Enabling Cyber-attack Mitigation Techniques in a Software Defined Network

Achilleas Pasias*, Thanasis Kotsiopoulos*†, Georgios Lazaridis*, Anastasios Drosou*,
Dimitrios Tzovaras* and Panagiotis Sarigiannidis†
*Information Technologies Institute, Centre for Research & Technology Hellas
57001 Thermi, Greece Email: pasiasach@iti.gr
†Dept. of Electrical and Computer Engineering, University of Western Macedonia, Karamanli & Ligeris Street
50100 Kozani, Greece

2021 IEEE International Conference on Cyber Security and Resilience (CSR) | 978-1-6654-0285-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSR51186.2021.9527932

***Abstract*—Software Defined Networking (SDN) is an innovative technology, which can be applied in a plethora of applications and areas. Recently, SDN has been identified as one of the most promising solutions for industrial applications as well. The key features of SDN include the decoupling of the control plane from the data plane and the programmability of the network through application development. Researchers are looking at these features in order to enhance the Quality of Service (QoS) provisioning of modern network applications. To this end, the following work presents the development of an SDN application, capable of mitigating attacks and maximizing the network's QoS, by implementing mixed integer linear programming but also using genetic algorithms. Furthermore, a low-cost, physical SDN testbed was developed in order to evaluate the aforementioned application in a more realistic environment other than only using simulation tools.**

I. Introduction

The integration of sophisticated and intelligent techniques into the traditional Industrial Control Systems (ICS) has resulted in the emergence of a distinct technological ecosystem. Sensors and actuators, inventory monitoring systems, industrial equipment, video surveillance cameras, and generic PCs and networking devices all fall under the umbrella of modern ICS.These provide advanced services and functionality, boost operational benefits such as power, liability, and protection, and make new infrastructure paradigms easier to introduce (e.g., the Smart Grid) [1].

Although the industrial revolution provided many benefits, it also presented new design issues and exposed ICS to a new wave of cyber-physical attacks. Several reports include different type of attacks conducted in ICS. The cyber-attack at the Ukrainian electricity grid [2] demonstrated the exceptional impact of cyber-physical attacks, where an ordinary malware infection may leave regions without electricity. For this reason, different countermeasures are taken in order to protect these types of critical infrastructures, such as the development of AI models in order to accurately detect cyber-threats [3], the reactive or proactive mitigation of the attacks [4] or the appliance of SDN in order to increase the survivability of the communication network [1]. Characteristically, in [5] the aspects of energy consumption of the communication infrastructure, the network QoS and security criteria are taken into account in order to mitigate the impact of a cyber-attack, while in [6], the authors propose an online routing self-trained decision-making algorithm based on Deep Reinforcement Learning that takes decisions based on the current state of the network by optimizing the QoS and security.

With the centralized control of network activity, the global visibility of the network state, and the run-time manipulation of traffic forwarding rules, SDN improves network security. The centralized networking of an SDN environment allows network-wide security policies to be enforced, thus reducing the possibility of policy collisions. In details, by decoupling the control plane (i.e., the software suite that makes routing decisions) from the forwarding plane, SDN enables the development of virtual networking services and the implementation of global networking decisions (i.e., the equipment that forwards the traffic) [7], [8]. SDN builds on the OpenFlow [9] open standard protocol, to allow communication with remote devices.

To this end, the authors of this paper propose a framework to mitigate cyber-attacks and to maximize the QoS of the SDN, called Electrical Data Analysis Engine (EDAE), which is evaluated upon of a physical SDN testbed in order to validate the aforementioned algorithms in a proof-of-concept scenario.

The rest of the paper is organized as follows: Section II analyzes the functionality and mathematical formulation of EDAE. Subsequently, section III presents the evaluation setup and the results of EDAE. Finally, section IV concludes the technologies and the results presented in the manuscript.

