Finite-time Average Consensus in a Byzantine Environment Using Set-Valued Observers

Daniel Silvestre, Paulo Rosa, João P. Hespanha, Carlos Silvestre

***Abstract*— This paper addresses the problem of consensus in the presence of Byzantine faults, modeled by an attacker injecting a perturbation in the state of the nodes of a network. It is firstly shown that Set-Valued Observers (SVOs) attain finite-time consensus, even in the case where the state estimates are not shared between nodes, at the expenses of requiring large horizons, thus rendering the computation problem intractable in the general case. A novel algorithm is therefore proposed that achieves finite-time consensus, even if the aforementioned requirement is dropped, by intersecting the set-valued state estimates of neighboring nodes, making it suitable for practical applications and enabling nodes to determine a stopping time. This is in contrast with the standard iterative solutions found in the literature, for which the algorithms typically converge asymptotically and without any guarantees regarding the maximum error of the final consensus value, under faulty environments. The algorithm suggested is evaluated in simulation, illustrating, in particular, the finite-time consensus property.**

I. Introduction

The problem of consensus in a Byzantine environment relates to a set of agents agreeing on a common value, using a distributed algorithm, and considering that an unknown number of those nodes can be malfunctioning or attacked by an intruder. We are interested in randomized gossip average consensus in which nodes are allowed to send messages to a random neighbor in order to compute the average of their initial state. By nature, such algorithms are designed to cope with "crash type" faults by using redundancy and randomization. However, Byzantine faults, such as an intruder in the system, can prevent convergence or drive the steady state of the system to any value [1].

The consensus problem has been widely studied when considering rather non-antagonistic failure models which include packet drops and nodes crashing but, to enable a more comprehensive model, Byzantine faults must be considered. The work [1] considers the problem of detecting and correcting the state of the system in the presence of a Byzantine fault. The case of malicious agents and faulty agents is studied and the authors provide, in both cases, bounds on the number of corrupted nodes to ensure detectability of the fault. In [1], the system dynamics are described by a linear time-invariant model that constrains the communication network to be fixed in all time slots. Here, however, a randomized gossip algorithm is considered, thus dropping the assumption that the same sets of nodes are involved in message exchanges at every time instant.

D. Silvestre, C. Silvestre are with the Dep. of Electrical and Computer Engineering, Instituto Superior Técnico, ISR, 1046-001 Lisboa, Portugal. {dsilvestre,cjs}@isr.ist.utl.pt. This work was supported by the FCT project [PEst-OE/EEI/LA0009/2011]. The work from Daniel Silvestre is partially funded with grant SFRH//BD/71206/2010, from Fundação para a Ciência e a Tecnologia.

P. Rosa is with Deimos Engenharia, Lisbon, Portugal. paulo.rosa@deimos.com.pt.

João P. Hespanha is with the Dept. of Electrical and Computer Eng., University of California, Santa Barbara, CA 93106-9560, USA. J. Hespanha was supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under grants No. W911NF-09-1-0553 and W911NF-09-D-0001. hespanha@ece.ucsb.edu

The problem of finite-time consensus in the presence of malicious agents have been addressed in [2], where the authors show that the topology of the network categorizes its ability to deal with attacks. Both the number of corrupted nodes and vertex-disjoint paths in the network influence its resilience. In [2], it is assumed a broadcast model where, at each transmission time, the nodes send to all their neighbors the same value and the agents objective is to compute some function of the initial state. The main difference is the communication model which we assume to be gossip where random pairs of nodes exchange information.

The choice for representing the set of possible states depends on a mathematical formulation that enables fast and non-conservative intersections and unions of sets, as those are major and normally time-consuming operations when implemented in a computer. One approach is to use the concept of zonotopes, described in [3] and further developed in [4] and [5]. In this article, an alternative approach is adopted, based on the concept of Set-Valued Observers (SVOs) first introduced in [6] and [7] and further information can be found in [8] and [9] and the references therein.

In [10] an SVO-based overlay technique is used to detect Byzantine faults in gossip randomized average consensus algorithms. The main interest is on considering the stochastic information to detect faults and on providing bounds on the attacker signal magnitude. In [10], each node runs an independent SVO to construct a set-valued state estimation. The focus herein is given on, in finite-time, either detecting a fault or returning the average consensus value using estimations obtained from local information and exchanged during communication time. We do not consider the case of compensating for faults.

The main contributions of the present work are as follows:

- Finite-time consensus is shown to be a property of SVOs for a sufficiently large horizon when not communicating estimates, in non-faulty scenarios;
- An algorithm is introduced that intersects neighbor estimates to produce less conservative bounds and reduce the time required to detect faults;
- Finally, it is also shown that this algorithm has the property of achieving finite-time consensus without the need to consider large horizons.

