Automatica 48 (2012) 2894–2901

Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

A control-theoretic approach to disseminating values and overcoming malicious links in wireless networks☆

Shreyas Sundaram [a,1], Shai Revzen [b], George Pappas [b]

[a] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada

[b] GRASP Laboratory and the Department of Electrical and Systems Engineering, University of Pennsylvania, PA 19104, USA

ARTICLE INFO

Article history:
Received 2 December 2010
Received in revised form
13 February 2012
Accepted 30 May 2012
Available online 18 July 2012

Keywords:
Security
Fault-tolerant systems
Information dissemination
Distributed systems
Network reliability

ABSTRACT

We consider a network in which every node has a value that it wishes to disseminate to all other nodes, despite an attack by an adversary that can falsify messages on a number of the links. To achieve this objective, we study a class of linear iterative strategies in which, at each time-step, each node in the network broadcasts a value to its neighbors that is a linear combination of its previous value and the values received from its neighbors. We take the number of unreliable links to be bounded, in that the number of incoming unreliable links to any node plus the total number of other nodes with incoming unreliable links is no greater than some nonnegative integer f. We show that the linear iterative strategy will be resilient to the unreliable links if and only if the vertex connectivity is at least $2f + 1$. If this condition is satisfied, we show that almost any choice of weights in the linear combinations will suffice to provide resilience. We further show that each node can identify the exact set of unreliable links that directly enter that node, and can communicate this information to the other nodes via the linear strategy.

1. Introduction

The problems of information dissemination (Giridhar & Kumar, 2005; Hromkovic, Klasing, Pelc, Ruzicka, & Unger, 2005; Lynch, 1996) and distributed consensus (Lynch, 1996; Olfati-Saber, Fax, & Murray, 2007) in networks have received considerable attention from researchers from a variety of disciplines, and, in recent years, *linear* strategies have gained prominence as a mechanism to achieve these tasks. In the communications community, these strategies (under the moniker of *network coding*) have been shown to maximize the rate at which information can be transmitted from source to sink nodes in the network (Koetter & Medard, 2003; Yeung, Li, Cai, & Zhang, 2005), and to be resilient to a fixed number of malicious links (or nodes) provided that the minimum cut between the source and sink nodes is sufficiently high (Jaggi et al., 2008; Koetter & Kschischang, 2008). The control community has focused on the case in which *all* nodes in the network have a *single* value to disseminate.[2] In this case, it has been shown that, if each node in the network updates its value to an appropriate weighted linear combination of its previous value and those of its neighbors, given certain conditions on the network topology, all of the nodes will asymptotically converge to the same value (e.g., see Olfati-Saber et al., 2007 and Ren, Beard, & Atkins, 2005 and the references therein). More generally, it was shown in Deb, Médard, and Choute (2006), Mosk-Aoyama and Shah (2006), and Sundaram and Hadjicostis (2008) that this linear iterative strategy will allow any node in the network to obtain *all* of the node values in a finite number of time-steps. Furthermore, it was shown in Sundaram and Hadjicostis (2011) that linear strategies are resilient to up to f malicious (and potentially colluding) nodes, provided that the network graph has (vertex) connectivity of at least $2f + 1$. This result was extended in Pasqualetti, Bicchi, and Bullo (2012) to analyze linear iterative strategies for asymptotic consensus in the presence of faulty agents (in addition to malicious agents), and Teixeira, Sandberg, and Johansson (2010) studied the problem of detecting attacks in networks of linear continuous-time systems.

The underlying network model for the linear iterations studied in the control literature is one in which each node can broadcast identical messages to all its neighbors. When technically feasible,

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Huijun Gao under the direction of Editor Ian R. Petersen. This material is based upon work supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

E-mail addresses: ssundara@uwaterloo.ca (S. Sundaram), shrevzen@seas.upenn.edu (S. Revzen), pappasg@seas.upenn.edu (G. Pappas).

[1] Tel.: +1 519 888 4567x36908; fax: +1 519 746 3077.

[2] For example, these values could represent measurements of the environment in a sensor network (Abdelzaher, He, & Stankovic, 2004), or optimization variables in a multi-agent system (Tsitsiklis, Bertsekas, & Athans, 1986).

0005-1098/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2012.06.072

these local broadcasts offer considerable bandwidth advantages over schemes that require different messages to be sent to each neighbor. Regardless of the network technology, it is likely that random network faults manifest themselves as intermittent communication (link) failures between specific transmitters and receivers, rather than the complete malfunction of a transmitter or receiver (a node failure). In the case of a concerted attack by an adversary, interception and corruption of messages – be they wireless radio packets or laser pulses on a fiber-optic cable – may be far easier to achieve than subversion of a network node. Thus, a distributed information dissemination algorithm that can tolerate and isolate faulty or malicious *links* would be of particular value in the design of reliable mesh networks.

The main contribution of this paper is to extend the analysis of linear iterative strategies with malicious nodes (studied in Pasqualetti et al. (2012) and Sundaram and Hadjicostis (2011)) to the case in which network *links* are malicious. While there are strong parallels between the two scenarios, some care must be taken to prove that nodes that have incoming malicious links can, in fact, recover the correct initial values of the other nodes. Our contribution introduces the notion of an "f-wise safe set" which is shown to characterize all malicious links that can be overcome by the linear iterative strategy.

Our analysis extends the discussion in Teixeira et al. (2010) to the case in which the malicious links can be arbitrarily placed in the network, as opposed to only on the outgoing links of a single node. The setting in this paper also differs from that traditionally considered in network coding (Jaggi et al., 2008), in which a set of sources wish to transmit a stream of information reliably to a set of sinks; our work considers the problem in which every node in the network has a single value to disseminate. As in Jaggi et al. (2008), our algorithm is capable of resisting "Byzantine failures", in which links are subverted to execute a coordinated *worst-case attack*. While this assumption may seem overly pessimistic, Internet Protocol networks are routinely attacked in a coordinated fashion by "bot-nets" that communicate with each other via a command-and-control channel to share a global state (Bailey, Cooke, Jahanian, Xu, & Karir, 2009)—an attack scenario not far removed from the one we propose.

2. Background

2.1. Notation

We use bold-face uppercase letters ($\mathbf{A}$) to indicate matrices, and bold-face lowercase letters ($\mathbf{a}$) to indicate vectors (the orientation will be clear from the context). The symbol $\mathbf{I}_N$ denotes the $N \times N$ identity matrix, $\mathbf{A}'$ indicates the transpose of matrix $\mathbf{A}$, and $\mathbf{e}_i$ denotes the column vector of appropriate size with a 1 in its ith position and zeros elsewhere. We denote the cardinality of a set $\mathcal{S}$ by $|\mathcal{S}|$. For a pair of sets $\mathcal{S}$ and $\mathcal{T}$, $\mathcal{S} \setminus \mathcal{T}$ denotes the set of elements of $\mathcal{S}$ that are not in $\mathcal{T}$. The set of nonnegative integers is denoted by $\mathbb{N}$, the set of reals by $\mathbb{R}$, and the set of complex numbers by $\mathbb{C}$.