II. Component Architecture

This section is devoted to thoroughly analyze the functionality of the developed component, namely as the Electrical Data Analysis Engine (EDAE), which is a component of the SDN-microSENSE H2020 project [10].

A. EDAE

The primary functions of EDAE are to continuously provide alternate network routes between critical applications in order to ensure that the QoS and security requirements are covered. Second, to isolate a network component that

has been identified as attacked. Third, to optimally reconnect the gateways with the sensing units, when a gateway is disconnected after a malfunction or a cyber-attack, in order to restore the data acquisition and in the Electrical Power Energy Systems (EPES) use case and to restore and maximize the redundant observability of the Electrical grid. In general EDAE consists of two algorithms. A genetic algorithm for the communication path reconstruction and device isolation. A Mixed Integer Linear Programming algorithm to perform the matchmaking between the leftover gateways and the sensing devices.

TABLE I: Table of Notations

SW	=	Set of communication network forwarding devices (switches)
BS	=	Set of buses in the power transmission network
SD	=	Set of sensing devices in the network
GW	=	Set of gateways in the network
L	=	Set of active communication paths between hosts
$DL_c(h_i)$	=	Delay constraint for host i
$AB_c(h_i)$	=	Available bandwidth constraint for host i
$PL_c(h_i)$	=	Packet loss constraint for host i
$JTR_c(h_i)$	=	Jitter constraint for host i
$SR(h_i)$	=	Security constraint for host i
L_{bs_i,bs_j}	=	Set of transmission lines between buses
k_{GW}	=	Device capacity of each Gateway
p_{h_i,h_j}	=	A full communication path between two hosts
bs_i	=	Bus number i
h_i	=	Host i
s_i	=	Switch i
a_i	=	A network asset could be switch or host
$< a_i, a_j >$	=	Link between two assets of the network
$sr_{<s_i,s_j>}$	=	Security risk level between two linked switches
sr_{s_i}	=	Security risk level of switch
sr_{h_i}	=	Security risk level of host
$PL: P_{h_i,h_j} \mapsto R^+$	=	Function that maps a full communication link to its packet loss
$B: P_{h_i,h_j} \mapsto R^+$	=	Function that maps a full communication link to its bandwidth
$AB: P_{h_i,h_j} \mapsto R^+$	=	Function that maps a full communication link to its available bandwidth
$JTR: P_{h_i,h_j} \mapsto R^+$	=	Function that maps a full communication link to its jitter
$DL: P_{h_i,h_j} \mapsto R^+$	=	Function that maps a full communication link to its delay
$SR: P_{h_i,h_j} \mapsto N$	=	Function that maps a full communication link to its security risk level
$threshold$	=	Used to avoid using paths that over-collateralize some of the KPIs
$x(a,b)$	=	Decision Variable of the MILP model

1) Rerouting GA: The goal of the algorithm is to find the optimal routing path between two hosts in order to meet the QoS and security demands imposed by the application the two hosts serve. The QoS and security demands required for each application are expressed in the genetic algorithm as constraints to the objective functions that aims to solve. The aspects of QoS that are subject of this manuscript are the delay, jitter, packet loss and bandwidth. Finally, the security requirement expresses numerically the sensitivity of the information that is processed and/or forwarded by the target application.

The problem is initially formulated as a multi-objective problem with constraints, where the objective functions and constraints are the following:

i) Delay objective function

$$min: \ J_{DL}(p) = max\left(threshold, \frac{DL\left(p_{h_i,h_j}\right)}{DL_c(h_i)}\right) \quad (1)$$

ii) Jitter objective function

$$min: \ J_{JTR}(p) = max\left(threshold, \ \frac{JTR\left(p_{h_i,h_j}\right)}{JTR_c(h_i)}\right) \quad (2)$$

iii) Packet loss objective function

$$min: \ J_{PL}(p) = \max\left(threshold, \ \frac{PL\left(p_{h_i,h_j}\right)}{PL_c(h_i)}\right) \quad (3)$$

