Resilient Desynchronization for Decentralized Medium Access Control

Daniel Silvestre, João Hespanha and Carlos Silvestre

***Abstract*— In Wireless Sensor Networks (WSNs), equally spaced timing for Medium Access Control (MAC) is fundamental to guarantee throughput maximization from all nodes. This motivated the so called *desynchronization* problem and its solution based on the fast Nesterov method. In this paper, we tackle the problem of constructing centralized and distributed versions of the optimal fixed-parameter Nesterov that are resilient to attacks to a subset of nodes. By showing a relationship between the variance of the attacker signal and how further away a node is, we are able to present a distributed algorithm that has minimal added complexity and performs the detection and isolation of the faulty node. Simulations are provided illustrating the successful detection and highlighting that without a correction mechanism (dependent on additional assumptions), there is a residual error that is not eliminated.**

***Index Terms*— Distributed control; Communication networks; Optimization algorithms**

I. Introduction

The problem of desynchronizing transmitters in Wireless Sensor Networks (WSNs) plays a key role in getting a fair Time Division Multiple Access (TDMA). In WSNs, in the absence of a centralized structure, the design of distributed algorithms capable of performing desynchronization at the layer of the Medium Access Control (MAC) is fundamental to the practical implementation of TDMA. In the literature, many authors have looked into this problem of how to devise distributed algorithms that can spread the transmitters evenly across the time slots [1], [2], [3], [4], [5].

Centralized solutions to the desynchronization problem often rely on a coordination channel, a central node or a global clock (for instance using GPS) [2]. It is also common that algorithms describe the mechanism for channel hopping at the physical layer to avoid excessive interference. The state-of-the-art is defined in IEEE 802.15.4e-2012 standard [6] where it is used the Time-Synchronized Channel Hoping (TSCH) protocol [2]. Various works have considered distributed desynchronization algorithms [1], [2], [4], [7], [8], [9], [10], [11], [12] that would enable a decentralized WSN MAC-layer coordination. These proposals follow a biological-inspired model, named Pulse-Coupled Oscillators (PCOs), where each node periodically sends a pulse signal and adjusts their transmissions based on the difference to the nodes transmitting before and after (immediate neighbors). These methods based on the seminal work by Mirollo and Strogatz [13] present features amenable to WSNs: limited listening [1], [14], [15] that enables power saving in wireless transceivers; algorithms capable of dealing with multi-hop networks and hidden nodes [1], [8]; scalability to a large number of nodes [4], [12]; and, fast convergence to steady-state [9], [10], [11], [15].

In [16] and later on [17], it was shown that the problem can be seen as the solution of a quadratic function and addressed both using optimization algorithms such as the Nesterov method and as the solution of a linear equation, respectively. Following the techniques in [17], it is possible to optimize the parameters of the Nesterov method and achieve a very efficient distributed iterative algorithm for the desynchronization problem. However, all these techniques lack the appropriate mechanisms to deal with faulty nodes or attackers that purposely transmit the periodic signals at different times to avoid convergence and the state of desynchrony in the network that corresponds to a fair schedule of the data transmissions.

Resilient algorithms have attracted attention from the control community especially when dealing with the consensus problem (see for example [18] and the references therein). In the consensus literature, such algorithms often rely on ignoring a number of minimum and maximum values, under the fear that these may have been manipulated. An example is the Mean-Subsequence Reduced (MSR) algorithm in [19] [20]. Many other works verse on the same key idea of removing large and small state values from neighbors: in second-order consensus with sampled data [21]; clock synchronization in WSNs [22]; leader-follower consensus to arbitrary reference values [23]; consensus with time-varying topologies [24]; and, consensus with quantized transmissions and communication delays [25]. However, this is not possible in the desynchronization problem given that each node only has two neighbors and the topology graph is not f-robust even if $f = 1$.

Similar ideas of the MSR algorithm have been investigated theoretically for the case of distributed optimization such as in [26]. Since the desynchronization will be performed using the Nesterov method, the work in [26] is closely related to the one presented herein. The authors have characterized the limitations in performance and the distance-to-optimality of

D. Silvestre is with the Department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Macau, Macau, China, and with the Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal, dsilvestre@isr.ist.utl.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 the National Science Foundation under Grants No. EPCN-1608880 and CNS-1329650. hespanha@ece.ucsb.edu

C. Silvestre is with the Department of Electrical and Computer Engineering of the Faculty of Science and Technology of the University of Macau, Macau, China, on leave from Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal, csilvestre@umac.mo

This work was partially supported by the project MYRG2016-00097-FST of the University of Macau; by the Portuguese Fundação para a Ciência e a Tecnologia (FCT) through Institute for Systems and Robotics (ISR), under Laboratory for Robotics and Engineering Systems (LARSyS) project UID/EEA/50009/2019.

this type of algorithms. In this paper, we leverage the specific structure of the problem to achieve a resilient algorithm and to avoid the need for an f-robust graph, which would increase the communication overhead.

