C. Alcaraz, "Cloud-Assisted Dynamic Resilience for Cyber-Physical Control Systems", *IEEE Wireless Communications*, vol. 25, pp. 76-82, 2018.
http://doi.org/10.1109/MWC.2018.1700231
NICS Lab. Publications: https://www.nics.uma.es/publications

Cloud-Assisted Dynamic Resilience for Cyber-Physical Control Systems

Cristina Alcaraz, *Member IEEE*

Abstract.— We are increasingly witness to the enormous security problems that cyber-physical control systems have and their susceptibility to certain types of attacks. An attractive way to coordinate the situation and ensure resilience in lineal times could be through redundancy-based restoration mechanisms. For this reason, we present in this paper a network infrastructure based on three layers, where the redundant support is primarily concentrated on a fog-based structure to protect a specific subset of cyber-physical control devices. The specification of the context and the abstract construction of the approach include a set of conceptual theories related to structural controllability, power dominance, supernode and opinion dynamics, where the validation of the approach is subject to a thereotical and practical analysis based on two threat case studies.

Index Terms—Control systems, resilience, structural controllability, power dominance, critical infrastructure protection

1 INTRODUCTION

Automated self-healing mechanisms for cyber-physical control systems (CPCS) are nowadays a primary criterion [1], considered so by such relevant standards and recommendations such as NIST-7826 (vol 1) [2] or NIST-SP 800-82 [3]. However, the implementation of these mechanisms can be an extremely complex task. The criticality of the context and the performance restrictions [1-4] may hinder the integration and adaptation of new restoration approaches. Until now, most solutions have been based on tree-like structures [5-6] or on redundancy solutions [7-12], without exploring the existence of new technologies and infrastructures for the control, or the possibility of expanding their capacity to benefit the restoration processes. In this regard, one alternative to look at the problem from a new angle, is to decouple the control network itself from the existing protection mechanisms and integrate the redundancy measures outside the monitoring system, but close to the application context, for example in the fog nodes [13-14]. These devices work as cache structures responsible for the general management of the local context and transferring backup instances to the third cloud-based layer [14] as illustrated in Figure 1.

Conceptually speaking, this paper models a cloud-assisted control industry, based on a set of conceptual theories: (i) structural controllability given by Lin in [15], (ii) the power dominting set (PDS) problem originally introduced by Haynes et al. in [16] but later simplified by Kneis et al. in [17], and (iii) the supernode theory [18]. Structural controllability is a concept derived from the traditional control theory given by Kalman in [19], but one which enables the graphical formulation of large distributions through graph theory. This also means that our models are constructed according to a digraph $G(V, E)$ with loops where V represents the control devices, and E, the communication and control links. To characterize the dominance properties in V and the different roles of its elements (e.g. controlled nodes –sensors or actuators- or controllers/driver nodes), the following two control rules are applied [17], where N_D comprises the minimum set of driver nodes capable of injecting control signals to other devices. That is:

CR1: *a vertex in N_D controls itself and all its neighbors*. The resulting set therefore includes the minimum set of driver nodes (denoted here by D_S), corresponding to the traditional dominance problem, and known as the dominating set (DS).
CR2: *If a controlled vertex $v_i \in N_D$ with outdegree $d^+ \geq 2$ is adjacent to $d-1$ controlled vertices, the remaining un-controlled vertex v_j becomes controlled as well, such that $v_j \in N_D$*. The result of this rule, previously combined with **CR1**, holds N_D such that $D_S \leq N_D$. Note that although the PDS problem was originally specified for observability, its application in this paper principally concerns its dual problem related to controllability [20].

The concept of supernode is also contemplated to establish the redundant pathways from those `fog' nodes deployed in a secondary (and intermediary) network (see Figure 1). A supernode is a conceptual node acting as server or proxy with the capacity to offer peer-to-peer communication [18],

ideal for a context composed of multiple redundant routes from external devices. However, to determine when to activate the resilience processes from the fog-computing network, a local detection method of threats is additionally considered. In our case, the mechanism implemented follows a dynamic model of opinion dynamics [21] in which large data sets can be generated to detect topological changes in $G(V, E)$. To process the data in the whole set and determine the existence of a threat, data-mining techniques, such as kmeans and k-nearest neighbor (knn), are also handled within our restoration approach.

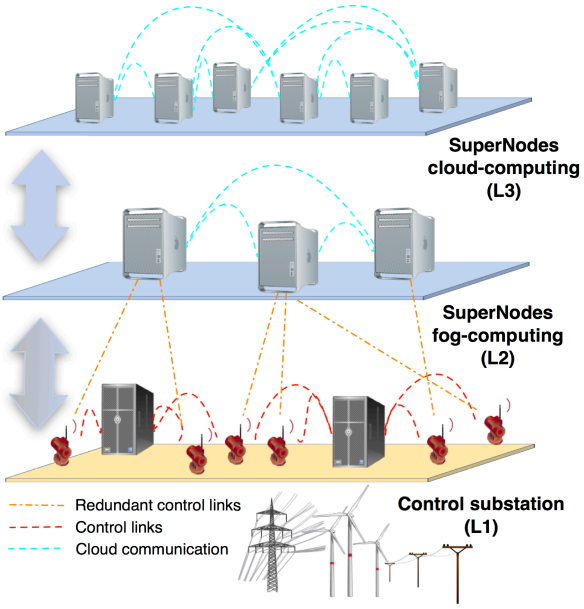

Figure 1. General architecture and control context

The paper is organized as follows. Section 2 specifies the control architecture and the detection and response components to threats defined in Section 3. This section also details the adversarial model and the general assumptions of the approach proposed in Section 4 together with its theoretical and practical validation. Section 5 concludes the paper and outlines future work.

