Resilient Desynchronization for Decentralized Medium Access Control

Daniel Silvestre, *Member, IEEE*, João P. Hespanha, *Fellow, IEEE*, and Carlos Silvestre, *Senior Member, IEEE*

***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 letter, 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*—Communication networks, fault tolerant systems, sensor networks.**

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].

Centralized solutions to the desynchronization problem often rely on a coordination channel, a central node or a global clock (for instance using GPS) [1]. The state-of-the-art is defined in IEEE 802.15.4e-2012 standard [5] where it is used the Time-Synchronized Channel Hoping (TSCH) protocol [1]. Various works have considered distributed desynchronization algorithms [1], [3], [6], [7], [8], [9], [10] 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 its 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 [11] present features amenable to WSNs: limited listening [12], [13] that enables power saving in wireless transceivers; algorithms capable of dealing with multi-hop networks and hidden nodes [7]; scalability to a large number of nodes [3], [10]; and, fast convergence to steady-state [8], [9], [13].

The PCO networks have recently attracted attention due to their application to modeling populations of fireflies, claps of an applauding audience, cells of cardiac and even circadian pacemakers. In [14], synchronization is shown for the weak assumption of the existence of a root node in static networks and for dynamic networks in [15]. Results on global synchronization were given in [16] and the interested reader is referred to that article for a classification of the different solutions based on centralized/decentralized, discrete/continuous state or local/global synchronization.

In [17] and later on [18], it was shown that the problem can be seen as the minimization of a quadratic function and addressed using the Nesterov method. Following the techniques in [18], 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.

Resilient algorithms have attracted attention from the control community especially when dealing with the consensus

Manuscript received March 17, 2020; revised May 30, 2020; accepted June 23, 2020. Date of publication June 30, 2020; date of current version July 15, 2020. The work of João P. Hespanha was supported by the National Science Foundation under Grant EPCN-1608880 and Grant CNS-1329650. This work was supported in part by the Project MYRG2018-00198-FST of the University of Macau; and in part by the Portuguese Fundação para a Ciência e a Tecnologia through Institute for Systems and Robotics, under Laboratory for Robotics and Engineering Systems Project under Grant UIDB/50009/2020. Recommended by Senior Editor J. Daafouz. *(Corresponding author: Daniel Silvestre.)*

Daniel Silvestre is with the Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Macau, China, and also with the Institute for Systems and Robotics, 1049-001 Lisbon, Portugal (e-mail: dsilvestre@isr.ist.utl.pt).

João P. Hespanha is with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA (e-mail: hespanha@ece.ucsb.edu).

Carlos Silvestre is with the Department of Electrical and Computer Engineering, Faculty of Science and Technology, University of Macau, Macau, China, on leave from Instituto Superior Técnico, Universidade de Lisboa, 1649 Lisbon, Portugal (e-mail: csilvestre@umac.mo).

Digital Object Identifier 10.1109/LCSYS.2020.3005819

problem (see for example [19] and the references therein). 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 [20]. Many other works verse on the same key idea: in second-order consensus with sampled data [21]; clock synchronization in WSNs [22]; consensus with time-varying topologies [23]; and, consensus with quantized transmissions and communication delays [24]. 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 to the MSR algorithm have been investigated for distributed optimization such as in [25]. Since the desynchronization will be performed using the Nesterov method, the work in [25] is closely related to the one presented herein. The authors have characterized the limitations in performance and the distance-to-optimality of this type of algorithms. In this letter, 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 [26], 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 [27] by allowing nodes to exchange estimates and generalized for linear gossip algorithms in [28]. 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 letter, 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 main contributions of this letter are:

- Interesting properties of the desynchronization algorithm employing the optimal fixed-rate Nesterov are shown;
- It is proposed a centralized and a distributed algorithm capable of eliminating the faulty nodes based on the variance of their transmission offsets;
- A theoretical result stating that the attacker signal has to converge to zero in order to remain undetected.

Notation: The transpose of a matrix A is denoted by $A^\intercal$. We let $\mathbf{1}_n := [1 \dots 1]^\intercal$ and $\mathbf{0}_n := [0 \dots 0]^\intercal$ 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 $\mathbf{e}_i$ 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* in the shared medium of an all-to-all communication network. 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 an internal clock, with the same speed, characterized by a *phase* variable $\theta_i(t)$ for each $i \in \{1, \dots, n\}$ that is defined as:

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

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 remainder of the division defined as $a \mod b = a - \lfloor a/b \rfloor b$ and $\lfloor a/b \rfloor$ rounds to the greatest integer that is smaller or equal than a/b. A pulse is emitted by node i each time the phase θ_i passes through zero following (1). The objective of a desynchronizing algorithm is to update the variable $\phi_i(t)$ such that their difference approaches $1/n$. Upon reception of the previous and next pulses (i.e., nodes use the offsets of only two neighbors in the update, meaning an interaction ring topology), agent i adjusts its phase offset ϕ_i according to some update equation.

