IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 18, 2021, accepted March 25, 2021, date of publication April 8, 2021, date of current version April 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3071995

Assessment of Connectivity-Based Resilience to Attacks Against Multiple Nodes in SDNs

DORABELLA SANTOS[1], AMARO DE SOUSA[2], CARMEN MAS-MACHUCA[3], (Senior Member, IEEE), AND JACEK RAK[4], (Senior Member, IEEE)

[1]INESC Coimbra, DEEC, 3030-290 Coimbra, Portugal
[2]Instituto de Telecomunicações, DETI, Universidade de Aveiro, 3810-193 Aveiro, Portugal
[3]Chair of Communication Networks, Technical University of Munich, 80333 Munich, Germany
[4]Faculty of Electronics, Telecommunications, and Informatics, Gdańsk University of Technology (GUT), 80-233 Gdańsk, Poland

Corresponding author: Jacek Rak (jrak@pg.edu.pl)

This article is based on work from COST Action CA15127 (“Resilient communication services protecting end-user applications from disaster-based failures” – RECODIS), supported by COST (European Cooperation in Science and Technology); *http://www.cost.eu*
This work was also partially supported by ERDF Funds through the Centre’s Regional Operational Program and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia, I.P. under the project CENTRO-01-0145-FEDER-029312.

ABSTRACT In Software Defined Networks (SDNs), the control plane of a network is decoupled from its data plane. For scalability and robustness, the logically centralized control plane is implemented by physically placing different controllers throughout the network. The determination of the number and placement of controllers is known as the Controller Placement Problem (CPP). In the regular (i.e., failure-free) state, the control plane must guarantee a given maximum delay between every switch and its primary controller and a given maximum delay between every pair of controllers. In general, these delay bounds allow multiple solutions and, so, other goals can be used to determine the best CPP solution. In this paper, we assess the connectivity-based resilience to malicious attacks against multiple network nodes of the CPP solutions obtained with three different aims: the regular state delay optimization without any concern about attacks, the regular state delay optimization taking into consideration the worst-case attacks and the resilience optimization to attacks against multiple nodes. We assess the CPP solutions considering attacks of targeted nature (when the attacker has complete knowledge of the data plane) and attacks of non-targeted nature (i.e., random and epidemic attacks). We present computational results providing an analysis of the CPP solutions to the different types of attacks. The main conclusion is that the connectivity-based resilience between the different CPP solutions strongly depends on the network topology, the regular state delay bounds and the type of attacks. Finally, we provide insights on how SDN operators can consider the conducted assessment when deciding the controller placements in their networks.

INDEX TERMS Resilience, software defined networking, attacks against nodes, malicious human activities, communication networks, optimization.

I. INTRODUCTION

In Software Defined Networks (SDNs), the control plane is decoupled from the data plane, allowing a more efficient centralized management of the network resources [1]. The data plane is provided by a set of switches (and interconnecting links) dedicated to forwarding packets of traffic flows. The control plane is provided by a set of controllers with a complete view of the current network traffic and of all supported services. When a packet of a new traffic flow reaches a switch, it queries one controller which replies with the routing decision on how to forward the new traffic flow.

Although the SDN control plane can be provided by a single controller, it is usually based on multiple controllers physically distributed over the network and where each switch is assigned a primary controller (i.e., the controller with which the switch interacts when it needs a routing decision). One main reason to deploy multiple controllers is to increase the control plane availability, i.e., to avoid the single point of failure. The other main reason is to increase the control

The associate editor coordinating the review of this manuscript and approving it for publication was Hocine Cherifi.

plane scalability, both in terms of query-response interaction times (i.e., the delays between the switches and their primary controllers) and in terms of controller processing capacity (i.e., the processing load required on the controllers for the rate at which they are queried by the switches).

So, an immediate concern that arises in the planning of an SDN is the determination of the number and placement of the controllers on a given data plane network. This problem is known as the Controller Placement Problem (CPP), which is a variant of the facility location problem shown to be $\mathcal{NP}$-hard in [2] (please see [3] on modeling and solving combinatorial optimization problems including the facility location problem). The CPP is addressed here in the context of networks covering large geographical areas (where the delays between the switches and their primary controllers are a concern) assuming unlimited controller processing capacities (i.e., the incoming query rate at each controller is not bounded by its processing capacity).

A physically distributed SDN control plane can operate either in a logically centralized or a logically distributed mode [4], [5]. As will be explained later, the main aim of this work is to assess the resilience of different controller placement alternatives to attacks against multiple nodes. In general, failure recovery can be implemented by either restoration or protection mechanisms and both alternatives are available in SDNs [6]. Although protection mechanisms are the only alternative to achieve recovery times within 50 ms required by carrier-grade networks (as shown in [6]), this alternative is only possible for single failures. In attacks against multiple nodes, it is no longer possible to guarantee protection to all service flows nor to guarantee the recovery times usually required in events involving single failures. So, to minimize the impact of attacks against multiple nodes, the SDN is assumed to operate with a restoration mechanism that, when the connectivity between a switch and its primary controller is lost, a new primary controller can be dynamically assigned to it. Moreover, we also consider a logically centralized control plane with a flat controller architecture [5] since, in this case, all controllers have the same complete view of the network state and, therefore, any controller can become the primary controller of any switch at any time.

In a logically centralized control plane, a routing decision on a particular controller (to an incoming query from a switch) is sent to all other controllers and, therefore, the resync time (i.e., the time interval until the common view of the network state is reached in all controllers) is, in the worst case, the longest delay among all pairs of controllers. The resync time can be very long in networks covering large geographical areas and, therefore, must be bounded in any controller placement solution.

So, the CPP variant addressed in this work is defined as follows. For a given data plane network and a desired number of controllers, the control plane provided by the selected controller placements must guarantee for the regular state (i.e., when all network elements are running without failure):

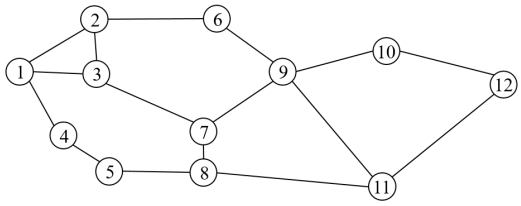

FIGURE 1. Data plane example.

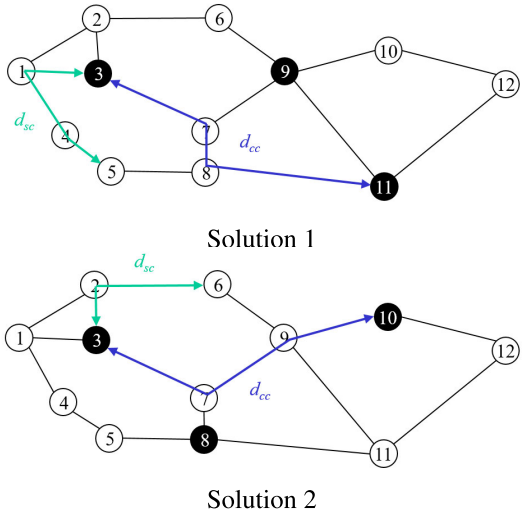

FIGURE 2. Two geographically distributed SDN control plane examples. Controller nodes highlighted in black. Worst switch-controller delay d_{sc} and controller-controller delay d_{cc} highlighted in green and blue, respectively.

(i) a given maximum delay between any switch and its primary controller (to bound the query-response interaction delays to an acceptable value), and
(ii) a given maximum delay between any pair of controllers (to bound the maximum resync time to an acceptable value).

Moreover, since all controllers have a complete and updated view of the network state, we assume that the primary controller of each switch is its closest controller (in terms of delay) both in the regular state and in any failure state. Finally, we assume that each controller is locally connected to a nearby switch, which is referred to as a controller node.

In general, the regular state delay bounds allow multiple controller placement alternatives. Consider the data plane example shown in Fig. 1 of a hypothetical network topology where the delay of each link is proportional to its length.

Fig. 2 shows two different control plane examples with three physically distributed SDN controllers each. In each example, the worst delay d_{sc} between any switch and its primary controller is highlighted in green and the worst delay d_{cc} between any pair of controllers is highlighted in blue. If the values of d_{sc} and d_{cc} of both cases are within the delay bounds to be guaranteed in the regular state, both solutions can be adopted leaving room to the use of other control plane design goals.

Multiple failures can be caused by different reasons, as natural disasters, technology related failures or human malicious activities [7], [8]. Multiple failures might involve either link failures only, or node and link failures (since a node failure implies that its connected links also fail). In malicious human attacks, node shutdowns are harder to achieve but they are the most rewarding from the attacker's perspective since the shutdown of a single node also shuts down all the incoming/outgoing links. Here, we consider the case of attacks against multiple nodes as they are the most harmful case. Moreover, in the case of a controller node, the shutdown of the switch also disconnects the locally connected SDN controller from the network, which means that the controller is no longer available.

In general, malicious human attacks aim to seriously disrupt services supported by telecommunication networks. In SDNs, the impact of such attacks is potentially more damaging due to the SDN architectural characteristics. Consider once again the two control plane examples shown in Fig. 2 and assume that nodes 9 and 11 are shut down by a malicious attack, splitting the data plane into two connected components. In the component composed by nodes 10 and 12, node 10 is a controller node in Solution 2 and, therefore, this component will keep supporting the services between the two nodes (after the recovery time required by the restoration mechanism). In the same component, on the other hand, the services will fail (sooner or later) in Solution 1 due to the lack of connectivity of the switches to a surviving controller node.

When addressing the resilience of networks to multiple failures, we can distinguish two types of problems. One is the post-disaster (i.e., reactive) problem which deals with how to recover the network as quickly as possible from a given event involving multiple failures. In general, the resilience assessment in the post-disaster problem considers the network state before the event, the network state immediately after the event, the desired network state after recovery and the required recovery time [9]. The other is the pre-disaster (i.e., proactive) problem which deals with how to set up the network in advance aiming to minimize the impact of different possible multiple failure events before the failures are detected, which is, in our case, before any measure being triggered by the operator (for example, the replacement of the shutdown nodes) besides the SDN restoration mechanism of the surviving switches being reassigned to the surviving controllers in the surviving network.

In this work, we address the pre-disaster problem considering that, as already mentioned, the multiple failures are caused by malicious attacks against multiple nodes. In this case, the dominant impact of the attacks is the connectivity disruption between switches (at the data plane) and between switches and primary controllers (at the control plane). So, in this work, the resilience is measured in terms of the impact of the attacks on the connectivity of the network, i.e., we consider the connectivity-based resilience. In this case, the primary aim of the resilience assessment is to measure how degraded the network state becomes after an attack and taking only into consideration the result of the restoration mechanism, which is limited as it cannot restore connectivity between nodes in different network components. By minimizing the impact of the attacks, the aim of the pre-disaster problem is not to address the recovery problem but, instead, it is to minimize the recovery effort required in the post-disaster problem. Moreover, the recovery time is not included in the resilience assessment of the pre-disaster problem as it depends on how the post-disaster problem is tackled afterwards.

From the perspective of an SDN operator, neither the type of attack nor the set of shutdown nodes are known in advance. The attack depends on the attacker's knowledge of the network, i.e., how much he/she knows about the data plane (the location of the switches and their interconnecting links) and about the control plane (the location of the controllers). Moreover, the attack also depends on the attacker's capacity to shut down each (known) node and/or the attacker's strategy on how the nodes are selected.

In this work, we consider the assessment of the connectivity-based resilience of different controller placements to different types of attacks against multiple nodes. We investigate attacks of targeted nature, which corresponds to the case when the attacker, having complete knowledge of the data plane topology, selects the most harmful nodes according to his/her strategy. We further distinguish these attacks as ''targeted attacks'' (*TAs*) when the attacker does not know the location of the controllers and ''controller targeted attacks'' (*CTAs*) when the attacker also knows the location of the controllers (in the latter case, the attacker only selects controller nodes as they are the most important nodes in keeping the network operational). We also consider attacks of non-targeted nature:

(i) ''random attacks'' (*RAs*), which corresponds to the case when the attacker discovers (by some means) some nodes that he/she is able to shut down, and
(ii) ''epidemic attacks'' (*EAs*), which corresponds to the case when a few initial nodes are discovered (and shut down), and, subsequently, the attack is propagated to some of the neighbor nodes of the previously shutdown nodes, in an iterative manner.

In this work, the assessment of the connectivity-based resilience to the different types of attacks is conducted on three different controller placement alternatives which were proposed in previous works. To this aim, we first define a connectivity-based resilience metric that measures the average connectivity impact of each type of attack in both the data and control planes. Then, we use this metric to assess the resilience of the CPP solutions obtained with three different aims: the optimization of the regular state delay without any concern about attacks, referred to as the *BasicRest* solution; the optimization of the regular state delay taking into consideration the worst-case attacks, referred to as the *Robust* solution; and the optimization of the resilience to attacks against multiple nodes, referred to as the *Optimal* solution.

We conducted a set of computational experiments to analyze and compare the connectivity-based resilience of the three CPP solutions for each type of attacks, on two well-known telecommunication network topologies (Germany50 with 50 nodes and Coronet CONUS with 75 nodes). As expected, the results show that the resilience is much higher to non-targeted attacks (*RAs* and *EAs*) than to attacks of targeted nature (*CTAs* and *TAs*) as, typically, the former ones do not split the network into many components and, in most cases, all components include a controller node. Concerning the attacks of targeted nature, the resilience differences between the different CPP solutions are strongly dependent on each particular case (i.e., the network topology and the regular state delay bounds): in some cases, the different CPP solutions present resilience values close to the optimal (i.e., optimizing the regular state delay provides a solution which is also optimal, or almost optimal, in terms of resilience to attacks) while in other cases, there are significant resilience gains when the controller placements are determined with the aim to optimize their resilience to the attacks.

