the network are less than $2f + 1 = 3$, since node E does not have any outgoing links to any other node. However, the approximate consensus is solvable using IABC algorithms under one (directed) faulty link, since the network satisfies the sufficient condition proved later. The proof is presented in A. Therefore, $2f + 1$ node- and edge-connectivity are not necessary for the existence of IABC algorithms.

4 Necessary Condition

For a correct iterative approximate consensus algorithm to exists in the presence of Byzantine link failures, the graph $G(\mathcal{V}, \mathcal{E})$ must satisfy the necessary condition proved in this section. We now define relations $\Rightarrow$ and $\not\Rightarrow$ that are used frequently in our proofs.

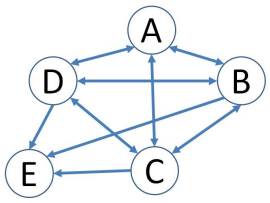

Fig. 1. Example Network

Definition 1. *For non-empty disjoint sets of nodes A and B in $G(\mathcal{V},\mathcal{E})$, $A \Rightarrow B$ iff there exists a node $i \in B$ that has at least $f+1$ incoming links from nodes in A, i.e., $|\{(j,i) \mid j \in A,\ (j,i) \in \mathcal{E}\}| > f$; $A \not\Rightarrow B$ iff $A \Rightarrow B$ is* not *true.*

Condition P : Consider graph $G(\mathcal{V},\mathcal{E})$. Denote by F a subset of $\mathcal{E}$ such that $|F| \leq f$. Let sets L, C, R form a partition of $\mathcal{V}$, such that both L and R are non-empty. Then, in $G' = (\mathcal{V}, \mathcal{E} - F)$, at least one of the two conditions below must be true: (i) $C \cup R \Rightarrow L$; (ii) $L \cup C \Rightarrow R$.

Theorem 1. *Suppose that a correct IABC algorithm exists for $G(\mathcal{V},\mathcal{E})$. Then G satisfies* Condition P.

Proof. The proof is by contradiction. Let us assume that a correct IABC algorithm exists, and for some node partition L, C, R and a subset $F \subseteq \mathcal{E}$ such that $|F| \leq f$, $C \cup R \not\Rightarrow L$ and $L \cup C \not\Rightarrow R$ in $G' = (\mathcal{V}, \mathcal{E} - F)$. Thus, for any $i \in L$, $|\{(k,i) \mid k \in C \cup R,\ (k,i) \in \mathcal{E} - F\}| < f+1$. Similarly, for any $j \in R$, $|\{(k,j) \mid k \in L \cup C,\ (k,j) \in \mathcal{E} - F\}| < f+1$.

Also assume that the links in F (if F is non-empty) all behave faulty, and the rest of the links are all fault-free in every iteration. Note that the nodes are not aware of the identity of the faulty links.

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

In the *Transmit Step* of iteration 1, each node k, sends to nodes in N_k^+ value $v_k[0]$; however, some values sent via faulty links may be tampered. Suppose that the faulty links in F (if non-empty) tamper the messages sent via them in the following way (i) if the link is an incoming link to a node in L, then $m^- < m$ is deliver to that node; (ii) if the link is an incoming link to a node in R, then $M^+ > M$ is deliver to that node; and (iii) if the link is an incoming link to a node in C, then some arbitrary value in interval $[m, M]$ is deliver to that node. This behavior is possible since links in F are Byzantine faulty by assumption. Note that $m^- < m < M < M^+$.

Consider any node $i \in L$. Recall that E_i^- the set of all the node i's incoming links. Let E_i' be the subset of E_i^- that are incident to nodes in $C \cup R$, i.e.,

$$E_i' = \{(j,i) \mid j \in C \cup R, (j,i) \in \mathcal{E}\}.$$

Since $|F| \leq f$, $|E_i^- \cap F| \leq f$. Moreover, by assumption $C \cup R \not\Rightarrow L$; thus, $|E_i' - F| \leq |E_i'| \leq f$. Node i will then receive m^- via the links in $E_i^- \cap F$ (if non-empty) and values in $[m, M]$ via the links in $E_i' - F$, and m via the rest of the links, i.e., links in $E_i^- - E_i' - F$.

Consider the following two cases:

- Both $E_i^- \cap F$ and $E_i' - F$ are non-empty:
  In this case, recall that $|E_i^- \cap F| \leq f$ and $|E_i' - F| \leq f$. From node i's perspective, consider two possible scenarios: (a) links in $E_i^- \cap F$ are faulty, and the other links are fault-free, and (b) links in $E_i' - F$ are faulty, and the other links are fault-free.
  In scenario (a), from node i's perspective, all the nodes may have sent values in interval $[m, M]$, but the faulty links have delivered m^- to node i. According to the validity property, $v_i[1] \geq m$. On the other hand, in scenario (b), all the nodes may have sent values m^- or m, where $m^- < m$; so $v_i[1] \leq m$, according to the validity property. Since node i does not know whether the correct scenario is (a) or (b), it must update its state to satisfy the validity property in both cases. Thus, it follows that $v_i[1] = m$.
- At most one of $E_i^- \cap F$ and $E_i' - F$ is non-empty:
  Recall that by assumption, $|E_i^- \cap F| \leq f$ and $|E_i' - F| \leq f$. Since at most one of the set is non-empty, $|(E_i^- \cap F) \cup (E_i' - F)| \leq f$. From node i's perspective, it is possible that the links in $(E_i^- \cap F) \cup (E_i' - F)$ are all faulty, and the rest of the links are fault-free. In this situation, the values sent to node i via all the fault-free links are all m, and therefore, $v_i[1]$ must be set to m as per the validity property.

Thus, $v_i[1] = m$ 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* property.

The above discussion implies that, at the end of iteration 1, 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 the interval $[m, M]$. 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 $t \geq 0$, $v_i[t] = m$ for all $\forall i \in L$, $v_j[t] = M$ for all $j \in R$ and $v_k[t] \in [m, M]$ for all $k \in C$.

Since both L and R are non-empty, the ϵ-*agreement* property is not satisfied. A contradiction. □

Theorem 1 shows that *Condition P* is necessary. However, *Condition P* is not intuitive. Below, we state an equivalent condition *Condition S* that is easier

to interpret. To facilitate the statement, we introduce the notions of "source component" and "link-reduced graph" using the following three definitions. The link-reduced graph is analogous to the similar concept introduced in our prior work on node failures [22, 21, 23].

Definition 2. Graph decomposition: *Let H be a directed graph. Partition graph H into non-empty strongly connected components, $H_1, H_2, \cdots, H_h$, where h is a non-zero integer dependent on graph H, such that*

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

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

It is known that the decomposition graph H^d is a directed *acyclic* graph [5].