In [17], it was shown that phase offsets for all nodes are characterized as the minimizers of a quadratic function as follows:

$$\text{minimize}_\phi \; g(\phi) := \frac{1}{2}\|D\phi - \frac{\mathbf{1}_n}{n} + \mathbf{e}_n\|_2^2 \tag{2}$$

where $\mathbf{1}_n$ is the vector of ones, $\mathbf{e}_n = (0, 0, \dots, 0, 1)^\intercal$, 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 [29], it is shown that the Nesterov method, that iteratively solves optimization problems such as the one in (2), given by the equations:

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

for a gradient step size β and a momentum term γ can be expressed as:

$$x^{(k+1)} = (A + BQC)x^{(k)} + BD^\intercal \mathbf{e}_n \tag{3}$$

where $x := \left[z^\intercal \; \xi^\intercal\right]^\intercal$, $Q = D^\intercal D$ and:

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

Remark that in [17] was shown that the PCO-based algorithm corresponds to a gradient descent and the Nesterov method improves its performance. Also, the above reformulation used the fact that $\nabla g(\phi) = D^\intercal D\phi + D^\intercal \mathbf{e}_n$ and that the system in (3) 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 (3) is of the form:

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

where T is given by:

$$T = \begin{bmatrix} (1+\gamma)(I-\beta Q) & -\gamma(I-\beta Q) \\ I & 0 \end{bmatrix}. \quad (4)$$

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 [29] 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/Fault Model: In this letter, we assume that the attacker is able to corrupt (3) by adding a signal $a^{(k)} \in \mathbb{R}^n$ that is nonzero in the entries corresponding the set of attacked nodes $\mathcal{A}$. Therefore, corrupted version of the optimal fixed-parameter Nesterov is:

$$x^{(k+1)} = (A + BQC)x^{(k)} + BD^{\intercal}\mathsf{e}_n + a^{(k)}. \quad (5)$$

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. 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 without posing additional defense characteristics in the nodes, e.g., the possibility of coordinated change to a different channel. In this letter, 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 (3) 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 $\mathrm{Var}(x_i^{(k)}) := \frac{1}{k+1}\sum_{\tau=0}^{k}(x_i^{(\tau)} - \mu_i)^2$ 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 1 (Properties of T): Consider matrix T defined in (4). Then, the following hold:

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

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

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

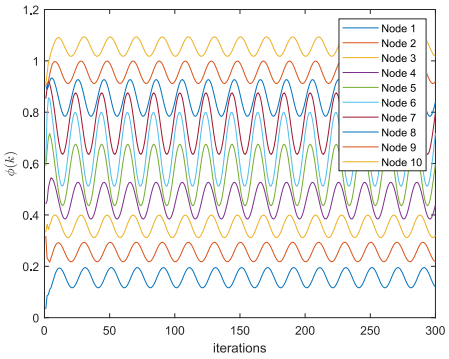

Fig. 1. Evolution of a 10 node network states when a cosine signal is injected in node 6 and using the optimal fixed-parameter Nesterov.

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. ■

The proof of Proposition 1 is now presented.

Proof: Let us first rewrite (5) 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 (3) 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 1, $\alpha^{(1)} = T a^{(\tau)}$ 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 cosine function. Since there are no defense mechanism, the sample variance of each $x_j^{(k)}$ is sorted as stated by Proposition 1. In contrast, Fig. 2 is the typical behavior of the desynchronization algorithm in the absence of an attack.

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

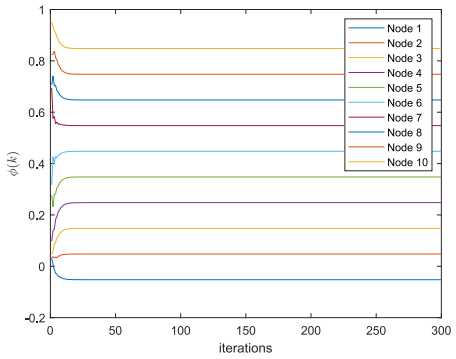

Fig. 2. Evolution of a 10 node network states when there is no attack using the optimal fixed-parameter Nesterov.

Theorem 1 (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 (5). 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 $k > \mathcal{T}$:

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

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 for other problems apart the 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 (which are also referred to as attackers) 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. This is introduced since it provides the necessary intuition for the distributed version, which is the main contribution of this letter. Notice that these algorithms resort to the results in Theorem 1 so the magnitude of the attacker signal has to converge to zero.