All three CPP solutions (*BasicRest*, *Robust* and *Optimal*) assume that, upon an attack against multiple nodes, the control plane reacts with a restoration mechanism as, for example, the one in [6]. The *BasicRest* (short, for Basic Restoration) solution considers the CPP problem, as defined by [2], with additional constraints imposing the regular state delay bounds. The *Robust* and *Optimal* solutions were proposed in [10] where the so-called robustness property is imposed to maximize the resilience to the most damaging type of attacks (i.e., *CTAs*).

In [11], an extended version of [10] reviews the way the resilience is evaluated and proposes a more efficient algorithm to compute the CPP solutions. In both works, though, the resilience of the solutions considers only attacks of targeted nature, the evaluation addresses only the resilience of the control plane and the *Optimal* solution is computed accordingly (i.e., aiming only at the optimization of the control plane resilience). On the other hand, both data and control planes are considered in [12] to evaluate the resilience of the controller placements but, as in the previous works, the evaluation is conducted only for attacks of targeted nature.

This work is an extension of [12]. Compared with [12], this work provides the following original contributions.

- We propose a connectivity-based resilience metric that considers three degradation parameters (also used in [12]) but now allowing the SDN operator to assign a weight to each parameter to represent its importance of the degradation impact on the services supported by the network.
- We show how the *Optimal* solution is computed to consider the new (proposed) resilience metric so that the controller placements are selected based on the weight values defined by the SDN operator.
- We provide a more comprehensive evaluation of the resilience of the different CPP solutions to different types of attacks (the previous work [12] has only considered attacks of targeted nature while here we also consider non-targeted attacks, both random and epidemic).
- We provide a more comprehensive evaluation of the resilience of the different CPP solutions to attacks against different numbers of nodes (the previous work [12] has only considered the evaluation of attacks against an expected maximum number of nodes but, in general, a different number of nodes can be targeted by an attacker).
- We provide insights on how SDN operators can consider the conducted assessment when deciding the controller placements in their networks.

The outline of the paper is as follows. Section II presents a review of the related scientific literature concerning the CPP, focusing on works that deal with the resilience of SDNs to single and multiple failures. In Section III, the different types of attacks against multiple nodes considered in this work are further detailed. Section IV describes how the connectivity-based resilience metric is computed and how the CPP solutions of the three considered optimization problems can be obtained. Section V describes the problem instances considered in the computational analysis and shows how the attacks against multiple nodes were generated to compute the resilience results. Section VI analyzes the computational results and provides insights on how SDN operators can consider the conducted analysis when deciding the controller placements in their networks. Finally, Section VII provides the concluding remarks.

II. LITERATURE REVIEW

The CPP defined by Heller [2] considered a physically distributed control plane, where each switch of the data plane had a single control connection to one of the controllers of the control plane. Based on this initial problem, some solutions have been proposed to maintain the control-data plane connectivity (i.e., between switches and controllers) in the case of single failure scenarios (i.e., link and node failures). Hu *et al.* [13] proposed a CPP solution aiming at minimizing the percentage of the expected control path loss (δ) due to any single failure scenario. Based on the min-cut algorithm proposed by Zhang *et al.* [14], the switches are grouped into partitions such that the connectivity within each partition and the assigned controller is improved. Mueller et al. [15] addressed the reliability of the CPP by considering path diversity and a list of backup controllers that can potentially be used when the primary controller or any component (link or node) of the control path to the primary controller fails. The CPP solution proposed by Vizarreta *et al.* [16] associates two different controllers to each switch by means of two disjoint control paths, which reduces δ significantly and supports fast control plane recovery.

The reliability of the control plane for multiple failures was also addressed. For example, Guo *et al.* [17] considered cascading failures, which are triggered by a node or link

failure and are propagated to other network components based on their dependence relations. Savas *et al.* [18] proposed the Recovery-Aware Switch-Controller Assignment and Routing (RASCAR) scheme, which enables fast data-path recovery after disasters affecting the set of links and nodes located within a circle of center c and radius r for modeling earthquakes. Hock *et al.* [19] developed a MATLAB framework named POCO (Pareto-based Optimal COntroller Placement framework) which is able to display the Pareto frontier enumerating all the feasible controller placements according to different objective functions (including controller failure tolerance) and visualize the different Pareto optimal placements. Li *et al.* [20] and Yang *et al.* [21] provided CPP solutions able to cope with k link failures. Perrot *et al.* [22] proposed a solution able to find the optimal number of controllers, their location, the assigned switches, and the several levels of backup controllers in case the primary controllers fail based on a given probability.

However, link and node failures are not the only threats of SDNs. In this work, we consider attacks, i.e., intended failures, that may also occur and operators should also consider them when designing their networks. Rueda et al. [23] proposed a CPP solution to improve the SDN control plane robustness against targeted attacks, which are defined according to the most harmful centrality metric (e.g., node degree centrality, node betweenness centrality). Two targeted attacks were considered, i.e., sequential attacks (i.e., the metric is computed after each attack to select the next targeted node) and simultaneous attacks (the metric is computed once and the components with the higher metrics are attacked). The Average Two-Terminal Reliability (ATTR), selected as the robustness metric, increases for the CPP solutions that are aware of the targeted attacks. Furthermore, Cosgaya et al. [24] considered different targeted attack scenarios and proposed a robust strategy to place the controllers preserving the desired control plane availability. Calle et al. [25] studied the issue of adding a number of additional controllers to the network in order to increase its resilience to targeted attacks. To that aim, an optimization model for solving the related CPP was proposed, which is based on an availability measure defined as the average number of switches that can still connect to a controller for a set of attacks against multiple network nodes. Pióro et al. [26] dealt with targeted attacks against multiple network nodes presenting an optimization approach useful for the controller placement of SDNs. That work proposed a probabilistic network availability measure to derive the most dangerous attacks based on the attacker's knowledge of the network.

In our previous works, Santos *et al.* [10] also studied the robustness of the control plane to attacks against multiple nodes. That work selected from different non-robust CPP solutions (aiming to minimize the average switch-to-controller or the average controller-to-controller delay), the ones maximizing the number of switches connected to a surviving controller for a given set of malicious attacks. In this case, the comparison metrics were the number of surviving nodes connected to at least one controller and the number of surviving nodes connected to its primary controller within the maximum allowed delay in the regular state. In order to qualify and quantify the robustness of the control plane, Santos *et al.* [12] proposed to select among the different CPP solutions, the ones compliant with the following robustness property: if any subset of controller nodes fail, there must still exist a path in the data plane from any switch to a surviving controller. Furthermore, this work extended the targeted attacks to include the ones based on the critical node detection [27]. It was shown that the robustness property could be achieved with a limited switch to controller delay penalty. Last but not least, Santos et al. [11] compared robust CPP solutions for SDN control plane operating with or without split-brain. It was shown that the split-brain architecture does not always provide the best robust solution despite requiring more controllers.

All previous works dealing with attacks against multiple nodes, though, have considered that the attacker has complete knowledge of the network topology (i.e., attacks of targeted nature). Moreover, almost all works have considered the impact of the attacks only in the control plane of the SDN. Here, we extend the resilience analysis of CPP solutions to attacks of non-targeted nature (random and epidemic attacks) proposing a connectivity-based resilience metric that measures the impact of the different types of attacks in both the data plane and control plane of the SDN. Moreover, we provide insights for SDN operators on how to evaluate the resilience of different controller placements depending on the type of attacks they are more concerned with.

III. ATTACKS AGAINST MULTIPLE NODES

An attack against multiple nodes involves uncertainty both on the targeted number of nodes, defined as p, and on how that number of nodes is selected. Concerning the uncertainty on the targeted number of nodes, the value of p is a parameter of the attack so that we can consider different values of p and evaluate the influence of its value on the resilience evaluation of the different CPP solutions.

Then, the uncertainty on how the nodes are selected depends on the type of attack. This section presents the different types of attacks, addressing in separate subsections the attacks of targeted and non-targeted nature.

A. ATTACKS OF TARGETED NATURE

An attack of targeted nature corresponds to the case when the attacker has complete knowledge of the data plane network and is able to shut down the most important nodes according to some strategy. Among such attacks, we further distinguish the cases when the attacker either knows or does not know the location of the SDN controllers. The first type is simply referred to as a "targeted attack" (*TA*) while the second type is referred to as a "controller targeted attack" (*CTA*). Note that, in practice, a *CTA* has a lower probability of being carried out than a *TA*, as SDN controllers are software-based systems which are harder to be identified.

1) TARGETED ATTACKS (*TAs*)

The most common strategies of *TAs* [7], [23] are based on three node centrality metrics from graph theory: node degree, node closeness and node betweenness. Consider the data plane topology modeled by a graph $G = (N, E)$ with a given set N of switches and a given set E of connecting links. The three node centrality metrics are defined as follows.

Node degree centrality, denoted as $deg(i)$, measures the centrality of node i by its number of neighbor nodes:

$$deg(i) = |N_i| \tag{1}$$

where N_i is the set of neighbor nodes of i in graph G and $|N_i|$ denotes the number of nodes in N_i.

Node closeness centrality, denoted as $clo(i)$, measures the centrality of node i by how close it is, on average, to all other nodes in graph G. A possible way to calculate the closeness centrality, as proposed in [28], is:

$$clo(i) = \sum_{j \in N \setminus \{i\}} \frac{1}{d_{ij}} \tag{2}$$

where d_{ij} is the number of links of any shortest path from node i to node j in graph G.

Node betweenness centrality, denoted as $btw(i)$, measures the centrality of node i by how frequent it is as an intermediate node in the shortest paths between all other node pairs:

$$btw(i) = \sum_{\substack{s<t \\ s,t \neq i}} \frac{\sigma_{st,i}}{\sigma_{st}} \tag{3}$$

where σ_{st} is the number of different shortest paths from node s to node t and $\sigma_{st,i}$ is the number of such shortest paths that include node i as an intermediate node.

In a centrality-based *TA*, the nodes are selected based on one of the previous centrality metrics assuming that shutting down more central nodes (i.e., nodes with higher centrality metric values) has a higher impact on the disruption of the services supported by the network.

For a targeted number of nodes p, the selection of the p nodes to be shut down can be done with two possible strategies [29]. In a 'simultaneous' *TA*, the node centrality metric values are calculated once for all network nodes and the p nodes with the highest values are shut down. In a 'sequential' *TA*, the node centrality metric values are recalculated each time a node is shut down by removing it from graph G and the process is repeated until p nodes are shut down.

Besides centrality-based *TAs*, another attack strategy has also been considered more recently in the evaluation of telecommunication networks to attacks against multiple nodes [30], which is based on an optimization problem known as the critical node detection (CND) problem. CND was first introduced in [27], where it was shown to be $\mathcal{NP}$-hard, and, since then, different variants of CND have been addressed in different contexts [31].

For a targeted number of nodes p in a CND-based *TA*, a set of p nodes (named critical nodes) is selected such that their removal from the network maximally reduces the number of node pairs that can still communicate in the surviving network. This CND variant has been addressed by exact methods based on integer linear programming [27], [32]–[34] and it was shown in [34] that, although the problem is $\mathcal{NP}$-hard, the optimal CND solutions can be computed for telecommunication network topologies up to 200 nodes.

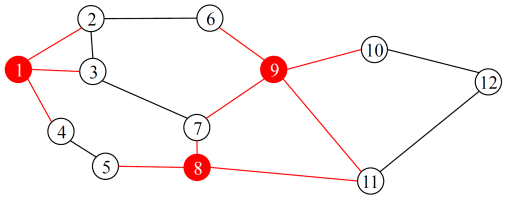

FIGURE 3. Nodes shut down by a CND-based *TA* to $p = 3$ nodes.

For illustration purposes, we present in Fig. 3 the nodes (highlighted in red) that are shut down by a CND-based *TA* for a targeted number of $p = 3$ nodes on the data plane topology previously presented in Fig. 1. By shutting down these nodes, the attacker is able to split the network into three components–one with two nodes (`4` and `5`), one with three nodes (`10`, `11` and `12`) and one with four nodes (`2`, `3`, `6` and `7`) – and the resulting number of node pairs that can communicate in the surviving network is 10, the lowest possible value in the shutdown of any set of three nodes of this topology.

In [30], a vulnerability evaluation of telecommunication topologies was conducted comparing the CND-based attack strategy with the previously described centrality-based attack strategies. The main conclusions are two-fold. One is that the CND-based attack is much more damaging (in terms of surviving connectivity between all node pairs) than centrality-based strategies. The other is that the most damaging centrality-based strategies are obtained with the 'sequential' variant. Although the work in [30] does not address the particular case of SDNs, since connectivity is a necessary condition to maintain the services after an attack, the same behavior is likely to happen in SDNs, the reason why in the computational results we only consider the 'sequential' strategy in the centrality-based *TAs*.

2) CONTROLLER TARGETED ATTACKS (*CTAs*)

For *CTAs*, the attacker also knows the location of the controllers. Hence, it can be assumed that the attacker targets only the controller nodes as they are the most important nodes in maintaining the network operational. In this case, if the attacker is able to shut down all controller nodes, he/she is 100% successful in disrupting all supported services. Otherwise, any of the three previously described centrality-based attacks can be conducted where the nodes to be selected are restricted to the set of controller nodes (i.e., if the attacker is only able to attack a subset of the controller nodes, the most central controller nodes are selected).

B. ATTACKS OF NON-TARGETED NATURE

Attacks of non-targeted nature correspond to the cases when the attacker does not have complete knowledge of the data plane and does not know the placement of the controllers. These attacks are further classified as ''Random Attacks'' and ''Epidemic Attacks''.

1) RANDOM ATTACKS (*RAs*)

A random attack (*RA*) corresponds to the case when the attacker discovers some nodes and is able to shut them down. By shutting down these nodes, the attacker aims to disrupt the services supported by the network as much as possible. From the perspective of the SDN operator, since it does not know how nodes are 'discovered' by the attacker, any set of uncorrelated nodes can be shut down at random.