Notation : The transpose of a matrix A is denoted by $A^\intercal$. For vectors a_i, $(a_1, \ldots, a_n) := [a_1^\intercal \ \ldots a_n^\intercal]^\intercal$. We let $1_n := [1 \ldots 1]^\intercal$ and $0_n := [0 \ldots 0]^\intercal$ indicate n-dimensional vector of ones and zeros, respectively, and I_n denotes the identity matrix of dimension n. Dimensions are omitted when clear from context. The vector e_i denotes the canonical vector whose components are equal to zero, except for the ith component. The symbol $\otimes$ denotes the kronecker product. The notation $||.||$ refers to $\|v\| := \sup_i |v_i|$ for a vector v.

II. Problem Statement

We consider a set of n_x agents, each of which with scalar state $x_i(k)$, $1 \leq i \leq n_x$ running a distributed iterative

algorithm that guarantees convergence of the state to its initial average value, i.e.,

$$\lim_{k\to\infty} x_i(k) = x_{\text{av}} := \frac{1}{n_x}\sum_{i=1}^{n_x} x_i(0). \tag{1}$$

We refer to this problem as the *average consensus problem*.

The communication topology is modeled by a graph $G = (V, E)$, where V represents the set of n_x agents, also denoted by nodes, and $E \subseteq V \times V$ is the set of communication links. Node i can send a message to node j, if $(i, j) \in E$. If there exists at least one $i \in V$ such that $(i, i) \in E$ we say that the graph has self-loops, which can model, for example, packet drops, since node i only has access to its own value at that transmission time. We associate to graph G a *weighted adjacency matrix* W with entries:

$$W_{ij} := \begin{cases} w_{ij}, \text{ if } (i,j) \in E \\ 0, \text{ otherwise} \end{cases}, \tag{2}$$

where the weights $w_{ij} \in [0, 1]$ represent link probabilities.

The dynamics of this consensus problem can be described by the following dynamic model:

$$S : \begin{cases} x(k+1) = A(k)x(k) + B(k)u(k) \\ y(k) = C(k)x(k) \end{cases} \tag{3}$$

where matrix $A(k)$ is selected randomly from a set $\{Q_{ij}, (i, j) \in E\}$, where each matrix Q_{ij} is symmetric and *column stochastic* (i.e. $1^\intercal Q_{ij} = 1^\intercal$) to preserve the average and selected with probability $\frac{w_{ij}}{n_x}$ and implementing the changes in the states x_i and x_j. The input $u(k)$ models the Byzantine behavior of nodes updating their state by something other than the "fault-free" averaging rule.

The output of the system $y(k)$, at time k, is composed of the states of the nodes to which the node running the observer communicated. In other words, if node i transmitted to node j at time k, then $y(k)$ will be the vector with the states x_i and x_j ($C(k) = [\mathsf{e}_i, \mathsf{e}_j]^\intercal$) and will only have its own state if the node did not communicate ($C(k) = [\mathsf{e}_i, \mathsf{e}_i]^\intercal$)[1]. With a slight abuse of notation, we use $y(k)$ to refer to the output of the system at time k and $y_k(x(0), u_k)$ to express the same output as a function of the initial state $x(0)$ and input u_k, where u_k denotes the sequence of inputs up to time k.

The focus is on *detectable* faults in the sense that we are interested in faults possible to be distinguished from the normal operation of the algorithm. The intuition behind this definition is that a fault is detectable if there is no possible set of initial conditions such that the sequence $y(0)\cdots y(N)$ can be generated without an attacker signal.

Definition 1: Take the consensus system (3). A nonzero fault signal u_k is said to be *undetectable in N iterations* if:

$$\forall k < N, \exists x(0), x'(0) \in W_o : y_k(x(0), u_k) = y_k(x'(0), 0)$$

where W_o is a set where initial states $x(0)$ and $x'(0)$ are known to belong. Otherwise, it is said to be *detectable*.

The main goal of this paper can therefore be stated as: to develop an algorithm that, in finite-time, either detects nonzero inputs $u(k)$ in (3) corresponding to a detectable fault as in Definition 1, or returns the final consensus value. The detection must not require knowledge of the matrices $B(k)$ (a fault being detectable restricts $B(k)$) and signal $u(k)$ in (3) and, instead, may only use the measurements y_k which stands for all the measurements up to time k taken from the perspective of the node running the observer.

Assumption 1: Each fault considered is defined by a sequence u_k, and is detectable in the sense of Definition 1.

The fault being detectable as in Assumption 1 relates to the observability of the system as in [11].

Assumption 2: $\forall k < N, \|x(k)\| < c$ for a constant c.

Assumption 2 is sustained by the fact that a non-faulty gossip algorithm has a bounded state. Therefore, there exists a constant c such that if the norm of the state is larger than c, one can trivially detect the occurrence of the fault.