2 Architecture and Dynamic Detection

The architecture proposed is a hierarchical scheme based on three layers (L1, L2 and L3) as depicted in Figure 1. One illustrates the behavior of a cyber-physical control subsystem (corresponding to the traditional control substations) with field devices such as remote terminal units, gateways, sensors and actuators (L1); whereas (ii) the two remaining networks comprise the logical functionalities of a cloud and fog-based system (L2 and L3, respectively). As indicated in the previous section, fog-computing can be considered as (private) a subpart of the cloud-like infrastructure, and comprises a few supernodes to locally manage the context and offer support and coverage in extreme situations.

The application scenarios are varied (e.g. power grids or water treatment control systems); and for all of them it is largely assumed that the communication channels are protected following a setting predefined and guided by a recognized standard such as IEC-62351-(1-8) [22], NIST-7628 [2] or NIST-SP 800-82 [3]. To manage the application context, the supernodes in fog-computing need to periodically receive information from the environment, and in particular from those drivers that hold the maximum control power ($\in N_D$) to reduce the system complexity and the communication overhead. With this information in hand, the supernodes in L2 can determine or predict the network structural degradation level (L1) and the moment to provisionally conduct the redundancy-based restauration processes. Moreover, as the general architecture contemplates a third layer with a broader vision of the entire control system, any other CPCS can be notified in advance of any suspicious change within a remote monitoring subsystem so as to prepare the response and address the threat.

For the local detection of topological deviations, we consider the technique of opinion dynamics [21] that help lead consensus among the diverse agents in N_D, by simply calculating an average of

weighted opinion dynamics in discrete time. That is, let $x_{n_d}(t)$ be the individual opinion of a driver node n_d in a context with n agents involved in the same opinon process at time t; and W, a weighted matrix of size $n\ x\ n$ containing the opinion of each n_d involved within the opinion process, therefore the opinion dynamic, at time t, can be computed as $x(t+1) = W(t, x(t))\, x(t)$. This also means that $\forall$ agent n_d in N_D), $x_{n_d}(t+1) = w_{i1}x_1(t) + w_{i2}x_2(t) + \cdots + w_{in}x_n(t)$ such that w_{ik} refers to the weighted influential opinion of n_{dk} in n_{di}; and $w_{ij} = 1$, the condition necessary to allow each agent to intercede with its own opinion at time t.

An example is illustrated in Figure 2, in which it is possible to visualize: (1) the pragmatic opinion of the entire network (left-hand figure) and (2) the simplified release produced by the subset of nodes $\in N_D$ (right-hand figure).

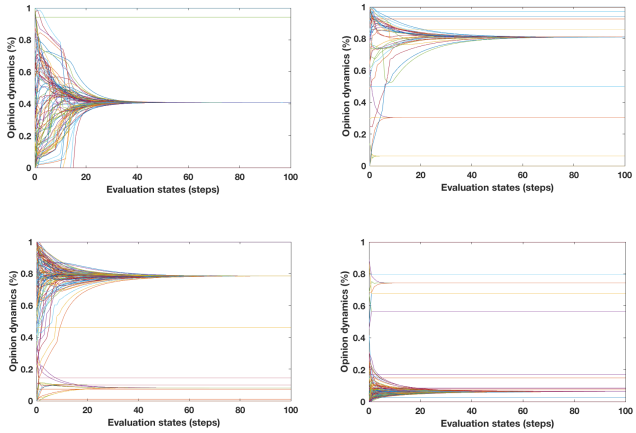

Figure 2. Opinion dynamics with 100 and 200 nodes in $\boldsymbol{G}$

The weights of W are frequently updated in each new state of the network by simply computing the edge betweeness centrality, whose value is defined as $E_{BC} = \sum_{s,t \in V} \frac{d(s,t|e)}{d(s,t)}$, such that $d(s,t)$ specifies the number of shortest paths between s and t, and $d(s,t|e)$ the number of routes passing through the link e. In this way, we are able to detect structural variations given that any topological change impacts on the shortest paths, through which the maximum control capacity flows [23].

3 Threat Model and Assumptions

The threat scenario is imagined for open control substations where attackers can penetrate alone, and modify the network topology to cause, for example, denial of service. In this respect, two main attack strategies can be applied:
[STG1]: produces Byzantine faults in which a few random edges are removed from the network and in particular, arbitrarily from a few nodes, so as to be immediately restored. In this way, we simulate an isolated anomalous behavior that can arise from an unprovoked fault in the system or a strategic attack. Within this category, a case is considered where attackers can very subtly, persistently target a particular node in order to isolate it (removing all of the edges) from the network.

[STG2]: addresses random δ combined attacks of types random and target such that $\delta \leq |V|$. For the target attacks, we consider the removing of a few random edges or the isolation of those nodes with the highest degree (known as the hubs), and those nodes with the highest strength ($\Sigma_{e_i \in E} E_{BC}(e_i)$) within the network. Nonetheless, it is also important to underline that we also explore (in experimentation phase) a variant of this strategy (STG2-1) so as to assess the behaviour of the system if the target nodes are precisely the observed nodes. To the contrary, the random attack consists in randomly invalidating edges or nodes.

With regard to general assumptions, we must comply with the control requirements and in such a way that restoration measures do not entail a modification in the two control rules, CR1 and CR2. To fulfil this first aim, we specify the three following redundancy principles:

[RP1]: to reduce implementation and maintenance costs in L1, the number of redundant links should

be bounded to those nodes that are not part of the set N_D., i.e. the set of controlled nodes, C, where $C \leftarrow V \,/\, N_D$.

[RP2]: the redundancy is concentrated in those supernodes, S, in L2 such that this set is, in turn, part of $\mathrm{N_D}$ ($\mathrm{S} \subset \mathrm{N_D}$). The reason for this condition is so a supernode in L2 is able to retake control of a sensor or actuator, at least, provisionally.