In modeling terms, it is similar to multiple unintended failures with the difference that single failures are much more likely than multiple failures in unintended events, while the shutdown of multiple nodes is more likely in *RAs*. The nodes that are shut down by a *RA* are selected as follows: for a target number of nodes p, a set of p nodes is randomly selected assuming the same selection probability for all network nodes.

2) EPIDEMIC ATTACKS (*EAs*)

An epidemic attack (*EA*), in turn, can be seen as an extension of a *RA*. At first, one node (or a group of nodes) is first discovered and shut down (i.e., similar to a *RA*). Then, the attack is extended to other nodes assuming that the intrusion of the attacker in the node's premises (either physically or by remote means) to shut down a given node i enables the attacker to discover, with some probability, other nodes among the neighbor nodes of i (i.e., the nodes with direct links with node i).

The way the set of shutdown nodes gets increased over time corresponds to the model of an epidemic spreading. As discussed in [35], the epidemic dynamics have been widely investigated in the literature and, in general, different epidemic models have been identified [36]–[38]. Although such models have been originally motivated to cover the different ways of modeling the spread of infections among living species, some of them have also been used in contexts more related to communication networks. Examples of such works are multiple failures propagation in GMPLS-based networks [39], random jamming activities for limited use in wireless sensor networks [40], epidemic dynamics in highly clustered networks [41], [42] and worm propagation in wireless sensor networks [43].

In general, an epidemic model considers that nodes can be in one of a set of different types of states and defines the possible transitions between states of each node. The simplest epidemic model is the *susceptible-infected* (SI) model where nodes can be either in the 'susceptible' state or in the 'infected' state [44]. In this model, when a node becomes 'infected', it remains in the 'infected' state forever. In our work, we consider an SI model where a node that is shut down by the attacker corresponds to a node in the 'infected' state, while a neighbor node of a shutdown node corresponds to a node in the 'susceptible' state, i.e., it can be selected by the attacker as the next node to be shutdown. Note that, although the result of an attack against a node is its shutdown, since the discovery of the neighbor nodes is retained by the attacker in all subsequent decisions, in modeling terms, it is equivalent as to consider that the node is in the 'infected' state forever.

Other epidemic models introduce other types of states. For example, in some models, the 'infected' state can be followed by the 'disabled' state or by the 'removed' state. Moreover, a node that reaches such states can either remain in these states forever or can return again to the 'susceptible' state, depending on the context. Such models either assume some self-healing capabilities of nodes after being 'infected' or some kind of counter measures to the epidemic spreading. In our case, a node which is shutdown is not able to self-recover. Moreover, we address the resilience evaluation of different controller placements to attacks against multiple nodes in a pre-disaster context, i.e., before the attack being detected and, therefore, before any counter measure being triggered by the SDN. So, models assuming counter measures during the attack propagation are outside the scope of our work.

The nodes that are shut down by an *EA* are selected as follows. For a targeted number of nodes p, one node v_1 is first 'infected' at random (assuming the same probability for all nodes) and the set V of 'susceptible' nodes is computed with the set of neighbor nodes of v_1. Then, the next 'infected' node v_2 is chosen randomly from V and set V is updated accordingly (i.e., with all nodes not yet 'infected' that are neighbor nodes of at least one 'infected' node). This process is repeated until p nodes become 'infected' (i.e., shut down).

At each step, a selection probability is assigned to each 'susceptible' node (i.e., each node in V) proportional to the number of 'infected' nodes it is neighbor to. In the first step (i.e., when selecting node v_2), all nodes of V have the same selection probability as they are neighbors of a single 'infected' node. In the subsequent steps, a 'susceptible' node which is neighbor to more 'infected' nodes has a higher probability of being 'infected' (i.e., shut down) as it has an higher probability of being discovered when the previous 'infected' nodes were selected.

IV. CONTROLLER PLACEMENTS AND ASSOCIATED CONNECTIVITY-BASED RESILIENCE TO ATTACKS

In this section, we first define in Section IV-A a connectivity-based resilience metric that measures, for a set of possible attacks, their average impact in the connectivity between switches (at the data plane) and between switches and primary controllers (at the control plane). Then, Section IV-B introduces the robustness property and describes an enumeration method to compute all CPP solutions compliant with it. Finally, Section IV-C describes how the three CPP solutions (*BasicRest*, *Robust* and *Optimal*) are computed.

A. CONNECTIVITY-BASED RESILIENCE METRIC

Consider a given CPP solution where the data plane topology is modeled again by a graph $G = (N, E)$ with a given set N of switches and a given set E of connecting links. The controller placements of the CPP solution guarantee in the regular state (i.e., before any attack) a given maximum delay between every switch and its primary controller defined as D_{sc}, and a given maximum delay between every pair of controllers defined as D_{cc}.

Let us consider a set $M_{A,p}$ of attacks of a given type A (where $A \in \{TA, CTA, RA, EA\}$) against a set of p nodes. Each attack $m \in M_{A,p}$ is defined by its set of shutdown nodes $N_m \subset N$, with $|N_m| = p$. The aim is to obtain a value representing the connectivity-based resilience of the CPP solution to the set of attacks defined in $M_{A,p}$.

Our resilience metric considers the following three parameters (also used in [12]), each one measuring a different connectivity degradation impact of each attack $m \in M_{A,p}$ computed in the network that survives attack m (i.e., the graph G without the nodes in N_m):

- n_{sp}^m the number of switch pairs that can communicate with each other as well as with a surviving controller (it represents the number of switch pairs that are still able to support data flows after restoration);
- n_{sc}^m the number of switches that can communicate with a surviving controller within the maximum delay D_{sc} (it represents the number of switches that, after restoration, can still be served by the control plane within the acceptable switch-control delay of the regular state);
- n_{pc}^m the number of switches that can communicate with their original primary controller in the regular state (it represents the number of switches not requiring the establishment of a control connection to another controller).

For illustration purposes, consider the evaluation of CPP Solution 1 shown in Fig. 2 to attack m given by the CND-based *TA*, which is shown in Fig. 3. Fig. 4 shows the network without the nodes that are shut down by this attack m. In this case, only two of the network components include a surviving controller and, therefore, $n_{sp}^m = 9$, i.e., the number of node pairs both belonging to one of these two components is equal to 9. Moreover, Fig. 4 shows that, besides switches 4 and 5 (which no longer can connect to any existing controller), the shortest path delay from switch 10 to its new primary controller 11 is higher than D_{sc}, and, therefore, $n_{sc}^m = 6$, i.e., the number of switches that can still be served with the acceptable switch-controller delay of the regular state is equal to 6. Finally, among the seven switches that still have a primary controller (2, 3, 6, 7, 10, 11 and 12), the primary controller of switches 6 and 7 have changed from the original controller 9 (before the attack, see Fig. 2) to controller 3 (after the attack) and the primary controller of switch 10 has also changed from 9 (before the attack) to 11 (after the attack). Therefore, $n_{pc}^m = 4$, i.e., the number of switches not requiring the establishment of a control connection to another controller is equal to 4.

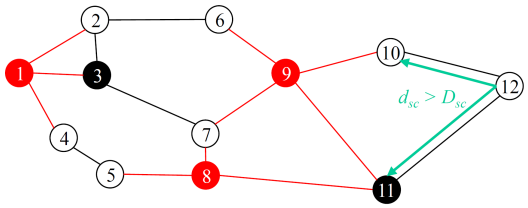

FIGURE 4. Evaluation of CPP Solution 1 (Fig. 2) to the CND-based *TA* (Fig. 3). Shutdown nodes highlighted in red. Surviving controller nodes highlighted in black. Switch-controller delays higher than D_{sc} highlighted in green.

The determination of parameters n_{sp}^m, n_{sc}^m and n_{pc}^m for each attack $m \in M_{A,p}$ is polynomial since these values can be computed with the shortest path lengths between every node pair. We efficiently compute these lengths by running a shortest path algorithm between all node pairs in graph G without the nodes in N_m, which has complexity $\mathcal{O}((|N| - p)^3)$, where p is the number of shutdown nodes.

Then, we compute from these three parameter values an average degradation value for each attack $m \in M_{A,p}$ taking into consideration the importance of the degradation impact represented by each individual parameter on the services supported by the SDN. We assume that the SDN operator assigns a weight to each parameter: α to parameter n_{sp}^m, β to parameter n_{sc}^m and γ to parameter n_{pc}^m (with $\alpha + \beta + \gamma = 1$). With these weights, we compute the average degradation value n_m of attack m as:

$$n_m = \alpha \times \frac{n_{sp}^m}{|N|(|N|-1)/2} + \beta \times \frac{n_{sc}^m}{|N|} + \gamma \times \frac{n_{pc}^m}{|N|} \quad (4)$$

In Eq. (4), each parameter is normalized (between 0 and 1) dividing its absolute value by its maximum value (number of node pairs in the case of n_{sp}^m and number of nodes in the other two cases). So, the average degradation value n_m is also normalized between 0 and 1. Finally, the connectivity-based resilience metric value, denoted as $r_{A,p}$, of a given CPP solution to the set of attacks $M_{A,p}$ is measured by the average value of n_m among all attacks $m \in M_{A,p}$:

$$r_{A,p} = \frac{1}{|M_{A,p}|} \sum_{m \in M_{A,p}} n_m \quad (5)$$

In the computational results, we consider that the most important parameter is n_{sp}^m as it represents the number of node pairs that can still support services after the attack, and services' support is the primary goal of any communications network. Then, we consider that the second most important parameter is n_{sc}^m as it represents the number of switches that can still be served (after the attack) within the same maximum switch-controller delay required in the regular state. Note that the impact of the switch-controller delay depends on the supported services as, for example, data services might suffer no significant impact while real-time services might be significantly degraded. In any case, some degree of service support can always be granted regardless of the switch-controller delay obtained after the attack, which is the reason

why we consider this parameter less important than n_{sp}^{m}. Finally, we consider that the least important parameter is n_{pc}^{m} as it only represents a temporary control plane unavailability at the switches while they do not reconnect to a new surviving controller. In practice, we assume that $\alpha > \beta > \gamma$ but the particular values of each weight are to be defined by the SDN operator.

B. ROBUSTNESS PROPERTY

The robustness property was first introduced in [10] in the CPP to maximize the resilience of the SDN to the most damaging *CTAs* that occur when the attacker knows the controller placements.

1) DEFINITION OF THE ROBUSTNESS PROPERTY

The robustness property is defined as follows: the set of controller placements is selected so that there is at least one routing path from each switch to each controller node, which does not include any of the other controller nodes. The result is that the shutdown of all but one controller nodes still allows all surviving switches to connect to the surviving controller and, consequently, no surviving switch is left without a primary controller.

Consider again the two controller placement solutions presented in Fig. 2. The set of controller nodes in Solution 1 is not compliant with the robustness property since any routing path from, for example, node `10` (or node `12`) to controller node `3` must include at least one of the controller nodes `9` or `11`. In this case, if controller nodes `9` and `11` are shut down, neither node `10` nor node `12` can connect to the surviving controller node `3`. On the other hand, the set of controller nodes in Solution 2 of Fig. 2 is compliant with the robustness property since any switch can reach any controller node by a routing path not including any of the other controller nodes. In this case, the shutdown of any combination of two controller nodes still leaves the surviving network fully connected and operational since all switches can connect with the surviving controller node.

2) ENUMERATION OF ROBUST CPP SOLUTIONS

In general, for a given data plane topology, the full enumeration of all controller placement alternatives compliant with the robustness property (in the sequel, named robust CPP solutions) might not be possible when the dimension of this set is too large. However, recall that, in practice, the CPP solutions are required to guarantee given maximum switch-controller (SC) and controller-controller (CC) delays in the regular state (denoted by D_{sc} and D_{cc} respectively). In this case, the full enumeration (or, at least, the enumeration of a large percentage of them) is possible for networks of typical size. A first method for the enumeration of all robust CPP solutions was first proposed in [10] and a more efficient method was proposed afterwards in [11].

The enumeration of all robust CPP solutions requires the resolution of an optimization problem defined as an integer linear programming (ILP) model (see [10] for details). For a given number of controllers C, given maximum D_{sc} and D_{cc} delays, the ILP model considers appropriate constraints guaranteeing that:

(i) C controller nodes are selected;
(ii) the delay from each switch to its closest controller node is at most D_{sc};
(iii) the delay between each pair of controller nodes is at most D_{cc};
(iv) the controller nodes are compliant with the robustness property.

The objective function of the ILP model is the minimization of the sum of the closeness centrality values of the controller nodes, as defined in Eq. (2).

While solving the ILP model is computationally expensive, checking if a given placement of C controllers is a valid robust CPP solution (i.e., if it is compliant with the maximum delays and with the robustness property) is computationally efficient. The maximum D_{sc} and D_{cc} delays are straightforwardly checked with the shortest path delays between all node pairs. The robustness property is checked by considering an auxiliary directed graph (where controller nodes only have incoming arcs) and checking if the shortest path from every switch to every controller node is less than infinity in the auxiliary graph. So, in the method proposed in [11], the enumeration of all robust CPP solutions is as follows:

Step 1: Solve the ILP model. If the ILP model is unfeasible, stop and return all found robust CPP solutions. Otherwise, set the solution of the ILP model as the current robust CPP solution.

Step 2: Run an exhaustive search starting from the current robust CPP solution to find as many other robust CPP solutions as possible.

Step 3: Add to the ILP model one constraint per robust CPP solution found in Step 2 (including the initial one), to eliminate them from the set of the ILP feasible solutions. Return to Step 1.

The method is an iterative process that ends when the ILP model becomes infeasible in Step 1 (meaning that there are no more robust CPP solutions). Otherwise, the solution of the ILP model computed in Step 1 is used in Step 2 as the starting point of an exhaustive search to find many other CPP solutions.

The exhaustive search in Step 2 is as follows. New solutions are computed by swapping a controller node of the current robust CPP solution with each of its neighbor nodes that are not controller nodes. The new solutions that are valid robust CPP solutions are stored and used to generate new solutions. The search ends when there are no new robust CPP solutions. The exhaustive search (in Step 2) is efficiently implemented using a breadth-first strategy, i.e., the next current robust CPP solution is the oldest one not yet used to generate new solutions.