Definition 3. Source component*: Let H be a directed graph, and let H^d be its decomposition as per Definition 2. Strongly connected component H_k of H is said to be a* source component *if the corresponding vertex c_k in H^d is* <u>not</u> *reachable from any other vertex in H^d.*

Definition 4. Link-Reduced Graph: *For a given graph $G(\mathcal{V}, \mathcal{E})$ and $F \subset \mathcal{E}$, a graph $G_F(\mathcal{V}, \mathcal{E}_F)$ is said to be a* link-reduced graph*, if $\mathcal{E}_F$ is obtained by first removing from $\mathcal{E}$ all the links in F, and* then *removing up to f other incoming links at each node in $\mathcal{E} - F$.*

Note that for a given $G(\mathcal{V}, \mathcal{E})$ and a given F, multiple link-reduced graphs G_F may exist.

Now, we state *Condition S*:

Condition S: Consider graph $G(\mathcal{V}, \mathcal{E})$. For any $F \subseteq \mathcal{E}$ such that $|F| \leq f$, every link-reduced graph G_F obtained as per Definition 4 must contain exactly one *source component*.

Then, we show that *Condition S* and *Condition P* specify the equivalent property of the graph.

Lemma 1. *Suppose that* Condition P *holds for graph $G(\mathcal{V}, \mathcal{E})$. Then G satisfies* Condition S.

Proof. By assumption, G contains at least two node, and so does G_F; therefore, at least one source component must exist in G_F. We now prove that G_F cannot contain more than one source component. The proof is by contradiction. Suppose

that there exists a subset $F \subset \mathcal{E}$ with $|F| \leq f$, and the link-reduced graph $G_F(\mathcal{V}, \mathcal{E}_F)$ corresponding to F such that the decomposition of G_F includes at least two source components.

Let the sets of nodes in two such source components of G_F be denoted L and R, respectively. Let $C = \mathcal{V} - L - R$. Observe that L, C, R form a partition of the nodes in $\mathcal{V}$. Since L is a source component in G_F, it follows that there are no directed links in $\mathcal{E}_F$ from any node in $C \cup R$ to the nodes in L. Similarly, since R is a source component in G_F, it follows that there are no directed links in $\mathcal{E}_F$ from any node in $L \cup C$ to the nodes in R. These observations, together with the manner in which $\mathcal{E}_F$ is defined, imply that (i) there are at most f links in $\mathcal{E} - F$ from the nodes in $C \cup R$ to each node in L, and (ii) there are at most f links in $\mathcal{E} - F$ from the nodes in $L \cup C$ to each node in R. Therefore, in graph $G' = (\mathcal{V}, \mathcal{E} - F)$, $C \cup R \not\Rightarrow L$ and $L \cup C \not\Rightarrow R$. Thus, $G = (\mathcal{V}, \mathcal{E})$ does not satisfies *Condition P*, since $F \subseteq \mathcal{E}$ and $|F| \leq f$, a contradiction. □

Lemma 2. *Suppose that* Condition S *holds for graph* $G(\mathcal{V}, \mathcal{E})$*. Then,* G *satisfies* Condition P*.*

Proof. The proof is by contradiction. Suppose that *Condition P* does not hold for graph $G = (\mathcal{V}, \mathcal{E})$. Thus, there exist a subset $F \subset \mathcal{E}$, where $|F| \leq f$, and a node partition L, C, R, where L and R are both non-empty, such that $C \cup R \not\Rightarrow L$ and $L \cup C \not\Rightarrow R$ in $G' = (\mathcal{V}, \mathcal{E} - F)$.

We now constructed a link-reduced graph $G_F(\mathcal{V}, \mathcal{E}_F)$ corresponding to set F. First, remove all links in F from $\mathcal{E}$. Then since $C \cup R \not\Rightarrow L$, the number of links at each node in L from nodes in $C \cup R$ is at most f; remove all these links. Similarly, for every node $j \in R$, remove all links from nodes in $L \cup C$ to j (recall that by assumption, there are at most f such links). The remaining links form the set $\mathcal{E}_F$. It should be obvious that $G_F(\mathcal{V}, \mathcal{E}_F)$ satisfies Definition 4; hence, G_F is a valid link-reduced graph.

Now, observe that by construction, in the link-reduced graph $G_F(\mathcal{V}, \mathcal{E}_F)$, there are no incoming links to nodes in R from nodes in $L \cup C$; similarly, in G_F, there are no incoming links to nodes in L from nodes in $C \cup R$. It follows that for each $i \in L$, there is no path using links in $\mathcal{E}_F$ from i to nodes in R; similarly, for each $j \in R$, there is no path using links in $\mathcal{E}_F$ from j to nodes in L. Thus, G_F must contain at least two source components. Therefore, the existence of G_F implies that G violates *Condition S*, a contradiction. □

Lemmas 1 and 2 imply that *Condition P* is equivalent to *Condition S*. An alternate interpretation of *Condition S* is that in every link-reduced graph G_F, non-fault-tolerant iterative consensus must be possible.

4.1 Useful Properties

Suppose $G(\mathcal{V}, \mathcal{E})$ satisfies *Condition P* and *Condition S*. We provide two lemmas below to state some properties of $G(\mathcal{V}, \mathcal{E})$ that are useful for analyzing the iterative algorithm presented later. Lemma 3 intuitively states that at least one node

can propagate its value to all the other nodes (over enough number of iterations). Lemma 4 states that each node needs to have enough incoming neighbors for achieving approximate consensus.

Lemma 3. *Suppose that graph $G(\mathcal{V}, \mathcal{E})$ satisfies* Condition S. *Then, in any link-reduced graph $G_F(\mathcal{V}, \mathcal{E}_F)$, there exists a node that has a directed path to all the other nodes in $\mathcal{V}$.*

Proof. Recall that *Condition S* states that any link-reduced graph $G_F(\mathcal{V}, \mathcal{E}_F)$ has a single source component. By the definition of source component, any node in the source component (say node s) has directed paths using edges in $\mathcal{E}_F$ to all the other nodes in the source component, since the source component is a strongly connected component. Also, by the uniqueness of the source component, all other strongly connected components in G_F (if any exist) are not source components, and hence reachable from the source component using the edges in $\mathcal{E}_F$. Therefore, node s also has directed paths to all the nodes in $\mathcal{V}$ that are not in the source component as well. Therefore, node s has directed paths to all the other nodes in $\mathcal{V}$. This proves the lemma. □

Lemma 4. *For $f > 0$, if graph $G = (\mathcal{V}, \mathcal{E})$ satisfies* Condition P*, then each node in $\mathcal{V}$ has in-degree at least $2f + 1$, i.e., for each $i \in \mathcal{V}, |N_i^-| \geq 2f + 1$.*

Proof. The proof is by contradiction. By assumption in the lemma, $f > 0$, and graph $G = (\mathcal{V}, \mathcal{E})$ satisfies *Condition P*.