III. Proposed Solution

The dynamics of the system can be cast into a Linear Parameter-Varying (LPV) model with uncertainty in the time-varying matrix $A(k)$ by rewriting them, as in [10], as a central matrix and a sum of uncertainties to comply with the framework of the SVOs, resulting in (3) being given by

$$x(k+1) = \Big(A_0 + \sum_{\ell=1}^{n_\Delta} \Delta_\ell(k) A_\ell\Big)x(k) + B(k)u(k) \tag{4}$$

where n_Δ is the number of required uncertainties and each $\Delta_\ell(k)$ is a scalar uncertainty with $|\Delta_\ell(k)| = 1$. The parameters $\Delta_\ell(k)$, as well as the matrices A_ℓ, are related in a straightforward manner to the matrices $U(k)$ in [10].

We use the Set-Valued Observers (SVOs) framework from [12] and [13] and define, at transmission time k,

$$X(k) := \text{Set}(M(k), m(k))$$

where $\text{Set}(M, m) := \{q : Mq + m \le 0\}$ represents a convex polytope and the operator $\le$ is applied component-wise. The aim of an SVO is to find an approximation of the smallest set containing all possible states of the system at time k, $\tilde{X}(k)$, with the knowledge that $\forall 0 \le i < N, x(k-i) \in \tilde{X}(k-i)$ and that the dynamics of the system are as in (4). The initial state $x(0) \in X(0)$ where $X(0) := \text{Set}(M_0, m_0)$ and we can always select M_0 and m_0 due to Assumption 2. For a given uncertainty instantiation $\Delta^\star$, the set $X(k+1) := \text{Set}(M_{\Delta^\star}(k+1), m_{\Delta^\star}(k+1))$, which contains all the possible states of the system at time $k+1$, can be computed as in [14] as the set of points, $\mathbf{x}$, satisfying the equation relating the current time and the previous time with $\mathbf{x}^-$

$$\begin{bmatrix} I & -A_0 - A_{\Delta^\star} \\ -I & A_0 + A_{\Delta^\star} \\ C(k+1) & 0 \\ -C(k+1) & 0 \\ 0 & M(k) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^- \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \\ y(k+1) \\ -y(k+1) \\ -m(k) \end{bmatrix} \tag{5}$$

where

$$A_{\Delta^\star} = \sum_{\ell=1}^{n_\Delta} \Delta_\ell^\star A_\ell$$

and $\mathbf{x}, \mathbf{x}^- \in \mathbb{R}^{n_x}$ and $\Delta_\ell^\star$ is the uncertainty instantiation for the current transmission time. Applying the Fourier-Motzkin elimination method [15], the dependence on $\mathbf{x}^-$ is removed and a set described by $M(k+1)\mathbf{x} \le -m(k+1)$ is obtained. The Fourier-Motzkin elimination method projects the points defined by (5) into the first n_x coordinates.

Let the coordinates of each vertex of the hypercube $H := \{\delta \in \mathbb{R}^{Nn_\Delta} : |\delta| \le 1\}$ be denoted by $\theta_i, i = 1, \cdots, 2^{Nn_\Delta}$. Using (5), we compute $X_{\theta_i}(k)$ with $\Delta^\star = \theta_i$. Thus, the set of all possible states at time $k+1$ can be obtained by

$$X(k+1) = \bigcup_{\theta_i \in H} \text{Set}(M_{\theta_i}(k+1), m_{\theta_i}(k+1))$$

[1] Alternatively, one can consider simply $C(k) = e_i^\intercal$, but this would imply that the size of vector $y(k)$ depends on k.

where we make the union for all the vertices θ_i and where M_{θ_i} and m_{θ_i} are obtained using (5). The convex hull, $\tilde{X}(k+1)$, of set $X(k+1)$ is then obtained by using the methods described in [16], since, in general, the set $X(k+1)$ is non-convex even if $X(k)$ is convex. For additional properties of the set $\tilde{X}(k)$, the interested reader is referred to [10] and references therein.

Notice that using the method provided before to compute $M(k)$ and $m(k)$ for the "fault-free" model gives a set of states that are compatible with the sequence of measurements obtained up to time k. If the intersection of these admissible outputs with the vector of measurements $y(k)$ results in an empty set, then a fault is detected.

A relevant issue regarding the SVOs is their decentralized construction, which is fundamental when dealing with distributed systems. A node only requires the following: signal $y(k)$ that it measures when communicating with others at time k; the matrix $C(k)$ by identifying which nodes it contacts; and its own previous set-valued state estimate. All the matrices $A_{\Delta^\star}$ can be determined if the node has access to the global network structure. Otherwise, all possible links between nodes can be considered, although this is only feasible in networks with a limited number of nodes. However, in a practical scenario, in order to optimize the convergence rate, the nodes will compute the matrix W in (2) in a distributed fashion [17] and the global network structure can be inferred as the support graph of the matrix W.