At the end of Step 2, it is not guaranteed that all valid robust CPP solutions are found, as other solutions might exist which cannot be reached only by swapping one controller from a

current node to a neighbor node on the already found robust CPP solutions. So, in Step 3, the ILP model is augmented with one constraint per robust CPP solution found to eliminate these solutions from the solution set of the model (returning to Step 1), the augmented ILP model is solved again (in Step 1) and the new solution, if it exists, is used as the starting point for a new exhaustive search (in Step 2).

In practice, the enumeration method also considers another input parameter l_{max} defining the maximum number of valid robust CPP solutions. In this case, the method stops either when it reaches a number of robust CPP solutions equal to l_{max} or when there are no more robust CPP solutions.

C. DETERMINATION OF CPP SOLUTIONS

Recall that we assume a logically centralized SDN control plane with a flat architecture (i.e., any controller can become the primary controller of any switch) and operating with a restoration strategy (i.e., the primary controller of each switch is its closest controller, in terms of the shortest path delay, both in the regular state and in any failure state). Then, for a given data plane network modeled as graph $G = (N, E)$ and a desired number of controllers C, a CPP solution is a set of C controller nodes guaranteeing, in the regular state, a given maximum delay D_{sc} between any switch and its primary controller as well as a given maximum delay D_{cc} between any pair of controllers. In this work, we consider three different CPP solutions, each one corresponding to the optimal solution of a different optimization problem.

1) *BasicRest* SOLUTION

The *BasicRest* (short for Basic Restoration) solution aims to minimize the average delay between switches and their primary controllers in the regular state without any concern with attacks against multiple nodes. This is the most natural goal (i.e., to optimize the regular state delay of the control plane) when no other objective is considered, as introduced in [2]. Note that the minimization of the average delay between all pairs of controllers could also be a goal but, in practice, it is not so relevant since the maximum delay (parameter D_{cc} imposed in all CPP solutions) is the main parameter that impacts the synchronization efficiency between controllers [45].

While in [2], the maximum delay parameters (D_{sc} and D_{cc}) are not considered, this CPP variant is defined in [10] by an ILP model that can be efficiently solved (always below one second of running time) for the network topologies considered in the computational results.

2) *Robust* SOLUTION

The *Robust* solution has the same aim as the *BasicRest* solution (i.e., to minimize the average delay between switches and their primary controllers in the regular state) but now imposing the robustness property in the selection of the C controller placements. The aim is to obtain a solution that optimizes the control plane delay in the regular state while guaranteeing the maximum resilience to the most damaging *CTAs* (i.e., targeted attacks when the attacker has full knowledge of both the data and control planes).

To compute the *Robust* solution, we resort to the enumeration method of all robust CPP solutions described in the previous subsection. For each found solution, we compute its average delay between switches and their primary controllers and the *Robust* solution is the one with the minimum average delay.

3) *Optimal* SOLUTION

The *Optimal* solution aims to maximize the resilience of the CPP solution to the most damaging types of attacks against multiple nodes. To this aim, the robustness property is again imposed (to guarantee the maximum resilience to *CTAs*) and the resilience is maximized to the next most damaging type of attacks (which is *TAs*) against a maximum expected number of $p = C - 1$ nodes. The rationale is that, since there are C controller nodes, the SDN operator aims to optimize the resilience of its network to attacks against the maximum number of nodes that cannot completely shutdown the network, which is $p = C - 1$.

To compute the *Optimal* solution, we consider the set of attacks $m \in M_{TA,C-1}$ composed by the 4 attacks described in Subsection III-A: the three centrality-based attacks using the 'sequential' strategy and the CND-based attack. The *Optimal* solution is computed resorting again to the enumeration method of all robust CPP solutions described in the previous subsection. For each obtained solution, we compute the connectivity-based resilience metric value $r_{TA,C-1}$, as defined in Eq. (5), and the *Optimal* solution is the one with the maximum value of $r_{TA,C-1}$.

V. PROBLEM INSTANCES AND ATTACKS AGAINST MULTIPLE NODES

In this section, we present the different problem instances considered in the computational results, we describe how the different types of attacks against multiple nodes were generated and how the resilience results were obtained for all problem instances and types of attacks (presented and analyzed in the next section).

A. PROBLEM INSTANCES

We have considered two networks, which are amongst the largest networks in the related literature: Germany50 [46] and Coronet CONUS [47]. The network topologies (shown in Fig. 5) are defined over two countries: Germany (Germany50 case) and the USA (Coronet CONUS case), where the geographical coordinates of all network nodes are known.

Based on the geographical coordinates of nodes, we have computed the length of each link (in km) as the shortest path length between the end node coordinates of the link over the Earth's surface. Then, we have computed the diameter of each network as the maximum shortest path length among all pairs of nodes. The topological characteristics of both networks are summarized in Table 1, showing the number of nodes

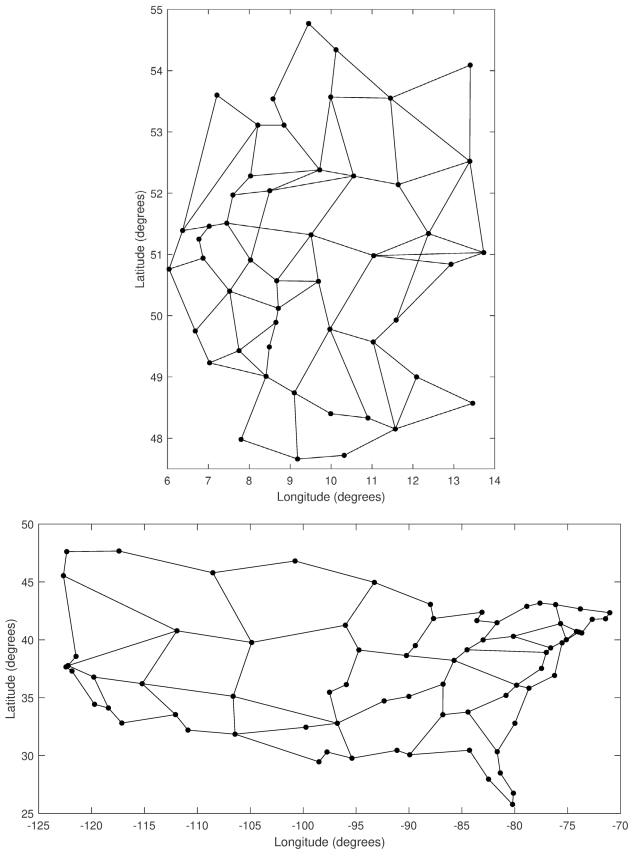

FIGURE 5. Germany50 (top) and Coronet CONUS (bottom).

TABLE 1. Topological characteristics of the networks.

Networks	#nodes	#links	avg deg	diameter [km]
Germany50	50	88	3.52	934
Coronet CONUS	75	99	2.64	6472

('#nodes'), number of bidirectional links ('#links'), average node degree ('avg deg') and diameter ('diameter [km]').

Note that both network topologies are 2-connected, meaning that no single (node or link) failure splits the network–a typical requirement in core telecommunication networks to guarantee full connectivity resilience to single unintended failures. However, the two networks have different topological characteristics. First, the average node degree is much higher for Germany50 than for Coronet CONUS and, consequently, Germany50 has a much higher level of connectivity (more routing paths, on average, between all node pairs) than Coronet CONUS. Furthermore, Germany50 has a more 'regular' shape (i.e., nodes with higher degree are mostly in the central area of the topology and nodes with lower degree are mostly in the border of the topology) than Coronet CONUS, which has many more nodes with low degree values in the center of the topology. As it will be shown later in this paper, these topology differences have an impact on the connectivity-based resilience between the different CPP solutions.

TABLE 2. Characterisation of problem instances.

Network	ID	C	D_{sc}	D_{cc}	#RPsols	$BasicRest$ has RP
Germany50	G1	4	30%	65%	26	Yes
	G2	4	35%	55%	156	Yes
	G3	4	40%	35%	201	Yes
	G4	6	25%	65%	227	Yes
	G5	6	30%	60%	7469	Yes
	G6	6	35%	45%	654	Yes
	G7	8	20%	75%	100	Yes
	G8	8	25%	65%	27603	Yes
	G9	8	30%	60%	100000	No
Coronet CONUS	C1	4	30%	55%	22	Yes
	C2	4	35%	40%	23	No
	C3	4	40%	35%	86	Yes
	C4	6	20%	80%	15	Yes
	C5	6	25%	60%	375	Yes
	C6	6	30%	50%	50	No
	C7	8	20%	65%	2109	Yes
	C8	8	25%	55%	506	Yes
	C9	8	30%	50%	505	No

As in other related works, we assume that the main source of delays is the propagation delay on links and, consequently, the delay over a routing path is proportional to its length (i.e., the sum of the lengths of all links in the routing path). So, we consider delay parameters (e.g., the maximum delay values D_{sc} and D_{cc}) given as percentages of the network diameter.

Note that, in practice, the maximum delay values D_{sc} and D_{cc}, which define the bounds to guarantee a proper SDN control plane performance in the regular state, are dependent not only on the number of controllers to be placed in the network but also on the types of services to be supported by the SDN. For this reason, we have defined 9 different problem instances for each network as shown in Table 2. Each instance has its own ID (Gx for Germany50 and Cx for Coronet CONUS), and they differ on the number of controllers to be placed in the network ($C = 4$, 6 and 8 controllers) and on the D_{sc} and D_{cc} values.

For each problem instance, we have computed the three CPP solutions (*BasicRest*, *Robust* and *Optimal*) as described in Section IV-C. Recall that both *Robust* and *Optimal* solutions use the method described in Section IV-B to enumerate the robust CPP solutions. Column "#RPsols" of Table 2 presents the number of solutions obtained by the enumeration method for each problem instance. We have run the method with a maximum number of $l_{max} = 100000$ solutions and it can be observed that the robust CPP solutions were fully enumerated for all cases except for instance G9 (of Germany50) where l_{max} was reached.

Recall also that both *BasicRest* and *Robust* solutions are computed with the same objective function (i.e., to minimize the average delay between switches and their primary controllers in the regular state). Although the robustness property is not imposed in the *BasicRest* solution, it may still be satisfied by the solution (in such cases, the *BasicRest* and *Robust* solutions are the same CPP solution). The last column of Table 2, named '*BasicRest* has RP', indicates for each problem instance if its *BasicRest* solution is com-

pliant with the robustness property. As shown in this column, there is only one case (instance G9) among all Germany50 instances and three cases (instances C2, C6 and C9) among all Coronet CONUS instances such that the *BasicRest* solution is not compliant with the robustness property. These results give a first indication that many optimal CPP solutions in terms of average delays are also compliant with the robustness property and these cases are more likely to happen in more connected and regular topologies as, e.g., for Germany50.

Concerning the average degradation value n_m of each attack m, defined in Eq. (4), used to compute the connectivity-based resilience of each type of attacks, defined in Eq. (5), recall that we have assumed that the most important parameter is n_{sp}^m, the second most important is n_{sc}^m and the third important parameter is n_{pc}^m, i.e., $\alpha > \beta > \gamma$. To quantify different importance levels of each parameter, we have considered in the computational results two sets of weights following from these assumptions. The first Weight Set (referred as WS1) is $\alpha = 0.5$, $\beta = 0.3$ and $\gamma = 0.2$, while the second Weight Set (referred as WS2) is $\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.1$.

Finally, note that both *BasicRest* and *Robust* solutions are computed independently of the resilience metric and, so, these solutions are unique for each problem instance. On the other hand, the objective function of the *Optimal* solution is the maximization of the connectivity-based resilience metric value which depends on the adopted set of weights. So, we compute one *Optimal* solution for each of the two considered sets of weights (WS1 and WS2).

B. GENERATION OF ATTACKS AGAINST MULTIPLE NODES

In the computational experiments, the sets $M_{A,p}$ of attacks for the resilience evaluation of the different CPP solutions were computed as follows. Concerning *CTAs*, since the attacker knows the location of the controllers and targets only such nodes, we have assumed attacks against p controller nodes, from $p = 1$ up to $p = C - 1$ (recall that if the attacker is able to shut down all C controller nodes, he/she is able to fully disrupt all services). Moreover, the set $M_{CTA,p}$ of attacks against p controller nodes was computed with the three centrality-based attacks (node degree, node closeness and node betweenness) in their 'sequential' variant, as described in Section III-A.

Concerning the other three types of attacks (*TAs*, *RAs* and *EAs*), we have assumed the attacks against p nodes, from $p = 1$ up to $p = 20\%$ of the network nodes (i.e., up to 10 nodes in Germany50 and 15 nodes in Coronet CONUS). Concerning *TAs*, the set $M_{TA,p}$ of attacks against p nodes was computed with the CND-based attack plus the three centrality-based attacks in their 'sequential' variant. In the non-targeted attacks, recall that they are of random nature. So, concerning *RAs*, the set $M_{RA,p}$ of attacks against p nodes was computed with 100 *RAs* generated randomly. Similarly, the set $M_{EA,p}$ of attacks against p nodes was computed with 100 *EAs* also randomly generated.