Suppose that there exists a node $i \in \mathcal{V}$ such that $|N_i^-| \leq 2f$. Define $L = \{i\}, C = \emptyset$, and $R = \mathcal{V} - \{i\}$. Note that sets L, C, R form a partition of $\mathcal{V}$. Now, define an edge set F such that $F \subseteq \mathcal{E}, |F| \leq f$, and F contains $\min(f, |N_i^-|)$ incoming links from nodes in R to node i.

Observe that $f > 0$, and $|L \cup C| = 1$. Thus, there can be at most 1 link from $L \cup C$ to any node in R in $G' = (\mathcal{V}, \mathcal{E} - F)$. Therefore, $L \cup C \not\Rightarrow R$ in $G' = (\mathcal{V}, \mathcal{E} - F)$. Then, recall that E_i^- is the set of all the node i's incoming links. Since $L = \{i\}$ and $C = \emptyset$, $E_i^- = \{(j, i) \mid j \in R\}$. Also, since $|E_i^-| = |N_i^-| \leq 2f$, and F contains $\min(f, |N_i^-|)$ links in E_i^-, $|E_i^- - F| \leq 2f - f = f$. Therefore, $C \cup R \not\Rightarrow L$ in $G(\mathcal{V}, \mathcal{E} - F)$. Thus, $G' = (\mathcal{V}, \mathcal{E})$ does not satisfy *Condition P*, a contradiction. □

5 Algorithm 1

We will prove that there exists a correct IABC algorithm particularly Algorithm 1 below that satisfies the termination, validity and ϵ-agreement properties provided that the graph $G(\mathcal{V}, \mathcal{E})$ satisfies *Condition S*. This implies that *Condition P* and *Condition S* ares also sufficient. Algorithm 1 has the iterative structure described in Section 3, and it is similar to algorithms that were analyzed in prior work as well [22, 21] (although correctness of the algorithm under the necessary condition (*Conditions P* and *S*) has not been proved previously).

Algorithm 1

1. *Transmit step:* Transmit current state $v_i[t-1]$ on all outgoing edges and self-loop.
2. *Receive step:* Receive values on all incoming edges and self-loop. These values form vector $r_i[t]$ of size $|N_i^-| + 1$ (including the value from node i itself). When a node expects to receive a message from an incoming neighbor but does not receive the message, the message value is assumed to be equal to its own state, i.e., $v_i[t-1]$.
3. *Update step:* Sort the values in $r_i[t]$ in an increasing order (breaking ties arbitrarily), and eliminate the smallest and largest f values. Let $N_i^*[t]$ denote the set of nodes from whom the remaining $|N_i^-| + 1 - 2f$ values in $r_i[t]$ were received. Note that as proved in Lemma 4, each node has at least $2f + 1$ incoming neighbors. Thus, when $f > 0$, $|N_i^*[t]| \geq 2$. Let w_j denote the value received from node $j \in N_i^*[t]$. Note that $i \in N_i^*[t]$. Hence, for convenience, define $w_i = v_i[t-1]$ to be the value node i receives from itself. Observe that if the link from $j \in N_i^*[t]$ is fault-free, then $w_j = v_j[t-1]$.
   Define

$$v_i[t] \;=\; T_i(r_i[t]) \;=\; \sum_{j \in N_i^*[t]} a_i \, w_j \tag{2}$$

   where

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

   The "weight" of each term on the right-hand side of (2) is a_i. Note that $|N_i^*[t]| = |N_i^-| + 1 - 2f$, and $i \notin N_i^*[t]$ because $(i, i) \notin \mathcal{E}$. Thus, the weights on the right-hand side add to 1. Also, $0 < a_i \leq 1$.[5]

Termination: Each node terminates after completing iteration t_{end}, where t_{end} is a constant defined later in Equation (9). The value of t_{end} depends on graph $G(\mathcal{V}, \mathcal{E})$, constants U and μ defined earlier in Section 3 and parameter ϵ in ϵ-agreement property.

6 Sufficiency (Correctness of Algorithm 1)

We will prove that given a graph $G(\mathcal{V}, \mathcal{E})$ satisfying *Condition S*, Algorithm 1 is correct, i.e., Algorithm 1 satisfies *termination, validity, ϵ-agreement* properties. Therefore, *Condition S* and *Condition P* are proved to be sufficient. We borrow the matrix analysis from the work on non-fault-tolerant consensus [9, 3]. The proof below follows the same structure in our prior work on node failures [21, 23]; however, such analysis has not been applied in the case of link failures.

In the rest of the section, we assume that $G(\mathcal{V}, \mathcal{E})$ satisfies *Condition S* and *Condition P*. We introduce standard matrix tools to facilitate our proof. Then, we use transition matrix to represent the *Update* step in Algorithm 1, and show how to use these tools to prove the correctness of Algorithm 1 in $G(\mathcal{V}, \mathcal{E})$.

[5] Although f and a_i may be different for each iteration t, for simplicity, we do not explicitly represent this dependence on t in the notations.

6.1 Matrix Preliminaries

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

Definition 5. *A vector is said to be* stochastic *if all 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.*

When presenting matrix products, for convenience of presentation, we adopt the "backward" product convention below, where $a \leq b$,

$$\Pi_{i=a}^{b}\mathbf{A}[i] = \mathbf{A}[b]\mathbf{A}[b-1]\cdots\mathbf{A}[a] \tag{3}$$

For a row stochastic matrix $\mathbf{A}$, coefficients of ergodicity $\delta(\mathbf{A})$ and $\lambda(\mathbf{A})$ are defined as follows [24]:

$$\delta(\mathbf{A}) = \max_{j} \max_{i_1,i_2} |\mathbf{A}_{i_1 j} - \mathbf{A}_{i_2 j}|$$
$$\lambda(\mathbf{A}) = 1 - \min_{i_1,i_2} \sum_{j} \min(\mathbf{A}_{i_1 j}, \mathbf{A}_{i_2 j})$$

Lemma 5. *For any p square row stochastic matrices $\mathbf{A}(1), \mathbf{A}(2), \ldots, \mathbf{A}(p)$,*

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

Lemma 5 is proved in [8]. Lemma 6 below follows from the definition of $\lambda(\cdot)$.

Lemma 6. *If all the elements in any one column of matrix $\mathbf{A}$ are lower bounded by a constant γ, then $\lambda(\mathbf{A}) \leq 1-\gamma$. That is, if $\exists g$, such that $\mathbf{A}_{ig} \geq \gamma \forall i$, then $\lambda(\mathbf{A}) \leq 1-\gamma$.*

It is easy to show that $0 \leq \delta(\mathbf{A}) \leq 1$ and $0 \leq \lambda(\mathbf{A}) \leq 1$, and that the rows of $\mathbf{A}$ are all identical iff $\delta(\mathbf{A}) = 0$. Also, $\lambda(\mathbf{A}) = 0$ iff $\delta(\mathbf{A}) = 0$.