2.2. Concepts from graph theory

We will follow the graph terminology from West (2001). A graph is an ordered pair $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$, where the vertex set $\mathcal{X} = \{v_1, \ldots, v_N\}$ represents the network nodes, and the (directed) edge set $\mathcal{E}$ contains ordered pairs of different vertices, representing (directed) network links. The vertices $\mathcal{N}_i = \{u | (u, v_i) \in \mathcal{E}\}$ are *neighbors of* v_i, and the *in-degree of* v_i is $\deg(v_i) \triangleq |\mathcal{N}_i|$. A *path* P from vertex v_0 to vertex v_t is a sequence of vertices $v_0, v_1, \ldots, v_t$ such that v_i is a neighbor of v_{i+1}, $i \in \{0, 1, \ldots, t-1\}$. A graph is *strongly connected* if, for every two vertices $v, u \in \mathcal{X}$, $v \neq u$, there is a path between v and u. If there exist $v, u \in \mathcal{X}$ such that there is no path from v to u, the graph is *disconnected*. A *vertex cut* is a set $\mathcal{S} \subset \mathcal{X}$ such that removing the vertices of $\mathcal{S}$ (and the associated edges) causes the graph to be disconnected. The *connectivity* of a graph is the smallest size of a vertex cut.

2.3. Linear iterative strategies for information dissemination

Unidirectional communication links in a network can be conveniently modeled via a directed graph $\mathcal{G}$. Suppose that each node v_i has some initial value, given by $x_i[0] \in \mathbb{R}$, and the goal is to broadcast all these values to some (or all) of the nodes in the network. The linear iterative strategy discussed in Section 1 operates as follows: at each time-step k, each node updates its value $x_i[k]$ to a weighted sum of the values of itself and its neighbors. A node is said to be malicious if it updates its value arbitrarily at each time-step (i.e., it does not necessarily follow the linear iterative strategy). As discussed in Sundaram and Hadjicostis (2011), the linear update for any node v_i can be modeled by

$$x_i[k+1] = \left(w_{ii} x_i[k] + \sum_{j \in \mathcal{N}_i} w_{ij} x_j[k] \right) + u_i[k], \tag{1}$$

where $u_i[k]$ is an arbitrary additive error (it is zero if node v_i operates as expected at time-step k). The following result from Sundaram and Hadjicostis (2011) reveals that the connectivity of the network completely characterizes the resiliency of the linear iterative strategy to malicious behavior by a certain number of nodes.

Theorem 1. *Let the graph of network $\mathcal{G} = \{\mathcal{X}, \mathcal{E}\}$ have connectivity κ. If $\kappa \leq 2f$, then, regardless of the choice of weights and the number of time-steps for which the linear strategy is run, there is a choice of f malicious nodes that can update their values to prevent some correctly functioning node from obtaining the initial values of certain other nodes. If $\kappa \geq 2f+1$, then, for almost any choice of weights, every node in the network can correctly determine all initial values after running the linear iteration for at most $|\mathcal{X}|$ time-steps, even when there are up to f malicious nodes that update their values arbitrarily (and possibly in a coordinated manner) at each time-step.*

3. Networks with malicious links

We wish to extend the paradigm of resilient networks based on linear iterative strategies to networks in which *links* may be subverted by an adversary. Suppose that link (v_l, v_i) in the network modifies the value that node v_i receives from node v_l at time-step k to be $x_l[k] + z_{li}[k]$, where $z_{li}[k]$ is an (arbitrary) additive error. Similarly to (1), the value of node v_i at time-step $k+1$ is

$$x_i[k+1] = \left(w_{ii} x_i[k] + \sum_{j \in \mathcal{N}_i} w_{ij} x_j[k] \right) + w_{il} z_{li}[k]. \tag{2}$$

Definition 1. Suppose that all nodes run the linear iteration for T time-steps in order to disseminate information (for some $T \in \mathbb{N}$). Link (v_l, v_i) is said to be *unreliable* if $z_{li}[k]$ is nonzero for at least one time-step k, $0 \leq k \leq T-1$.

When there are multiple unreliable links entering a node, there is the possibility that the errors that they introduce cancel each other out in expression (2). This motivates the following definition.

Definition 2. A set of unreliable links $\mathcal{L}$ into the same node v_i is *malicious* if $\sum_{l \in \mathcal{L}} w_{il} z_{li}[k]$ is nonzero for at least one time-step k, $0 \leq k \leq T-1$.

Comparing (2) to (1), we note that a malicious error introduced by a link into node v_i reduces mathematically to a malicious error introduced by v_i itself, with $u_i[k] = w_{il}z_{li}[k]$. This leads to the following corollary of Theorem 1.

Corollary 1. *If $\kappa \leq 2f$, then, regardless of the choice of weights and the number of time-steps for which the linear strategy is run, there is a choice of f malicious links that can modify their values to prevent some correctly functioning node from obtaining the initial values of certain other nodes.*

The proof follows by selecting f nodes whose malicious behavior would disrupt the network (as guaranteed by Theorem 1), then choosing one link into each of these nodes to be malicious, with additive errors chosen so as to mimic the appropriate malicious node behavior (see Pasqualetti et al., 2012 and Sundaram & Hadjicostis, 2011 for an explicit strategy for the values $u_i[k]$).

Remark 1. This model for unreliable links is quite general, and allows each such link (v_l, v_i) to arbitrarily affect the value that node v_i receives from node v_l via appropriate choices of the error $z_{li}[k]$ at each time-step. In fact, we will assume that the adversary is *strong* in that he/she is aware of the state of the entire network and can use that information to decide what errors to inject on the unreliable links. Viewed as a set of colluding malicious links, such "Byzantine" adversaries are common in the literature on fault-tolerant distributed algorithms (Lynch, 1996). The reason for considering such strong adversaries is to enforce worst-case guarantees—if the distributed algorithm can tolerate a Byzantine adversary, it can tolerate less powerful adversaries as well. Having said this, it was shown in Sundaram and Hadjicostis (2011) that the strategy for the f malicious nodes specified by Theorem 1 does not require them to have knowledge of the global state in order to disrupt networks of connectivity less than $2f + 1$; they only have to coordinate their actions between themselves. This fact holds for the setting considered in this paper as well, by the translation of attack strategies from the malicious node case to the malicious link case described above. However, we will also show in what follows that the proposed algorithm is resilient to at least f Byzantine links if the connectivity is $2f + 1$ or more. In summary, even worst-case attackers cannot disrupt the network if the connectivity is sufficiently high, whereas non-worst-case attackers can disrupt the network if the connectivity is low.

3.1. A first attempt to handle link failures

A reasonable first attempt to directly show resilience to unreliable links via the framework in Sundaram and Hadjicostis (2011) would be to replace each link of the form (v_i, v_j) in the network with a new node v_{ij} and two new edges (v_i, v_{ij}) and (v_{ij}, v_j), and then translate malicious link behavior on (v_i, v_j) to malicious node behavior on v_{ij}. While intuitive, this approach presents some complications. First, adding these additional nodes automatically drops the connectivity of the new network down to 1 (since each new node has only one incoming edge), which violates the connectivity requirements presented in Sundaram and Hadjicostis (2011), and prevents those results from being directly applied to the new network. One could then argue that the new nodes do not contribute any initial values of their own, and can thus be disregarded in the analysis. However, the graph-theoretic tools used in Sundaram and Hadjicostis (2011) would first have to be significantly extended to handle such situations. A second complication is that each new node v_{ij} imposes an additional 'delay' of one time-step in the transfer of information between v_i and v_j if modeled in the same way as all other nodes in the network.