[RP3]: the redundancy in S has to accomplish the two control rules, and concretely CR2 [23]: if there exists an uncontrolled node u_i ($\in C^- \leftarrow (V/N_D)/C$), then it is required to find a supernode $s \in N_D$ that satisfies: ($|C_s^-| \geq 1$ and $|N_D| \geq 0$) or ($|C_s^-| = 0$ and $|N_D| = 0$), where $C_s^- \leftarrow (\forall\, s_i \in V, (s, s_i) \in E) \,/\, ((\forall\, s_i \in V, (s, s_i) \in E) \cap N_D)$. Note that C^- embodies the set of unobserved and uncontrolled devices by, at least, a driver node in N_D.

Lastly, the implementation should not modify the power-law structure of the underlying infrastructure either, given that most of critical infrastructures and their monitoring systems obey distributions of kind $y \propto x^{-\alpha}$ [16]. For this configuration and for the opinion dynamics, it is further assumed that the information gathered from L1 is completely trustworthy where the driver nodes are trusted entities.

4 SUPERNODE-BASED RESILIENCE AND CONTEXT

In this section, an automated resilience mechanism located in the fog-computing (L2) is proposed, the activation of which relies on: (i) the opinion dynamic received from the context in L2, and (ii) the clustering and classification of the opinion dynamic at time t. The data-mining techniques applied for the anomaly detection in the whole of L1 are based on kmeans and knn, useful for processing and analyzing large data sets.

4.1 Commissioning Phase and Redundancy

Considering the assumptions of Section 4, any node deployed in the network (L1) needs to comply with, at least: the two rules of control, the power-law conditions, and the three redundancy principles RP1-2-3. However, the construction of a redundancy-based system also implies redesigning the original versions of CR1 and CR2 [24] to consider, from the commissioning phase, all the redundant pathways required from the supernodes in L2 [PR2].

Algorithm 1 outlines the new pseudo-code of CR1-2 and comprises: (i) the extension of $\mathrm{G(V, E)}$ in $\mathrm{G(V', E)}$ to include the supernodes within V and facilitate the implementation and experimentation in Matlab of the whole process; and, on the other hand, (ii) the creation of a new network $\mathrm{G^r(V', E')}$ equivalent to $\mathrm{G(V, E)}$, but containing the supernodes and the redundant links thereby mapping the entire system and ensuring the compliance of CR1-2 from a global perspective. Therefore, $\mathrm{S} \subset \mathrm{N_D}$ and $\mathrm{S} \subset \mathrm{V'}$.

Algorithm 1. *CR1-2 with Redundancy in fog-computing*

Input: $\mathrm{G}(V, E)$, S
Output: $\mathrm{G}(V', E)$, $G^r(V', E')$, N_D
Local: $C \leftarrow \emptyset, N_D \leftarrow \emptyset, N_D^* \leftarrow \emptyset, G^r(V, E) \leftarrow \mathrm{G}(V, E)$

$N_D \leftarrow$ **CR1** as defined in [24] but including S
$N_D^* \leftarrow N_D$
$\mathrm{G}(V', E)$, $G^r(V', E)$, $V' \leftarrow$ extend G and G^r updating V' with the new supernodes in **L2**
while ($N_D^* \neq \emptyset$) **do** {/*Establish redundant pathways*/
  $n_d \leftarrow$ randomly select one candidate $\in N_D{}^*$
  $G^r(V', E') \leftarrow$ *search and connect supernodes*
    $(\mathrm{G}(V', E), G^r(V', E), n_d, S, N_D)$
  $N_D^* \leftarrow N_D^* \,/\, n_d$
}
$N_D \leftarrow$ **CR2** as defined in [24], but ∀ uncontrolled node $\mathrm{u_i} \in C^-$ when executing the algorithm **CR2** of [24] has to be linked with a supernode in S such that Algorithm 2 is newly invoked and

$G^r(V', E')$ is upgraded.

return $\mathrm{G}(V', E)$, $G^r(V', E')$

Algorithm 2, in contrast, delimits the redundancy in those nodes in S (by complying with RP2-3) to avoid, as much as possible, any new increase of driver nodes, especially, when Algorithm CR2 (also sketched in [24]) is applied.

Algorithm 2. *Search and connect supernodes*

Input: $\mathrm{G}(V', E)$, $G^r(V', E)$, n_d, S, N_D
Output: $G^r(V', E')$
Local: $S' \leftarrow S$, $G^r(V', E) \leftarrow \mathrm{G}(V', E)$, $S^* \leftarrow \emptyset$

while $(S' \neq \emptyset)$ **do** {
  $s \leftarrow$ randomly select one candidate $\in S'$
  $D \leftarrow (\forall\, s_i \in V', (s, s_i) \in E'$ in $\mathrm{G}^r) \cap N_D$ /*driver children*/
  $C \leftarrow (\forall\, s_i \in V', (s, s_i) \in E'$ in $\mathrm{G}^r) \,/\, D$ /*controlled children*/
  if (**RP3** is met in relation to D and C)$\{S^* \leftarrow S^* \cup s\}$
  $S' \leftarrow S' \,/\, \mathrm{s}$
}
if $(S^* \neq \emptyset)\{s \leftarrow$ randomly select one candidate $\in S^*\}$
else$\{s \leftarrow$ randomly select one candidate $\in S\}$
$G^r(V', E') \leftarrow$ update E by connecting $(s, c_i) \in E'\ \forall\, c_i \in C$ such that $C \leftarrow (\forall\, c_i \in V', (n_d, c_i) \in E'$ in $\mathrm{G}^r) \,/\, D$, being $D \leftarrow (\forall\, c_i \in V', (n_d, c_i) \in E'$ in $\mathrm{G}^r) \cap N_D$

return $G^r(V', E')$

4.2 Restoration, Correctness and Complexity

The procedure to repair the control through redundant pathways is proposed in Algorithm 3, the technique of which focuses on four specific scenarios:

Optimal* scenario: tries to find an existing father driver $\mathrm{n_d} \in \mathrm{N_D}$ with the ability to offer coverage to a determined uncontrolled node $\mathrm{u_i}$ in $\mathrm{C^-}$, thereby offering full linear response $O(n)$.