Another approach to the design of resilient consensus algorithms, is based on a per-node defense mechanism that enables each node to estimate the state of all remaining nodes. In the work in [27], Set-valued Observers are used to produce sets where the remaining state values must belong. Detection is performed when the received value is outside of the set. This has also been refined in [28] by allowing nodes to exchange estimates and generalized for linear gossip algorithms in [29]. However, both detection and identification of the faulty node incur in a high computational cost, albeit with theoretical guarantees of no false detection. In this paper, a defense mechanism is devised that does not increase the communication and complexity apart from a constant irrespective of the size of the network.

The intuitive idea used in this paper is that the Nesterov method naturally attenuates the variance introduced by faulty nodes. Exploring this key idea, a centralized algorithm is devised that can be extended to the distributed case if nodes use the periodic messages to encode 4 values identifying the node they believe is the faulty one.

The main contributions of this paper are:

- Interesting properties of the desynchronization algorithm employing the optimal fixed-rate Nesterov are shown;
- Based on the previous, a centralized algorithm that is capable of eliminating the faulty nodes based on the variance of their transmission times;
- A resilient distributed version of the desynchronization is introduced that is based on a voting scheme among the nodes.

Notation : The transpose of a matrix A is denoted by $A^\intercal$. We let $1_n := [1 \dots 1]^\intercal$ and $0_n := [0 \dots 0]^\intercal$ indicate n-dimension vector of ones and zeros, and I_n denotes the identity matrix of dimension n. Dimensions are omitted when no confusion arises. The vector e_i denotes the canonical vector whose components equal zero, except component i that equals one. The Euclidean norm for vector x is represented as $\|x\|_2 := \sqrt{x^\intercal x}$.

II. Desynchronization Problem Review

To select their medium access time, nodes in a WSN periodically broadcast a *fire message* or a *pulse*. The nomenclature comes from the relationship with how biological systems such as firefly desynchronize their light pulses to attract a suitable mate. Each node in the network possesses a *phase* variable $\theta_i(t)$ for each $i \in \{1, \cdots, n\}$ that is defined as:

$$\theta_i(t) = \frac{t}{T} + \phi_i(t) \mod 1,$$

where the phase offset $\phi_i \in [0, 1]$ corresponds to the difference between each consecutive pulse message, and where the notation $\mod$ stands for the modulo arithmetics. The objective of a desynchronizing algorithm is to update the variable $\phi_i(t)$ of each node i such that the pulses occurring whenever $\theta_i(t) = 0$ become separated by $1/n$. A pulse is sent from node i to its neighbors, each time the phase θ_i passes through zero. Upon reception of the previous and next node to transmit (i.e., neighbor nodes if we were placing all of the nodes in the circle at their phase value), agent i adjusts its phase offset ϕ_i according to some update equation.

In [16], it was shown that finding phase offsets for all nodes is characterized as the minimizers of the following quadratic function:

$$\underset{\phi}{\text{minimize}} \quad g(\phi) := \frac{1}{2}\|D\phi - \frac{1_n}{n} + \mathsf{e}_n\|_2^2$$

where 1_n is the vector of ones, $\mathsf{e}_n = (0, 0, \cdots, 0, 1)$, and

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & & \vdots \\ 0 & \cdots & 0 & 0 & -1 & 1 \\ 1 & \cdots & 0 & 0 & 0 & -1 \end{bmatrix}.$$

III. Desynchronization using the optimal fixed-parameter Nesterov

In [30], it is shown that the Nesterov method given by the equations:

$$\text{Nesterov}: \begin{array}{l} z^{(k+1)} = \xi^{(k)} - \beta \nabla g(\xi^{(k)}) \\ \xi^{(k)} = (1+\gamma) z^{(k)} - \gamma z^{(k-1)} \end{array}$$

can be expressed as:

$$x^{(k+1)} = (A + BQC)x^{(k)} + BD^\intercal \mathsf{e}_n \tag{1}$$

where $x := \begin{bmatrix} z^\intercal & \xi^\intercal \end{bmatrix}^\intercal$, $Q = D^\intercal D$ and:

$$A = \begin{bmatrix} (1+\gamma)I_n & -\gamma I_n \\ I_n & 0_n \end{bmatrix}, B = \begin{bmatrix} -\beta I_n \\ 0_n \end{bmatrix},$$
$$C = \begin{bmatrix} (1+\gamma)I_n & -\gamma I_n \end{bmatrix}.$$

Remark that in the above formulation it was used the fact that $\nabla g(\phi) = D^\intercal D\phi + D^\intercal \mathsf{e}_n$ and that the system in (1) corresponds to a second order dynamical system where the first n entries of x have the ϕ_i variables for each agent.