6.2 Correctness of Algorithm 1

Denote by $v[0]$ the column vector consisting of the initial states at all nodes. The i-th element of $v[0]$, $v_i[0]$, is the initial state of node i. Denote by $v[t]$, for $t \geq 1$, the column vector consisting of the states of all nodes at the end of the t-th iteration. The i-th element of vector $v[t]$ is state $v_i[t]$.

For $t \geq 1$, define $F[t]$ to be the set of all links behaving faulty in iteration t. Recall that link (j,i) is said to be faulty in iteration t if the value received by node i is different from what node j sends in iteration t. Then, define N_i^F as

the set of all nodes whose outgoing links to node i is faulty in iteration t, i.e., $N_i^F = \{j \mid j \in N_i^-,\ (j,i) \in F[t]\}$.[6]

Define N_i^r as a subset of incoming neighbors at node i of size at most f, i.e.,[7]

$$N_i^r \subseteq N_i^- \quad \text{such that } |N_i^r| \leq f$$

Now, we state the key lemma that helps prove the correctness of Algorithm 1. In particular, Lemma 7 allows us to use results for non-homogeneous Markov chains to prove the correctness of Algorithm 1. The proof is presented in Appendix B.

Lemma 7. *The* Update *step in iteration t ($t \geq 1$) of Algorithm 1 at the nodes can be expressed as*

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

where $\mathbf{M}[t]$ is an $n \times n$ row stochastic transition matrix with the following property: there exist a constant β ($0 < \beta \leq 1$) that depends only on graph $G(\mathcal{V}, \mathcal{E})$, and N_i^r such that for each $i \in \mathcal{V}$, and for all $j \in \{i\} \cup (N_i^- - N_i^F - N_i^r)$,

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

Matrix $\mathbf{M}[t]$ is said to be a transition matrix for iteration t. Aa the lemmas states, $\mathbf{M}[t]$ is a row stochastic matrix. The proof of Lemma 7 shows how to construct a suitable row stochastic matrix $\mathbf{M}[t]$ for each iteration t. $\mathbf{M}[t]$ depends not only on t but also on the behavior of the faulty links in iteration t.

Theorem 2. *Algorithm 1 satisfies the* Termination, Validity*, and* ϵ-agreement *properties.*

Proof. Sections 6.3, 6.4 and 6.5 provide the proof that Algorithm 1 satisfies the three properties for iterative approximate consensus in the presence of Byzantine links. This proof follows a structure used to prove correctness of other consensus algorithms in our prior work [21, 23]. □

6.3 Validity Property

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

$$v[t] = (\Pi_{u=1}^{t}\mathbf{M}[u])v[0] \tag{5}$$

[6] N_i^F may be different for each iteration t. For simplicity, the notation does not explicitly represent this dependence.

[7] As will be seen later, N_i^r corresponds to the links removed in some link-reduced graph. Thus, the superscript r in the notation stands for "removed." N_i^r may be different for each t. For simplicity, the notation does not explicitly represent this dependence.

Since each $\mathbf{M}[u]$ is row stochastic as shown in Lemma 7, the matrix product $\Pi_{u=1}^{t}\mathbf{M}[u]$ is also a row stochastic matrix. Thus, (5) implies that the state of each node i at the end of iteration t can be expressed as a convex combination of the initial states at all the nodes. Therefore, the validity property is satisfied.

6.4 Termination Property

Algorithm 1 terminates after t_{end} iterations, where t_{end} is a finite constant depending only on $G(\mathcal{V},\mathcal{E}), U, \mu$, and ϵ. Recall that U and μ are defined as upper and lower bounds of the initial inputs at all nodes, respectively. Therefore, trivially, the algorithm satisfies the termination property. Later, using (9), we define a suitable value for t_{end}.

6.5 ϵ-agreement Property

The proof below follows the same structure in our prior works on node failures [21, 23] for proving correctness of other consensus algorithms with Byzantine nodes.

Denote by R_F the set of all the link-reduced graph of $G(\mathcal{V},\mathcal{E})$ corresponding to some faulty link set F. Let

$$r = \sum_{F\subset\mathcal{E},\ |F|\leq f} |R_F|$$

Note that r only depends on $G(\mathcal{V},\mathcal{E})$ and f, and is a finite integer.

Consider iteration t ($t \geq 1$). Recall that $F[t]$ denote the set of faulty links in iteration t. Then for each link-reduced graph $H[t] \in R_{F[t]}$, define connectivity matrix $\mathbf{H}[t]$ as follows, where $1 \leq i, j \leq n$:

- $\mathbf{H}_{ij}[t] = 1$, if either $j = i$, or edge (j, i) exists in link-reduced graph H;
- $\mathbf{H}_{ij}[t] = 0$, otherwise.

Thus, the non-zero elements of row $\mathbf{H}_i[t]$ correspond to the incoming links at node i in the link-reduced graph $H[t]$, or the self-loop at i. Observe that $\mathbf{H}[t]$ has a non-zero diagonal.

Based on *Condition S* and Lemma 7, we can show the following key lemmas.

Lemma 8. *For any $H[t] \in R_{F[t]}$, and $k \geq n$, $\mathbf{H}^k[t]$ has at least one non-zero column, i.e., a column with all elements non-zero.*

Proof. $G(\mathcal{V},\mathcal{E})$ satisfies the *Condition S*. Therefore, by Lemma 3, there exists at least one node p in the link-reduced graph $H[t]$ that has directed paths to all the nodes in $H[t]$ (consisting of the edges in $H[t]$). $\mathbf{H}^k_{jp}[t]$ of product $\mathbf{H}^k[t]$ is 1 if and only if node p has a directed path to node j consisting of at most k edges in $H[t]$. Since the length of the path from p to any other node in $H[t]$ is at most n, and p has directed paths to all the nodes, for $k \geq n$ the p-th column of matrix $\mathbf{H}^k[t]$ will be non-zero.[8] □

[8] That is, all the elements of the column will be non-zero. Also, such a non-zero column will exist in $\mathbf{H}^{n-1}[t]$, too. We use the loose bound of n to simplify the presentation.

For matrices $\mathbf{A}$ and $\mathbf{B}$ of identical dimension, we say that $\mathbf{A} \leq \mathbf{B}$ iff $\gamma \mathbf{A}_{ij} \leq \mathbf{B}_{ij}$ for all i, j. Lemma below relates the transition matrices with the connectivity matrices. Constant β used in the lemma below was introduced in Lemma 7.