To avoid these difficulties, we will work with unreliable links directly; we will show that, after some appropriate manipulations, some of the key tools developed for the malicious node case in Sundaram and Hadjicostis (2011) can be adapted to identify unreliable links as well. After we develop the model further, we will discuss some of the additional technical complications that arise in the translation. We will also show that the class of unreliable links that can be overcome in a graph of $2f + 1$ connectivity is richer than the class of malicious nodes that can be overcome; we will introduce the notion of an *f-wise safe* set to capture this richer class.

3.1.1. Nodes cannot distinguish remote link failures

There are certain link failures that *cannot* be distinguished by each node using only the linear iteration values it receives. Specifically, consider a node v_i with incoming links from nodes v_j and v_n. From (2), we see that the value of $x_i[k + 1]$ would be the same when (i) (v_n, v_i) is malicious and introduces error $z_{ni}[k]$, or (ii) (v_j, v_i) is malicious and introduces error $\frac{w_{in}}{w_{ij}}z_{ni}[k]$. Note that *only* node v_i experiences any difference between case (i) and case (ii). Consider some other node v_r: since node v_i only transmits its current value at each time-step, and since the above discussion shows that malicious errors introduced by one link will be indistinguishable from malicious errors caused by a different link, it is impossible for node v_r to identify which link was malicious based purely on the values that it receives from the linear iteration.

For all intents and purposes, the above discussion reveals that, from the perspective of any given node, all errors entering into any other node can be aggregated into a single injected error. To make this more formal, let $\mathcal{Q} \subset \mathcal{E}$ be any set of links in the network. Define the sets $\mathcal{L}_i(\mathcal{Q}) \subset \mathcal{E}$ and $\chi_i(\mathcal{Q}) \subset \mathcal{X}$:

$$\begin{aligned} \mathcal{L}_i(\mathcal{Q}) &= \{(u, v) \in \mathcal{Q} \mid v = v_i\}, \\ \chi_i(\mathcal{Q}) &= \{v \in \mathcal{X} \mid v \neq v_i \wedge \exists u : (u, v) \in \mathcal{Q}\}. \end{aligned} \tag{3}$$

The set $\mathcal{L}_i(\mathcal{Q})$ contains all links in $\mathcal{Q}$ that end at node v_i, and the set $\chi_i(\mathcal{Q})$ contains all nodes except v_i that have incoming links in $\mathcal{Q}$. For notational brevity we will omit the $(\mathcal{Q})$ parameter whenever it can be understood from the context. We will also find it convenient to work with the following definition.

Definition 3. For any positive integer f, a set of links $\mathcal{Q} \subset \mathcal{E}$ is *f-wise safe* if, for all $v_i \in \mathcal{X}$, the inequality $|\chi_i(\mathcal{Q})| + |\mathcal{L}_i(\mathcal{Q})| \leq f$ holds.

The definition is motivated by the observation that, for a set $\mathcal{F}$ of unreliable links, and a node v_i, all unreliable links entering into any node in $\chi_i(\mathcal{F})$ will be indistinguishable based on the information that node v_i receives.

3.2. Linear system model for linear iterative strategies with unreliable links

At each time-step of the linear iteration, node v_i has access to its own value, as well as the values of its neighbors, possibly corrupted by the unreliable links $\mathcal{F}$. Let us denote $j_0 \triangleq i$, and let $j_1, \ldots, j_{|\mathcal{N}_i|}$ be some enumeration of $\mathcal{N}_i$. Let $\mathbf{z}_i[k]$ denote the $|\mathcal{L}_i(\mathcal{F})|$ column vector of errors injected by links in $\mathcal{L}_i(\mathcal{F})$ at time-step k, and let $\mathbf{x}[k] = [x_1[k], x_2[k], \ldots, x_N[k]]'$ denote the vector of values at all nodes. The values seen by node v_i can then be represented by the $\deg(v_i) + 1$ vector $\mathbf{y}_i[k]$ given by

$$\mathbf{y}_i[k] = \mathbf{C}_i\mathbf{x}[k] + \mathbf{E}_{\mathcal{L}_i(\mathcal{F})}\mathbf{z}_i[k], \tag{4}$$

where $\mathbf{C}_i$ is a $(\deg(v_i)+1) \times N$ binary matrix with a single '1' in each row denoting the entries of the state-vector $\mathbf{x}[k]$ that are available to node v_i. The matrix $\mathbf{E}_{\mathcal{L}_i}$ (omitting the $\mathcal{F}$ from $\mathcal{L}_i(\mathcal{F})$ for brevity) is a $(\deg(v_i)+1) \times |\mathcal{L}_i|$ binary matrix. A '1' in row j and column m of $\mathbf{E}_{\mathcal{L}_i}$ indicates that element j of $\mathbf{y}_i[k]$ is affected by element m of $\mathbf{z}_i[k]$,

i.e., the mth unreliable link in $\mathcal{L}_i(\mathcal{F})$ affects the value received by v_i from its jth neighbor.

Let $\mathbf{w}_i$ denote the $1 \times (\deg(v_i)+1)$ row vector containing all of the weights that node v_i uses to multiply the values of its neighbors (and itself). With these weights, the value of node v_i at time-step $k+1$ is given by

$$x_i[k+1] = \mathbf{w}_i\mathbf{y}_i[k] = \mathbf{w}_i\mathbf{C}_i\mathbf{x}[k] + \mathbf{w}_i\mathbf{E}_{\mathcal{L}_i}\mathbf{z}_i[k].$$

The lth element of the row vector $\mathbf{w}_i\mathbf{C}_i$ is zero if node v_l is not a neighbor of node v_i, and w_{il} otherwise. The vector $\mathbf{w}_i\mathbf{E}_{\mathcal{L}_i}$ selects those weights in $\mathbf{w}_i$ that multiply the values received on unreliable links. Defining $\mathbf{g}_i \triangleq \mathbf{w}_i\mathbf{E}_{\mathcal{L}_i}$, we write the evolution of the values of all nodes in the system as

$$\mathbf{x}[k+1] = \mathbf{W}\mathbf{x}[k] + \mathbf{G}(\mathcal{F})\mathbf{z}[k]. \tag{5}$$

The vector $\mathbf{z}[k]$ is the stacked vector $[\,\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\,]'$ of all link errors; its dimension is $|\mathcal{F}| = \sum_{j=1}^{N} |\mathcal{L}_j|$. The matrix $\mathbf{G}(\mathcal{F}) \triangleq diag(\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_N)$ is block structured, of size $N \times |\mathcal{F}|$, with the $1 \times |\mathcal{L}_i|$ row vectors $\mathbf{g}_i$ placed diagonally and zeros elsewhere. The entries of $\mathbf{G}(\mathcal{F})$ map a vector of link errors to the state changes it induces in the nodes that receive these link errors. Entry w_{ij} of matrix $\mathbf{W}$ satisfies $w_{ij} = 0$ whenever $v_j \notin \mathcal{N}_i \cup \{v_i\}$.