The iterative algorithm in (1) is of the form:

$$x^{(k+1)} = Tx^{(k)} + b$$

where T is given by:

$$T = \begin{bmatrix} (1+\gamma)(I - \beta Q) & -\gamma(I - \beta Q) \\ I & 0 \end{bmatrix}. \tag{2}$$

Let us define the sorted spectra of Q as $0 < m \leq \lambda_2^Q \leq \cdots \leq \lambda_{n-1}^Q \leq L$, which are all real eigenvalues given that Q is the Laplacian matrix of the ring network. According to [30] and using $\kappa = L/m$, if one selects $\frac{1}{3} < \beta = \frac{4}{3L+m} < \frac{4}{9}$ and $\gamma = \frac{\sqrt{3\kappa+1}-2}{\sqrt{3\kappa+1}+2} < 1$ then the best worst-case convergence rate of $1 - \frac{2}{\sqrt{3\kappa+1}}$ is achieved and the optimal fixed-parameter Nesterov is always convergent.

a) Attacker model: In this paper, we assume that the attacker is able to corrupt (1) by adding a signal $a^{(k)}$ that is nonzero in the entries corresponding the set of attacked nodes $\mathcal{A}$. Therefore, corrupted version of the optimal fixed-parameter Nesterov is:

$$x^{(k+1)} = (A + BQC)x^{(k)} + BD^{\intercal}\mathsf{e}_n + a^{(k)}. \qquad (3)$$

Moreover, as seen from Figure 1, it is made the assumption that the signal $a^{(k)}$ does not change the relative order of the states. Assuming that $x^{(0)}$ is sorted, this corresponds to restricting $a^{(k)}$ such that $\forall k \geq 0, \forall i < j : x_i^{(k+1)} < x_j^{(k+1)}$. Notice that an attacking signal not satisfying this condition, even though very easy to detect by the adjacent nodes in the ring to the two nodes that broke the relative ordering, it is hard to counter (the attacker can make its state ordering arbitrary in each iteration) without posing additional defense characteristics in the nodes, e.g., the possibility of coordinated change to a different channel. In this work, the objective of the attacker is to prevent desynchronization among the healthy nodes while remaining undetected.

b) Properties of the Nesterov method: In the next proposition, it is shown that nodes further away from an attacker have a smaller variance caused by the injection of the signal of that particular attacker.

Proposition 1 (Variance-distance relationship): Consider a set of n nodes running the optimal fixed-parameter Nesterov in (1) for some initial condition $x^{(0)} \in [0,1]^n$, attacker set $\mathcal{A} = \{i\}$ and injected signal $a^{(k)}$. Then, there exists $\mathcal{T}$ such that for $k \geq \mathcal{T}$, it holds:

- $\mathrm{Var}(x_i^{(k)}) > \mathrm{Var}(x_{i+1}^{(k)}) > \cdots > \mathrm{Var}(x_n^{(k)})$;
- and, $\mathrm{Var}(x_i^{(k)}) > \mathrm{Var}(x_{i-1}^{(k)}) > \cdots > \mathrm{Var}(x_1^{(k)})$;

where Var stands for the sample variance of the signal from the initial time up to the current one.

Before introducing the proof of Proposition 1, it is useful to prove some properties of matrix T, which are summarized in the next lemma.

Lemma 2 (Properties of T): Consider matrix T defined in (2). Then, the following hold:

i) $T\mathsf{1}_{2n} = \mathsf{1}_{2n}$;
ii) $|T_{ij}| < 1$.