iv) Available bandwidth objective function

$$min: \ J_{AB}(p) = -\frac{AB\left(p_{h_i,h_j}\right)}{max\left(ab\left\langle h_i, h_j\right\rangle\right)} \quad (4)$$

v) Security objective function

$$min: \ J_{security}(p) = \ \frac{SR\left(p_{h_i,h_j}\right)}{SR_c(h_i)} \quad (5)$$

$$s.t = \begin{cases} DL\left(p_{h_i,h_j}\right) \leq min\left(DL_c(h_i), \ DL_c(h_j)\right) \\ AB\left(p_{h_i,h_j}\right) \geq max\left(AB_c(h_i), \ AB_c(h_j)\right) \\ PL\left(p_{h_i,h_j}\right) \leq min\left(PL_c(h_i), \ PL_c(h_j)\right) \\ JTR\left(p_{h_i,h_j}\right) \leq min\left(JTR_c(h_i), \ JTR_c(h_j)\right) \\ SR\left(p_{h_i,h_j}\right) \leq min\left(SR_c(h_i), \ SR_c(h_j)\right) \\ \forall p_{h_i,h_j} \in \left(L + p - p^{'}\right) \end{cases}$$

The traditional online algorithms and their alternatives (such as Djikstra, LARAC, SSR+DCCR) used to find communication paths are not scalable with respect to the number of objectives and/or the scale of the network [11]. Hence, genetic algorithms were used in order to to find a near optimal solution as soon as possible. For this reason, PaDe [12] multi-objective genetic algorithm is used since it allows the parallel solving of multi-objective problems. For each chromosome in the population, PaDe decomposes a multi-objective problem into single-objective ones and using the asynchronous generalized island model [13] to distribute the solution process to multiple processors. At the end of the evolution, the population is set as the best individual in each single-objective island. PaDe is not suitable to solve constrained problems; therefore, the problem should be transformed into a non-constrained one. A solution to this is to add an additional objective, which will quantify the violation of the constraints and will be used as a penalty to the overall objective function. The additional, objective function is

the norm of the total constraint violation. A constrain violation is calculated as (e.g. for latency and available bandwidth):

$$C\_vl_i = \max\left(0, LT\left(p_{h_i,h_j}\right) - \min\left(LT_c\left(h_i\right), LT_c\left(h_j\right)\right)\right) \quad (6)$$

$$C\_vl_i = \max\left(0, \max\left(AB_c\left(h_i\right), AB_c\left(h_j\right) - AB\left(p_{h_i,h_j}\right)\right)\right) \quad (7)$$

Therefore, the additional objective function is formed as:

$$min:\ J_{violation}\left(p\right) = \sqrt{\sum_{k\ \forall p_{h_i,h_j} \in (L+p-p')} C\_vl_k^2} \quad (8)$$

Since, the problem is multi-objective, six dimensional specifically; there will be probably multiple solutions, each one being optimal at different objectives or combinations of objectives. After the final population of solutions has been calculated, the Pareto Front of solutions with respect to the function below is calculated:

$$\begin{aligned}(\underset{p}{arg\,min}:\ & \alpha J_{LT}\left(p\right) + \beta J_B\left(p\right) + \gamma J_{PL}\left(p\right) \\ & + \delta J_{JTR}\left(p\right) + \varepsilon\ J_{security}\left(p\right) + \zeta\ J_{violation}) \end{aligned} \quad (9)$$

Where α + β + γ + δ + ε + ζ = 1. The values of α , β , γ , δ , ε , ζ define the user's preference. The solutions with the minimal value are chosen in order to construct the Pareto Front of solutions with respect to the given policy. Finally, an additional filtering step takes place in order to select the path that makes the lowest allocation of resources.