The SVOs can be used as an overlay to a consensus algorithm that detects Byzantine faults such as in [10]. In this paper, a novel algorithm is introduced that intersects local estimates among neighbors to obtain improved estimates. In the process, the set of possible states is reduced and the consensus solution is reached in finite time.

We define the set generated by the SVO at each node i as $X_i(k)$. In general, the result of the Fourier-Motzkin elimination method produces a polytopic set with a bounded number of vertices. However, transmitting $X_i(k)$ would mean communicating the matrix $M_i(k)$ and vector $m_i(k)$. Since the dimension of $M_i(k)$ depends on the number of vertices, we might need to communicate a large amount of information, which may not be feasible in many applications.

For that reason, we can overbound this uncertainty set by a hyper-parallelepiped $Set\left(\hat{M}_i(k), \hat{m}_i(k)\right)$, with

$$\hat{M}_i(k) = I \otimes \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

and $\hat{m}_i(k) = z_i(k) \in \mathbb{R}^{2n_x}$, where $\hat{m}_i(k)$ is defined such that $Set\left(\hat{M}_i(k), \hat{m}_i(k)\right)$ contains $X_i(k)$. Using this approach, z_i will be the only vector that we need to transmit between neighbors. Thus, the z_i's represent state boundaries for the other agents and are obtained through the previously described algorithm to compute the SVO (5) using local information available when communicating with neighbors.

The algorithm can be briefly described as, in each discrete time instant, each node that does not communicate with its neighbors updates its set-valued state estimates of the corresponding SVO using (5).

If node i and j communicate, then they intersect both set-valued state estimates followed by a consensus phase. Notice that z_i and z_j are state boundaries from the perspective of node i and j. Thus, $s^\star$ — the concatenation of $[z_i]_{2i-1}$ and $[z_i]_{2i}$ for each node i — is such that $s^\star \in \text{Set}(\hat{M}_i, z_i)$ and $s^\star \in \text{Set}(\hat{M}_j, z_j)$ (see Fig. The concept of $s^\star$ and state

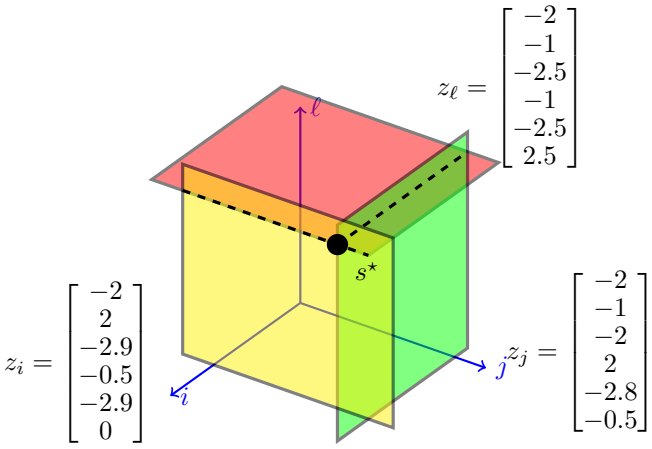

Fig. 1: Example of the set-valued estimates boundaries of node i (yellow), node j (green) and node ℓ (red), where for each node there is no uncertainty regarding its own state and where $s^\star$ represents the full state of the system that is contained in all three state boundaries.

boundaries are illustrated in Fig. 1. A fault is declared by node i, whenever it receives z_j with $[z_i]_{2j-1} > [z_j]_{2j-1} \vee [z_i]_{2j} > [z_j]_{2j}$ since there is not a vector $s^\star$ satisfying the observations made by node i and j.

The intersection step is described using the maximum function (z variables were defined to have the maximum of the boundary with a negative sign, see Fig. 1 for a numeric example) by operating on the state of both nodes i and j

$$z_i(k) = z_j(k) = \max(z_i(k), z_j(k)) \tag{6}$$

where the max function operates row-wise.

At each time k, the consensus phase runs in both communicating nodes and is defined for node i communicating with node j by the following linear iteration, similarly to what is done in [17]:

$$z_i(k+1) = \left[\left(\frac{1}{2}(e_i - e_j)(e_j - e_i)^\intercal + I_{n_x}\right) \otimes I_2\right] z_i(k) \tag{7}$$

where, as previously mentioned, the variable z_i is the vector-valued estimate of node i of all the states of the nodes. It should be noticed that, for node i, we have $[z_i]_{2i} = [z_i]_{2i-1}$ since the node has access to its own state at all time instants and thus there is no uncertainty associated to it.

IV. Main Properties

This section starts by providing a result regarding the convergence of the proposed algorithm to robust consensus for detecting Byzantine faults and performing the detection using the information of the estimates of each node.