Proof: i) We start by computing:

$$(I - \beta Q)\mathsf{1}_n = \mathsf{1}_n - \mathsf{0}_n = \mathsf{1}_n$$

Thus, the submatrix corresponding to the first n rows of T multiplied by the vector of ones is equal to

$$\begin{aligned}(1+\gamma)(I-\beta Q)\mathsf{1}_n - \gamma(I-\beta Q)\mathsf{1}_n &= (1+\gamma)\mathsf{1}_n - \gamma\mathsf{1}_n \\ &= \mathsf{1}_n\end{aligned}$$

and multiplying the remaining n rows of T by $\mathsf{1}_{2n}$ also satisfies i) trivially.

ii) In the first $n \times n$ block of matrix T we either have diagonal entries equal to $(1+\gamma)(1-2\beta) \in (0, \frac{2}{3})$ given the bounds for β and γ or off-diagonal elements $(1+\gamma)\beta \in (0,1)$. In the block corresponding to $-\gamma(I-\beta Q)$ either has the diagonal elements equal to $-\gamma(1-2\beta) \in (-1,0)$ or $-\beta\gamma \in (-1,0)$. The remaining two blocks are trivially either zero or one, and the property holds. ∎

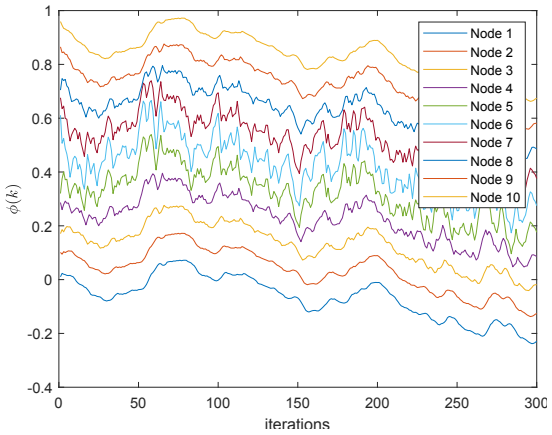

Fig. 1. Evolution of a 10 node network states when a random noise is introduced in node 6 and using the optimal fixed-parameter Nesterov.

The proof of Proposition 1 is now presented.

Proof: Let us first rewrite (3) to the following format:

$$\begin{aligned}x^{(k)} &= T^k x^{(0)} + \sum_{\tau=0}^{k-1} T^{k-1-\tau}\left(BD^{\intercal}\mathsf{e}_n + a^{(\tau)}\right) \\ &= T^k x^{(0)} + \sum_{\tau=0}^{k-1} T^{k-1-\tau}BD^{\intercal}\mathsf{e}_n + \sum_{\tau=0}^{k-1} T^{k-1-\tau}a^{(\tau)} \\ &= z^{(k)} + \sum_{\tau=0}^{k-1} T^{k-1-\tau}a^{(\tau)}\end{aligned}$$

In the above equation, since the Nesterov method in (1) is convergent it means that:

$$\forall j \leq n : \lim_{k\to\infty} \mathrm{Var}(z_j^{(k)}) = 0.$$

Therefore, for any $\epsilon > 0$ there exists $\mathcal{T}$ such that $\mathrm{Var}(z_j^{(\mathcal{T})}) < \epsilon$. From Lemma 2, $\alpha^{(1)} = Ta^{(\tau)}$ will have only three nonzero entries, namely $\alpha_{i-1}^{(1)}$, $\alpha_i^{(1)}$ and $\alpha_{i+1}^{(1)}$, which satisfies $\alpha_i^{(1)} > \alpha_{i-1}^{(1)}$ and $\alpha_i^{(1)} > \alpha_{i+1}^{(1)}$ since the non-attacked entries make weighted averages with zero entries. Moreover, we have that $\alpha_{i-1}^{(1)} = \alpha_{i+1}^{(1)}$. The same reasoning applies to any $\alpha^{(k)}$, which means that $\sum_{\tau=0}^{k-1} T^{k-1-\tau}a^{(\tau)}$ inherits the same property. For a sufficiently large $\mathcal{T}$, the conclusion follows. ∎

In order to illustrate the aforementioned results, an example of a 10 node network is depicted in Fig. 1 for the case where $\mathcal{A} = \{6\}$ and $a_6^{(k)}$ follows a random uniform distribution. Since there are no defense mechanism, the sample variance of each $x_j^{(k)}$ is sorted as given by Proposition 1.

The effect stated in Proposition 1 is better visualized when the attacker uses an oscillatory $a^{(k)}$ such as a cosine. In Figure 2, it is depicted the damped oscillation as we move away from the attacker in the ring.

We can therefore state the main result regarding a variance-based detector that is going to be proposed in the next section.