2) Matchmaking Mixed Integer Linear Programming Algorithm: The proposed MILP algorithm focuses on the matchmaking between gateways and sensing units. Concretely, let $A = (SD_1, SD_2, \ldots, SD_N)$, a set of N available Sensing Devices. Let $B = (GW_1, GW_2, \ldots, GW_M)$, a set of M available Gateways. The decision variables are the sets of SD-GW pairs:

$$\{a, b\} \forall a \in A, \forall b \in B \quad (10)$$

with binary values:

$$\{a, b\} = \begin{cases} 1, & \text{if SD a sends data to GW b} \\ 0, & \text{if SD a does not send data to GW b} \end{cases}$$

The objective function of the problem is to select the SD-GW pairs which minimize the average communication network latency. In this case, the objective function is described as:

$$min \Sigma_{a \in A, b \in B} latency(a, b) * \{a, b\} \quad (11)$$

A GW cannot accept data from j+1 SDs if the connected SDs exceed the amount of devices that a GW can host.

$$\Sigma_{a \in A}\{a, b\} <= k_B ... \forall b \in B \quad (12)$$

Finally, each SD should send data to at most one GW:

$$\Sigma_{b \in B}\{a, b\} <= 1 ... \forall a \in A \quad (13)$$

3) Worfklow: EDAE's consists of three parallel workflows, the QoS maintenance workflow, the Cyber-attack mitigation workflow and Observability restoration worfklow. **QoS maintenance:** This workflow is executed periodically and it's objective is to find which paths violate the constraints of the applications in order to construct alternative paths.
Cyber-attack mitigation: This workflow is triggered when event is received from a risk assessment framework. EDAE updates the security status of the assets in the network topology based on the risk assessment framework and forces the re-construction process for the paths that violate the security requirements of the applications. Additionally, this workflow can apply mitigation policies specified by the security administrator such as re-routing the traffic of the affected asset to a Honeypot, but this functionality is out of scope of this manuscript.
Observability restoration: This workflow is executed when a gateway is out of operation. The goal of this workflow is to assign as many sensor devices to the leftover gateways minimizing the experienced latency between the new pairs resulted from the matchmaking process.

III. Evaluation

This section describes in details the evaluation framework and the results of EDAE. Concretely, the Evaluation scenarios subsection, provides details regarding the evaluation flow, while the Evaluation setup subsection analyzes the SDN testbed topology setup, the network traffic generation, the selection of the application constrains and details regarding the executed cyber-attacks. Last but not least, subsection Results, describes thoroughly the outcomes of the executed scenarios.

A. Evaluation scenarios

Two scenarios were utilized in order to assess the performance of EDAE from multiple views. The first scenario is called the "Non-attack scenario" while the second one is called the "Attack scenario". The status of each switch link (delay, bytes received/sent etc.) was captured through the provided API of the controller with a sampling ratio of 5 seconds and were stored to a timeseries database to be processed by the re-routing algorithms studied in the present manuscript.
Non-attack scenario: This scenario aims to evaluate the capability of EDAE's rerouting functionality to construct communication paths that serve the QoS requirements of the underlying applications. Therefore, in this scenario there isn't presence of cyber-attacks and it is assumed that no vulnerable asset exists in the topology. EDAE's rerouting algorithm is compared with the approach presented at [14]. The scenario is executed for 24 hours and every two minutes the paths were re-constructed.
Attack scenario: The aim of this scenario is twofold. The first goal is to evaluate EDAE's capability to assist in the mitigation of cyber-attacks, while in the same time manages the network resources to serve the QoS of the applications. The second goal is to evaluate the quality of the solutions provided by EDAE's MILP matchmaking algorithm and compare it with the method proposed at [15]. For this reason, we are executing

3 different cyber-attacks for the ModBus protocol. The attacks are executed from a malicious insider, from the gateways to the emulated smart meters.