Lemma 9. *For any $t \geq 1$, there exists a link-reduced graph $H[t] \in R_{F[t]}$ such that $\beta \mathbf{H}[\mathbf{t}] \leq \mathbf{M}[t]$, where $\boldsymbol{H}[t]$ is the connectivity matrix for $H[t]$.*

Proof. First, let us construct a link-reduced graph $H[t]$ by first removing $F[t]$ from $G(\mathcal{V}, \mathcal{E})$. Recall that $F[t]$ is the set of faulty links in iteration t. Then for each i, remove a set of at most f node i's incoming links as defined in Lemma 7 (N_i^r). As a result, we have obtained a link-reduced graph $H[t]$ such that $\mathbf{M}_{ij}[t] \geq \beta$, if $j = i$ or edge (j, i) is in the link-reduced graph $H[t]$.

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

The statement in the lemma then follows from the above two observations. □

Lemma 10. *For any $z \geq 1$, at least one column in the matrix product $\Pi_{t=u}^{u+rn-1} \mathbf{H}[t]$ is non-zero.*

Proof. Since $\Pi_{t=u}^{u+rn-1} \mathbf{H}[t]$ consists of rn connectivity matrices corresponding to link-reduced graphs, and the number of all link-reduced graphs for F ($|F| \leq f$) is r, connectivity matrices corresponding to at least one link-reduced graph, say matrix $\mathbf{H}_*$, will appear in the above product at least n times.

Now observe that: (i) By Lemma 8, $\mathbf{H}_*^n$ contains a non-zero column, say the k-th column is non-zero, and (ii) by definition, 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.[9] □

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

$$\mathbf{Q}(i) = \Pi_{t=(i-1)rn+1}^{irn} \mathbf{M}[t] \tag{6}$$

From (5) and (6) observe that

$$v[krn] = \left(\Pi_{i=1}^{k} \mathbf{Q}(i) \right) \; v[0] \tag{7}$$

Lemma 11. *For $i \geq 1$, $\mathbf{Q}(i)$ is a scrambling row stochastic matrix, and*

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

[9] The product $\Pi_{t=u}^{u+rn-1} \mathbf{H}[t]$ can be viewed as the product of n instances of $\mathbf{H}_*$ "interspersed" with matrices with non-zero diagonals.

Proof. $\mathbf{Q}(i)$ is a product of row stochastic matrices ($\mathbf{M}[t]$); therefore, $\mathbf{Q}(i)$ is row stochastic. From Lemma 9, for each $t \geq 1$,

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

Therefore,

$$\beta^{rn}\; \mathit{\Pi}_{t=(i-1)rn+1}^{irn}\; \mathbf{H}[t] \;\leq\; \mathit{\Pi}_{t=(i-1)rn+1}^{irn}\; \mathbf{M}[t] \;=\; \mathbf{Q}(i)$$

By using $u = (i-1)n+1$ in Lemma 10, we conclude that the matrix product on the left side of the above inequality contains a non-zero column. Therefore, since $\beta > 0$, $\mathbf{Q}(i)$ on the right side of the inequality also contains a non-zero column.

Observe that rn is finite, and hence, β^{rn} is non-zero. Since the non-zero terms in $\mathbf{H}[t]$ matrices are all 1, the non-zero elements in $\mathit{\Pi}_{t=(i-1)rn+1}^{irn}\mathbf{H}[t]$ must each be ≥ 1. Therefore, there exists a non-zero column in $\mathbf{Q}(i)$ with all the elements in the column being $\geq \beta^{rn}$. Therefore, by Lemma 6, $\lambda(\mathbf{Q}(i)) \leq 1-\beta^{rn}$, and $\mathbf{Q}(i)$ is a scrambling matrix. □

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

$$\begin{aligned}
\delta(\mathit{\Pi}_{u=1}^{t}\mathbf{M}[u]) &= \delta\left(\left(\mathit{\Pi}_{u=(\lfloor \frac{t}{rn} \rfloor)rn+1}^{t}\mathbf{M}[u]\right)\left(\mathit{\Pi}_{u=1}^{\lfloor \frac{t}{rn} \rfloor}\mathbf{Q}(i)\right)\right) \quad \text{by definition of } \mathbf{Q}(u)\\
&\leq \lambda\left(\mathit{\Pi}_{u=(\lfloor \frac{t}{rn} \rfloor)rn+1}^{t}\mathbf{M}[u]\right)\left(\mathit{\Pi}_{u=1}^{\lfloor \frac{t}{rn} \rfloor}\lambda\left(\mathbf{Q}(u)\right)\right) \quad \text{by Lemma 5}\\
&\leq \mathit{\Pi}_{u=1}^{\lfloor \frac{t}{rn} \rfloor}\lambda\left(\mathbf{Q}(u)\right) \quad \text{because } \lambda(\cdot) \leq 1\\
&\leq (1-\beta^{rn})^{\lfloor \frac{t}{rn} \rfloor} \quad \text{by Lemma 11} \qquad (8)
\end{aligned}$$

Observe that the upper bound on right side of (8) depends only on graph $G(\mathcal{V},\mathcal{E})$ and t, and is independent of the input states, and the behavior of the faulty links. Moreover, the upper bound on the right side of (8) is a non-increasing function of t. Define t_{end} as the smallest positive integer such that the right hand side of (8) is smaller than $\frac{\epsilon}{n\max(|U|,|\mu|)}$. Recall that U and μ are defined as the upper and lower bound of the inputs at all nodes. Thus,

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

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

Recall that $\mathit{\Pi}_{u=1}^{t}\mathbf{M}[u]$ is an $n\times n$ row stochastic matrix. let $\mathbf{M}^* = \mathit{\Pi}_{u=1}^{t}\mathbf{M}[u]$. From 5, we have $v_j[t] = \mathbf{M}_j^* v[0]$. That is, the state of any node j can be obtained as the product of the j-th row of $\mathbf{M}^*$ and $v[0]$. Now, consider any two nodes j, k, we have

$$
\begin{aligned}
|v_j[t] - v_k[t]| &= |\mathbf{M}_j^* v[0] - \mathbf{M}_k^* v[0]| \\
&= |\Sigma_{i=1}^n \mathbf{M}_{ji}^* v_i[0] - \Sigma_{i=1}^n \mathbf{M}_{ki}^* v_i[0]| \\
&= |\Sigma_{i=1}^n \left(\mathbf{M}_{ji}^* - \mathbf{M}_{ki}^*\right) v_i[0]| \\
&\leq \Sigma_{i=1}^n |\mathbf{M}_{ji}^* - \mathbf{M}_{ki}^*||v_i[0]| \\
&\leq \Sigma_{i=1}^n \delta(\mathbf{M}^*)|v_i[0]| \\
&\leq n\delta(\mathbf{M}^*)\max(|U|, |\mu|) \\
&\leq n\delta(\Pi_{u=1}^t \mathbf{M}[u])\max(|U|, |\mu|) \qquad (10)
\end{aligned}
$$

Therefore, by (9) and (10), we have

$$|v_j[t_{end}] - v_k[t_{end}]| < \epsilon \qquad (11)$$

Since the output of the nodes equal its state at termination (after t_{end} iterations). Thus, (11) implies that Algorithm 1 satisfies the ϵ-agreement property.

