Resilient Desynchronization for Decentralized Medium Access Control

D. Silvestre, J. Hespanha and C. Silvestre

2020 Conference on Decision and Control
Jeju Island, Korea

December 14-18 2020

Outline

Motivation

- Wireless Sensor Networks (WSNs) - a network composed of nodes using a wireless medium in Time Division Multiple Access (TDMA).
- No centralized infrastructure - implies the need for a decentralized algorithm to perform desynchronization of transmissions.
- Applicable to surveillance - a group of vigilant robots that want to periodically visit sites to be monitored.

Traditional Solution

- A WSN can run Time-Synchronized Channel Hoping (TSCH) protocol established in IEEE 802.15.4e-2012 standard [1].
- Solution is inspired in biological agents modeled as Pulse-Coupled Oscillators (PCOs). In a sense similar to how fireflies adjust their firing rate depending on other fireflies.
- In [2], the desynchronization is performed using the Nesterov method applied to an optimization formulation.

Intuition behind PCOs

- Assume an internal clock of each node that broadcast a pulse whenever its phase θ_i reaches 1, i.e., every T time units.
- Each nodes in the ring network adjusts its phase offset ϕ_i attempting to desynchronize from the others.
- Phase offsets are changed in a consensus-like [3] iteration from the offsets of the two neighbors.
- $\theta_i(t) = \frac{t}{T} + \phi_i(t) \mod 1,$

Intuition behind PCOs

- Assume an internal clock of each node that broadcast a pulse whenever its phase θ_i reaches 1, i.e., every T time units.
- Each nodes in the ring network adjusts its phase offset ϕ_i attempting to desynchronize from the others.
- Phase offsets are changed in a consensus-like [3] iteration from the offsets of the two neighbors.
- $\phi_i = \phi_{i-1} + \frac{T}{n}$

Intuition behind PCOs

- Assume an internal clock of each node that broadcast a pulse whenever its phase θ_i reaches 1, i.e., every T time units.
- Each nodes in the ring network adjusts its phase offset ϕ_i attempting to desynchronize from the others.
- Phase offsets are changed in a consensus-like [3] iteration from the offsets of the two neighbors.
- $\phi_i^{(k)} = (1-\alpha)\phi_i^{(k-1)} + \frac{\alpha}{2}\left(\phi_{i-1}^{(k-1)} + \phi_{i+1}^{(k-1)}\right)$

Optimization formulation

- A desynchronization state is a minimizer of the function:

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

- Matrix D represents the network. Example for 4 nodes:

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 \end{bmatrix}$$

- Nesterov method becomes:

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

Problem Statement

- Attacker model:

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

- where matrices A, B and C implement the Nesterov method, $Q = D^{\intercal}D$ and $a^{(k)}$ is the attacker signal.

Resilient Desynchronization Problem

Can we devise a lightweight technique to detect the presence of an attacker?

- Yes, by exploiting some properties of the Nesterov method.

Possible Solutions in the Literature

MSR algorithm Discard f largest and smallest neighbor values

- Not possible since number of neighbors is 2, so $f = 1$ removes all neighbors.

Fault Detection Employ distributed fault detection like using a bank of Kalman Filters or Set-Valued Observers [4].

- Adds additional communication overhead and computational complexity.

Properties of the Desync Nesterov algorithm

- If one injects a signal in node i then:
  - $\mathrm{Var}(x_i^{(k)}) > \mathrm{Var}(x_{i+1}^{(k)}) > \cdots > \mathrm{Var}(x_n^{(k)})$;
  - $\mathrm{Var}(x_i^{(k)}) > \mathrm{Var}(x_{i-1}^{(k)}) > \cdots > \mathrm{Var}(x_1^{(k)})$;

for sample variance $\mathrm{Var}(x_i^{(k)}) := \frac{1}{k+1} \sum_{\tau=0}^{k} \left(x_i^{(\tau)} - \mu_i \right)^2$.

- This is due to the properties of the transition matrix T of the algorithm satisfying:
  1. $T 1_{2n} = 1_{2n}$;
  2. $|T_{ij}| < 1$.