After the detection of each cyber-attack, EDAE blocks the communication from the gateway, to all the other network components. Afterwards, EDAE genetic algorithm recalculates the new paths and EDAE's MILP proceeds to the matchmaking of the leftover gateways with the sensing units. This scenario is executed for 4 hours and every five minutes an attack is randomly selected and executed from one of the three gateways. Each gateway is also randomly selected to perform the attack. The network statistics (e.g. jitter, latency) during the performance of the attack are send to EDAE in order to perform the evaluation experiments. More information about the cyber-attacks can be found in subsection Executed Cyber-attacks. It has to be noted also that the detection of the cyber-attacks is out of the scope of this work.

B. Evaluation Setup

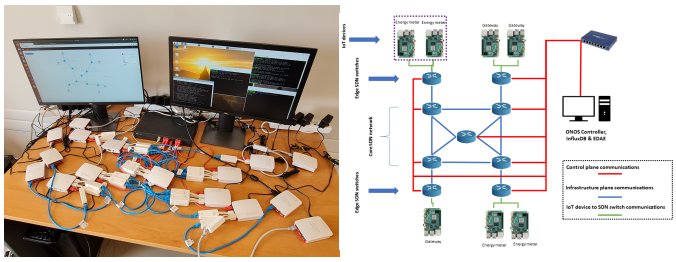
1) Physical SDN testbed setup: This section will describe the development and the implementation of an SDN-enabled testbed setup, used in order to evaluate and analyze the performance of the EDAE tool in a proof-of-concept SDN environment.

The physical SDN testbed, which was developed in the scope of our research in order to be able to evaluate the SDN functionalities and the developed algorithms, consists of the following components, nine SDN Switches, the Open Network Operating System (ONOS) acting as the SDN Controller and seven host devices.

SDN Controller: The SDN Controller, which can programmably construct the data flows, is the central system of the SDN-enabled network. The SDN Controller used in our experimental testbed is ONOS. The ONOS controller is the most widely used open-source SDN Controller for developing next-generation SDN solutions, and it is written in Java [16]. ONOS is installed as a Docker container on a computer running a Linux-based Operating System (OS).

SDN Switches: Open vSwitch, also known as OVS, is a multilayer software switch which interconnects virtual devices in the same host or between different hosts, licensed under open-source Apache 2.0 license. OVS allows programmers to create forwarding functions in order to automate and control network traffic and supports standard management interfaces. The software-based Open vSwitch, which runs on a Raspberry Pi 3B+ board, forms an SDN Switch for the network. The SDN-enabled switch includes a built-in Ethernet port and with the use of four USB-to-Ethernet adapters, it is able to expand its network capabilities. The proposed SDN-enabled network consists of nine SDN Switches. The five of them are part of the core network, where the system administrator is able to program different flow routes, while the four of them are part of the edge network, allowing different kind of devices to connect to the SDN-enabled network. Each SDN Switch is connected to another switch or host, using wired connections (Ethernet).

Host Devices: In the scope of our experiments, seven host devices are used in the SDN topology. More specifically, four Raspberry Pi 3B+ boards are programmed to act as Modbus TCP/IP energy meters (servers), while the other three hosts are acting as gateways (clients), requesting energy meter data and producing Modbus network traffic in the SDN-enabled network.



(a) (b)

Fig. 1: SDN testbed a) Physical b) SDN topology

2) Traffic Engineering: The network traffic between the hosts of the network are mainly ModBus/TCP packets created using the server-client stack communication offered by PyModbus [17] python library. The QoS metrics (available bandwidth, delay, jitter, packet loss) for each link between the switches was the same and optimal prior to any traffic generation. Therefore, diversity between the status of the switch links had to be induced. The NetEm package [18] from Linux foundation was used in order to insert delay, jitter and packet loss over the network. Additionally, in order to affect the bandwidth of the links, TCP/IP traffic was generated between the hosts of the network. Subsequently, to allow the status of the links to variate over time, multiple profiles were created for each QoS metric that were sampled for each link every 5 minutes. Specifically, there were 9 different traffic generation profiles for the bandwidth metric (combination of 3 profiles for the frequency of POST action and 3 of the volume of the data), 3 different profiles were created for delay and jitter and 4 profiles for packet loss. In Table II the different profiles and the distribution of probabilities to be sampled are shown for each QoS metric. It should be noted that the choice of the profiles is made in such a way that the network will always contain a portion of switch links that will be congested.