Algorithm 1 Centralized Detector

```
/* Initialize vectors */
v^(0) = 0_n, μ^(0) = 0_n
for each k > 0 do
    /* Update average and variance */
    Update v^(k) and μ^(k) using (6)
    /* Label an attacker */
    i* = arg max_i v_i(k)
    /* Signal neighbors to freeze update */
    Update neighbors using (7)
end for
```

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:

$$\begin{aligned} v^{(k)} &= v^{(k-1)} + \left(x^{(k)} - \mu^{(k-1)}\right)\left(x^{(k)} - \mu^{(k)}\right) \\ \mu^{(k)} &= \mu^{(k-1)} + \frac{1}{k}\left(x^{(k)} - \mu^{(k-1)}\right) \end{aligned} \tag{6}$$

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.,

$$x_{\text{prev}}^{(k+1)} = x_{\text{prev}}^{(k)}, \qquad x_{\text{next}}^{(k+1)} = x_{\text{next}}^{(k)} \tag{7}$$

The algorithm described is summarized in Algorithm 1.

We remark that the above algorithm implements the detection proposed by Theorem 1. Multiple attacked nodes can be stopped by selecting more than one node in the arg max operation of Algorithm 1.

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:

$$\begin{aligned} z_i^{(k)} &= z_i^{(k-1)} + \left(x_{[i-1:i+1]}^{(k)} - v_i^{(k-1)}\right)\left(x_{[i-1:i+1]}^{(k)} - v_i^{(k)}\right) \\ v_i^{(k)} &= v_i^{(k-1)} + \frac{1}{k}\left(x_{[i-1:i+1]}^{(k)} - v_i^{(k-1)}\right) \end{aligned} \tag{8}$$

where $z_i, v_i \in \mathbb{R}^3$ 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 $x^{(k)}$.

Algorithm 2 Decentralized Detector

```
/* Initialize local vectors */
v_i^(0) = 0_n, μ_i^(0) = 0_n, max_vars_i = 1
for each k > 0 do
    /* Algorithm run by each node */
    for each i do
        /* Update average and variance */
        Update v^(k) and μ^(k) using (8)
        if k mod round = 0 then
            /* Label the attacker */
            ψ_i^ℓ = ψ_i^r = arg max z_i^(k)
            Send ψ_i^ℓ and ψ_i^r
            if frozen & ψ_i^ℓ < max_vars_i then
                Resume using (5)
            end if
        else
            /* Send received values */
            Send ψ_{i+1}^ℓ and ψ_{i-1}^r
            if ψ_{i+1}^ℓ = ψ_{i-1}^r then
                /* Node i freezes update */
                Update using (7)
                max_vars_i = ψ_{i+1}^ℓ
            end if
        end if
    end for
end for
```

Since there is no centralized detector, nodes perform a voting scheme that takes 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 by using (7). The algorithm described is summarized in Algorithm 2.

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 [30]. The first setup considered the case where node 6 introduces random noise.

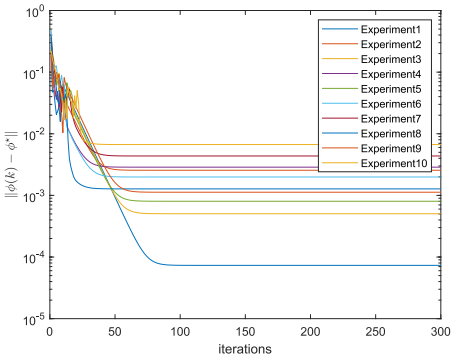

Fig. 3. Logarithmic evolution of the error for a 10 node network states when a random noise is introduced in node 6 and using the resilient optimal fixed-parameter Nesterov.

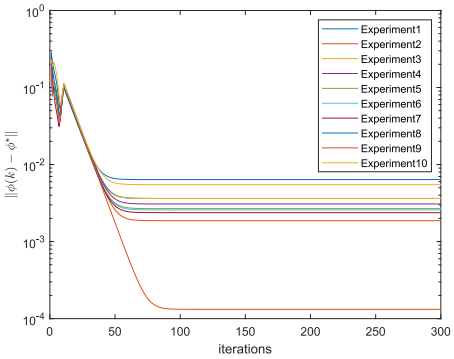

Fig. 4. Logarithmic evolution of the error for a 10 node network states when a cosine noise is introduced in node 6 and using the resilient optimal fixed-parameter Nesterov.

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, but at some point, the decision remains constant and the algorithm desynchronizes the remaining nodes. The non-vanishing residual error 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. 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 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.