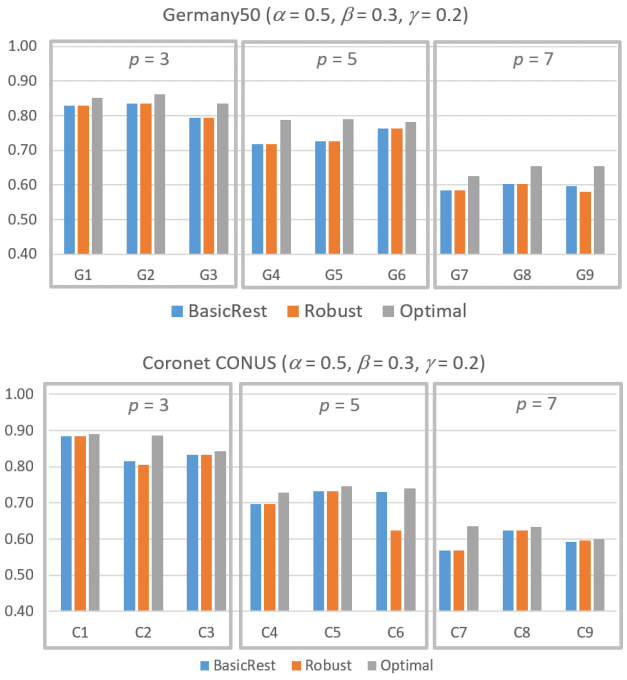

FIGURE 6. Connectivity-based resilience, using WS1, of all Germany50 and Coronet CONUS instances to *TAs* against $p = C - 1$ nodes.

VI. ANALYSIS OF COMPUTATIONAL RESULTS

In this section, we present and analyze the computational results comparing the connectivity-based resilience of the different CPP solutions to the different types of attacks. Section VI-A addresses the evaluation of the CPP solutions to attacks of targeted nature, which are the most disruptive forms of an attack. Then, Section VI-B presents the evaluation of the CPP solutions to attacks of non-targeted nature (random and epidemic attacks). Finally, Section VI-C discusses how an SDN operator can use the findings of the conducted resilience analysis in the context of malicious attacks against multiple nodes.

A. ATTACKS OF TARGETED NATURE

First, consider the resilience analysis for Targeted Attacks (*TAs*). Recall that the *Optimal* solution maximizes the resilience to *TAs* against a maximum expected number of $p = C - 1$ nodes, where C is the number of controller nodes. So, we start the analysis with the results obtained for the connectivity-based resilience metric of the three CPP solutions to *TAs* against $p = C - 1$ nodes. These results are shown in Fig. 6 for Germany50 (top chart) and Coronet CONUS (bottom chart) using WS1 (i.e., $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$) as the set of weights of the resilience metric.

As expected, the resilience of the *Optimal* (grey bar) solution is better (in all problem instances of both networks) than the resilience of the other two solutions (i.e., *BasicRest* and *Robust*). Moreover, the resilience difference between the *Optimal* solution and the best of the other two varies between instances as there are cases with sig-

nificant differences and other cases with small differences. Note that the resilience differences are higher, on average, for the Germany50 instances than for the Coronet CONUS instances, indicating that the connectivity-based resilience gains obtained by the *Optimal* solution are higher in more connected and regular topologies as is the case of Germany50.

Concerning the other two CPP solutions, the resilience of *BasicRest* (blue bar) and *Robust* (orange bar) solutions is different only for G9 (in Germany50) and C2, C6 and C9 (in Coronet CONUS) while, as already seen in Section V, the two CPP solutions are the same for the other instances. Nevertheless, comparing the two solutions in the cases where they are different, the resilience of the *BasicRest* solution is better in G9, C2 and C6, and only slightly worse in C9. This has to do with the fact that the *Robust* solution ensures the robustness property (i.e., if any $C-1$ controller nodes fail, the surviving nodes can still connect to the surviving controller) and, consequently, the controllers are placed further apart, leading to higher SC (switch-controller) delays, on average, after the attacks. The *BasicRest* solution has an average n_{sp}^m value which is either equal or very close to that of the *Robust* solution but the average n_{sc}^m value is better (i.e., higher) since the solution minimizes the SC delays, leading to better connectivity-based resilience, on average, among these cases.

For the interested reader, we present the results when the resilience metric value is computed using WS2 ($\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.1$) in Fig. 19 of the Appendix. Comparing the results in the Appendix with the ones in Fig. 6, we observe that by giving a higher weight to parameter n_{sp}^m, the absolute resilience values become slightly higher and the resilience differences between the different CPP solutions become smaller. Nevertheless, in qualitative terms, the different weight values do not significantly change the conclusions drawn before and this also stands for the resilience assessment, presented next, to attacks against other values of p nodes.

Since the *Optimal* solution is computed to an expected maximum number of nodes $p = C-1$ but, in general, a different number of nodes can be targeted by an attacker, the next analysis is concerned with the resilience assessment of the three CPP solutions to attacks against a number of nodes $p \neq C-1$. As described in the previous section, we have computed the resilience considering the values of p from 1 up to 10 (for Germany50) or 15 (for Coronet CONUS). The obtained results show that, indeed, the resilience of the *Optimal* solution is better, on average, for such cases, although it depends on each particular instance and each particular value of p. In particular:

- In Germany50 instances, the resilience of the *Optimal* solution is always similar or better in all the instances except for G2, G4 and G6. The *Optimal* solution is also similar or better than the other solutions in G2 except for $p = 4$, in G4 except for $p = 7$ and 10 and in G6 except for $p = 4$.

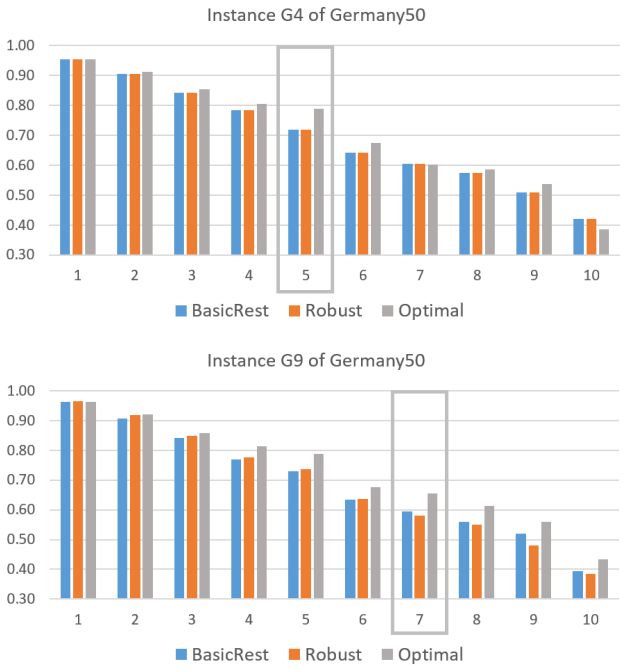

FIGURE 7. Connectivity-based resilience, using WS1, of instances G4 and G9 to *TAs* against $p = 1, \ldots, 10$ nodes.

- In Coronet CONUS instances, the resilience of the *Optimal* solution is always similar or better in all the instances except for C2, C3 and C4. In C2, the resilience of the *Optimal* solution is always much better for $p \geq 2$, while for $p = 1$ it is similar and for $p = 4$ it is slightly worse. In C3, the resilience of the *Optimal* solution is always similar or better, except for $6 \leq p \leq 10$ where it is slightly worse. In C4, the optimal solution is always similar or better, except for $7 \leq p \leq 12$, where it is slightly worse.

For illustrative purposes, we show the resilience results (using WS1) of the three CPP solutions for all values of p (including the expected maximum number of nodes $p = C-1$, highlighted with a box) of two Germany50 instances (G4 and G9, in Fig. 7) and two Coronet CONUS instances (C3 and C6, in Fig. 8). As expected, the connectivity-based resilience decreases in all cases for the attacks against a higher number of nodes as all degradation parameters used to compute the resilience value suffer a higher reduction of their values.

Instances G9 (Fig. 7) and C6 (Fig. 8) are the two examples where the resilience of the *Optimal* solution is better than of the other two solutions for all values of p. On the other hand, instances G4 (Fig. 7) and C3 (Fig. 8) are examples where the *Optimal* solution (which was computed for attacks against $C-1$ nodes) is worse than the other solutions for at least one value of p.

Consider now the resilience analysis for Controller Targeted Attacks (*CTAs*). Since in this type the attacker targets only controller nodes, it is the only type of attack where the attacked nodes depend on the number and location of the

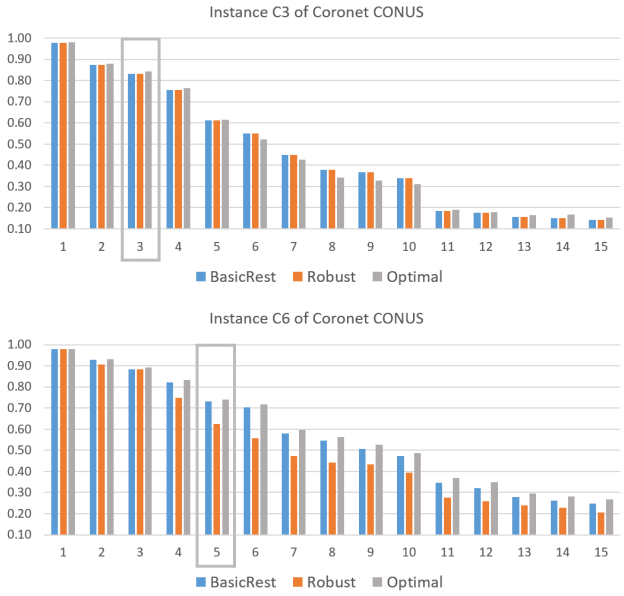

FIGURE 8. Connectivity-based resilience, using WS1, of instances C3 and C6 to *TAs* against $p = 1, \ldots, 15$ nodes.

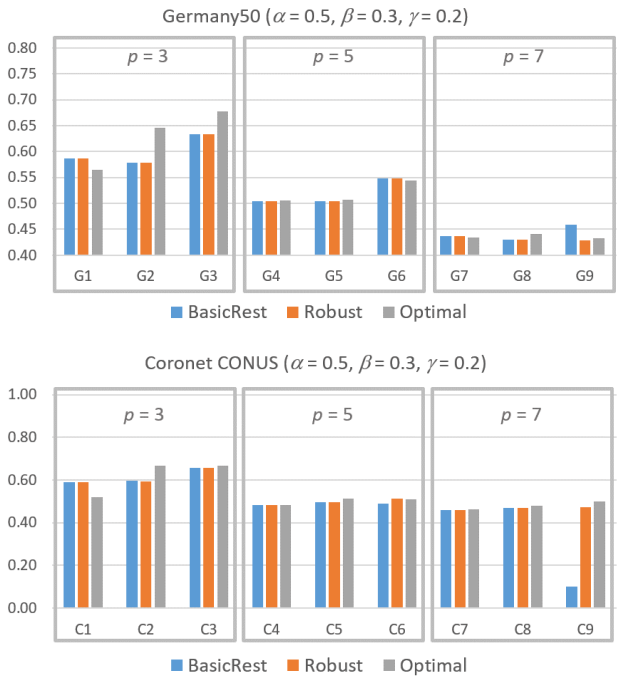

FIGURE 9. Connectivity-based resilience, using WS1, of all Germany50 and Coronet CONUS instances to *CTAs* against $p = C - 1$ controller nodes.

controllers of each CPP solution. As explained in the previous section, we consider attacks up to $p = C - 1$ controller nodes (as shutting down all C controller nodes causes the network to fail completely). The resilience results, using WS1, of the three CPP solutions to *CTAs* against $p = C - 1$ controller nodes are presented in Fig. 9 for Germany50 (top chart) and Coronet CONUS (bottom chart).

For Germany50 (top chart of Fig. 9), we can see that the resilience of the *Optimal* solution is better in instances G2, G3 and G8 and is similar in instances G4, G5, G6 and G7, when compared to the resilience of the other solutions. Note that the *Optimal* solution is computed to optimize the resilience to Targeted Attacks (*TAs*). Therefore, it does not always guarantee that it is also optimal to *CTAs*, as is the case of the instances G1 and G9.

For Coronet CONUS (bottom chart of Fig. 9), we can see that the resilience of the *Optimal* solution is better in instances C2, C5 and C9 and is similar in instances C3, C4, C6, C7 and C8, when compared to the resilience of the other solutions. In this case, there is only one instance (instance C1) such that the *Optimal* solution is outperformed. In the particular case of instance C9 (one of the instances where the *BasicRest* and the *Robust* solutions are not the same), we can observe that by not guaranteeing the robustness property, the *BasicRest* solution has significant smaller connectivity-based resilience than the other solutions.

For the interested reader, we present the resilience results when the resilience metric value is computed using WS2 in Fig. 20 of the Appendix. Comparing the results in the Appendix to the ones in Fig. 9, and contrary to the case of *TAs*, now the resilience values using WS2 are much higher than the ones using WS1. This is due to the fact that the value of parameter n_{sp}^{m} is maximum in all solutions compliant with the robustness property and has a much higher weight in WS2 ($\alpha = 0.7$) than in WS1 ($\alpha = 0.5$). Nevertheless, again the different weight values do not significantly change the conclusions drawn before and this also stands for the resilience results to attacks against other values of p controller nodes.

For the values of $p < C - 1$, in general, we draw the same conclusions as we did for $p = C - 1$, i.e., there are many cases where the resilience of the *Optimal* solution is better but there are also a significant number of cases where the *Optimal* solution is outperformed by the other solutions. For illustrative purposes, we show the resilience results, using WS1, of the three CPP solutions for all values of $p \leq C - 1$ (including the expected maximum number of nodes $p = C - 1$, highlighted with a box) of two Germany50 instances (G4 and G8, in Fig. 10) and two Coronet CONUS instances (C2 and C6, in Fig. 11). In this case, the instances G8 (Fig. 10) and C2 (Fig. 11) are two examples where the resilience of the *Optimal* solution is better than the other solutions for all values of p. In the other hand, in the instances G4 (Fig. 10) and C6 (Fig. 11), the *Optimal* solution is worse than the other solutions for some values of p.

B. ATTACKS OF NON-TARGETED NATURE

First, consider the resilience analysis for Random Attacks (*RAs*). The resilience results, using WS1, of the three CPP solutions for *RAs* against $p = C - 1$ controller nodes are presented in Fig. 12 for Germany50 (top chart) and Coronet CONUS (bottom chart).