TABLE II: Distribution of Probabilities

Metric	Categories	Distribution
Delay (ms)	[0, 1.5, 3]	[0.5, 0.33, 0.17]
Jitter (ms)	[0.5, 1, 1.5, 2]	Uniform
Packet loss (%)	[0.15,0.25,0.35,0.45]	[0.4,0.3,0.15,0.15]
Traffic-freq (ms)	[0.1, 0.3, 0.7]	Uniform
Traffic-load (MBps)	[4, 8, 16]	Uniform
Security (Categorical)	[1, 2, 3, 4, 5]	[0.4,0.2,0.15,0.15,0.1]

3) Executed Cyber-attacks: The executed cyber-attacks are implemented in the emulated servers and clients of the PyModbus library. For the execution of the attacks, the widely known SMOD penetration tool was selected and the attacks

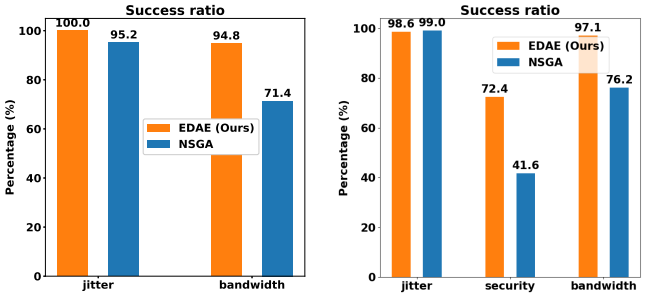
were: the UID Brute Force attack, the function enumeration attack and the Write All Registers DoS attack [19].

The security aspect, in which the two scenarios differ, introduces a new metric, the security status of each switch. In the attack scenario the switches were assigned with an additional property, the security status. The security status was constant over time and was selected randomly for each switch as shown in Table II.

4) Selection of constraints: In order to make the evaluation framework specified for EPES applications, the types of the various hosts and their respective QoS requirements were selected based on sources that were found in the literature regarding the EPES QoS requirements [20], [21], [22], [23], [24]. The four most demanding applications, namely the PMU to PDC data transfer, the transient stability, the wide area situational awareness and the communication of distributed energy resources, were selected and included in the analysis.

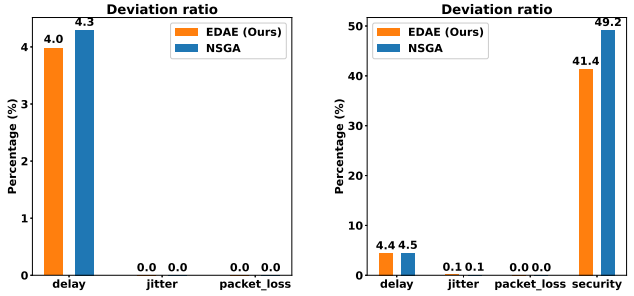
C. Results

The results are presented for each algorithm of EDAE separately.



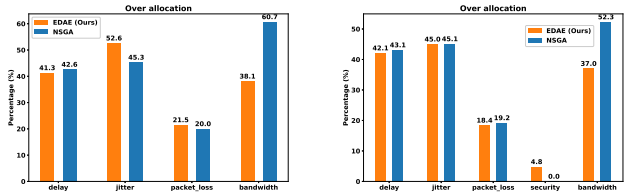
(a) (b)

Fig. 2: Success ratios a) on non-attack scenario b) on attack scenario



(a) (b)

Fig. 3: Deviation ratios a) on non-attack scenario b) on attack scenario



(a) (b)

Fig. 4: Over allocation ratios a) on non-attack scenario b) on attack scenario

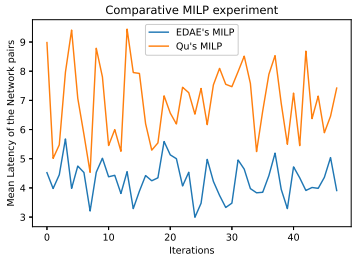


Fig. 5: Mean latency of the new gateway - sensing units pairs for EDAE's MILP and Qu's MILP.