Theorem 1: Take the consensus algorithm defined by equations (6) and (7). If the matrix of probabilities W is strongly connected, then the algorithm converges in:

- *expectation*
- *mean square sense*
- *almost surely*.

Proof: The proof follows a similar reasoning as in [17]. We start by noticing that from equation (6) and that, for node i, $[z_i]_{2i-1}$ and $[z_i]_{2i}$ are always equal to the node state. Thus,

we stack the values z_i and prove convergence of the whole system. Let us introduce variable $\mathbf{z}$:

$$\mathbf{z} = \begin{bmatrix} [z_1]_1 \\ [z_1]_2 \\ [z_2]_3 \\ [z_2]_4 \\ \vdots \\ [z_m]_{2n_x-1} \\ [z_m]_{2n_x} \end{bmatrix} \tag{8}$$

with $\mathbf{z} \in \mathbb{R}^{2n_x}$, where n_x is the number of nodes. Then, one can write

$$\mathbf{z}(k+1) = U_k \mathbf{z}(k) \tag{9}$$

where U_k is a matrix randomly selected from $\{Q_{ij}\}$ with

$$Q_{ij} = \left(\frac{1}{2}(e_i - e_j)(e_j - e_i)^\intercal + I_{n_x} \right) \otimes I_2 \tag{10}$$

for each pair of nodes i and j communicating with each other with probability w_{ij} from the probability matrix W.

Let us define

$$R = \mathbb{E}[U_k].$$

where $\mathbb{E}$ is the expected value operator. Then

$$\mathbb{E}[\mathbf{z}(k+1)] = R^k \mathbb{E}[\mathbf{z}(0)] \tag{11}$$

due to the probability distributions w_{ij} being independent. Rearranging using the transformation $T^\intercal Q_{ij} T$ with

$$T_{ij} = \begin{cases} 1, & \text{if } j = 2i-1 \wedge i \leq n_x \\ 1, & \text{if } j = 2(i-n_x) \wedge i > n_x \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

we get

$$T^\intercal R T = I_2 \otimes \left((1 - \frac{1}{n_x}) I_{n_x} + \frac{1}{n_x} W \right)$$

The eigenvalues of R are the eigenvalues of $(1 - \frac{1}{n_x}) I_{n_x} + \frac{1}{n_x} W$ counted twice. We can use the fact

$$\lambda((1 - \frac{1}{n_x}) I_{n_x} + \frac{1}{n_x} W) = (1 - \frac{1}{n_x}) I_{n_x} + \frac{1}{n_x} \lambda(W)$$

and since W is a doubly stochastic matrix with a strongly connected support graph with all but one eigenvalues less than 1. The $\lambda(W) = 1$ is associated to the eigenvector $\mathbf{1}_{n_x}$. Thus, $\lim_{k\to\infty} R^k = I_2 \otimes \mathbf{1}_{n_x}/n_x$ which proves the convergence in expectation.

In order to prove convergence in the mean square sense, let us compute

$$\mathbb{E}[z(k+1)^\intercal z(k+1)] = R_2 \mathbb{E}[z(k)^\intercal z(k)]$$

where $R_2 = R$ due to the fact that $Q_{ij}^\intercal Q_{ij} = Q_{ij}$. Therefore, we use the same argument and the algorithm converges in the mean square sense. Almost surely convergence is given by Borel-Cantelli first lemma [18], [19] due to (11), which means exponential rate convergence. ■

The previous theorem shows the asymptotic convergence of the algorithm, which is useful when considering small horizons. In the next theorem, we show an important feature of the proposed algorithm, when applied to Byzantine fault detection in networks, although its verification may be costly in terms of needed computational power.

Theorem 2: Consider the detection scheme of using a single centralized SVO for the whole network without sharing node measurements. There exists a sufficiently large horizon $N^\star$ such that $X(N^\star)$, constructed using (5), is a set with a finite number of points.

Proof: Let us rewrite the matrix in (5) recursively:

$$\underbrace{\begin{bmatrix} \mathcal{R}_1 & 0 & 0 & \cdots & 0 \\ 0 & \mathcal{R}_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \mathcal{R}_N & 0 \\ 0 & 0 & \cdots & 0 & M_0 \end{bmatrix}}_{\mathbf{M}_{\mathbf{\Delta^\star}}} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \vdots \\ \mathbf{x_k} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ y(k+1) \\ -y(k+1) \\ 0 \\ 0 \\ \vdots \\ y(1) \\ -y(1) \\ m_0 \end{bmatrix} \tag{13}$$

where

$$\mathcal{R}_n = \begin{bmatrix} I & -\mathcal{A}_n \\ -I & \mathcal{A}_n \\ C(k+1-(n-1)) & 0 \\ -C(k+1-(n-1)) & 0 \end{bmatrix}$$

and $\mathcal{A}_n$ represents the matrix $A_0 + A_{\Delta^\star}$ with a $\Delta^\star$ that accumulates the uncertainties for n periods of time, i.e., the parameter $\Delta^\star$ is the uncertainty instantiation for the respective horizon (see [13]).