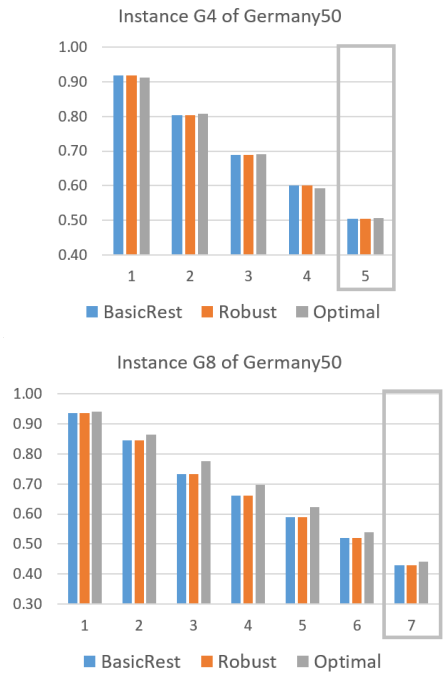

FIGURE 10. Connectivity-based resilience, using WS1, of instances G4 and G8 to *CTAs* against $p = 1, \ldots, C - 1$ nodes.

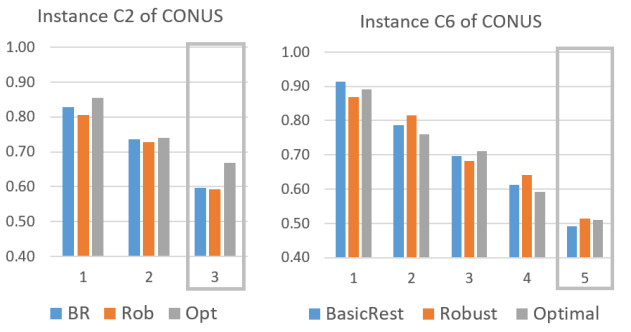

FIGURE 11. Connectivity-based resilience, using WS1, of instances C2 and C6 to *CTAs* against $p = 1, \ldots, C - 1$ nodes.

As expected, the *RAs* are much less disruptive than the attacks of targeted nature. Moreover, in each of the instances, the connectivity-based resilience values of the three solutions (presented in Fig. 12) are similar, showing that, for the expected maximum number $p = C - 1$ of controller nodes, the impact of Random Attacks (*RAs*) is not very different between the three CPP solutions. Although not shown, the resilience results when the connectivity-based resilience metric value is computed using WS2 are very similar to the ones shown in Fig. 12. This is because *RAs* tend to equally affect all degradation parameters and, therefore, the resilience metric values are very close between the two sets of weights (WS1 and WS2).

Since other number of nodes can be targeted by an attacker in a *RA*, the next analysis is concerned with the resilience

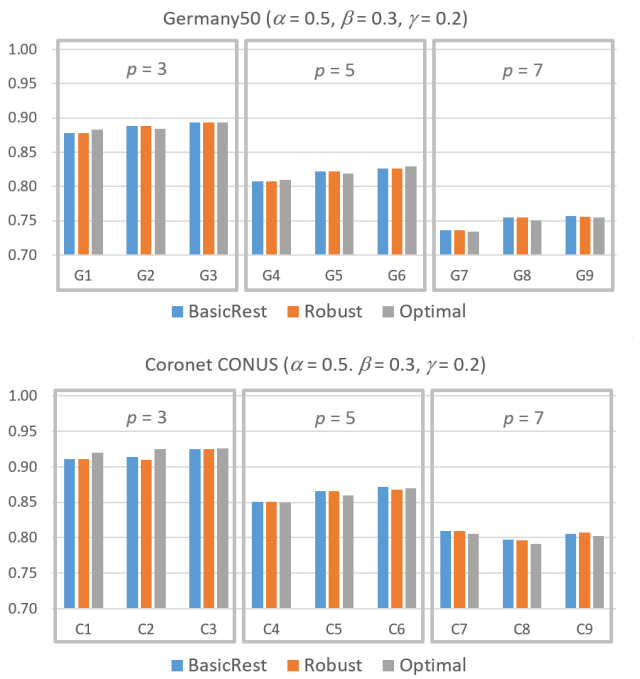

FIGURE 12. Connectivity-based resilience, using WS1, of all Germany50 and Coronet CONUS instances to *RAs* against $p = C - 1$ nodes.

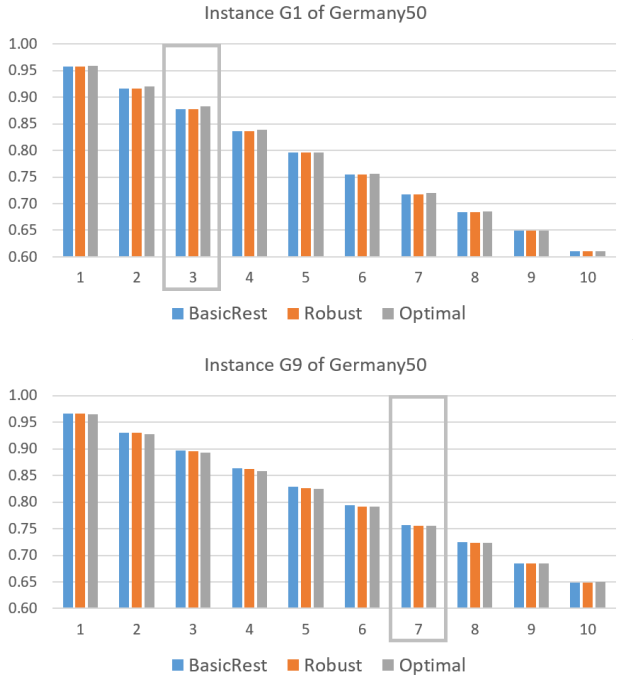

FIGURE 13. Connectivity-based resilience, using WS1, of instances G1 and G9 to *RAs* against $p = 1, \ldots 10$ nodes.

assessment of the three CPP solutions to attacks against $p \neq C-1$ controller nodes. In fact, the computational results show that the resilience of the different solutions is also similar for each problem instance and each value of p. For illustrative purposes, we show in Fig. 13 the resilience results, using WS1, of the three solutions for all values of p (including the

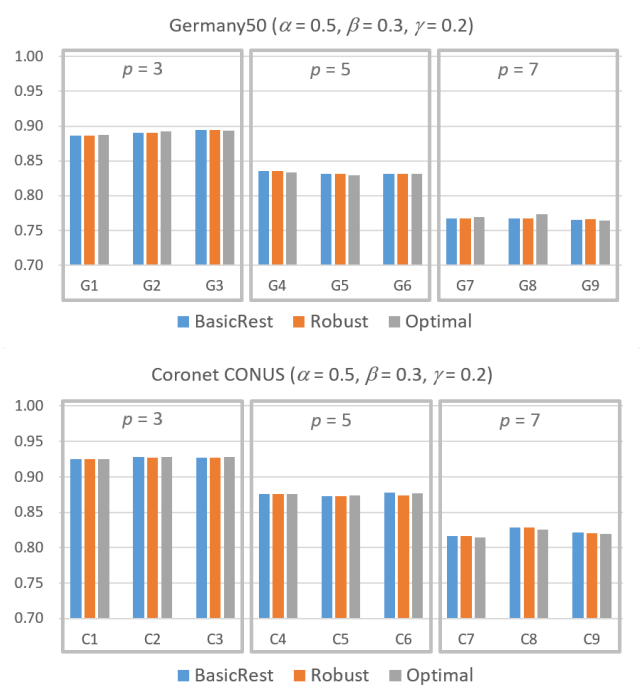

FIGURE 14. Connectivity-based resilience, using WS1, of all Germany50 and Coronet CONUS instances to *EAs* against $p = C - 1$ nodes.

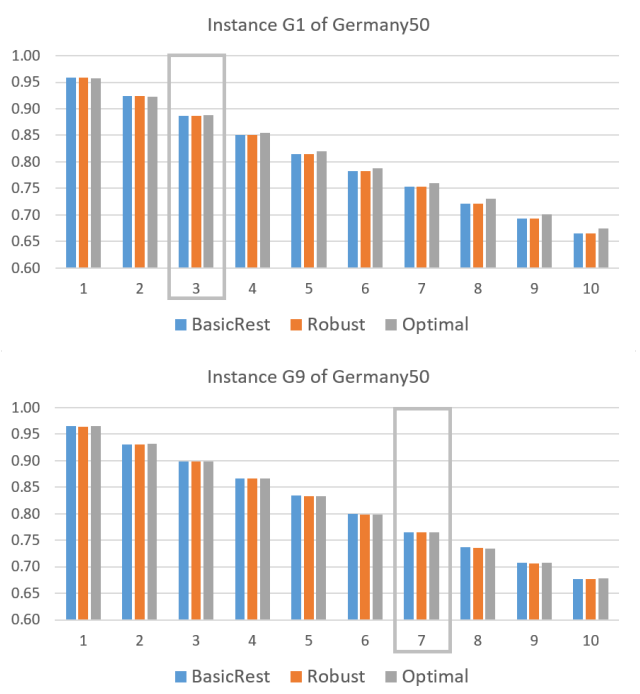

FIGURE 15. Connectivity-based resilience, using WS1, of instances G1 and G9 to *EAs* against $p = 1, \ldots 10$ nodes.

expected maximum number of nodes $p = C - 1$, highlighted with a box) of the Germany50 instances G1 and G9. Instance G1 is an example where the resilience of the *Optimal* solution is better than (or equal to) the best of the other solutions for all values of p. Instance G9 is an example where the resilience of the *Optimal* solution is worse than at least one of the other solutions for some values of p. Note, though, that the resilience differences are always small in both cases and for all values of p.

Consider now the resilience analysis for Epidemic Attacks (*EAs*). The resilience results, using WS1, of the three CPP solutions for *EAs* against $p = C - 1$ controller nodes are presented in Fig. 14 for Germany50 (top chart) and Coronet CONUS (bottom chart).

The *EAs* are less disruptive, in general, than the *RAs*, since each attacked node is adjacent to a previous attacked node. It means that the partitioning of the network into different components with *EAs* is less frequent to happen than with *RAs*. Moreover, the connectivity-based resilience values of the three solutions (presented in Fig. 14) are even more similar for each instance than the values obtained previously when considering *RAs*. These results clearly indicate that, for the expected maximum number $p = C - 1$ of nodes, the impact of *EAs* is almost the same among all CPP solutions. Like with *RAs*, the resilience results for *EAs* when the resilience metric value is computed using WS2 are also very similar to the values shown in Fig. 14.

Concerning the resilience of *EAs* against a number of nodes $p \neq C - 1$, the computational results also show that the resilience of the different solutions is almost the same for each instance of both networks and each value of p. For illustrative purposes, we show in Fig. 15 the resilience results, using WS1, of the three solutions for all values of p of the Germany50 instances G1 and G9. Like in the case of *RAs*, G1 is an example where the resilience of the *Optimal* solution is better than (or equal to) the best of the other solutions for all values of p and G9 is an example where the resilience of the *Optimal* solution is worse than at least one of the other solutions for some values of p. Comparing the results in Fig. 15 with the ones in Fig. 13 (which considers the same instances for *RAs*), we can see that the resilience differences among the different solutions of each value of p are even smaller for *EAs* than for *RAs*.

C. VIEWPOINT OF THE SDN OPERATOR

This section presents two analyses of interest to an SDN operator at evaluating the impact of the controller placements in the connectivity-based resilience of its network to attacks against multiple nodes. The first analysis focuses on a given number of controllers while the second analysis investigates the possibility of considering a different number of controllers for given maximum regular state delay constraints.

1) ANALYSIS OF ONE INSTANCE OVER ALL TYPES OF ATTACKS

In this analysis, the SDN operator aims to compare the three different CPP solutions for its particular case defined by the required number of controllers C and the required maximum delays (D_{sc} and D_{cc}) in the regular state. As the operator

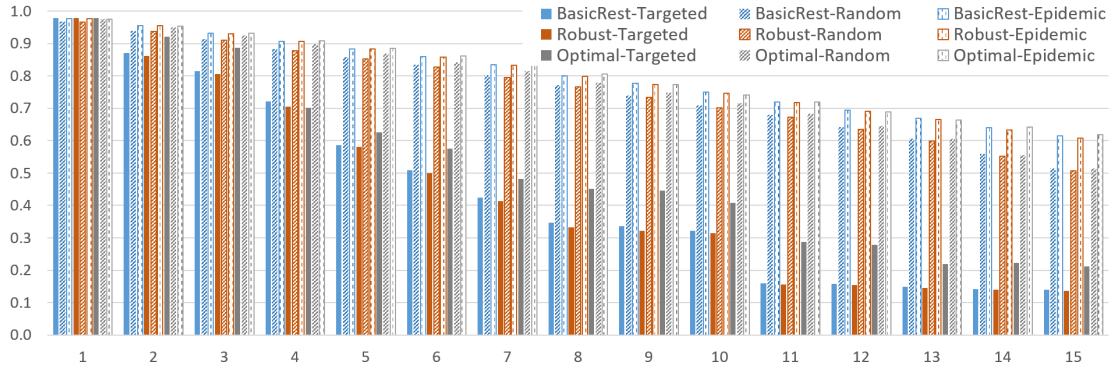

FIGURE 16. Connectivity-based resilience, using WS1, of instance C2 to all types of attacks against $p = 1, \ldots 15$ nodes.

cannot foresee the attacks that the network will have to face, this analysis aims at evaluating the resilience over all types of attacks up to x shutdown nodes. In this analysis, we exclude the *CTAs* as they might be of little interest to the operator due to the fact that the probability of the attacker knowing the controller locations is very small.

Let us take as an example instance C2 of Coronet CONUS with $x = 15$. Fig. 16 presents the connectivity-based resilience of the three CPP solutions (*BasicRest* in blue, *Robust* in orange and *Optimal* in grey) to the three types of attacks (*TA* in color, *RA* with diagonal dashes and *EA* with vertical dashes).