We will also find it convenient to work with a modified form of $\mathbf{Gz}[k]$ that captures a node-centric view from some given node v_i. For convenience, we will take $i = 1$, but the same analysis holds for any other node. Letting $\chi_1(\mathcal{F}) = \{v_{j_1}, v_{j_2}, \ldots, v_{j_r}\}$, (5) can be written as

$$\mathbf{x}[k+1] = \mathbf{W}\mathbf{x}[k] + \underbrace{\left[\mathbf{B}_{\mathcal{L}_1} \mid \mathbf{B}_{\chi_1}\right]}_{\triangleq \mathbf{B}_1(\mathcal{F})} \mathbf{u}_1(\mathcal{F})[k], \tag{6}$$

$$\mathbf{B}_{\mathcal{L}_1} \triangleq \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{0}_{(N-1)\times|\mathcal{L}_1|} \end{bmatrix}, \qquad \mathbf{B}_{\chi_1} \triangleq \begin{bmatrix} \mathbf{e}_{j_1} & \cdots & \mathbf{e}_{j_r} \end{bmatrix}, \tag{7}$$

$$\mathbf{u}_1(\mathcal{F})[k] \triangleq \left[\mathbf{z}_1'[k], \mathbf{g}_{j_1}\mathbf{z}_{j_1}[k], \ldots, \mathbf{g}_{j_r}\mathbf{z}_{j_r}[k]\right]'. \tag{8}$$

The vector $\mathbf{u}_1(\mathcal{F})[k]$ consists of the link errors on $\mathcal{L}_1(\mathcal{F})$ and the aggregate errors at all vertices with one or more incoming unreliable links. The matrix $\mathbf{B}_1$ takes the error values in $\mathbf{u}_1$ and applies them to the appropriate elements of $\mathbf{x}[k]$. The link errors $\mathbf{z}_1[k]$ of $\mathcal{L}_1$ are multiplied by $\mathbf{g}_1$, and the vertex errors $\mathbf{g}_{j_i}\mathbf{z}_{j_i}[k]$ are added to the appropriate element of $\mathbf{x}[k]$ via the j_ith column of the identity matrix—the vector $\mathbf{e}_{j_i}$. The form of the vector $\mathbf{u}_1[k]$ captures the fact that node v_1 only sees only the aggregate effect of errors entering into other nodes, but is directly affected by the errors entering into itself (via the values that it receives from its neighbors). We will later show how each node can determine the identities of the unreliable links entering into other nodes (i.e., eliminate the uncertainty due to the aggregation) by applying a slight variant of the linear iterative strategy.

Consider again (4), which models the values seen by node v_i at each time-step. We may rewrite this equation with respect to $\mathbf{u}_1[k]$ as

$$\mathbf{y}_1[k] = \mathbf{C}_1\mathbf{x}[k] + \mathbf{D}_1(\mathcal{F})\mathbf{u}_1[k], \tag{9}$$
$$\mathbf{D}_1(\mathcal{F}) \triangleq \left[\mathbf{E}_{\mathcal{L}_1(\mathcal{F})} \mid \mathbf{0}_{(\deg(v_1)+1)\times(|\mathcal{F}|-|\mathcal{L}_1(\mathcal{F})|)}\right].$$

Eqs. (6) and (9) together define a linear system that captures the linear iterative strategy with unreliable links $\mathcal{F}$. We will also find it helpful to discuss systems of this form for arbitrary subsets of links $\mathcal{Q} \subseteq \mathcal{E}$. We extend the notation $\mathbf{B}_1(\mathcal{Q})$, $\mathbf{D}_1(\mathcal{Q})$ to indicate matrices analogous to $\mathbf{B}_1(\mathcal{F})$ and $\mathbf{D}_1(\mathcal{F})$ of (6) and (9).

Remark 2. Modeling linear iterative strategies with malicious nodes as linear systems was also the starting point of the analysis in Sundaram and Hadjicostis (2011). A branch of control theory known as structured system theory (Dion, Commault, & van der Woude, 2003) was then applied to analyze the resulting system and relate the topology of the network to system properties. Structured system theory deals with linear systems in which each entry of the system matrices is either fixed at zero or taken to be an independent free parameter. Unfortunately, the linear system given by (6) and (9) does not satisfy these conditions, since each nonzero entry in the matrices $\mathbf{B}_{\chi_1}$, $\mathbf{C}_1$ and $\mathbf{D}_1$ is set to 1 (and thus are not free parameters), and each nonzero entry in $\mathbf{B}_{\mathcal{L}_1}$ is the same as some nonzero entry in $\mathbf{W}$. Thus, standard results from structured system theory cannot be directly applied to analyze this system, as was done in Sundaram and Hadjicostis (2011)—we will have to first manipulate the given system into a form that is amenable to analysis.

3.3. Data dissemination despite unreliable links

Hereafter, we will use the notation $\Theta = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ to denote the discrete time linear system given by

$$\mathbf{x}[k+1] = \mathbf{A}\,\mathbf{x}[k] + \mathbf{B}\,\mathbf{u}[k]$$
$$\mathbf{y}[k] = \mathbf{C}\,\mathbf{x}[k] + \mathbf{D}\,\mathbf{u}[k],$$

with *state* $\mathbf{x} \in \mathbb{R}^N$, *input* $\mathbf{u} \in \mathbb{R}^m$, and *output* $\mathbf{y} \in \mathbb{R}^p$. The following property of linear systems will prove to be key to our analysis.

Definition 4 (*Hautus, 1983, Rappaport & Silverman, 1971 and Silverman, 1976*)**.** The linear system $\Theta = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ is said to be *strongly observable* if there exists a positive integer $T \leq \dim \mathbf{x}$ such that the outputs $\mathbf{y}[0], \mathbf{y}[1], \ldots, \mathbf{y}[T-1]$ are sufficient to recover the initial state $\mathbf{x}[0]$, regardless of the unknown inputs $\mathbf{u}[0], \mathbf{u}[1], \ldots, \mathbf{u}[T-1]$.

Theorem 2. *Suppose that there exists a weight matrix $\mathbf{W}$ such that, for all possible $2f$-wise safe sets $\mathcal{Q} \subseteq \mathcal{E}$, the system $\Theta(\mathcal{Q}) = (\mathbf{W}, \mathbf{B}_1(\mathcal{Q}), \mathbf{C}_1, \mathbf{D}_1(\mathcal{Q}))$ is strongly observable. Then, there exists an integer $T \leq N$ such that, if the nodes run the linear iteration for T time-steps with the weight matrix $\mathbf{W}$, node v_1 can correctly obtain the initial values $\mathbf{x}[0]$, despite the actions of any f-wise safe set $\mathcal{F}$ of unreliable links.*