7 Summary

This paper explores approximate consensus problem under transient Byzantine link failure model. We address a particular class of iterative algorithms in arbitrary directed graphs, and prove a necessary and sufficient condition for the graphs to be able to solve the approximate consensus problem iteratively.

References

1. I. Abraham, Y. Amit, and D. Dolev. Optimal resilience asynchronous approximate agreement. In OPODIS, 2004.
2. M. Biely, U. Schmid, and B. Weiss. *Synchronous consensus under hybrid process and link failures.* Theoretical Computer Science, 412(40):5602 5630, 2011.
3. D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Optimization and Neural Computation Series. Athena Scientific, 1997.
4. B. Charron-Bost and A. Schiper. The Heard-Of model: computing in distributed systems with benign faults. Distributed Computing, 22(1):4971, April 2009.
5. S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms.* McGraw-Hill Higher Education, 2006.
6. 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*, May 1986.
7. M. J. Fischer, N. A. Lynch, and M. Merritt. Easy impossibility proofs for distributed consensus problems. PODC '85, 1985. ACM.
8. J. Hajnal. Weak Ergodicity in non-homogeneous Markov Chains. In *Proceedings of the Cambridge Philosophical Society*, volume 54, pages 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. D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. IEEE Symposium on Foundations of Computer Science, Oct. 2003.
11. L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Trans. on Programming Languages and Systems*, 1982.
12. H. J. LeBlanc, H. Zhang, X. Koutsoukos, S. Sundaram. Resilient Asymptotic Consensus in Robust Networks. Selected Areas in Communications, IEEE Journal on , vol.31, no.4, pp.766,781, April 2013.
13. D. S. Lun, M. Médard, R. Koetter, and M. Effros. On coding for reliable communication over packet networks. Physical Communication, 2008.
14. M. Pajic, S. Sundaram, J. Le Ny, G. J. Pappas, and R. Mangharam. Closing the Loop: A Simple Distributed Method for Control over Wireless Networks. international conference on Information Processing in Sensor Networks, 2012.
15. N. Santoro, and P. Widmayer. Time is not a healer. in: Proc. 6th Ann. Symposium on Theoretical Aspects of Computer Science, STACS '89, 1989.
16. N. Santoro and P. Widmayer. Agreement in synchronous networks with ubiquitous faults. Theor. Comput. Sci. 384 (2-3) (2007) 232249.
17. I. D. Schizas, A. Ribeiro, and G. B. Giannakis. Consensus in Ad Hoc WSNs With Noisy Links- Part I: Distributed Estimation of Deterministic Signals. IEEE Transactions on Signal Processing, 2008.
18. U. Schmid, B. Weiss, I. Keidar. Impossibility results and lower bounds for consensus under link failures. SIAM Journal on Computing 38 (5) 19121951, 2009..
19. S. Sundaram, S. Revzen, and G. Pappas. A control-theoretic approach to disseminating values and overcoming malicious links in wireless networks Automatica, 2012.
20. S. Sundaram and C. N. Hadjicostis. Distributed function calculation via linear iterative strategies in the presence of malicious agent. IEEE Transactions on Automatic Control, 2011.
21. L. Tseng and N. H. Vaidya. Iterative approximate byzantine consensus under a generalized fault model. In International Conference on Distributed Computing and Networking (ICDCN), January 2013.
22. N. H. Vaidya, L. Tseng, and G. Liang. Iterative Approximate Byzantine Consensus in Arbitrary Directed Graphs. PODC '12, 2012. ACM.
23. N. H. Vaidya. Iterative Byzantine Vector Consensus in Incomplete Graphs. In International Conference on Distributed Computing and Networking (ICDCN), January 2014.
24. J. Wolfowitz. Products of Indecomposable, Aperiodic, Stochastic Matrices. In *Proceedings of the American Mathematical Society*, volume 14, pages 733–737, 1963.

Appendix

A Example Network

Lemma 12. *The graph in Figure 1 satisfies* Condition P *when* $f = 1$.

Proof. Denote by G the graph in Figure 1. First observe that a clique of 4 nodes satisfies *Condition P* when $f = 1$. Thus, for G, we only need to consider the case when node E is in either L or R; otherwise, some node in L (or R) from the clique (formed by nodes A, B, C, D) will have at least $f + 1 = 2$ incoming links from R (or L) excluding link in F.

Without loss of generality, consider the case when E is in L. Consider the following cases:

- One of the nodes A, B, C, D is in L: say node X is in L besides E. Then node X has at least $f+1$ incoming links from R excluding link in F.
- Two of the nodes A, B, C, D are in L: say nodes X_1, X_2 are in L besides E. Then either node X_1 or X_2 has at least $f+1$ incoming links from R excluding link in F.
- Three of the nodes A, B, C, D are in L: say node Y is the only node in R, since all the other nodes are in L. Then node Y has at least $f+1$ incoming links from L excluding link in F.

In every case, either $L \cup C \rightarrow R$ or $C \cup R \rightarrow L$. Thus, G satisfies *Condition P*. □

B Proof of Lemma 7

We prove the following Lemma in Section 6.

Lemma 7. *The* Update *step in iteration t ($t \geq 1$) of Algorithm 1 at the nodes can be expressed as*

$$v[t] = \boldsymbol{M}[t]v[t-1] \tag{12}$$

where $\mathbf{M}[t]$ is an $n \times n$ row stochastic transition matrix with the following property: there exist a constant β ($0 < \beta \leq 1$) that depends only on graph $G(\mathcal{V}, \mathcal{E})$, and N_i^r such that for each $i \in \mathcal{V}$, and for all $j \in \{i\} \cup (N_i^- - N_i^F - N_i^r)$,

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

Proof. We prove the correctness of Lemma 7 by constructing $\mathbf{M}_i[t]$ for $1 \leq i \leq n$ that satisfies the conditions in Lemma 7. Recall that $F[t]$ denotes the set of faulty links in the t-th iteration.

Consider a node i in iteration t ($t \geq 1$). In the *Update* step of Algorithm 1, recall that the smallest and the largest f values are removed from $r_i[t]$ by node i. Denote by $\mathcal{S}$ and $\mathcal{L}$, respectively, the set of nodes[10] from whom the smallest and the largest f values were received by node i in iteration t. Define sets $\mathcal{S}_g$ and $\mathcal{L}_g$ to be subsets of $\mathcal{S}$ and $\mathcal{L}$ that contain all the nodes from whom node i receives the correct value in $\mathcal{S}$ and $\mathcal{L}$, respectively. That is, $\mathcal{S}_g = \{j \mid j \in \mathcal{S},\ (j,i) \in \mathcal{E} - F[t]\}$ and $\mathcal{L}_g = \{j \mid j \in \mathcal{L},\ (j,i) \in \mathcal{E} - F[t]\}$.