Take a sufficiently large $N^\star$ such that the following sequence of time instants $\{c_k : 0 \leq c_k \leq N^\star\}$ is present due to the randomness of the node communications.

The sequence is constructed as follows with respective to node i running the SVO:

- Start with the first-degree neighbors and there exists a communication between i and all of them where only the state is transmitted and not the estimates, i.e., $\forall j : (i,j) \in E$ we have $A(k) = Q_{ij} \vee A(k) = Q_{ji}$;
- with each neighbor of i there exists a communication at time even and at time odd a communication between that neighbor and a second-degree neighbor and this pattern is repeated for the number of second-degree neighbors of each of our node neighbor, i.e., $\forall j : (i,j) \in E, \forall \ell : (j,\ell) \in E$ such that $A(2k) = Q_{ij} \vee A(2k) = Q_{ji}$ and $A(2k+1) = Q_{j\ell} \vee A(2k+1) = Q_{\ell j}$;
- repeat the same as before for the third-degree neighbors with communication between the nodes happening at each multiple of three communication instants. The number of communications must be equal to the number of possible paths with length 2.
- we continue with the same reasoning until all the nodes are included in the sequence.

Since when a node is involved in a communication there is no uncertainty, the sequence was constructed such that with the first condition all the neighbor states can be determined. With the second condition all the second-degree neighbor states can be determined. The same applies for any degree neighbors. This implies that for a specific instantiation of $\Delta^\star$, the system in (13) either:

- has only one solution;
- is infeasible.

Thus, the set $X(k)$ is a union of at most card(Δ) points. ■

Remark 3: Notice that in Theorem 2, it is not possible to get $X(k)$ to be a singleton without imposing additional conditions on the graph. From the perspective of a node i, every first-degree neighbors should have common neighbors among them. Otherwise, even though the state is restricted to a point, it is not possible to associate the state with the

node as they form an isomorphism from the node i point of view. However, to get finite-time consensus it is only needed to compute the average of one of the points in $X(k)$.

The previous result shows that SVOs have an intrinsic property that can be used to compute the average consensus. Theorem 2 assumes that estimates are not shared between neighbors at the expenses of considering a large horizon $N^\star$. In practice, however, the applicability of this result is questionable, as $N^\star$ can be arbitrarily large and represent a prohibitive computational burden since the SVO complexity grows exponentially with the horizon. The result is interesting in the scenario where the node running the SVO is controlling the network and is allowed to impose a given communication pattern. Progress is made in the next theorem to drop the horizon condition by taking advantage of state sharing between nodes.

Theorem 4: Consider the algorithm described in (6) and (7). There exists a sufficiently large running time $\tilde{N}$ such that $X(\tilde{N})$, constructed using (5), is a singleton.

Proof: Take a sufficiently large $\tilde{N}$ containing a sequence of time instants $\{c_k : 0 \leq c_k \leq \tilde{N}\}$ that fulfills

- every transmission shares one of the nodes involved in the previous transmission, i.e.,
$$\forall k \in \{c_k\} : \\ A(k) = Q_{ij}, A(k+1) = Q_{i\ell} \vee A(k+1) = Q_{\ell i}$$
for any node ℓ;
- there exists a time instant such that before and after that time, all the nodes in the network were involved in the communication, i.e.,
$$\exists k_c : \forall i \exists k_i \leq k_c : (A(k_i) = Q_{i\ell} \vee A(k_i) = Q_{\ell i}) \\ \wedge \\ \exists k_i' \geq k_c : (A(k_i') = Q_{i\ell} \vee A(k_i') = Q_{\ell i})$$
for any node ℓ.

It should be noticed that such a sequence exists due the random behavior of the communications.

Define a function
$$V(k) = \text{card}([z_i(k)]_{2i-1} - [z_i(k)]_{2i})$$
where the function $\text{card}(x)$ counts the number of non-zero entries of the vector x and i is a node involved in communication at time k. Function $V(k)$ counts, therefore, the number of uncertain states of the last node i involved in a communication at time k. Recall that from equation (6), both nodes i and j involved in the communication have the same estimates of the states for all the nodes in the network.

Notice that
$$V(k+1) - V(k) \leq 0$$
for all time instants $k \leq k_c$, since every transmission is assumed to include one node involved in the previous communication. Thus, at time k_c, $V(k_c) = 0$ using the two conditions of the sequence, which means that the two nodes communicating at time k_c have access to the full state of the network, regardless of the horizon of the SVOs.