Proof. The proof of this result is very similar to the proof of robustness in the malicious *node* case presented in Pasqualetti et al. (2012) and Sundaram and Hadjicostis (2011). Specifically, note that any $2f$-wise safe set $\mathcal{Q}$ essentially defines a system $\Theta(\mathcal{Q})$ with $2f$ inputs. Similarly, for an f-wise safe set $\mathcal{F}$, $\Theta(\mathcal{F})$ has f inputs. Consider two (possibly different) f-wise safe sets $\mathcal{F}_1$ and $\mathcal{F}_2$, with corresponding systems $\Theta(\mathcal{F}_1)$ and $\Theta(\mathcal{F}_2)$ having different initial states $\mathbf{x}^1[0]$ and $\mathbf{x}^2[0]$, respectively. By linearity, the difference in the outputs of the two systems over any T time-steps can be generated by another system $\Theta(\mathcal{F}_1 \cup \mathcal{F}_2)$, with initial state $\mathbf{x}^1[0] - \mathbf{x}^2[0]$. Noting that the set $\mathcal{F}_1 \cup \mathcal{F}_2$ is $2f$-wise safe, we see that the outputs generated by the two systems with different initial states must be different; otherwise, the system $\Theta(\mathcal{F}_1 \cup \mathcal{F}_2)$, with initial state $\mathbf{x}^1[0] - \mathbf{x}^2[0]$, produces the all-zero output with a nonzero initial state, which contradicts the assumption of strong observability. Thus, for any f-wise safe set $\mathcal{F}$, the output of system (6) and (9) uniquely specifies the initial state $\mathbf{x}[0]$, despite the actions of the unreliable links in $\mathcal{F}$. □

A specific procedure was provided in Sundaram and Hadjicostis (2011) to recover all of the initial values (in finite time) despite malicious nodes, and that can be adapted in a straightforward manner to the case in which the links are unreliable.[3]

[3] The procedure described in Sundaram and Hadjicostis (2011) assumes that all nodes know the weight matrix $\mathbf{W}$; this assumption of full knowledge of the network topology is standard in the literature on Byzantine fault-tolerant systems, due to the strong adversarial model. There have been some investigations of fault-tolerant consensus and broadcast algorithms that are agnostic of the network topology, but, as expected, such algorithms require more stringent conditions on the network topology than simply having a connectivity of $2f+1$ (e.g., see Koo, 2004, LeBlanc, Zhang, Sundaram, & Koutsoukos, 2012, and Zhang & Sundaram, 2012).

3.4. Strong observability and invariant zeros

Theorem 2 requires that the system $\Theta(\mathcal{Q})$ be strongly observable for any $2f$-wise safe set $\mathcal{Q}$; here, we will show that, when $\kappa \geq 2f+1$, one can find a weight matrix $\mathbf{W}$ such that this condition is satisfied. First, we recall some classical definitions and results.

Definition 5. The matrix $\mathbf{P}(z) = \begin{bmatrix} \mathbf{W} - z\mathbf{I}_N & \mathbf{B}_1(\mathcal{Q}) \\ \mathbf{C}_1 & \mathbf{D}_1(\mathcal{Q}) \end{bmatrix}$ is called the *matrix pencil* of $\Theta(\mathcal{Q})$.

Definition 6 (*Schrader & Sain, 1989*)**.** The complex number $z_0 \in \mathbb{C}$ is an *invariant zero* of Θ if $\text{rank}\,(\mathbf{P}(z_0)) < \max_{z\in\mathbb{C}} \text{rank}(\mathbf{P}(z))$.

Theorem 3 (*Hautus, 1983, Rappaport & Silverman, 1971 and Silverman, 1976*)**.** *The system Θ is strongly observable if and only if it has no invariant zeros.*

To prove robustness of the linear iterative strategy to any f-wise safe set of unreliable links, we see from Theorems 2 and 3 that we should choose $\mathbf{W}$ such that the system $\Theta(\mathcal{Q})$ has no invariant zeros for all possible $2f$-wise safe sets $\mathcal{Q}$. To this end, for any set $\mathcal{Q}$ of links, assume (without loss of generality) that the output values available to node v_1 (given by (9)) are arranged such that the outputs corresponding to links in $\mathcal{L}_1(\mathcal{Q})$ are at the bottom of $\mathbf{y}[k]$. Define $\mathcal{L}_1^c \triangleq \mathcal{E} \setminus \mathcal{L}_1$. This means that the $\mathbf{C}_1(\mathcal{Q})$ and $\mathbf{D}_1(\mathcal{Q})$ matrices in (9) can be written as

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{H}_1^c \\ \mathbf{H}_1 \end{bmatrix}, \qquad \mathbf{D}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{|\mathcal{L}_1|} & \mathbf{0} \end{bmatrix}. \tag{10}$$

The matrix $\mathbf{H}_1^c$ corresponds to all neighbors of node v_1 (including itself) that are not involved in links in $\mathcal{L}_1$.

Next, define the matrix $\mathbf{W}_{\mathcal{L}_1^c}$, which is obtained from $\mathbf{W}$ by setting the entries corresponding to links in $\mathcal{L}_1$ to zero (i.e., this zeroes out $|\mathcal{L}_1|$ entries in the first row of $\mathbf{W}$). Thus, $\mathbf{W}_{\mathcal{L}_1^c}$ is the weight matrix for a graph $\mathcal{G}_{\mathcal{L}_1^c} = \{\mathcal{X}, \mathcal{L}_1^c\}$.

Lemma 1. *For any set $\mathcal{Q}$ of links, the invariant zeros of $\Theta(\mathcal{Q}) = (\mathbf{W}, \mathbf{B}_1, \mathbf{C}_1, \mathbf{D}_1)$ are exactly the invariant zeros of $\Theta_1(\mathcal{Q}) = (\mathbf{W}_{\mathcal{L}_1^c}, \mathbf{B}_{\chi_1}, \mathbf{H}_1^c, \mathbf{0})$ (where $\mathbf{B}_{\chi_1}$ is defined in (7)).*

Proof. Using (6) and (10), the matrix pencil for the set $\Theta(\mathcal{Q})$ is given by

$$\mathbf{P}(z) = \begin{bmatrix} \mathbf{W} - z\mathbf{I}_N & \mathbf{B}_1 \\ \mathbf{C}_1 & \mathbf{D}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{W} - z\mathbf{I}_N & \mathbf{B}_{\mathcal{L}_1} & \mathbf{B}_{\chi_1} \\ \mathbf{H}_1^c & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_1 & \mathbf{I}_{|\mathcal{L}_1|} & \mathbf{0} \end{bmatrix}.$$

Now, note that $\mathbf{W} - \mathbf{B}_{\mathcal{L}_1}\mathbf{H}_1 = \mathbf{W} - \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{H}_1 = \mathbf{W}_{\mathcal{L}_1^c}$, because each row of the matrix $\mathbf{H}_1$ has a single 1, and these are located in the columns corresponding to neighbors of v_1 that are involved in links in $\mathcal{L}_1$. Furthermore, the vector $\mathbf{g}_1$ contains all of the weights associated with those links. Thus, subtracting $\mathbf{B}_{\mathcal{L}_1}$ times the last block-row of $\mathbf{P}(z)$ from the first block-row, we obtain

$$\begin{aligned} \text{rank}\,\mathbf{P}(z) &= \text{rank} \begin{bmatrix} \mathbf{W}_{\mathcal{L}_1^c} - z\mathbf{I}_N & \mathbf{0} & \mathbf{B}_{\chi_1} \\ \mathbf{H}_1^c & \mathbf{0} & \mathbf{0} \\ \mathbf{H}_1 & \mathbf{I}_{|\mathcal{L}_1|} & \mathbf{0} \end{bmatrix} \\ &= |\mathcal{L}_1| + \text{rank} \begin{bmatrix} \mathbf{W}_{\mathcal{L}_1^c} - z\mathbf{I}_N & \mathbf{B}_{\chi_1} \\ \mathbf{H}_1^c & \mathbf{0} \end{bmatrix}. \end{aligned}$$