Centralized Resilient Desync Nesterov algorithm

Steps:

1. The central node computes the average and sample variance of all nodes using:
   - $v^{(k)} = v^{(k-1)} + \left(x^{(k)} - \mu^{(k-1)}\right)\left(x^{(k)} - \mu^{(k)}\right);$
   - $\mu^{(k)} = \mu^{(k-1)} + \frac{1}{k}\left(x^{(k)} - \mu^{(k-1)}\right);$
2. Label an attacker:
   - $i^\star = \arg\max_i v_i(k);$
3. Nodes with previous and next phase values do:
   - $x_{\text{prev}}^{(k+1)} = x_{\text{prev}}^{(k)}, \qquad x_{\text{next}}^{(k+1)} = x_{\text{next}}^{(k)};$

Distributed Resilient Desync Nesterov algorithm

Steps:

1. Node i keeps the average and variance for the immediate neighbors
2. Label an attacker after a voting scheme:
   - $i^\star = \arg\max_i z_i(k)$;
3. Nodes with previous and next phase values do:
   - $x_{\text{prev}}^{(k+1)} = x_{\text{prev}}^{(k)}, \qquad x_{\text{next}}^{(k+1)} = x_{\text{next}}^{(k)}$;

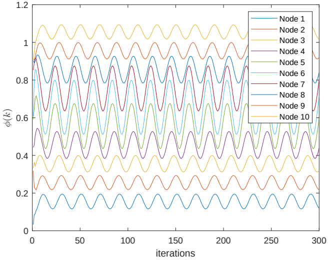
Main Result

An undetected attack signal $\alpha^{(k)}$ must have sample variance bounded by a sequence converging to zero.

Simulation Results (1/2)

Setup: A 10-node network running the Desync Nesterov algorithm with node i subject to a faulty signal.

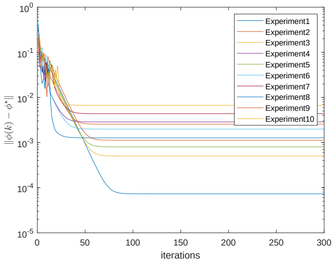
- Inserting a sinusoidal signal prevents convergence.
- The amplitude decreases as we move further away from the corrupted node.
- This property enables the proposed resilient algorithm.



Simulation Results (2/2)

Setup: A 10-node network running the Resilient Desync Nesterov algorithm with node i subject to a faulty signal.

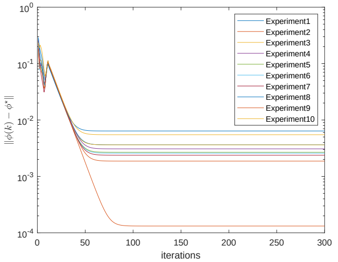
- Corrupting node i with a uniform random signal;
- The error at first is not monotonic;
- The stopping of neighbors updates leads the remaining nodes to converge.



Simulation Results (2/2)

Setup: A 10-node network running the Resilient Desync Nesterov algorithm with node i subject to a faulty signal.

- Corrupting node i with a sinusoidal signal;
- The behavior is clearer;
- Without a correction mechanism there is a residual error to the optimal desynchronization state.



Concluding Remarks

- We have shown theoretical properties of the Nesterov method when applied to the Desynchronization problem.
- As a consequence, variance is larger in nodes close to the attacked one.
- We present both a centralized and distributed version based on these theoretical results.
- Undetected attacks are characterized by signals with bounded variance by a sequence converging to zero.
- Additional correction mechanisms are needed if we want to have optimal desynchronization.

References

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*, pp. 1–225, 2012.

D. Silvestre, J. Hespanha, and C. Silvestre, "Desynchronization for decentralized medium access control based on gauss-seidel iterations," in *2019 American Control Conference (ACC)*, Jul. 2019, pp. 4049–4054. DOI: 10.23919/ACC.2019.8814471.

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*, vol. 6, no. 2, pp. 474–486, Jun. 2019, ISSN: 2325-5870. DOI: 10.1109/TCNS.2018.2839341.

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, 2017, ISSN: 0005-1098. DOI: http://doi.org/10.1016/j.automatica.2016.12.011.

The end

- Thank you for your time.

Resilient Desynchronization for Decentralized Medium Access Control

D. Silvestre, J. Hespanha and C. Silvestre

2020 Conference on Decision and Control
Jeju Island, Korea

December 14-18 2020