Since, for every node ℓ, $\exists k_i' \geq k_c : A(k_i') = Q_{i\ell}$, the full state is passed to all the remaining nodes. ■

Remark 5: Notice that the sequence construction made in Theorem 4 can be implemented in practice by a token-passing scheme and obtain finite-time consensus regardless of the chosen horizon, if no fault is detected. Mechanisms for fault robustness in a token-based gossip algorithm are outside the scope of this paper and also further work is needed to evaluate its effects on the convergence rate.

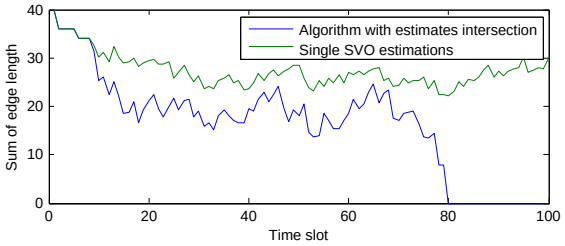

Fig. 2: Typical behavior of the size of the SVO.

V. Simulation Results

In this section, we show simulation results to illustrate the finite-time consensus property derived in the previous section. Focus is given to how a measure of the set dimension evolves with the algorithm as opposed to a setting where nodes just run SVOs without sharing their estimates. Simulations illustrating the fault detection capabilities can be found in [10]. The simulations also indicate how likely it is to find a sequence of transmissions that produce finite-time consensus when using randomized gossip algorithms.

We consider a 5-node network with nodes labelled $i, i \in \{1,2,3,4,5\}$, initial state $x_i(0) = i-1$ and a nominal bound for the state magnitude of $|x_i| \leq 5$. To assess the algorithm in a disadvantageous scenario, we considered the horizon $N = 1$, which is a worst-case scenario, as the algorithm only takes into account the dynamics of the system with one time step from the last estimate and discards prior observations. Each experiment lasts 300 communications and the results shown correspond to 1000 Monte-Carlo runs. The following probability matrix is used:

$$W = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0.25 & 0 & 0.25 \\ 0.5 & 0.25 & 0 & 0.125 & 0.125 \\ 0 & 0 & 0.125 & 0.25 & 0.625 \\ 0 & 0.25 & 0.125 & 0.625 & 0 \end{bmatrix}$$

Our experiment setting does not include any fault and, at each time instant, we compute a measure of the size of the SVO. Since the hypervolume is always zero, we present the average sum of the length of uncertainties as the size of the sets across the network, which by definition if zero, then all nodes have reached consensus.

A typical run is depicted in Fig. 2 where finite-time consensus is achieved. All the simulations share the same behavior but with different times to reach consensus. In comparison, the same measure is calculated for the case of independent SVOs. As expected, the estimates using the algorithm are less conservative as they incorporate the measurements performed by the node itself and the estimation set transmitted by its neighbors. In this particular run, consensus was achieved by all the nodes at iteration 80.

Using a 1000 Monte-Carlo run, in Fig. 3 is shown the histogram for the stopping time of the algorithm when using a horizon of 1. The experiments where consensus was not achieved in less than 300 communications are not represented in the histogram and corresponded to 21.9%. Simulations were repeated using the same sequence of communications and a horizon of 5, decreasing such percentage to 13.4%. This fact is justified by the smaller sets generated by each node. In essence, to get 100% of the experiments to

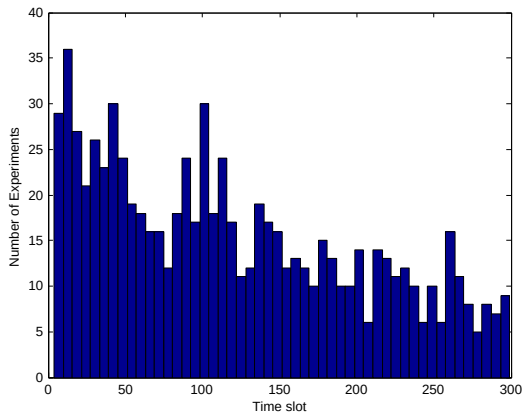

Fig. 3: Histogram for the stopping time with the proposed algorithm.

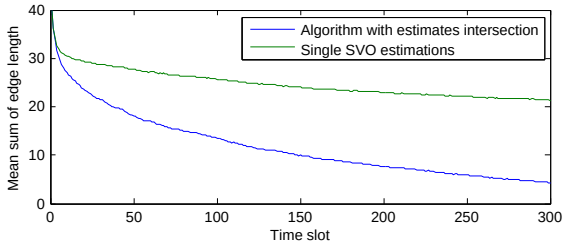

Fig. 4: Evolution of the mean sum of edges of all node set-valued state estimations.

end in finite-time, we either have to increase the time of the simulation or increase the horizon, or both.