The values of z for which this last matrix loses rank are exactly the invariant zeros of $\Theta_1(\mathcal{Q})$. □

It is instructive to consider a graph-theoretic interpretation of the set $(\mathbf{W}_{\mathcal{L}_1^c(\mathcal{Q})}, \mathbf{B}_{\chi_1(\mathcal{Q})}, \mathbf{H}_1^c(\mathcal{Q}), \mathbf{0})$; this set of matrices corresponds to graph $\mathcal{G}_{\mathcal{L}_1^c(\mathcal{Q})}$ (i.e., the graph $\mathcal{G}$ with edges in $\mathcal{L}_1(\mathcal{Q})$ removed), and with inputs affecting only the nodes in $\chi_1(\mathcal{Q})$. This is exactly the model considered in Sundaram and Hadjicostis (2011) (which studied the issue of linear iterative strategies with malicious nodes), and the following result was proved in Lemmas 2 and 3 of that paper.[4]

Lemma 2. *Let the graph of network $\mathcal{G}$ have connectivity κ, and let $\mathcal{S} = \{v_{j_1}, v_{j_2}, \ldots, v_{j_{|\mathcal{S}|}}\}$ be any set of nodes. Let $\mathbf{B}_{\mathcal{S}} = \begin{bmatrix} \mathbf{e}_{j_1} & \mathbf{e}_{j_2} & \cdots & \mathbf{e}_{j_{|\mathcal{S}|}} \end{bmatrix}$, and let $\mathbf{C}_i$ be the $(\deg(v_i)+1) \times 1$ matrix with a single 1 in each row indicating the neighbors of node v_i (and v_i itself). If $\kappa \geq |\mathcal{S}| + 1$, then, for almost any choice of weights, the set $(\mathbf{W}, \mathbf{B}_{\mathcal{S}}, \mathbf{C}_i, \mathbf{0})$ will have no invariant zeros.*

The term *almost any* in the above lemma means that the set of weights for which the result does not hold lies on an algebraic variety, and thus has measure zero in the space of all possible weights. Using the above lemma, we obtain the following result.

Lemma 3. *Let the graph of network $\mathcal{G}$ have connectivity κ, and let $\mathcal{Q}$ be any set of links. If $\kappa \geq |\mathcal{L}_1(\mathcal{Q})| + |\chi_1(\mathcal{Q})| + 1$, then, for almost any choice of weights, the system $\Theta(\mathcal{Q})$ will have no invariant zeros.*

Proof. From Lemma 1, the invariant zeros of $\Theta(\mathcal{Q})$ are the invariant zeros of $\Theta_1(\mathcal{Q})$. Recall that the set $(\mathbf{W}_{\mathcal{L}_1^c(\mathcal{Q})}, \mathbf{B}_{\chi_1(\mathcal{Q})}, \mathbf{H}_1^c(\mathcal{Q}), \mathbf{0})$ corresponds to a graph $\mathcal{G}_{\mathcal{L}_1^c(\mathcal{Q})}$, which was obtained by removing $|\mathcal{L}_1(\mathcal{Q})|$ links from $\mathcal{G}$. Let the connectivity of $\mathcal{G}_{\mathcal{L}_1^c(\mathcal{Q})}$ be denoted by κ^c; since removing an edge decreases the connectivity by at most 1 (e.g., see West, 2001), we have

$$\begin{aligned} \kappa^c &\geq \kappa - |\mathcal{L}_1(\mathcal{Q})| \geq |\mathcal{L}_1(\mathcal{Q})| + |\chi_1(\mathcal{Q})| + 1 - |\mathcal{L}_1(\mathcal{Q})| \\ &= |\chi_1(\mathcal{Q})| + 1. \end{aligned}$$

Thus, system $\Theta_1(\mathcal{Q})$ satisfies all of the conditions in Lemma 2 (with $\mathcal{G}_{\mathcal{L}_1^c(\mathcal{Q})} \to \mathcal{G}$, $\chi_1(\mathcal{Q}) \to \mathcal{S}$, $\mathbf{H}_1^c(\mathcal{Q}) \to \mathbf{C}_1$), which proves the lemma. □

We now come to the following theorem.

Theorem 4. *Let the graph of the given network $\mathcal{G}$ have connectivity κ. If $\kappa \geq 2f+1$, then there exists a positive integer $T \leq N$ such that, for almost any choice of weights, every node in the network can correctly determine $\mathbf{x}[0]$ after running the linear iteration for at most T time-steps, despite any errors introduced by any f-wise safe set of unreliable links.*

Proof. From Lemma 3, we see that, for almost any choice of weight matrix $\mathbf{W}$, the set $\Theta(\mathcal{Q}) = (\mathbf{W}, \mathbf{B}_i(\mathcal{Q}), \mathbf{C}_i, \mathbf{D}_i(\mathcal{Q}))$ will have no invariant zeros, for any particular $2f$-wise safe set $\mathcal{Q}$. The set of weights for which this property does not hold has measure zero, and thus it will hold generically and simultaneously for all possible $2f$-wise safe sets $\mathcal{Q}$. From Theorems 2 and 3, node v_1 can correctly obtain all of the initial values after running the linear iteration for at most N time-steps, despite the actions of any f-wise safe set $\mathcal{F}$ of unreliable links. The same analysis holds simultaneously for every node from the generic nature of the property. Thus every node can obtain all of the initial values after at most N time-steps of the linear iteration. □

Remark 3. As discussed in Sundaram and Hadjicostis (2011), the connectivity bound $\kappa \geq 2f+1$ is fundamental in order to tolerate up to f Byzantine nodes; in other words, there are no algorithms that can overcome f carefully chosen Byzantine

[4] While the exact statement provided in Sundaram and Hadjicostis (2011) focuses on the case $|\mathcal{S}| = 2f$, the proof extends directly to the more general statement provided in Lemma 2.

nodes if the connectivity is $2f$ or less. The result in Sundaram and Hadjicostis (2011) showed that linear iterative strategies are capable of achieving this bound, under the wireless broadcast model of communication. Theorem 4 shows that this is also true for Byzantine links. Specifically, as long as a link error is able to induce arbitrary changes in the incident node value, no algorithm can tolerate a carefully chosen f-wise safe set of Byzantine links if the connectivity is $2f$ or less. On the other hand, the linear strategy is able to overcome any f-wise safe set of Byzantine links if $\kappa \geq 2f+1$.

3.5. Identifying local unreliable links

Each node v_i only sees the *aggregate* effect of links entering into other nodes (see Section 3), and thus the linear iteration does not allow v_i to uniquely identify such links as malicious or even unreliable. However, each node can uniquely identify the unreliable links that are directly incident to it, as shown by the following results.