Optimal scenario: aims to search a supernode predefined from the commissioning phase, with the capability to give coverage to a determined $\mathrm{u_i}$ in $\mathrm{C^-}$. That is, if there is no driver node $\mathrm{n_d} \in \mathrm{N_D}$ such that $(\mathrm{n_d}, \mathrm{u_i}) \in \mathrm{E}$, then it is necessary to find a supernode s in S that guarantees $(\mathrm{s}, \mathrm{u_i}) \in \mathrm{E'}$ in $\mathrm{G^r}$. As the redundant edge is implicit to $\mathrm{G^r}$, the complexity of its activation in $\mathrm{G(V', E')}$ becomes lineal, $\mathrm{O(n)}$.

Suboptimal and non-optimal scenario: if there is no suitable supernode in L2 to provisionally offer control to $\mathrm{u_i}$ in $\mathrm{C^-}$, the system has to pursue a candidate in S that achieves at least RP3 (suboptimal scenario). But even so, if the system does not find an appropriate candidate, it forces the link establishment by randomly choosing a supernode in S (non-optimal scenario). Either way, the computational cost remains lineal in both scenarios but with the disadvantage of existing in the worst and non-optimal scenario, and an increase of resources in $\mathrm{E'}$, probably affecting CR2.

The algorithm is also able to distinguish the nodes that are part of the system from those that have decided to (temporarily or definitively) leave the network through $\mathrm{L_v}$, thereby favoring the mobility in the field (e.g. robots). This also means that any new incoming node in the system (L1) can require executing Algorithm 2 and the updating of CR2 from [24] to, at least, fulfil the redundancy principles (RP1-2-3) and the second dominance rule (CR2). As for the automatization of Algorithm 3, it depends largely on a predictor of class labels based on knn, the value of which also relies on the opinion dynamic average received from driver nodes in $\mathrm{N_D}$ at time t. As the opinion dynamic ranges between 0 and 1, any test value greater than zero underlines a true positive, and therefore a clear, abnormal situation.

Algorithm 3. *Restoration mechanism*

Input: $G(V', E)$, $G^r(V', E')$, N_D, S, L_v
Output: $G(V', E')$
Local: $C \leftarrow V' / N_D$

while $(C \neq \emptyset)$ **do** {
  $c \leftarrow$ randomly select one candidate $\in C$
  if $(c \notin Lv)$ **and** $((((\forall v_i \in V', (v_i, c) \in E \text{ in } G) \cap N_D) / S) = \emptyset)$ {
    $S_r \leftarrow ((\forall s_i \in V' \text{ and } S, (s_i, c) \in E' \text{ in } G^r) \cap N_D) \cap S$
    if $(S_r \neq \emptyset)$ {/*__optimal scenario__*/
      $s \leftarrow$ randomly select one candidate $\in S_r$
      $G(V', E') \leftarrow (s, c) \in E'$
    }**else**{/*__suboptimal scenario__*/
      $S^* \leftarrow S$; found $\leftarrow$ **false**
      while $(S^* \neq \emptyset)$ and **not**(found) **do**{
        $s \leftarrow$ randomly select one candidate $\in S^*$
          $D \leftarrow (\forall s_i \in V', (s, s_i) \in E' \text{ in } G^r) \cap N_D$
        $C^* \leftarrow (\forall s_i \in V', (s, s_i) \in E' \text{ in } G^r) / D$
        if (**RP3** is met in relation to D and C^*){
          $G(V', E') \leftarrow (s, c) \in E'$
          $G^r\left(V', E'\right) \leftarrow (s, c) \in E'$
          found $\leftarrow$ **true**
        }
        $S^* \leftarrow S$ / s
      }
      if not(found){/*__non-optimal scenario__*/
        $s \leftarrow$ randomly select one candidate $\in S$
        $G(V', E') \leftarrow (s, c) \in E'$
        $G^r\left(V', E'\right) \leftarrow (s, c) \in E'$
      }
    }
  }
  $C \leftarrow C$ / c
}
return $G(V', E')$

The correctness of Algorithm 3 can be shown if the following prerequisites are met:

- the algorithm (locally or remotely) retakes the control without infringing, as far as possible, CR1 and CR2 (resilience and control);
- the algorithm finishes in a finite time (termination); and
- the algorithm ceases in a finite time and futher ensures control at any time (validity).

The former condition is satisfied by default. The restoration is reached by an existing subnet of nodes with redundant routes in S, complying in addition with CR1 and CR2 from the commissioning phase. However, and apart from this, the approach always first tries to find those father drivers with the maximum capacity to offer direct support in the field. Otherwise, the system requires: (i) exploring the existence of a new supernode in L1 that gives coverage, fulfiling, at least, RP3; and in the case of no success, (ii) provisionally force a remote connection from L2.

The termination can, in contrast, be proved through induction but previously considering the following statements: (i) the predictor of knn is greater than 0 and C^- may be empty (precondition); and (ii) $C^- = \emptyset$ for any controller in C, and in such a way that the control is always ensured, either in local or in remote (postcondition). Based on these two conditions, the induction can finally be demonstrated as follows:

Case 1: $\exists$ an uncontrolled node u_i in C^-. At this point, it is worth mentioning that Algorithm 3 previously runs the set of controlled nodes (denoted as C) to detect the existence of uncontrolled nodes (C^-) such that $(((\forall v_i \in V', (v_i, c) \in E \text{ in } G) \cap N_D) / S) = \emptyset$, and hence $c \in C^-$.