An important issue is the influence of the intersection step on the size of the set-valued state estimates. Fig. 4 depicts the mean of the sum of edges length for the 1000 Monte-Carlo runs for both the case of a centralized SVO and estimate sharing using the intersection algorithm. Since the gossip random consensus algorithm is stable [17], the size of the generated set converges to a point (the consensus value) and the sum of edge lengths goes to zero asymptotically when in a non-faulty scenario and subject to a horizon smaller than $N^\star$. The measure of the sum of edges captures the size of the set-valued estimates, and correspondingly, how conservative they are. Fig. 4 shows that, in average, less conservatism is attained by exchanging set-valued estimates, and that they converge much faster to zero since the conditions of the Theorem 4 are less restrictive.

VI. Conclusions

In this paper, two methods are developed to address the problem of distributively detecting Byzantine faults in randomized gossip algorithms, namely one where a single node is running an SVO and another where each node has an SVO and set-state estimates are shared upon communication.

For the single node case, the method is guaranteed to detect Byzantine faults, but may require a large computational burden. In the absence of fault, it is shown that average consensus can be determined in finite-time using only measurements available to the node. The result is more applicable in situations where one node is able to control/command the sequence of communications.

In order to drop the requirement of a large horizon, an algorithm is presented where each node computes its own set-valued state estimates and performs an intersection with state estimates received by the neighbors. Besides reducing the computational burden, this method also achieves finite-time average consensus for any horizon value, provided that the algorithm runs for sufficiently large number of observations, and each node computes less conservative set-valued estimates. The result is relevant in practice to determine a stopping time in a faulty environment, which is not a straightforward issue due to the iterative nature and uncertainty generated by the random choice of communicating nodes. If conditions for finite-time convergence are not met within the time that the algorithm is running, asymptotic convergence of the state of the nodes is also provided.

References

[1] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90 –104, jan. 2012.

[2] S. Sundaram and C. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *Automatic Control, IEEE Transactions on*, vol. 56, no. 7, pp. 1495–1508, July 2011.

[3] D. Bertsekas and I. Rhodes, "Recursive state estimation for a set-membership description of uncertainty," *IEEE Transactions on Automatic Control*, vol. 16, no. 2, pp. 117 – 128, apr 1971.

[4] C. Combastel, "A state bounding observer for uncertain non-linear continuous-time systems based on zonotopes," in *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05.*, dec. 2005, pp. 7228 – 7234.

[5] T. Alamo, J. Bravo, and E. Camacho, "Guaranteed state estimation by zonotopes," *Automatica*, vol. 41, no. 6, pp. 1035 – 1043, 2005.

[6] H. Witsenhausen, "Sets of possible states of linear systems given perturbed observations," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 556 – 558, oct 1968.

[7] F. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 22 – 28, feb 1968.

[8] F. Schweppe., *Uncertain Dynamic Systems*. Prentice-Hall, 1973.

[9] M. Milanese and A. Vicino, "Optimal estimation theory for dynamic systems with set membership uncertainty: An overview," *Automatica*, vol. 27, no. 6, pp. 997 – 1009, 1991.

[10] D. Silvestre, P. Rosa, R. Cunha, J. P. Hespanha, and C. Silvestre, "Gossip average consensus in a byzantine environment using stochastic set-valued observers," in *52nd IEEE Conference on Decision and Control.*, 2013, Florence, Italy.

[11] M. Grewal and K. Glover, "Identifiability of linear and nonlinear dynamical systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 6, pp. 833–837, 1976.

[12] P. Rosa and C. Silvestre, "On the distinguishability of discrete linear time-invariant dynamic systems," in *50th IEEE Conference on Decision and Control*, December 2011.

[13] P. Rosa, "Multiple-Model Adaptive Control of Uncertain LPV Systems," Ph.D. dissertation, Technical University of Lisbon, Lisbon, Portugal, 2011.

[14] J. Shamma and K.-Y. Tu, "Set-valued observers and optimal disturbance rejection," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 253 –264, feb 1999.

[15] S. Keerthi and E. Gilbert, "Computation of minimum-time feedback control laws for discrete-time systems with state-control constraints," *IEEE Transactions on Automatic Control*, vol. 32, no. 5, pp. 432 – 435, may 1987.

[16] P. Rosa, C. Silvestre, J. Shamma, and M. Athans, "Fault detection and isolation of LTV systems using set-valued observers," *49th IEEE Conference on Decision and Control*, pp. 768–773, December 2010, Atlanta, Georgia, USA.

[17] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508 – 2530, Jun. 2006.

[18] E. Borel, "Les probabilités dénombrables et leurs applications arithmetiques," *Rend. Circ. Mat. Palermo (2)*, vol. 27, pp. pp. 247–271, 1909.

[19] F. P. Cantelli, "Sulla probabilità come limite della frequenza," *Atti Accad. Naz. Lincei*, vol. 26:1, pp. pp. 39–45, 1917'.