Corollary 2 (*To* Theorem 4)**.** *Let the graph of network $\mathcal{G}$ have connectivity $\kappa \geq 2f+1$, and let T be the integer specified in Theorem 4. Assume that the set of unreliable links during any T contiguous time-steps is f-wise safe. Then, for almost any choice of weights, every node in the network can correctly determine $\mathbf{x}[k]$, $k \in \mathbb{N}$, after running the linear iteration for at most $T+k$ time-steps.*

Proof. From Theorem 4, note that each node can recover $\mathbf{x}[0]$ after running the linear iteration for at most T time-steps. This also means that each node can recover $\mathbf{x}[k]$ after running the linear iteration for $T+k$ time-steps, simply by viewing $\mathbf{x}[k]$ as the initial values of a linear iteration starting at time-step k. □

Let $\mathcal{F}[k]$ denote the set of all links that are unreliable during the first k time-steps of the linear iteration. Thus, $\mathcal{L}_i(\mathcal{F}[k])$ is the set of incoming links to v_i that are unreliable during the first k time-steps.

Theorem 5. *Let the graph of the given network $\mathcal{G}$ have connectivity $\kappa \geq 2f+1$, and let T be the positive integer specified in Theorem 4. Assume that the set of unreliable links during any T contiguous time-steps is f-wise safe. Then, for almost any choice of weights, each node v_i in the network can determine $\mathcal{L}_i(\mathcal{F}[k])$ after running the linear iteration for at most $T+k$ time-steps, despite any errors injected by the links in $\mathcal{F}[T+k]$.*

Proof. Note from Corollary 2 that each node v_i can determine $\mathbf{x}[k]$ after running the linear iteration for at most $T+k$ time-steps. Rearranging (4) (the model of values seen by node v_i), we obtain $\mathbf{y}_i[k] - \mathbf{C}_i\mathbf{x}[k] = \mathbf{E}_{\mathcal{L}_i}\mathbf{z}_i[k]$. The left-hand side of this equality is known to v_i after $T+k$ time-steps: it is the difference between the actual values of node v_i's neighbors and the values that v_i received. Thus, if component l of $\mathbf{y}_i[k] - \mathbf{C}_i\mathbf{x}[k]$ is nonzero, this indicates a discrepancy in the received and actual values from that neighbor, and v_i can conclude that link l is unreliable (thereby obtaining $\mathcal{L}_i(\mathcal{F}[k])$). □

Remark 4. It is worth noting that each node can also use (6) to calculate $\mathbf{x}[k+1] - \mathbf{W}\mathbf{x}[k] = \mathbf{B}_i(\mathcal{F})\mathbf{u}_i(\mathcal{F})[k]$, and any nonzero element on the left-hand side indicates a node with incident malicious links.

Note that Theorem 5 and Corollary 2 allow the set of unreliable links $\mathcal{F}[k]$ to violate the property of being f-wise safe for $k > T$; as long as the set of unreliable links during any T *contiguous* time-steps is f-wise safe, the initial values and the set of locally unreliable links can be identified.

3.6. Identifying unreliable links incident on other nodes

Given that nodes can communicate reliably using the linear iterative strategy, we now show that each node can inform the other nodes about any incoming unreliable links by sending additional information via the linear strategy.[5]

Consider p phases of the linear iteration ($p \in \mathbb{N}$), each of which consists of T time-steps. In each phase, the linear iterative strategy is used by the network to disseminate a new set of values to all the nodes. However, after the first phase, we allow the nodes to append information about any incoming unreliable links from previous phases to their values. Specifically, at the start of phase $p+1$ (i.e., at time-step $(p+1)T$), each node will have been able to identify all incoming unreliable links up to time-step pT (from Theorem 5). The following result provides a means for each node v_i to encode information about $\mathcal{L}_i(\mathcal{F}[pT])$ in the value it transmits in the T steps of phase $p+1$.

Theorem 6. *Let the graph of the given network $\mathcal{G}$ have connectivity $\kappa \geq 2f+1$, and let T be the positive integer specified in Theorem 4. Assume that the set of unreliable links during any T contiguous time-steps is f-wise safe. At the beginning of phase $p+1$, we assume that all errors have been corrected and that all nodes have identified their incoming unreliable links up to time pT. Let $\mathcal{F}'$ denote the set of unreliable links during phase $p-1$. The additional information providing all nodes with the exact identity of all links $\mathcal{F}'$ can be encoded in $\sum_{v\in\mathcal{X}} \log_2 \binom{\deg(v)}{f}$ bits, and can be made available to all nodes no later than the end of phase $p+1$.*

Proof. All nodes are aware of the structure of graph $\mathcal{G}$ in its entirety, and each node knows the degrees of all other nodes. The number of unreliable links $|\mathcal{L}_i|$ coming into any node during phase $p-1$ (time steps $(p-1)T$ to $pT-1$) is at most f, because the set of unreliable links is f-wise safe. Each node therefore knows *a priori* the value of $\binom{\deg(v)}{f}$ for each node $v \in \mathcal{X}$. Each node with incoming unreliable links can now encode which of this collection of possible sets of unreliable links was actually found via a bit vector of length $\log_2 \binom{\deg(v)}{f}$, and append this vector to its initial value in phase $(p+1)$. This information can then be reliably recovered (along with the initial values) by the other nodes at the end of the phase. □

4. Discussion of other failure models

We now briefly discuss some common fault cases captured by our model.

4.1. Permanent failure

One special case that is often used in theoretical analyses (Koo, 2004) is that of permanent failures—the set $\mathcal{F}$ of nodes or links under control of the adversary is constant. In this case, our approach provides a means to recover from the interference injected by the adversary after T steps, and to disseminate $\mathcal{F}$ to all nodes after an additional T steps.

[5] If the network has a combination of malicious links and nodes, it will be impossible for any node v_i to differentiate between a malicious node changing its own state, and a malicious link inducing a state change in that node (since that node is the only one that sees the value incoming on that link). In such cases, v_i would have to settle for simply recovering the set of nodes that had their states changed (i.e., as described in Remark 4), without further delving into the cause of the state change (i.e., node induced or link induced).

4.2. Stochastic failure

At the other extreme, the subset of failed links may be stochastically random. If the failure probability of a link at each time-step is q and links fail independently of each other and over time, then the probability of a link having a failure in a phase of T steps is $\bar{q} = 1 - (1 - q)^T$. The expected number of links failing in a phase is $m\bar{q}$, with m the total number of links in the network. The probability of having more than f links fail is a binomially distributed variable which can be bounded (e.g., using Hoeffding's inequality), providing a level of confidence that the number of failures will not exceed a given value in a given interval.

4.3. Spatio-temporal models

Physical networks often have link failures associated with some local disturbances. These include local weather conditions (in large networks), interference caused by people or equipment moving around, encryption key compromise in secured networks, narrow-band noise in frequency multiplexed networks, and synchronization failures in frequency-hopping networks. Our algorithm would preserve system-wide communication integrity as long as the disturbance was local enough to harm only a few links at a time. If the disturbance moves around slowly enough that f or fewer links are adversely affected within a time window of T steps, our algorithm would correct the errors—even if the total number of failed links over the system lifetime is arbitrarily high.

4.4. Subversive attack

Consider networks with links that can be subverted by an adversary – for example by installing malware on a poorly protected machine and intercepting link traffic—and at some later time be subjected to a process that reclaims them to function correctly again – for example by running a virus scanning and removal tool. As long as link reclamation is rapid enough to ensure that the collection of malicious links remains f-wise safe in every window of T time-steps, our algorithm will correct all errors injected by the adversary.