Construction of $\mathbf{M}_i[t]$ differs somewhat depending on whether sets $\mathcal{S}_g, \mathcal{L}_g$ and N_i^F are empty or not. We divide the possibilities into 3 separate cases:

[10] Although $\mathcal{S}$ and $\mathcal{L}$ may be different for each iteration t, for simplicity, we do not explicitly represent this dependence on t in the notations $\mathcal{S}$ and $\mathcal{L}$.

– Case I: $\mathcal{S}_g \neq \emptyset, \mathcal{L}_g \neq \emptyset$, and $N_i^F \neq \emptyset$.
– Case II: $\mathcal{S}_g \neq \emptyset, \mathcal{L}_g \neq \emptyset$, and $N_i^F = \emptyset$.
– Case III: at most one of $\mathcal{S}_g$ and $\mathcal{L}_g$, and $N_i^F = \emptyset$.

Observe that if $\mathcal{S}_g$ ($\mathcal{L}_g$) is empty, then $N_i^F = \emptyset$ and $\mathcal{L} = \mathcal{L}_g$ ($\mathcal{S} = \mathcal{S}_g$), since there are at most f faulty links and $|\mathcal{S}| = |\mathcal{L}| = f$. Therefore, the 3 cases above cover all the possible scenarios.

Case I

In Case I, $\mathcal{S}_g \neq \emptyset, \mathcal{L}_g \neq \emptyset$, and $N_i^F \neq \emptyset$. Let $m_{\mathcal{S}}$ and $m_{\mathcal{L}}$ be defined as shown below. Recall that the incoming links from the nodes in $\mathcal{S}_g$ and $\mathcal{L}_g$ to node i are all fault-free, and therefore, for any node $j \in \mathcal{S}_g \cup \mathcal{L}_g$, $w_j = v_j[t-1]$ (in the notation of Algorithm 1). That is, the value received by node i from node j is exactly the state at node j in iteration $t-1$.

$$m_{\mathcal{S}} = \frac{\sum_{j \in \mathcal{S}_g} v_j[t-1]}{|\mathcal{S}_g|} \quad \text{and} \quad m_{\mathcal{L}} = \frac{\sum_{j \in \mathcal{L}_g} v_j[t-1]}{|\mathcal{L}_g|}$$

Now, consider any node $k \in N_i^F$. By the definition of sets $\mathcal{S}_g$ and $\mathcal{L}_g$, $m_{\mathcal{S}} \leq w_k \leq m_{\mathcal{L}}$. Therefore, we can find weights $S_k \geq 0$ and $L_k \geq 0$ such that $S_k + L_k = 1$, and

$$w_k = S_k \; m_{\mathcal{S}} + L_k \; m_{\mathcal{L}} \tag{13}$$

$$= \frac{S_k}{|\mathcal{S}_g|} \sum_{j \in \mathcal{S}_g} v_j[t-1] + \frac{L_k}{|\mathcal{L}_g|} \sum_{j \in \mathcal{L}_g} v_j[t-1] \tag{14}$$

Clearly, at least one of S_k and L_k must be $\geq 1/2$.

We now define elements $\mathbf{M}_{ij}[t]$ of row $\mathbf{M}_i[t]$:

– For $j \in N_i^*[t] - N_i^F$: In this case, either the edge (j,i) is fault-free, or $j = i$. For each such j, define $\mathbf{M}_{ij}[t] = a_i$. This is obtained by observing in (2) that the contribution of such a node j to the new state $v_i[t]$ is $a_i \; w_j = a_i \; v_j[t-1]$. The elements of $\mathbf{M}_i[t]$ defined here add up to

$$|N_i^*[t] - N_i^F| \; a_i$$

– For $j \in \mathcal{S}_g \cup \mathcal{L}_g$: In this case, the edge (j,i) is a fault-free.
For each $j \in \mathcal{S}_g$,

$$\mathbf{M}_{ij}[t] \; = \; a_i \sum_{k \in N_i^F} \frac{S_k}{|\mathcal{S}_g|}$$

and for each node $j \in \mathcal{L}_g$,

$$\mathbf{M}_{ij}[t] \; = \; a_i \sum_{k \in N_i^F} \frac{L_k}{|\mathcal{L}_g|}$$

To obtain these two expressions, we represent value w_k sent via faulty link (k, i) for each $k \in N_i^F$ using (14). Recall that this node k contributes $a_i w_k$ to (2). The above two expressions are then obtained by summing (14) over all the nodes in N_i^F, and replacing this sum by equivalent contributions by nodes in $\mathcal{S}_g$ and $\mathcal{L}_g$.

The elements of $\mathbf{M}_i[t]$ defined here add up to $a_i \sum_{k \in N_i^F}(S_k + L_k) = |N_i^F|\, a_i$

– For $j \in \mathcal{V} - ((N_i^* - N_i^F) \cup \mathcal{S}_g \cup \mathcal{L}_g)$: These nodes have not yet been considered above. For each such node j, define $\mathbf{M}_{ij}[t] = 0$.

With the above definition of $\mathbf{M}_i[t]$, it should be easy to see that $\mathbf{M}_i[t]\, v[t-1]$ is, in fact, identical to $v_i[t]$ obtained using (2). Thus, the above construction of $\mathbf{M}_i[t]$ results in the values sent via faulty links to (2) being replaced by an equivalent contribution from the nodes in $\mathcal{L}_g$ and $\mathcal{S}_g$.

Properties of $\boldsymbol{M}_i[t]$: First, we show that $\mathbf{M}[t]$ is row stochastic. Observe that all the elements of $\mathbf{M}_i[t]$ are non-negative. Also, all the elements of $\mathbf{M}_i[t]$ above add up to

$$|N_i^*[t] - N_i^F|\, a_i + |N_i^F|\, a_i = |N_i^*[t]|\, a_i = 1$$

because $a_i = 1/|N_i^*[t]|$ as defined in Algorithm 1. Thus, $\mathbf{M}_i[t]$ is a stochastic row vector.