1) Rerouting GA: Three KPIs were defined in order to asses the performance of the aforementioned genetic algorithms. Each KPI is applicable for all the metrics. The first KPI is the success ratio (SR) and expresses the percentage of the applications for which the constraints of the target metric is covered during the experiment. The second KPI is the deviation ratio (DR) and expresses the mean deviation from covering the constraint requirements for the applications that the constraints are not covered. Finally, the third KPI is the over allocation ratio (OAR) and expresses the mean percentage that each constraints is over-covered for the applications that the constraints are covered.

Both algorithms perform equally well and have a success ratio close to 100% with respect to delay, jitter and packet loss metrics in both scenarios. In Fig. 2 (a) its is observed that EDAE has greater success on finding routes that satisfy the bandwidth and jitter QoS requirements of the applications. Finally, in Fig. 2 (b) it is shown that EDAE can still provide routes that meet the QoS of the applications with greater success than the NSGA but is also able to find much more secure routes with respect to the security requirements (business value) of the application.

In Fig. 3 the results are similar for all the metrics except for the security. In the attack scenario it is observed that EDAE is capable to assign more secure paths to the applications even though the paths do not cover the security requirements. The results show that whenever a path that covers the constraints cannot be found, EDAE will find a path that deviates less from the specified constraints of the applications.

Finally, the plots presented in Fig. 4 demonstrate the effec-

tiveness of the threshold parameter included in the formulation of the objective functions of EDAE and the extra filtering step when choosing the final solution from the Pareto front. EDAE produces paths that do not over exceed the requirements of the applications. This explains the big difference in the bandwidth SR between the two algorithms. EDAE will use switch links with high available bandwidth for demanding applications and switch links with low available bandwidth for less demanding applications by means of bandwidth.

2) EDAE's Mixed Integer Linear Programming Algorithm: The evaluation of EDAE's MILP formulation is implemented in terms of the mean latency of the new matchmaking between the gateway-sensing unit devices pairs. For this reason, EDAE's MILP is compared with the Qu's MILP presented in [15]. It has to be noted that Qu's MILP considers the sensing units to be in neighbours, but for our scenario we consider that each sensing unit does not belong to a neighbourhood. Fig. 5 depicts the mean latency of the new gateway-sensing unit pairs for all the iterations that a Gateway went off due to a cyberattack. The results indicate that EDAE's MILP is able to pair the leftover gateways with the sensing units with the minimum average latency compared to Qu's approach.

IV. Conclusion

This paper presents thoroughly the EDAE, a component of the SDN-microSENSE H2020 project and the evaluation results on a real SDN testbed. The primary functions of EDAE are to provide an alternate network route from a measurement device to a control/monitoring unit in case the control/monitoring system's redundancy prevents the information from being forwarded or if it is under attack. The evaluation results indicate that both the EDAE's rerouting algorithm and EDAE's MILP formulation achieve better results in terms of QoS of the network with the two comparison methods. As future work, the authors intend to evaluate EDAE in a large scale SDN enabled electrical grid and apply also detection methods of the ModBus cyber-attacks, which are presented in this paper. Moreover, the physical SDN testbed, which was used for the evaluation of EDAE, will be further extended and upgraded, in order to support wireless mesh networks and other wireless communications instead of wired. To this end, the evaluation of EDAE in a new physical environment could be performed.