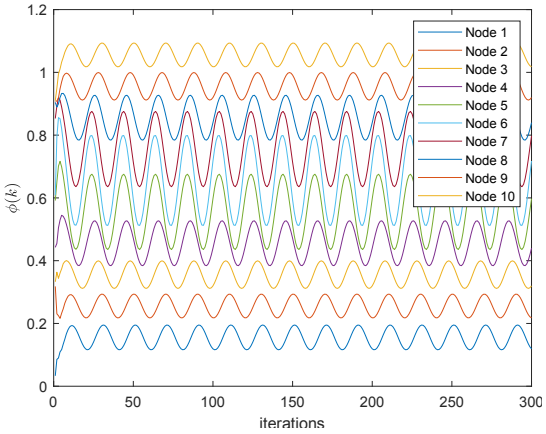

Fig. 2. Evolution of a 10 node network states when a cosine signal is injected in node 6 and using the optimal fixed-parameter Nesterov.

Theorem 3 (Vanishing attacks): Consider a resilient mechanism removing the state of the largest variance node from the updates, operating over the optimal fixed-parameter Nesterov algorithm described by (3). A successful attack $a^{(k)}$ on node i has to satisfy $\forall k \geq 0$:

$$\begin{aligned}\mathrm{Var}(\zeta_i^{(k)}) &\leq \max_{j\neq i} \mathrm{Var}(x_j^{(k)}) - \mathrm{Var}(z_i^{(k)}) - 2\mathrm{Cov}(z_i^{(k)}, \zeta_i^{(k)}) \\ &\leq \omega^{(k)}\end{aligned}$$

where $\zeta^{(k)} = \sum_{\tau=0}^{k-1} \alpha^{(\tau)}$ and $\lim_{k\to\infty} \omega^{(k)} = 0$.

Proof: We begin by noticing that:

$$\forall j \leq n : \mathrm{Var}(x_j^{(k)}) = \mathrm{Var}(z_j^{(k)}) + \mathrm{Var}(\zeta_j^{(k)}) + 2\mathrm{Cov}(z_j^{(k)}, \zeta_j^{(k)}).$$

For the attack to be successful it has to be undetected or otherwise the resilient mechanism will exclude the node from future updates. Therefore, $\mathrm{Var}(x_i^{(k)}) \leq \max_{j\neq i} \mathrm{Var}(x_j^{(k)})$ from which the first inequality is derived.

Given the convergent properties of Nesterov method, there exists $\mathcal{T}$ for any $\epsilon > 0$ such that for all $k > \mathcal{T}$, $\mathrm{Var}(z_i^{(k)}) < \epsilon$. Thus, for $k > \mathcal{T}$:

$$\begin{aligned}\mathrm{Var}(\zeta_i^{(k)}) &\leq \max_{j\neq i} \mathrm{Var}(x_j^{(k)}) \\ &\leq \max_{j\neq i} \mathrm{Var}(\zeta_j^{(k)}) + \epsilon\end{aligned}$$

Since ϵ can be made arbitrarily small and given the ordering property in Proposition 1, $a_i^{(k)} > 0$ translates into:

$$\mathrm{Var}(\zeta_i^{(k+1)}) - \mathrm{Var}(\zeta_i^{(k)}) > \max_{j\neq i} \mathrm{Var}(\zeta_j^{(k+1)}) - \max_{j\neq i} \mathrm{Var}(\zeta_j^{(k+1)})$$

which implies $\lim_{k\to\infty} a_i^{(k)} = 0$ and in turn the existence of $\omega^{(k)}$ such that $\lim_{k\to\infty} \omega^{(k)} = 0$, leading to the conclusion. ∎

Remark that as long as the characteristics proven in this section hold, the resilient algorithm will work in different problems and is not limited to the case of desynchronization. However, given the ring structure of neighbor relationship, most resilient algorithms based on the MSR concept cannot be used since for $f = 1$ they would require removing 2 neighbors from each node resulting in no edges in the network.

IV. Resilient Nesterov for Desynchronization

In this section, an algorithm is suggested to avoid the persistent errors caused by faulty nodes in the ring network. We first address the case where there is a centralized processing unit gathering the node states and deciding which nodes to stop taking into account their neighbor values.

A. Centralized version

A centralized detector maintains a vector $v, \mu \in \mathbb{R}^n$ (v tracks the sample variance over time while μ computes the average) following the equations:

$$\begin{aligned}v^{(k)} &= v^{(k-1)} + \left(x^{(k)} - \mu^{(k-1)}\right)\left(x^{(k)} - \mu^{(k)}\right) \\ \mu^{(k)} &= \mu^{(k-1)} + \frac{1}{k}\left(x^{(k)} - \mu^{(k-1)}\right)\end{aligned}$$

and $v^{(0)} = \mu^{(0)} = 0_n$.