Given this, Algorithm 3 checks the existence of a father driver in N_D that may offer coverage to u_i such that $(((\forall v_i \in V', (v_i, u_i) \in E \text{ in } G) \cap N_D) \ / \ S) \neq \emptyset$. If this condition exists, then it is possible to confirm the existence of driver node N_D that can give full controllability in the field without activating redundancy mechanisms. In these circumstances, C^- is updated as follows: $C^- \leftarrow C^- / u_i$ such that $C \leftarrow C \cup u_i$, and the postcondition is finally reached (Optimal* scenario). If not, then three further situations can occur:

Optimal scenario: $\exists$ a s_r in S where $(s_r, u_i) \in E'$, and $G(V', E')$ is updated in such a way that $C^- \leftarrow C^- / u_i$ and therefore $C \leftarrow C \cup u_i$ making the post-condition feasible.

Suboptimal scenario: $\exists$ a s_r in S where $(s_r, u_i) \in E'$, and $G(V', E')$ is updated in such a way that $C^- \leftarrow C^- / u_i$ and therefore $C \leftarrow C \cup u_i$ and the post-condition is satisfied. At this point, it is worth mentioning that Algorithm 3 previously runs the set of controlled nodes (C) to detect the existence of uncontrolled nodes (C^-) such that $(((\forall v_i \in V', (v_i, c) \in E \text{ in } G) \cap N_D) \ / \ S) = \emptyset$, and hence $c \in C^-$.

Non-optimal scenario: to attend to extreme situations that do not contemplate the two previous scenarios, we consider the case where the system coerces the remote control by choosing a supernode $s \in S$. Evidently, this configuration may collide with CR2, but the criticality of the context and the underlying system push us to temporarily assume this risk and take this decision. In this situation, C^- is upgraged by establishing a link between s and u_i.

Induction: in step k ($k \geq 1$) of while of Algorithm 3 we assume that $|C^-| > 0$ due to the current state of a particular node c in C. At this point, one of the four scenarios defined in Case 1 can arise, reducing the cardinality of C^- and increasing the cardinality of C in each step. When $C = \emptyset$, Algorithm 3 terminates, showing the viability of the approach for general cases. The validity, corresponding to the third prerequisite, is also satisfied because the algorithm finishes in a finite time t and the control is always resumed, either from a father driver in N_D or from a supernode in S.

4.3 Practical Validation and Results

Two case studies have been designed to explore the behavior of the sytem and its optimization in threat situations, further taking into account the two threat strategies STG1 and STG2 together with the weak threat model specified in Section 3. For the experiments, the Matlab environment has been considered so as to simulate power-law networks with diverse dimensionality. In our case, the construction of the power-law structure is based on a PLOD distribution [25] since its method arbitrarily establishes outdegree values (d^+) to each node of the system according to $\beta x^{-\alpha}$. To conceptually characterize these control structures with respect to real CPCS, α has been computed with a small value, i.e. with 0.1, and for a number of nodes ranging from 10 to 350 with 5 additional nodes in L2.

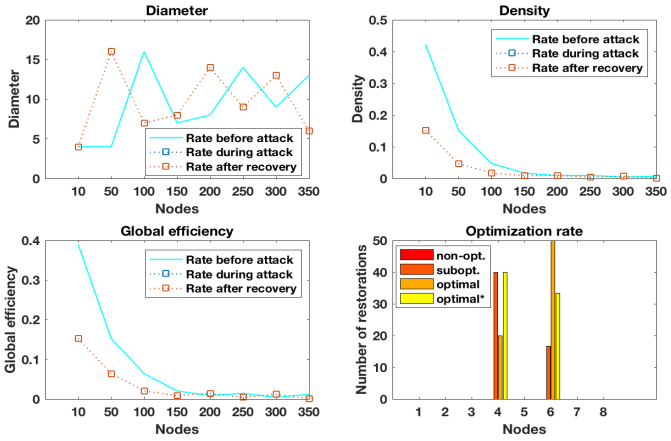

Figure 3. Network degradation and optimization in **STG1**

Figure 3 illustrates, through diameter, density and global efficiency (i.e. the average inverse shortest path length in $G(V', E')$, which is inversely related to the characteristic path length), the deterioration of the network after attacks of type STG1, as well as its recovery. To simplify the representation

associated with the resilience optimization, we show, in just one graph (i.e. in the fourth picture of Figure 3), the cases where the system requires the states of the controlled nodes to be verified and computes the repair processes, if and only if, they are necessary.