VI. Discussion and Future Work

In this letter, we addressed the desynchronization problem in the presence of an attacker 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. The need for a low computational complexity precluded the use of techniques based on reachability analysis or through sharing of public/private keys and encryption.

In this letter, it is first shown that the Nesterov method attenuates the variance of nodes that are further away from the injected signal in the ring. It is proposed a variance-based method followed by a voting 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) application of this type of procedures to consensus algorithms since the attenuation of variance is caused by the averaging property of the Nesterov method, opening the possibility to consider network topologies that are not f-robust; ii) if each transmitter can estimate which node has the maximum variance based solely on the local data and without additional information exchange would lead to a direct implementation in the state-of-the-art standard.

References

[1] A. Tinka, T. Watteyne, and K. Pister, "A decentralized scheduling algorithm for time synchronized channel hopping," in *Ad Hoc Networks* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), J. Zheng, D. Simplot-Ryl, and V. C. M. Leung, Eds. Berlin, Germany: Springer, 2010, pp. 201–216. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-17994-5_14#citeas

[2] 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 J.*, vol. 14, no. 2, pp. 482–489, Feb. 2014.

[3] R. Pagliari and A. Scaglione, "Scalable network synchronization with pulse-coupled oscillators," *IEEE Trans. Mobile Comput.*, vol. 10, no. 3, pp. 392–405, Mar. 2011.

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

[5] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer*, IEEE Standard 802.15.4e-2012, Apr. 2012.

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

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

[8] R. Leidenfrost and W. Elmenreich, "Firefly clock synchronization in an 802.15.4 wireless network," *EURASIP J. Embedded Syst.*, vol. 2009, no. 1, Jul. 2009, Art. no. 186406. [Online]. Available: https://jes-eurasipjournals.springeropen.com/articles/10.1155/2009/186406#citeas

[9] 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.

[10] Y.-W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 5, pp. 1085–1099, May 2005.

[11] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1645–1662, 1990.

[12] R. Pagliari, Y.-W. P. Hong, and A. Scaglione, "Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 4, pp. 564–575, May 2010.

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

[14] A. V. Proskurnikov and M. Cao, "Synchronization of pulse-coupled oscillators and clocks under minimal connectivity assumptions," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5873–5879, Nov. 2017.

[15] J. Klinglmayr, C. Bettstetter, M. Timme, and C. Kirst, "Convergence of self-organizing pulse-coupled oscillator synchronization in dynamic networks," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1606–1619, Apr. 2017.

[16] H. Gao and Y. Wang, "On the global synchronization of pulse-coupled oscillators interacting on chain and directed tree graphs," *Automatica*, vol. 104, pp. 196–206, Jun. 2019.

[17] N. Deligiannis, J. F. C. Mota, G. Smart, and Y. Andreopoulos, "Fast desynchronization for decentralized multichannel medium access control," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3336–3349, Sep. 2015.

[18] D. Silvestre, J. Hespanha, and C. Silvestre, "Desynchronization for decentralized medium access control based on gauss-seidel iterations," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 4049–4054.

[19] D. Silvestre, J. P. Hespanha, and C. Silvestre, "Broadcast and gossip stochastic average consensus algorithms in directed topologies," *IEEE Trans. Control Netw. Syst.*, vol. 6, no. 2, pp. 474–486, Jun. 2019.

[20] A. Haseltalab and M. Akar, "Convergence rate analysis of a fault-tolerant distributed consensus algorithm," in *Proc. 54th IEEE Conf. Decis. 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," *Syst. Control Lett.*, vol. 79, pp. 23–29, May 2015.

[22] Y. Kikuya, S. M. Dibaji, and H. Ishii, "Fault tolerant clock synchronization over unreliable channels in wireless sensor networks," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1551–1562, Dec. 2018.

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

[24] S. M. Dibaji, H. Ishii, and R. Tempo, "Resilient randomized quantized consensus," *IEEE Trans. Autom. Control*, vol. 63, no. 8, pp. 2508–2522, Aug. 2018.

[25] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1063–1076, Mar. 2019.

[26] 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 *Proc. 52nd IEEE Conf. Decis. Control*, Dec. 2013, pp. 4373–4378.

[27] D. Silvestre, P. Rosa, J. P. Hespanha, and C. Silvestre, "Finite-time average consensus in a Byzantine environment using set-valued observers," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 3023–3028.

[28] D. Silvestre, P. Rosa, J. P. Hespanha, and C. Silvestre, "Stochastic and deterministic fault detection for randomized gossip algorithms," *Automatica*, vol. 78, pp. 46–60, Apr. 2017.

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

[30] D. Silvestre, "Optool—An optimization toolbox for iterative algorithms," *SoftwareX*, vol. 11, Jan.–Jun. 2020, Art. no. 100371.