At each time instant k, the detector finds the node $i^\star$ corresponding to the largest entry of $v(k)$ and signals the nodes just before and after $i^\star$ to stop updating their states, i.e.,

$$x_{\text{prev}}^{(k+1)} = x_{\text{prev}}^{(k)}, \qquad x_{\text{next}}^{(k+1)} = x_{\text{next}}^{(k)} \tag{4}$$

Notice that the solution adopted in (4) does not make any additional assumptions as it is based solely on having nodes stopping their updates in order to avoid propagating the effects of the attack through the neighbor network topology (a ring). With the additional assumption of having a mechanism to remove the attacker from the medium, the process can be restarted with one less node and appropriately redefining matrix Q and the sizes of A, B and C. Multiple attacked nodes can be stopped by having the detector stop other nodes with non vanishing variances.

B. Distributed version

In this section, we make the implicit assumption that either there is a single attacked node or that the attackers cannot alter the content of the pulse messages. The distributed version requires that each node i computes the variance of itself and its two neighbors. The local computation at node i, assuming neighbors $i-1$ and $i+1$ (the indices are computed using modulo operation such that $i-1$ for node $i=1$ is n) are:

$$z_i^{(k)} = z_i^{(k-1)} + \left(x_{[i-1:i+1]}^{(k)} - \nu_i^{(k-1)}\right)\left(x_{[i-1:i+1]}^{(k)} - \nu_i^{(k)}\right)$$

$$\nu_i^{(k)} = \nu_i^{(k-1)} + \frac{1}{k}\left(x_{[i-1:i+1]}^{(k)} - \nu_i^{(k-1)}\right)$$

where $z_i, \nu_i \in \mathbb{R}^3$ and the notation $x_{[i-1:i+1]}^{(k)}$ stands for the indices between $i-1$ and $i+1$ computed with modulo operation from vector $x^{(k)}$.

Since there is no centralized detector that can activate the nodes to stop their updates, we will introduce a distributed

decision process similar to a maximization consensus but taking into account that the attacker node can falsify data and send corrupted variance computations to its neighbors.

Let us define ψ^ℓ and ψ^r as the vectors aggregating the two new pieces of information that each node has to store for the voting scheme. The superscripts ℓ and r serve to identify that the entries in each of these vectors correspond to the *left* and *right* votes of a node and comprise the node id and correspondent variance. At each iteration, each node j sends ψ_j^ℓ and ψ_j^r in its pulse message. At the end of the cycle, node j makes:

$$\psi_j^\ell = \psi_{j+1}^\ell, \qquad \psi_j^r = \psi_{j-1}^r$$

such that vector ψ_j^ℓ has a shift to the left if they are represented as columns, and conversely for ψ_j^r.

Nodes decide to stop updating their states if they receive a value equal to their vote. This corresponds to changing their update rule to (4). However, the nodes that are immediate neighbors might stop the update at different time steps depending on the location of the faulty node. After stopping updates, the stopped nodes will keep sending votes and resume the update if the variance of the detected node drops below what was stored. This mechanism prevents that a false detection causes the desynchronization error to persist while at the same time forcing the faulty node to reduce the variance of their states below the previous value when they were excluded.

V. Simulation Results

The main objective of this section is to provide simulation results that show the effectiveness of the resilient algorithm and expose the need for additional correction mechanisms to further reduce the error of the steady-state of the algorithm. The simulations considered a 10-node network executing the proposed resilient algorithm and resorting to the toolbox [31]. The first setup considered the case where node 6 introduces a random noise instead of applying the correct update form of the algorithm.

Figure 3 illustrates the logarithmic evolution of the error for the network when the resilient algorithm is used in 10 consecutive experiments using different uniform random initial conditions, when the attacker kept using its signal even after detection. One important aspect is that the error will typically oscillate while the nodes are deciding when to freeze their state around the one they consider the attacker and that at some point the decision remains constant and the algorithm executes and desynchronizes the remaining nodes. The residual error that does not vanish is caused by the difference between the two nodes freezing in the last decision. The error would go to zero according to the typical behavior of the Nesterov method if one considered exclusively the desynchronization among the non-faulty nodes and a transmission cycle of length equal to the difference between the two freezing nodes. Methods to correct this issue are subject of future research.