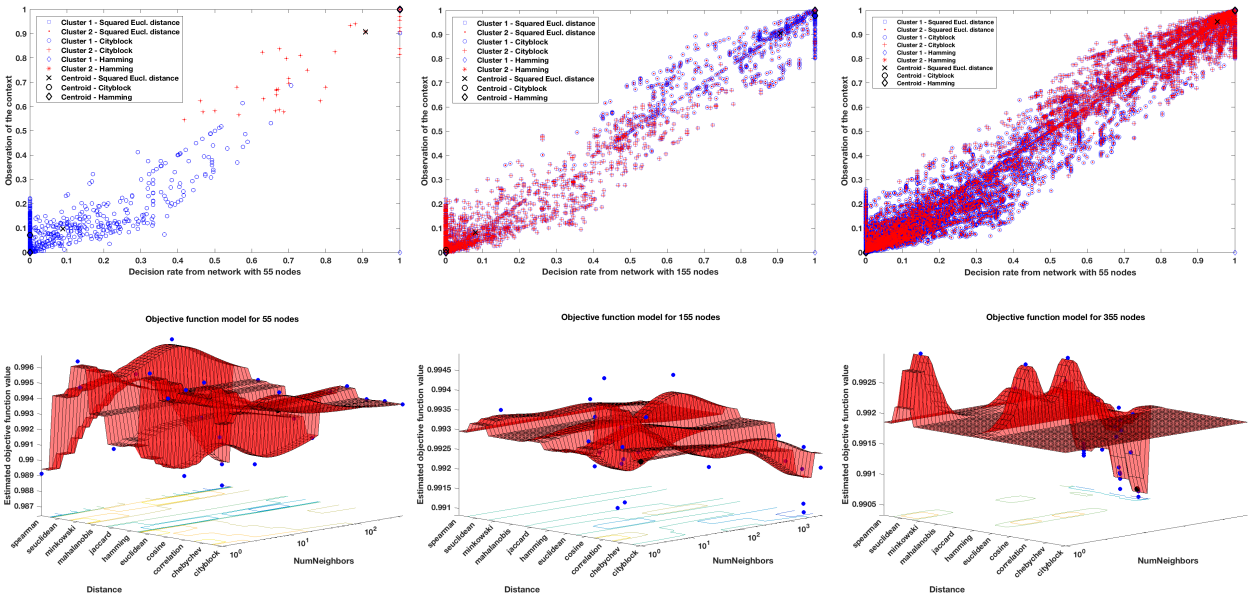

Figure 4. Kmeans and knn in STG1

In any case, the results clearly reveal the influence of Byzantine events in power-law contexts where the three
aforementioned statistical indicators (i.e. the diameter, the density and the global efficiency), suffer significant variations with respect to the original states. As for the restoration optimization, we observe the ease with which system self-repairs the control without requiring Algorithm 3 to be executed in full and in all the cases. This is mainly because adversaries attack target nodes belonging to N_D whose affected nodes can continue their operational tasks after the threat, and thanks in part to CR1 (the proof of this is later reflected in an experimental study for STG2-1). Moreover, the results indicate that the system is, in the worst cases, able to enter the optimal*, optimal and suboptimal solutions, offering local and remote controllability in lineal times.

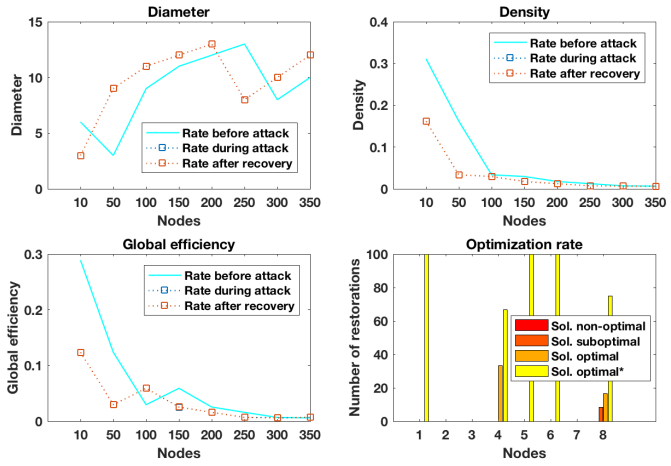

Figure 5. Network degradation and optimization in **STG-2**

With relation to Figure 3, Figure 4 illustrates the classification and the clustering of opinon dynamics received from observation network (L1). This classification mode, in turn, allows the real state of the entire system to be assessed, determining when to activate the control recovery mechanism. For example, through kmeans and knn it is possible to measure the general deviations between opinion

dynamics and detect or even predict the structural anomalies according to local information received from the main control agents ($\in N_D$). In our case, the repair processes are activated according to the predictor of class labels based on the seuclidean[1] metric provided by knn, and in such a way that if the prediction is greater than zero, the system is able to predict a possible opinion discrepancy. Nonetheless, it is also feasible to apply, as an anomaly indicator, the sums of point-to-centroid distances for a determined cluster or for all the clusters provided by the kmeans.