In this particular instance, we can immediately observe that the resilience of the three CPP solutions is much worse (i.e., lower) for *TAs* than for the other two types of attacks in all values of $p \leq x$. Moreover, the resilience is also slightly worse for *RAs* than for *EAs*. Concerning the resilience between the three CPP solutions, it can be easily observed that although the resilience for *RAs* and *EAs* is similar between the three solutions, the *Optimal* solution offers a higher resilience for *TAs* than the other two solutions against any number of shutdown nodes, except $p = 4$. Furthermore, the variation of the connectivity-based resilience value for the *Robust* and the *Optimal* solutions with respect to the *BasicRest* solution can also be evaluated. The distribution of the variation over the 15 attack scenarios of this instance is presented in Fig. 17, which clearly shows that the *Optimal* solution improves the resilience metric for *TAs* by an average of 30% with respect to the *BasicRest* solution, whereas for the *Robust* solution and for the other attack types (*RAs* and *EAs*), the difference is within $\pm$ 5%.

So, the conclusion for this particular problem instance is that, if the SDN operator aims to obtain the best connectivity-based resilience for any type of attack, the *Optimal* solution should be adopted as it offers a significant resilience improvement to *TAs* and a resilience slightly better than the other two CPP solutions for *RAs* and *EAs*.

However, this conclusion cannot be generalized as other instances may show different behaviors. For example, the improvement for two other instances (C6 and C7) are presented in Fig. 21 of the Appendix. In these two instances,

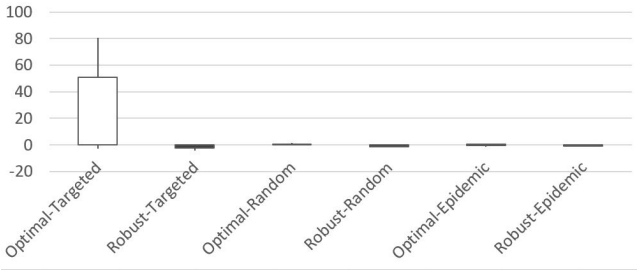

FIGURE 17. Connectivity-based resilience improvement (in %) of ***Optimal*** and ***Robust*** solutions of C2 with respect to the ***BasicRest*** solution over the 15 attack scenarios.

although the *Optimal* solution outperforms the *BasicRest* solution for *TAs*, it offers much less connectivity-based resilience gains for this type of attacks. When the operator is concerned with the three types of attacks, the *Optimal* solution is still the one providing the best connectivity-based resilience among the three solutions although followed closely by the *BasicRest* solution. However, for an operator concerned only with attacks of non-targeted nature (i.e., *RAs* and/or *EAs*), the *BasicRest* solution results in slightly better average resilience and, therefore, this solution is preferable as it optimizes the regular state delay of the SDN.

2) ANALYSIS OF DIFFERENT INSTANCES WITH THE SAME MAXIMUM DELAYS IN THE REGULAR STATE

In this analysis, we consider an SDN operator that, given its network and the required maximum delays (D_{sc} and D_{cc}) in the regular state, can deploy a different number of controllers and aims to analyze the connectivity-based resilience gains that can be obtained by deploying more controllers. In this analysis, we assume that the operator is only interested in the resilience of its network to *TAs* as they are much more damaging than *RAs* and *EAs*.

Let us consider as an example two instances with a different number of controllers C and with the same maximum delays: Coronet CONUS instances C6 and C9 with 6 and 8 controllers, respectively, and with the same $D_{sc} = 30\%$ and $D_{cc} = 50\%$ maximum delays (refer to Table 2). Fig. 18 presents the resilience of the three CPP solutions of these two instances to *TAs* against up to 15 nodes.

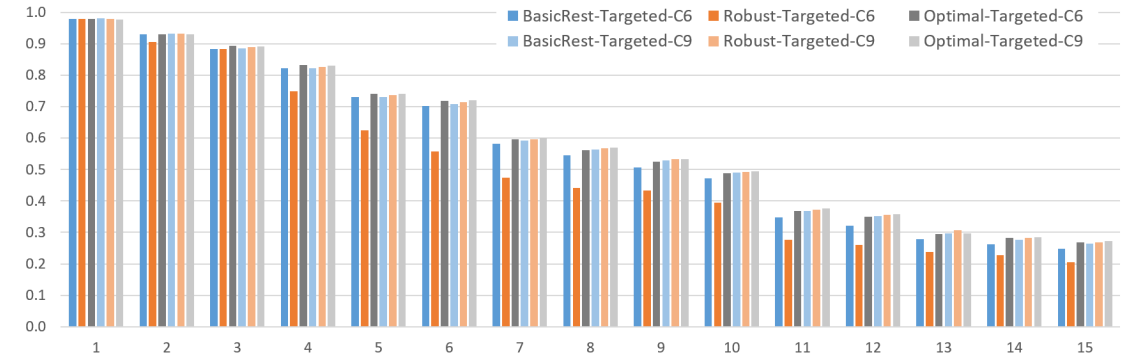

FIGURE 18. Connectivity-based resilience, using WS1, of instances C6 (in color) and C9 (in dashed) to *TAs* against $p = 1, \ldots 15$ nodes.

When comparing the resilience of each CPP solution between the two instances, Fig. 18 shows that for the *BasicRest* and the *Robust* solutions, the two additional controllers in instance C9 improve the resilience of the solutions in many of the attack scenarios. On the other hand, in the case of the *Optimal* solutions, not only do they offer a better resilience than the other two solutions but also the resilience of these solutions between the two instances is very similar in all attack scenarios. In this example, the SDN operator should use the *Optimal* solution with 6 controllers (instead of 8 controllers) as it implies network resource savings (as the control traffic is lower, for example) without degrading the connectivity-based resilience of the solution to attacks of *TA* type.

Again, this conclusion cannot be generalized to all cases. In our problem instances, there are two other examples of pairs of instances with a different number of controllers and the same maximum delays in the regular state: the pair G4 and G8 and the pair G5 and G9. Both pairs involve instances of Germany50 and, in both cases, one instance considers $C = 6$ controllers and the other instance considers $C = 8$ controllers (again, refer to Table 2).

In the Appendix, we present the resilience of the three CPP solutions to *TAs* against up to 10 nodes of instances G4 and G8 (Fig. 22) and instances in G5 and G9 (Fig. 23). In these cases, the resilience obtained by the *Optimal* solution with 6 controllers is similar to the one with 2 additional controllers only for the attack scenarios up to 5 nodes. For higher numbers of shutdown nodes, the 2 additional controllers indeed enable obtaining resilience gains for *TAs*. In such cases, the decision on how many controllers should be deployed depends on the importance given by the SDN operator to the connectivity-based resilience of its network to attacks against a higher number of nodes.

VII. CONCLUSION

In this work, we have addressed the Controller Placement Problem (CPP) of SDNs considering that in the regular state the control plane must guarantee a given maximum delay between every switch and its primary controller and a given maximum delay between every pair of controllers. Since in general these delay bounds allow multiple solutions, we have investigated the connectivity-based resilience to attacks against multiple network nodes of the CPP solutions obtained with three different aims: the regular state delay optimization without any concern about attacks, the regular state delay imposing the robustness property and the resilience optimization to attacks against multiple nodes. Moreover, the resilience assessment has considered attacks of targeted nature (when the attacker has complete knowledge of the data plane) and attacks of non-targeted nature (i.e., random and epidemic attacks).

To this aim, we have first defined a connectivity-based resilience metric that measures the average impact of each type of attack in both the data and control planes. The proposed metric considers three degradation parameters (the number of switch pairs able to support traffic flows after an attack, the number of switches served by the control plane within the maximum acceptable delay after an attack and the number of switches that maintain their primary controller after an attack) and allows the SDN operator to quantify the importance of each parameter on the services supported by its network.

Then, we have used the proposed metric to assess the connectivity-based resilience of the three CPP solutions to the different types of attacks against different numbers of nodes. To this aim, we have considered different problem instances defined over Germany50 and Coronet CONUS networks, which are amongst the largest network topologies considered in the related literature.

The main conclusion of the conducted analysis is that the connectivity-based resilience strongly depends on the network topology, the considered regular state delay bounds and the type of attacks: in attacks of non-targeted nature, the different CPP solutions present similar resilience values while in the attacks of targeted nature, there are cases with significant resilience gains when the controller placements are selected taking into consideration the attacks. In particular, we have shown that the *Optimal* solution is the best when attacks of targeted nature are the main concern of

the SDN operator. This is very clear when the operator is concerned with attacks against a number of nodes $p = C - 1$ (recall that C is the number of controllers) and it is also the case, on average, for attacks against other values of p. On the other hand, for attacks of non-targeted nature, since they are much less disruptive than attacks of targeted nature, the resilience of the *BasicRest* and *Optimal* solutions become closer and, in this case, there are many cases where the *BasicRest* solution is preferable as it optimizes the SDN control plane performance in the regular state. The *Robust* solution, which is proposed to maximize the resilience of the most damaging type of attacks targeting only controller nodes, in practice, corresponds to the *BasicRest* solution as, in many cases, the latter solution is compliant with the robustness property.

Finally, we have also discussed the viewpoint of the SDN operator on how the findings of the conducted resilience assessment can be used in the context of malicious attacks against multiple nodes. First, we have shown how the SDN operator can determine which of the three CPP solutions is the best depending on its particular view on the connectivity-based resilience of its network. Then, we have shown how the SDN operator can determine when an additional number of controllers can obtain resilience gains for attacks against multiple nodes.

APPENDIX

This Appendix presents additional computational results supporting the conclusions that are taken in the resilience assessments conducted in Section VI.

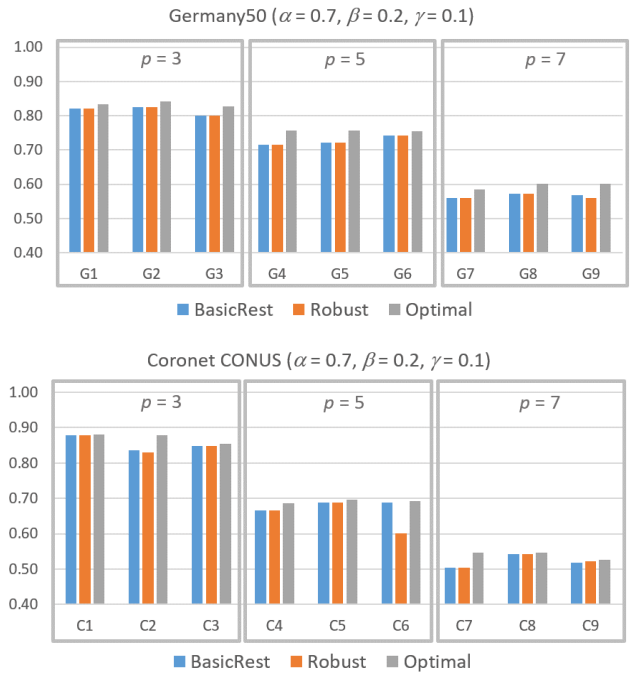

FIGURE 19. Connectivity-based resilience, using WS2, of all Germany50 and Coronet CONUS instances to *TAs* against $p = C - 1$ nodes.

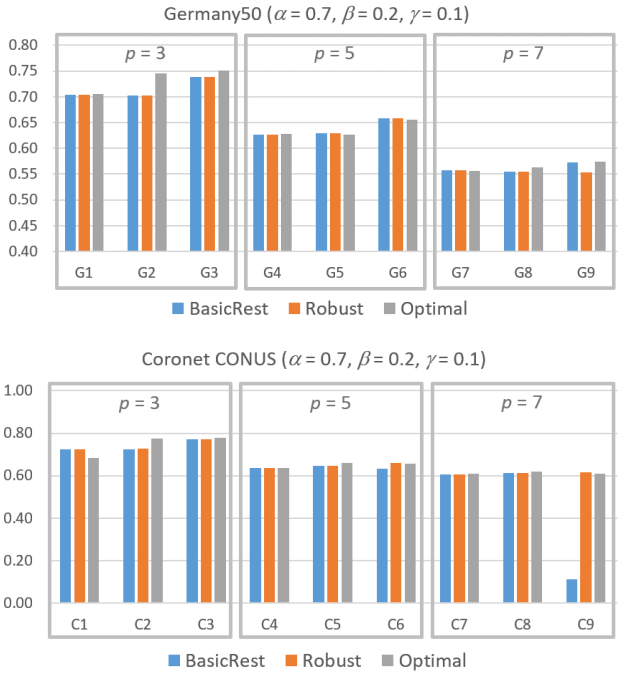

FIGURE 20. Connectivity-based resilience, using WS2, of all Germany50 and Coronet CONUS instances to *CTAs* against $p = C - 1$ controller nodes.

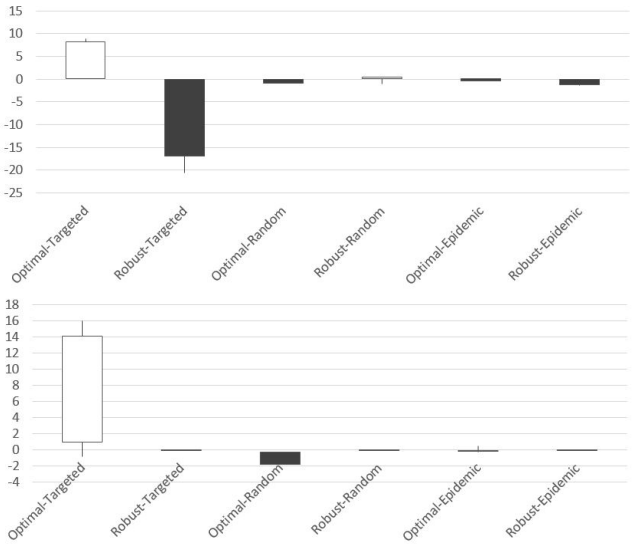

FIGURE 21. Connectivity-based resilience improvement (in %) of *Optimal* and *Robust* solutions of C6 (top) and C7 (bottom) with respect to the *BasicRest* solution over the 15 attack scenarios.