The second example considered the case where the attacker injects a cosine multiplied by the current difference

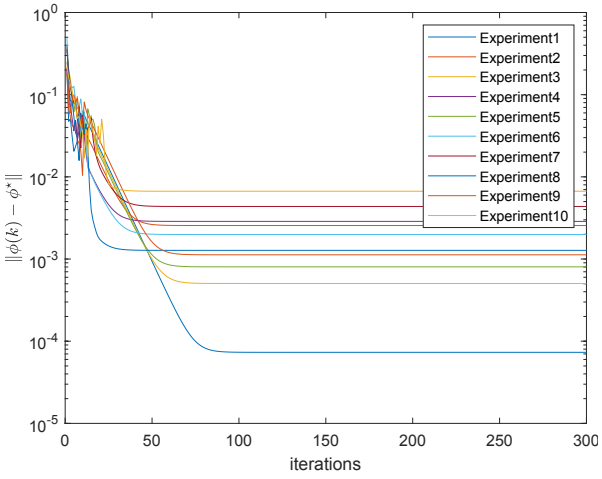

Fig. 3. Logarithmic evolution of the error for a 10 node network states when a random noise is introduced in node 6 and using the resilient optimal fixed-parameter Nesterov.

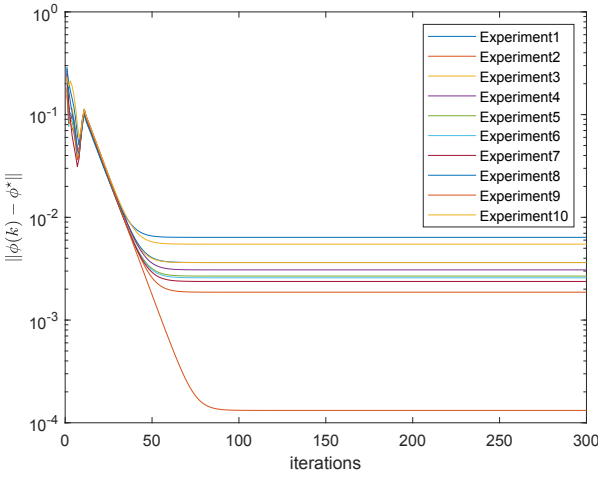

Fig. 4. Logarithmic evolution of the error for a 10 node network states when a cosine noise is introduced in node 6 and using the resilient optimal fixed-parameter Nesterov.

of its neighbors. This case shows that the attacker was able to maintain the non-faulty nodes with a higher error before the attack was detected. The results are depicted in Fig. 4.

The main conclusion to be drawn from Fig. 4 is that the resilient algorithm now has a error behavior very similar for multiple initial conditions. It still presents a decrease followed by an increase when the nodes are switching between who they label as the attacker. Once that decision remains constant, the error is reduced until the final value corresponds to the error between the two nodes that froze their states. If one added the possibility of these two nodes communicating between them, then the effect of the attacker can be perfectly removed.

VI. Discussion and Future Work

In this paper, we addressed the desynchronization problem in the presence of an attacker or faulty transmitter in the

context of a WSN. Given recent advances proposing the use of distributed algorithms to increase convergence, it is of interest to equip these algorithms with resilient mechanisms to faulty equipment or attackers. Given the need for a lightweight distributed solution at the MAC layer without increasing the number of possible neighbors, it prevented the use of resilient consensus algorithms based on discarding large and small state values or other techniques based on reachability analysis or through sharing of public/private keys and encryption. To this end, the Nesterov method is firstly shown to attenuate the variance of nodes that are further away from the injected signal in the ring.

In this work, it is proposed a variance-based method followed by a maximum consensus-type algorithm that enables the distributed decision of which nodes should stop updating their states to avoid propagating the injected signal. It is shown that, in order to remain undetected, the attacker has to inject a signal that must converge to zero. In simulation, it is tested against random uniform noise and oscillatory signals from the faulty node and detection and isolation occurs, leading the remaining nodes to achieve desynchronization.

We envision as topics for future work: i) the current method was tested for a single faulty node and ways to generalize the decision are required. If nodes are not attackers, this should not cause an issue as the voting still works but under more evolved strategies it is not sufficient; ii) application of this type of procedure to consensus algorithms since the attenuation of variance is caused by the averaging property of the algorithm and not specific of the Nesterov method. This would open the possibility to address faults in network topologies that are not f-robuts; iii) if each transmitter can estimate which node has the maximum variance based solely on the local data and without requiring additional information exchange would also bring the possibility of implementation directly in the state-of-the-art standard.

REFERENCES

[1] J. Degesys and R. Nagpal, "Towards desynchronization of multi-hop topologies," in *Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, Oct 2008, pp. 129–138.

[2] A. Tinka, T. Watteyne, and K. Pister, *A Decentralized Scheduling Algorithm for Time Synchronized Channel Hopping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 201–216.