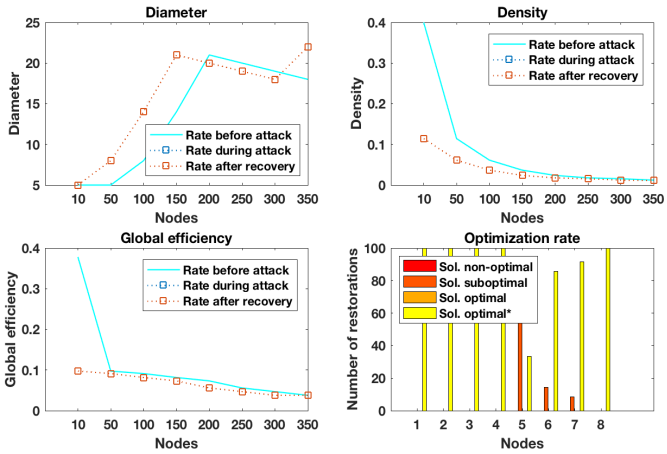

Figure 6. Network degradation and optimization in **STG2-1**

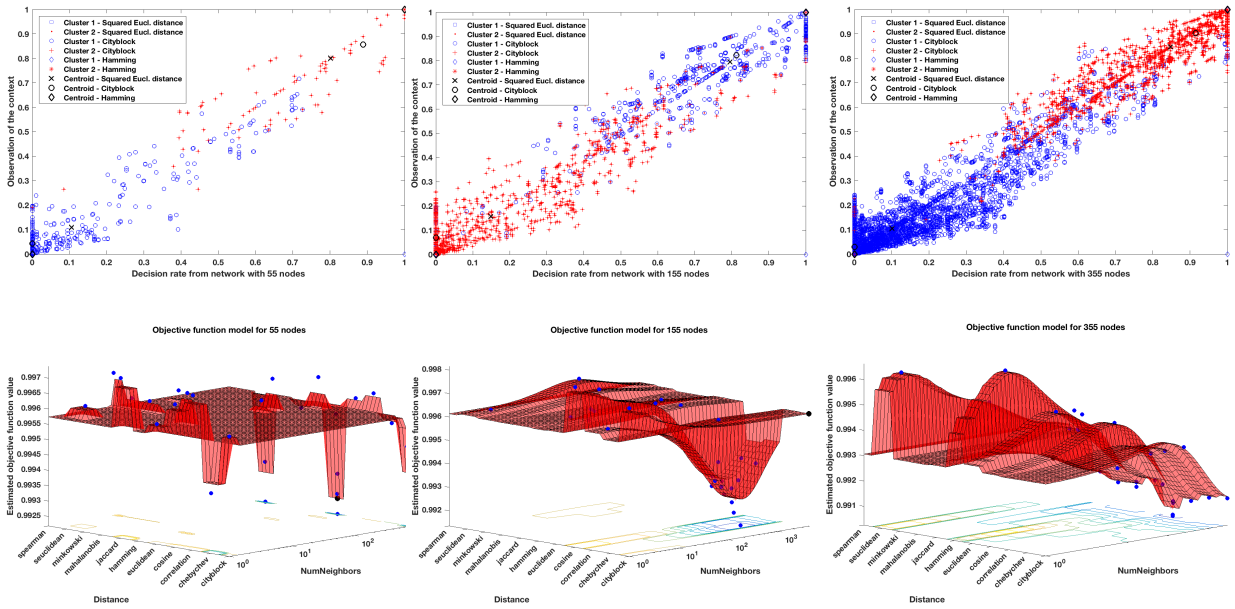

Figure 7. Kmeans and knn in STG2

Similarly, Figure 5 together with Figure 7 depict the malicious effects of attacks of the class STG2, manifesting once more the spoilage of the network and its optimal strength to respond to unforeseen events. As is true in Figure 3, the density, the diameter and global efficiency after the threat and after the repair, are not really significant given that most optimal restoration processes are based on local connections with father drivers or on redundant measures. Moreover, comparing the results of optimization in Figure 3 and Figure 5, we can observe that the effect of a Byzantine fault can be much more shocking from the point of view of the optimal response, than a target attack on the main controllers (e.g. the hubs or the nodes with the highest strength), probably because a Byzantine fault can be an unforeseen event, a dangerous fault or a persistent and advanced strategic threat.

On the other hand, the variant fo STG2, STG2-1, has also been simulated in which the system only

[1] Seuclidean: $d(x,c) = (x-c)(x-c)'$ and hamming: $d(x,y) = \frac{1}{p}\Sigma_{j=1}^{p} F\{x_i \neq y_i\}$.

targets those controlled nodes not included in N_D, and whose attacks aim to remove (i) a few random edges of arbitrary nodes, or definitively, (ii) isolate them. The proof of this experimentation is represented in Figure 6, in which we show not only the feasibility of the approach to be useful in critical scenarios but also for the ability of the sytem to find optimal* solutions and for all cases. Lastly, Figure 5, Figure 7 and Table 1 indicate that the best data-mining metrics to detect structural anomalies according to opinion dynamics are principally seuclidean and hamming. However, these distance metrics greatly depend on the size of the network, the number of control agents involved in the detection process and the decomposition of the network itself after a threat. The accuracy of data clustering in kmeans seems to be similar for the three metrics considered: seuclidean, hamming and cityblock2, except for cosine and correlation since their metrics do not permit some values of opinion dynamics ranging between 0 and 1 to be computed. Lastly, Figure 4 states that the dispersion of the data for a network with 355 nodes is not as clustered as the data represented in Figure 7 showing the effect of those attacks of class STG2. But even so, all the results keep approximate distances with regards to their centroids.

Table 1. Best observed and estimated distances with knn

Nodes	Threat strategy	Distance method	Best observed	Best estimated
55 (includes the 5 supernodes)	STG1	seuclidean	0.99471	0.99471
		hamming	0.98942	0.98967
		spearman	0.99643	0.98872
	STG2	seuclidean	0.99609	0.99609
		euclidean	0.99219	0.99227
155 (includes the 5 supernodes)	STG1	seuclidean	0.99348	0.99348
		hamming	0.9926	0.9926
		euclidean	0.99138	0.99147
		chebychev	0.99083	0.99146
	STG2	seuclidean	0.99626	0.99626
		hamming	0.99614	0.99646
		cosine	0.99255	0.99265
		correlation	0.99135	0.99242
355 (includes the 5 supernodes)	STG1	seuclidean	0.99299	0.99299
		hamming	0.99213	0.99255
		mahalanobis	0.99058	0.99062
		jaccard	0.99033	0.99041
	STG2	seuclidean	0.99628	0.99628
		hamming	0.99384	0.99525
		Cosine	0.99025	0.99059

5 Conclusions and future work

A redundancy-based resilience approach has been presented, analyzed and simulated in this paper, the support of which is concentrated on a secondary network. This network, serving as a dedicated network sublayer, has the ability to manage the context by periodically receiving consensual information from the driver nodes contained in the control network itself, and discerning opinion discrepancies through data-mining techniques such as kmeans and k-nearest neighbor. Any evidence that indicates a topological change is then managed in time using the implicit redundancy and the role of a few dedicated supernodes, also working as driver nodes. The experiments state the lineal optimization of the approach both for target and Byzantine scenarios, where the control is always retaken either from a local or remote perspective.

In future work, we intend to integrate the approach in a small proof-of-concept for refinements and extensions, and propose new restoration solutions also working in lineal times but this time, without redundant resources.