5. Summary

We have shown that, in networks that use local broadcasts and have sufficient connectivity, a linear iterative strategy can be used to counteract and identify link failures. We proved resilience to a strong adversary: malfunctioning links may be malicious and collude, potentially using information about the network that the nodes themselves do not possess—yet the network will prevail. The phase-by-phase approach taken by our algorithm allows the failure model to be extended to cases of stochastic link failures and spatio-temporally correlated failures. Our approach extends work in resilient distributed function calculation to wireless networks, showing that, with little communication overhead, networks can resist both link failures and active attacks.

References

Abdelzaher, T., He, T., & Stankovic, J. (2004). Feedback control of data aggregation in sensor networks. In *Proceedings of 43rd IEEE conference on decision and control* (pp. 1490–1495).

Bailey, M., Cooke, E., Jahanian, F., Xu, Y., & Karir, M. (2009). A survey of botnet technology and defenses. In *Conference for homeland security, cybersecurity applications & technology* (pp. 299–304).

Deb, S., Médard, M., & Choute, C. (2006). Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE Transactions on Information Theory*, *52*(6), 2486–2507.

Dion, J.-M., Commault, C., & van der Woude, J. (2003). Generic properties and control of linear structured systems: a survey. *Automatica*, *39*(7), 1125–1144.

Giridhar, A., & Kumar, P. R. (2005). Computing and communicating functions over sensor networks. *IEEE Journal on Selected Areas in Communications*, *23*(4), 755–764.

Hautus, M. L. J. (1983). Strong detectability and observers. *Linear Algebra and its Applications*, *50*, 353–368.

Hromkovic, J., Klasing, R., Pelc, A., Ruzicka, P., & Unger, W. (2005). *Dissemination of information in communication networks*. Springer-Verlag.

Jaggi, S., Langberg, M., Katti, S., Ho, T., Katabi, D., Médard, M., & Effros, M. (2008). Resilient network coding in the presence of Byzantine adversaries. *IEEE Transactions on Information Theory*, *54*(6), 2596–2603.

Koetter, R., & Kschischang, F. R. (2008). Coding for errors and erasures in random network coding. *IEEE Transactions on Information Theory*, *54*(8), 3579–3591.

Koetter, R., & Médard, M. (2003). An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, *11*(5), 782–795.

Koo, C.-Y. (2004). Broadcast in radio networks tolerating Byzantine adversarial behavior. In *Proceedings of the 23rd ACM symposium on principles of distributed computing* (pp. 275–282).

LeBlanc, H.J., Zhang, H., Sundaram, S., & Koutsoukos, X. (2012). Consensus of multi-agent networks in the presence of adversaries using only local information. In *Proceedings of 1st conference on high confidence networked systems* (pp. 1–10).

Lynch, N. A. (1996). *Distributed algorithms*. Morgan Kaufmann Publishers, Inc.

Mosk-Aoyama, D., & Shah, D. (2006). Information dissemination via network coding. In *Proceedings of the 2006 IEEE international symposium on information theory* (pp. 1748–1752).

Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, *95*(1), 215–233.

Pasqualetti, F., Bicchi, A., & Bullo, F. (2012). Consensus computation in unreliable networks: a system theoretic approach. *IEEE Transactions on Automatic Control*, *57*(1), 90–104.

Rappaport, D., & Silverman, L. M. (1971). Structure and stability of discrete-time optimal systems. *IEEE Transactions on Automatic Control*, *16*(3), 227–233.

Ren, W., Beard, R.W., & Atkins, E.M. (2005). A survey of consensus problems in multi-agent coordination. In *Proceedings of the American control conference* (pp. 1859–1864).

Schrader, C. B., & Sain, M. K. (1989). Research on system zeros: a survey. *International Journal of Control*, *50*(4), 1407–1433.

Silverman, L. M. (1976). Discrete Riccati equations: alternative algorithms, asymptotic properties and system theory interpretations. In C. T. Leondes (Ed.), *Control and dynamic systems, Vol. 12* (pp. 313–386). Academic Press.

Sundaram, S., & Hadjicostis, C. N. (2008). Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, *26*(4), 650–660.

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

Teixeira, A., Sandberg, H., & Johansson, K.H. (2010). Networked control systems under cyber attacks with applications to power networks. In *Proceedings of the American control conference* (pp. 3690–3696).

Tsitsiklis, J. N., Bertsekas, D. P., & Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, *31*, 803–812.

West, D. B. (2001). *Introduction to graph theory*. Upper Saddle River, New Jersey: Prentice-Hall Inc.

Yeung, R. W., Li, S.-Y. R., Cai, N., & Zhang, Z. (2005). Network coding theory. In *Foundations and trends in communications and information theory, Vol. 2* (pp. 241–381). Now Publishers Inc.

Zhang, H., & Sundaram, S. (2012). Robustness of information diffusion algorithms to locally bounded adversaries. In *Proceedings of the American control conference* (pp. 5855–5861).

Shreyas Sundaram is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Waterloo. He received his M.S. and Ph.D. degrees in Electrical Engineering from the University of Illinois at Urbana-Champaign in 2005 and 2009, respectively. He was a Postdoctoral Researcher in the GRASP Laboratory at the University of Pennsylvania from 2009 to 2010. His research interests include the analysis of dynamics on networks, secure and fault-tolerant control, network science, distributed control, and the application of graph theory to systems analysis. He received the M.E. Van Valkenburg Graduate Research Award and the Robert T. Chien Memorial Award from the University of Illinois, both for excellence in research. He was a finalist for the Best Student Paper Award at the 2007 and 2008 American Control Conferences.

Shai Revzen received his B.Sc. in Computer Science and Math, with a minor in Physics from the Amirim honours program of the Hebrew University in Jerusalem in 1993, and his M.Sc. in Computer Science from the Hebrew University in 2002. In 2009 he received his Ph.D. in Integrative Biology from the University of California at Berkeley. He worked for Harmonic Inc. (HLIT) as Chief Architect in the Convergent Systems division, focusing on satellite communications and MPEG encoding. He also co-founded Bio-Signal Analysis, a biomedical start-up company. Since 2009, Shai has been a Postdoctoral Research Associate at the University of Pennsylvania, working with Professors Daniel E. Koditschek (Electrical and Systems Engineering), Mark Yim (Mechanical Engineering), and George Pappas (Computer Science). He is currently a Visiting Assistant Professor in the EECS department of the University of Michigan. Shai's research interests include biologically inspired robotics and control, animal locomotion, and dynamical systems.

George Pappas received his Ph.D. degree in Electrical Engineering and Computer Sciences from the University of California, Berkeley (where he received the Eliahu Jury Award for Excellence in Systems Research), in 1998. He is currently the Joseph Moore Professor of Electrical and Systems Engineering at the University of Pennsylvania, Philadelphia. He is a member of the General Robotics, Automation, Sensing and Perception (GRASP) Laboratory and serves as the Deputy Dean for Research in the School of Engineering and Applied Science. His current research interests include hybrid and embedded systems, hierarchical control systems, distributed control systems, and nonlinear control systems, with applications to robotics, unmanned aerial vehicles, biomolecular networks, and green buildings. Dr. Pappas has received numerous awards, including the National Science Foundation (NSF) CAREER Award in 2002, the NSF Presidential Early Career Award for Scientists and Engineers in 2002, the 2009 George S. Axelby Outstanding Paper Award, and the 2010 Antonio Ruberti Outstanding Young Researcher Prize.