[3] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. J. Pister, "A Realistic Energy Consumption Model for TSCH Networks," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 482–489, Feb 2014.

[4] R. Pagliari and A. Scaglione, "Scalable network synchronization with pulse-coupled oscillators," *IEEE Transactions on Mobile Computing*, vol. 10, no. 3, pp. 392–405, March 2011.

[5] D. Buranapanichkit and Y. Andreopoulos, "Distributed time-frequency division multiple access protocol for wireless sensor networks," *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 440–443, October 2012.

[6] IEEE, "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–225, April 2012.

[7] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. H. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 81–97, September 2008.

[8] A. Motskin, T. Roughgarden, P. Skraba, and L. Guibas, "Lightweight coloring and desynchronization for networks," in *IEEE INFOCOM 2009*, April 2009, pp. 2383–2391.

[9] C. M. Lien, S. H. Chang, C. S. Chang, and D. S. Lee, "Anchored desynchronization," in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 2966–2970.

[10] R. Leidenfrost and W. Elmenreich, "Firefly clock synchronization in an 802.15.4 wireless network," *EURASIP Journal on Embedded Systems*, vol. 2009, no. 1, p. 186406, Jul 2009.

[11] J. Klinglmayr and C. Bettstetter, "Self-organizing synchronization with inhibitory-coupled oscillators: Convergence and robustness," *ACM Trans. Auton. Adapt. Syst.*, vol. 7, no. 3, pp. 30:1–30:23, Oct. 2012.

[12] Y.-W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 1085–1099, May 2005.

[13] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, 1990.

[14] R. Pagliari, Y. W. P. Hong, and A. Scaglione, "Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 4, pp. 564–575, May 2010.

[15] Y. Wang, F. Nunez, and F. J. Doyle, "Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5293–5306, Oct 2012.

[16] N. Deligiannis, J. F. C. Mota, G. Smart, and Y. Andreopoulos, "Fast desynchronization for decentralized multichannel medium access control," *IEEE Transactions on Communications*, vol. 63, no. 9, pp. 3336–3349, Sept 2015.

[17] D. Silvestre, J. Hespanha, and C. Silvestre, "Desynchronization for decentralized medium access control based on gauss-seidel iterations," in *American Control Conference (ACC)*, July 2019, pp. 4049–4054.

[18] D. Silvestre, J. P. Hespanha, and C. Silvestre, "Broadcast and gossip stochastic average consensus algorithms in directed topologies," *IEEE Transactions on Control of Network Systems*, pp. 1–1, 2018.

[19] A. Haseltalab and M. Akar, "Approximate byzantine consensus in faulty asynchronous networks," in *American Control Conference (ACC)*, July 2015, pp. 1591–1596.

[20] ——, "Convergence rate analysis of a fault-tolerant distributed consensus algorithm," in *54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 5111–5116.

[21] S. M. Dibaji and H. Ishii, "Consensus of second-order multi-agent systems in the presence of locally bounded faults," *Systems & Control Letters*, vol. 79, pp. 23–29, 2015.

[22] Y. Kikuya, S. M. Dibaji, and H. Ishii, "Fault tolerant clock synchronization over unreliable channels in wireless sensor networks," *IEEE Transactions on Control of Network Systems*, pp. 1–1, 2018.

[23] J. Usevitch and D. Panagou, "Resilient leader-follower consensus to arbitrary reference values," in *American Control Conference (ACC)*. IEEE, 2018.

[24] D. Saldana, A. Prorok, S. Sundaram, M. F. M. Campos, and V. Kumar, "Resilient consensus for time-varying networks of dynamic agents," in *American Control Conference (ACC)*, May 2017, pp. 252–258.

[25] S. M. Dibaji, H. Ishii, and R. Tempo, "Resilient randomized quantized consensus," *IEEE Transactions on Automatic Control*, 2017.

[26] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, pp. 1–1, 2018.

[27] 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*, Dec 2013, pp. 4373–4378.

[28] D. Silvestre, P. Rosa, J. P. Hespanha, and C. Silvestre, "Finite-time average consensus in a byzantine environment using set-valued observers," in *American Control Conference*, June 2014, pp. 3023–3028.

[29] ——, "Stochastic and deterministic fault detection for randomized gossip algorithms," *Automatica*, vol. 78, pp. 46 – 60, 2017.

[30] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 57–95, 2016.

[31] D. Silvestre, "Optool—an optimization toolbox for iterative algorithms," *SoftwareX*, vol. 11, p. 100371, 2020.