FIGURE 22. Connectivity-based resilience, using WS1, of instances G4 (in dark colors) and G8 (in light colors) to *TAs* against $p = 1, \ldots 10$ nodes.

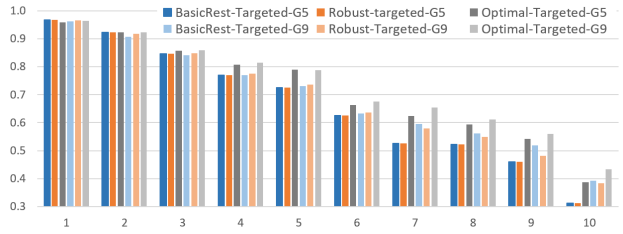

FIGURE 23. Connectivity-based resilience, using WS1, of instances G5 (in dark colors) and G9 (in light colors) to *TA*s against $p = 1, \ldots 10$ nodes.

REFERENCES

[1] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): A survey," *J. Netw. Comput. Appl.*, vol. 80, pp. 200–218, Feb. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516303186

[2] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2012, pp. 7–12, doi: 10.1145/2342441.2342444.

[3] L. Wolsey, *Integer Programming*, 2nd ed. Hoboken, NJ, USA: Wiley, Sep. 2020, doi: 10.1002/9781119606475.

[4] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," *Comput. Netw.*, vol. 112, pp. 279–293, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S138912861630411X

[5] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, "Multi-controller based software-defined networking: A survey," *IEEE Access*, vol. 6, pp. 15980–15996, 2018.

[6] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "OpenFlow: Meeting carrier-grade recovery requirements," *Comput. Commun.*, vol. 36, no. 6, pp. 656–665, Mar. 2013.

[7] J. Rak and D. Hutchison, Eds., "Guide to disaster-resilient communication networks," in *Computer Communications and Networks*. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-44685-7.

[8] J. Rak, D. Hutchison, E. Calle, T. Gomes, M. Gunkel, P. Smith, J. Tapolcai, S. Verbrugge, and L. Wosinska, "RECODIS: Resilient communication services protecting end-user applications from disaster-based failures," in *Proc. 18th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2016, pp. 1–4.

[9] B. Cai, M. Xie, Y. Liu, Y. Liu, and Q. Feng, "Availability-based engineering resilience metric and its corresponding evaluation methodology," *Rel. Eng. Syst. Saf.*, vol. 172, pp. 216–224, Apr. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832017307573

[10] D. Santos, A. de Sousa, and C. Mas-Machuca, "Robust SDN controller placement to malicious node attacks," in *Proc. 21st Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2018, pp. 1–8.

[11] D. Santos, A. de Sousa, and C. Mas-Machuca, "The controller placement problem for robust SDNs against malicious node attacks considering the control plane with and without split-brain," *Ann. Telecommun.*, vol. 74, nos. 9–10, pp. 575–591, Oct. 2019.

[12] D. Santos, A. de Sousa, and C. Mas-Machuca, "Combined control and data plane robustness of SDN networks against malicious node attacks," in *Proc. CNSM*, Rome, Italy, Nov. 2018, pp. 54–62.

[13] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, 2013, pp. 672–675.

[14] Y. Zhang, N. Beheshti, and M. Tatipamula, "On resilience of split-architecture networks," in *Proc. IEEE Global Telecommun. Conf. GLOBECOM*, Dec. 2011, pp. 1–6.

[15] L. F. Muller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspary, and M. P. Barcellos, "Survivor: An enhanced controller placement strategy for improving SDN survivability," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 1909–1915.

[16] P. Vizarreta, C. Mas-Machuca, and W. Kellerer, "Controller placement strategies for a resilient SDN control plane," in *Proc. 8th Int. Workshop Resilient Netw. Design Model. (RNDM)*, Sep. 2016, pp. 253–259.

[17] M. Guo and P. Bhattacharya, "Controller placement for improving resilience of software-defined networks," in *Proc. 4th Int. Conf. Netw. Distrib. Comput.*, Dec. 2013, pp. 23–27.

[18] S. S. Savas, M. Tornatore, F. Dikbiyik, A. Yayimli, C. U. Martel, and B. Mukherjee, "RASCAR: Recovery-aware switch-controller assignment and routing in SDN," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1222–1234, Dec. 2018.

[19] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. 25th Int. Teletraffic Congr. (ITC)*, Sep. 2013, pp. 1–9.

[20] L. Li, N. Du, H. Liu, R. Zhang, and C. Yan, "Towards robust controller placement in software-defined networks against links failure," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, Apr. 2019, pp. 216–223.

[21] S. Yang, L. Cui, Z. Chen, and W. Xiao, "An efficient approach to robust SDN controller placement for security," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 3, pp. 1669–1682, Sep. 2020.

[22] N. Perrot and T. Reynaud, "Optimal placement of controllers in a resilient SDN architecture," in *Proc. 12th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Mar. 2016, pp. 145–151.

[23] D. F. Rueda, E. Calle, and J. L. Marzo, "Improving the robustness to targeted attacks in software defined networks (SDN)," in *Proc. 13th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Mar. 2017, pp. 1–8.

[24] S. G. Cosgaya, E. Calle, and J. L. Marzo, "Resilient controller location under target attacks," in *Proc. 20th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2018, pp. 1–5.

[25] E. Calle, S. G. Cosgaya, D. Martinez, and M. Pioro, "Solving the backup controller placement problem in SDN under simultaneous targeted attacks," in *Proc. 11th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Oct. 2019, pp. 1–7.

[26] M. Pióro, M. Mycek, and A. Tomaszewski, "Using probabilistic availability measures for predicting targeted attacks on network nodes," in *Proc. 4th Cyber Secur. Netw. Conf. (CSNet)*, Oct. 2020, pp. 1–8.

[27] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos, "Detecting critical nodes in sparse graphs," *Comput. Oper. Res.*, vol. 36, no. 7, pp. 2193–2200, Jul. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305054808001494

[28] Y. Rochat, "Closeness centrality extended to unconnected graphs: The harmonic centrality index," in *Applications of Social Network Analysis (ASNA)*. Zürich, Switzerland, Aug. 2009.

[29] D. F. Rueda, E. Calle, and J. L. Marzo, "Robustness comparison of 15 real telecommunication networks: Structural and centrality measurements," *J. Netw. Syst. Manage.*, vol. 25, no. 2, pp. 269–289, Apr. 2017.

[30] A. de Sousa and D. Santos, *Vulnerability Evaluation of Networks to Multiple Failures Based on Critical Nodes and Links*. Cham, Switzerland: Springer, 2020, pp. 63–86, doi: 10.1007/978-3-030-44685-7.

[31] M. Lalou, M. A. Tahraoui, and H. Kheddouci, "The critical node detection problem in networks: A survey," *Comput. Sci. Rev.*, vol. 28, pp. 92–117, May 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574013716302416

[32] M. Di Summa, A. Grosso, and M. Locatelli, "Branch and cut algorithms for detecting critical nodes in undirected graphs," *Comput. Optim. Appl.*, vol. 53, no. 3, pp. 649–680, Dec. 2012, doi: 10.1007/s10589-012-9458-y.

[33] A. Veremyev, V. Boginski, and E. L. Pasiliao, "Exact identification of critical nodes in sparse networks via new compact formulations," *Optim. Lett.*, vol. 8, no. 4, pp. 1245–1259, Apr. 2014, doi: 10.1007/s11590-013-0666-x.

[34] D. Santos, A. de Sousa, and P. Monteiro, "Compact models for critical node detection in telecommunication networks," *Electron. Notes Discrete Math.*, vol. 64, pp. 325–334, Feb. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1571065318300349

[35] M. Manzano, E. Calle, J. Ripoll, A. M. Fagertun, and V. Torres-Padrosa, "Epidemic survivability: Characterizing networks under epidemic-like failure propagation scenarios," in *Proc. 9th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Mar. 2013, pp. 95–102.

[36] T. G. Lewis, *Network Science: Theory and Applications*. Hoboken, NJ, USA: Wiley, 2009.

[37] M. E. J. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford Univ. Press, 2010.

[38] R. Cohen and S. Havlin, *Complex Networks: Structure, Robustness and Function*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[39] E. Calle, J. Ripoll, J. Segovia, P. Vilà, and M. Manzano, "A multiple failure propagation model in GMPLS-based networks," *IEEE Netw.*, vol. 24, no. 6, pp. 17–22, Nov. 2010.

[40] M. López, A. Peinado, and A. Ortiz, "A SEIS model for propagation of random jamming attacks in wireless sensor networks," in *Proc. Int. Joint Conf. SOCO CISIS ICEUTE*, M. Graña, J. M. López-Guede, O. Etxaniz, Á. Herrero, H. Quintián, and E. Corchado, Eds. Cham, Switzerland: Springer, 2017, pp. 668–677.

[41] A. Bishop, I. Z. Kiss, and T. House, "Consistent approximation of epidemic dynamics on degree-heterogeneous clustered networks," in *Complex Networks and Their Applications VII*, L. M. Aiello, C. Cherifi, H. Cherifi, R. Lambiotte, P. Lió, and L. M. Rocha, Eds. Cham, Switzerland: Springer , 2019, pp. 376–391.

[42] M. Manzano, E. Calle, J. Ripoll, A. M. Fagertun, V. Torres-Padrosa, S. Pahwa, and C. Scoglio, "Epidemic and cascading survivability of complex networks," in *Proc. 6th Int. Workshop Reliable Netw. Design Modeling (RNDM)*, Nov. 2014, pp. 187–193.
[43] L. Feng, L. Song, Q. Zhao, and H. Wang, "Modeling and stability analysis of worm propagation in wireless sensor network," *Math. Problems Eng.*, vol. 2015, pp. 1–8, Aug. 2015.
[44] Z. Tao, F. Zhongqian, and W. Binghong, "Epidemic dynamics on complex networks," *Prog. Natural Sci.*, vol. 16, no. 5, pp. 452–457, 2006.
[45] T. Zhang, A. Bianco, and P. Giaccone, "The role of inter-controller traffic in SDN controllers placement," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 87–92.
[46] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "SNDlib 1.0—Survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.
[47] *Sample Optical Network Topology Files*. Accessed: Apr. 9, 2021. [Online]. Available: http://www.monarchna.com/topology.html

DORABELLA SANTOS received the M.Sc. degree in mathematics and the Ph.D. degree in electrotechnical engineering from the University of Aveiro, Portugal, in 2003 and 2007, respectively. She was a Postdoctoral Researcher with the Instituto de Telecomunicações, Aveiro, from 2008 to 2018. She has been a Research Assistant with the Institute for Systems Engineering and Computers at Coimbra (INESC Coimbra), since December 2018. She has coauthored several scientific publications in international journals, conference proceedings, and book chapters. Her research interests include optimization problems for telecommunication networks involving network design, traffic engineering, and more recently, software-defined networking. She also served as a TPC Member for some international conferences.

AMARO DE SOUSA received the five-year B.S. degree in electronics and telecommunications engineering from the University of Aveiro, Portugal, in 1989, the M.Sc. degree in telecommunications engineering from the University College of North Wales, Bangor, U.K., in 1991, and the Ph.D. degree in electrical engineering from the University of Aveiro, in 2001. He is currently an Assistant Professor with the Department of Electronics, Telecommunications and Informatics, University of Aveiro. He is also a Senior Researcher and the Coordinator of the Applied Mathematics Group, Instituto de Telecomunicações–Pole of Aveiro, Portugal. He has been involved in different European Union funded and Portuguese funded projects in the last 20 years. He has authored over 100 publications including papers in refereed international journals and conferences and book chapters. His research interests include advanced services and protocols for telecommunications, traffic engineering and network design, and optimization algorithms for efficient management of telecommunication network resources.

CARMEN MAS-MACHUCA (Senior Member, IEEE) received the Dipl.-Ing. degree (master's) from the Universitat Politècnica de Catalunya, UPC, Spain, in 1995, and the Dr.-Ing. degree (Ph.D.) from the École Polytechnique Fédérale de Lausanne, EPFL, Switzerland, in 2000. She is currently a Privat Dozent/Adjunct Teaching Professor with the Chair of Communication Networks, Technical University of Munich (TUM), Germany. She has published more than 150 peer-reviewed articles. Her main research interests include techno-economic studies, network planning and resilience, SDN/NFV optimization problems, and next generation converged access networks. She is currently the NoF'21General Co-Chair, the ONDM'21 TPC Co-Chair, and the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT (IEEE TNSM) Associate Co-Editor of the Special Issue on "Design and Management of Reliable Communications Network."

JACEK RAK (Senior Member, IEEE) received the M.Sc., Ph.D., and D.Sc. (Habilitation) degrees from the Gdańsk University of Technology, Gdańsk, Poland, in 2003, 2009, and 2016, respectively.

He is currently an Associate Professor and the Head of the Department of Computer Communications, Gdańsk University of Technology. From 2016 to 2020, he was leading the COST CA15127 Action "*Resilient Communication Services Protecting End-user Applications from Disaster-based Failures*" (RECODIS) involving over 170 members from 31 countries. He has authored over 100 publications, including the monograph *Resilient Routing in Communication Networks* (Springer, 2015) and co-edited the book *Guide to Disaster-Resilient Communication Networks* (Springer, 2020). His main research interests include resilience of communication networks and networked systems.

Prof. Rak is a member on the Editorial Board of *Optical Switching and Networking*, (Elsevier) and the Founder of the Workshop on *Resilient Networks Design and Modeling* (RNDM). He has also served as a TPC Member for numerous conferences and journals. Recently, he has been the General Chair of ITS-T'17 and MMM-ACNS'17, the General Co-Chair of NETWORKS'16, the TPC Chair of ONDM'17, and the TPC Co-Chair of IFIP Networking'19.