Acknowledgment

The aforementioned work effort in this paper is conducted under the framework of the SDN-microSENSE project, a Horizon 2020 program, funded by the European Union under the grant agreement No. 833955. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

References

[1] H. Sándor, B. Genge, Z. Szántó, L. Márton, and P. Haller, "Cyber attack detection and mitigation: Software defined survivable industrial control systems," *International Journal of Critical Infrastructure Protection*, vol. 25, pp. 152–168, 2019.

[2] A. Cherepanov, "Blackenergy by the sshbeardoor: attacks against ukrainian news media and electric industry," *We Live Security*, vol. 3, 2016.

[3] A. Valdes and K. Skinner, "Adaptive, model-based monitoring for cyber attack detection," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2000, pp. 80–93.

[4] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 580–591, 2014.

[5] K. Papachristou, T.-I. Theodorou, S. Papadopoulos, A. Protogerou, A. Drosou, and D. Tzovaras, "Routing policy verification for enhanced energy quality of service and security monitoring in iot networks," in *Proceedings of the 23rd Pan-Hellenic Conference on Informatics*, 2019, pp. 43–48.

[6] T. A. Q. Pham, Y. Hadjadj-Aoul, and A. Outtagarts, "Deep reinforcement learning based qos-aware routing in knowledge-defined networking," in *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Springer, 2018, pp. 14–26.

[7] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015.

[8] D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 2013, pp. 55–60.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.

[10] "Sdn microsense – project," https://www.sdnmicrosense.eu/, (Accessed on 04/28/2021).

[11] J. W. Guck, A. Van Bemten, M. Reisslein, and W. Kellerer, "Unicast qos routing algorithms for sdn: A comprehensive survey and performance evaluation," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 388–415, 2018.

[12] A. Mambrini and D. Izzo, "Pade: A parallel algorithm based on the moea/d framework and the island model," 09 2014, pp. 711–720.

[13] D. Izzo, M. Ruciński, and F. Biscani, *The Generalized Island Model*, 01 2012, vol. 415, pp. 151–169.

[14] D. Li, X. Wang, Y. Jin, and H. Liu, "Research on QoS routing method based on NSGAII in SDN," in *Journal of Physics Conference Series*, ser. Journal of Physics Conference Series, vol. 1656, Sep. 2020, p. 012027.

[15] Y. Qu, X. Liu, D. Jin, Y. Hong, and C. Chen, "Enabling a resilient and self-healing pmu infrastructure using centralized network control," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, ser. SDN-NFV Sec'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 13–18.

[16] C. M. Iurian, I. A. Ivanciu, B. M. Marian, D. Zinca, and V. Dobrota, "An sdn architecture for iot networks using onos controller," in *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2020, pp. 1–6.

[17] "Pymodbus - a python modbus stack — pymodbus 2.5.0 documentation," https://pymodbus.readthedocs.io/en/latest/readme.html, (Accessed on 04/27/2021).

[18] S. Hemminger, "Network emulation with netem," *Linux Conf Au*, 05 2005.

[19] "Github - theralfbrown/smod-1: Modbus penetration testing framework," https://github.com/theralfbrown/smod-1, (Accessed on 04/27/2021).

[20] S. Grewal, M. Soni, and D. Jain, "Requirements and challenges of pmus communication in wams environment," *Far East Journal of Electronics and Communications*, vol. 13, pp. 121–135, 12 2014.

[21] W. Tao, M. Ma, C. Fang, W. Xie, M. Ding, D. Xu, and Y. Shi, "Design and application of a distribution network phasor data concentrator," *Applied Sciences*, vol. 10, no. 8, 2020.

[22] P. Kansal and A. Bose, "Bandwidth and latency requirements for smart transmission grid applications," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1344–1352, 2012.

[23] T. O. of the General Counsel. Communications requirements of smart grid technologies. [Online]. Available: https://www.energy.gov/gc/downloads/communications-requirements-smart-grid-technologies

[24] SPP. Spp pmu communications handbook. [Online]. Available: https://www.spp.org/spp-documents-filings/?id=185197