[2] Cityblock: $d(x,c) = \Sigma_{j=1}^{p}|x_j - c_j|$, cosine: $d(x,c) = 1 - \frac{xc'}{\sqrt{(xx')(cc')}}$ and correlation is similar to cosine but correlating vectors of type $(x - \vec{x})$ where $\vec{x} = \frac{1}{p}(\Sigma_{j=1}^{p} x_j)\vec{1}$, and $\vec{1}$ is a row vector initialized with n ones.

ACKNOWLEDGMENT

This work has been partially supported by the research projects PERSIST (TIN2013-41739-R) and SMOG (TIN2016-79095-C2-1-R), both financed by the Ministerio de Economía y Competitividad.

REFERENCES

[1] C. Alcaraz, and J. Lopez, "Wide-area situational awareness for critical infrastructure protection", In *IEEE Computer*, vol. 46, no. 4, IEEE Computer Society, pp. 30-37, 2013.

[2] SGIP, "Guidelines for smart grid cybersecurity - volume 1 - smart grid cybersecurity strategy, architecture, and high-Level requirements", *NIST-7628 (rev. 1)*, Sept., 2014.

[3] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn. "Guide to Industrial Control Systems (ICS) security", *NIST-SP 800-82 (rev. 2)*, May, 2015.

[4] C. Alcaraz, and J. Lopez, "Analysis of requirements for critical control systems", In *International Journal of Critical Infrastructure Protection*, vol. 5, Elsevier, pp. 137–145, 2012.

[5] K. Nakayama, N. Shinomiya, and H. Watanabe, "An autonomous distributed control method for link failure based on tie-set graph theory", In *IEEE Transactions on Circuits and Systems I*, vol. 59, issue 11, pp. 2727–2737, 2012.

[6] M. Marchese and M. Mongelli, "Simple protocol enhancements of rapid spanning tree protocol over ring topologies", In *Computer Network*, vol. 56, issue 4, pp. 1131–1151, 2012.

[7] M. Médard, S. G. Finn, and R. A. Barry, "Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs", In *IEEE/ACM Trans. Netw.*, vol. 7, issue 5, pp. 641–652, 1999.

[8] C. Alcaraz and S. Wolthusen, "Recovery of structural controllability for control systems", In 8th IFIP WG 11.10 International Conference on Critical Infrastructure, Springer, vol. 441, pp. 47–63, 2014.

[9] W. Quattrociocchi, G. Caldarelli, and A. Scala, "Self-healing networks: Redundancy and structure", *PLoS ONE*, vol. 9, issue 2:e87986, 2014.

[10] B. Wang, L. Gao, Y. Gao, and Y. Deng, "Maintain the structural controllability under malicious attacks on directed networks", EPL (Europhysics Letters), vol. 101, issue 5:58003, 2013.

[11] W-X. Wang, X. Ni, Y-C. Lai, and Celso G., "Optimizing controllability of complex networks by minimum structural perturbations", Phys. *Rev. E*, vol. 85:026115, 2012.

[12] J. Ding, Y-Z. Lu, and J. Chu, "Recovering the controllability of complex networks", In *9th World Congress the International Federation of Automatic Control*, pp. 10894–10901, 2014.

[13] S. Yi, Z. Hao, Z. Qin and Q. Li, "Fog Computing: Platform and Applications," *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies*, Washington, DC, 2015, pp. 73-78.

[14] R. Roman, J. Lopez, and M. Mambo, Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges, CoRR, vol. abs/1602.00484, 2016.

[15] C.-T. Lin. Structural controllability. *IEEE Transactions on Automatic Control*, vol. 19, issue 3, pp. 201–208, 1974.

[16] T. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning, "Domination in graphs applied to electric power networks", *SIAM Journal on Discrete Mathematics*, vol. 15, isseue 4, pp. 519–529, 2002.

[17] J. Kneis, D. Mölle, S. R., and P. Rossmanith, "Parameterized power domination complexity", *Information Processing Letters*, vol. 98, issue 4, pp. 145–149, 2006.

[18] H. Samuel, W. Zhuang, and B. Preiss, "Improving the dominating-set routing over delay-tolerant mobile ad-hoc networks via estimating node intermeeting times", EURASIP Journal on *Wireless Communications and Networking*, Hindawi Publishing Corp., pp. 1–12, 2011.

[19] R. E. Kalman, "Mathematical description of linear dynamical systems", *Journal of the Society of Industrial and Applied Mathematics Control Series A*, vol. 1, pp. 152–192, 1963.

[20] N. J. Cowan, E. J. Chastain, D. A. Vilhena, J. S. Freudenberg, and C. T. Bergstrom, "Nodal Dynamics, Not Degree Distributions, Determine the Structural Controllability of Complex Networks", *PLoS ONE*, vol. 7, issue 6:e38398+, 2012.

[21] A. Das, S. Gollapudi, and K Munagala, "Modeling opinion dynamics in social networks", In *Proceedings of the 7th ACM international conference on Web search and data mining*, ACM, pp. 403-412, 2014.

[22] IEC-62351 Parts 1-8: Information Security for Power System Control Operations, International Electrotechnical Commission, http://www.iec.ch/smartgrid/standards/, retrieved April 2017, 2007-2011.

[23] C. Alcaraz, and J. Lopez, "Safeguarding Structural Controllability in Cyber-Physical Control Systems", In *the 21st* European Symposium on Research in Computer Security, Springer vol. 9879, pp. 471-489, 2016

[24] C. Alcaraz, E. Etcheves Miciolino, and S. Wolthusen, "Structural Controllability of Networks for Non-Interactive Adversarial Vertex Removal", In 8th International Conference on Critical Information Infrastructures Security, Springer, vol. 8328, pp. 120-132, 2013.

[25] C. Palmer and J. Steffan. Generating network topologies that obey power laws. In Conference on Global Telecommunications, vol. 1, pp. 434–438, 2000.