Recall that from the above discussion, for $k \in N_i^F$, one of S_k and L_k must be $\geq 1/2$. Without loss of generality, assume that $S_s \geq 1/2$ for all nodes $s \in N_i^F$. Consequently, for each node $j \in \mathcal{S}_g$, $\mathbf{M}_{ij}[t] \geq \frac{a_i}{|\mathcal{S}_g|} S_s \geq \frac{a_i}{2|\mathcal{S}_g|}$. Also, for each node j in $N_i^*[t] - N_i^F$, $\mathbf{M}_{ij}[t] = a_i$. Thus, if β is chosen such that

$$0 < \beta \leq \frac{a_i}{2|\mathcal{S}_g|} \tag{15}$$

and N_i^r is defined to be $\mathcal{L}_g$, then the condition in the lemma holds for node i. That is, for all $j \in \{i\} \cup (N_i^- - N_i^F - N_i^r)$,

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

Case II

Now, we consider the case when $\mathcal{S}_g \neq \emptyset, \mathcal{L}_g \neq \emptyset$, and $N_i^F = \emptyset$. That is, when each of $\mathcal{S}$ and $\mathcal{L}$ contains at least one node from which the node i receives correct value, and node i receives correct value(s) from all the node(s) in $N_i^*[t]$. In fact, the analysis of Case II is very similar to the analysis presented above in Case I. We now discuss how the analysis of Case I can be applied to Case II. Rewrite (2) as follows:

$$v_i[t] = \frac{a_i}{2} v_i[t-1] + \frac{a_i}{2} v_i[t-1] + \sum_{j \in N_i^*[t]-\{i\}} a_i w_j \tag{16}$$

$$= a_i w_z + a_i w_i + \sum_{j \in N_i^*[t]-\{i\}} a_i w_j \tag{17}$$

In the above equation, z is to be viewed as a "virtual" incoming neighbor of node i, which has sent value $w_z = \frac{v_i[t-1]}{2}$ to node i in iteration t. With the above rewriting of state update, the value received by node i from itself should be viewed as $w_i = \frac{v_i[t-1]}{2}$ instead of $v_i[t-1]$. With this transformation, Case II now becomes identical to Case I, with virtual node z being treated as an incoming neighbor of node i.

In essence, a part of node i's contribution (half, to be precise) is now replaced by equivalent contribution by nodes in $\mathcal{L}_g$ and $\mathcal{S}_g$. We now define elements $\mathbf{M}_{ij}[t]$ of row $\mathbf{M}_i[t]$:

- For $j = i$: $\mathbf{M}_{ij}[t] = \frac{a_i}{2}$. This is obtained by observing in (2) that node i's contribution to the new state $v_i[t]$ is $a_i \frac{v_i[t-1]}{2}$.
- For $j \in N_i^*[t]-\{i\}$: In this case, j is a node from which node i receives correct value. For each such j, define $\mathbf{M}_{ij}[t] = a_i$. This is obtained by observing in (2) that the contribution of node j to the new state $v_i[t]$ is $a_i w_j = a_i v_j[t-1]$.
- For $j \in \mathcal{S}_g \cup \mathcal{L}_g$: In this case, j is a node in $\mathcal{S}$ or $\mathcal{L}$ from which node i receives correct value.
  For each $j \in \mathcal{S}_g$,
  $$\mathbf{M}_{ij}[t] \;=\; \frac{a_i}{2} \frac{S_z}{|\mathcal{S}_g|}$$
  and for each node $j \in \mathcal{L}_g$,
  $$\mathbf{M}_{ij}[t] \;=\; \frac{a_i}{2} \frac{L_z}{|\mathcal{L}_g|}$$
  where S_z and L_z are chosen such that $S_z + L_z = 1$ and $w_z = \frac{v_i[t-1]}{2} = \frac{S_z}{2} m_{\mathcal{S}} + \frac{L_z}{2} m_{\mathcal{L}}$. Note that such S_z and L_z exist because by definition of $\mathcal{S}_g$ and $\mathcal{L}_g$, $v_i[t-1] \geq w_j,\ \forall j \in S_g$ and $v_i[t-1] \leq w_j,\ \forall j \in L_g$. Then the two expressions above are obtained by replacing the contribution of the virtual node z by an equivalent contribution by the nodes in $\mathcal{S}_g$ and $\mathcal{L}_g$, respectively.
- For $j \in \mathcal{V} - (N_i^*[t] \cup \mathcal{S}_g \cup \mathcal{L}_g)$: These nodes have not yet been considered above. For each such node j, define $\mathbf{M}_{ij}[t] = 0$.

Properties of $\boldsymbol{M}_i[t]$: By argument similar to that in *Case I*, $\mathbf{M}_i[t]$ is row stochastic. Without loss of generality, suppose that $S_z \geq 1/2$. Then for each node $j \in \mathcal{S}_g$, $\mathbf{M}_{ij}[t] = \frac{a_i}{2|\mathcal{S}_g|} S_z \geq \frac{a_i}{4|\mathcal{S}_g|}$. Also, for node j in $N_i^*[t] - \{i\}$, $\mathbf{M}_{ij}[t] = a_i$,

and $\mathbf{M}_{ii}[t] = \frac{a_i}{2}$. Recall that by definition, $|\mathcal{S}_g| \geq 1$. Hence, if β is chosen such that

$$0 < \beta \leq \frac{a_i}{4|\mathcal{S}_g|} \tag{18}$$

and N_i^r is defined to be equal to $\mathcal{L}_g$, then the condition in the Lemma 7 holds for node i. That is, $\mathbf{M}_{ij}[t] \geq \beta$ for $j \in \{i\} \cup (N_i^- - N_i^F - N_i^r)$.

Case III

Here, we consider the case when at most one of $\mathcal{S}_g$ and $\mathcal{L}_g$ is empty, and $N_i^F = \emptyset$. Without loss of generality, suppose that $\mathcal{S}$ contains only nodes whose outgoing links to node i is faulty in iteration t, i.e., $\mathcal{S} = \{j \mid (j,i) \in F[t]\}$. Since there are at most f faulty links and $|\mathcal{S}| = f$, $\mathcal{L} = \mathcal{L}_g$. That is, the value received from each node in $\mathcal{L}$ by node i is correct.

In this case, define $\mathbf{M}_{ij}[t] = a_i$ for $j \in N_i^*[t]$; define $\mathbf{M}_{ij} = 0$ for all other nodes j.

Properties of $\boldsymbol{M}_i[t]$:

All the elements of $\mathbf{M}_i[t]$ are non-negative. The elements of $\mathbf{M}_i[t]$ defined above add up to

$$|N_i^*[t]|\ a_i = 1$$

Thus, $\mathbf{M}_i[t]$ is a stochastic row vector.

In Case III, recall that for any node j in $N_i^*[t]$, $\mathbf{M}_{ij}[t] = a_i$. Thus, if β is chosen such that

$$0 < \beta \leq a_i \tag{19}$$

and N_i^r is defined to be equal to $\mathcal{L}$, then the condition in the Lemma 7 holds for node i.

Putting Cases Together

Now, let us consider Cases I-III together. From the definition of a_i in Algorithm 1, observe that $a_i \geq \frac{1}{|N_i^-|+1}$ (because $f \geq 0$). Let us define

$$\alpha = \min_{i \in \mathcal{V}} \frac{1}{|N_i^-| + 1}$$

Moreover, observe that $|\mathcal{S}_g| \leq n$ and $|\mathcal{L}_g| \leq n$. Then define β as

$$\beta = \frac{\alpha}{4n} \tag{20}$$

This definition satisfies constraints on β in Cases I through III (conditions (15), (18) and (19)). Thus, Lemma 7 holds for all three cases with this choice in (20). □