Fault-Tolerant SDN Solution for Cybersecurity Applications

ATHANASIOS LIATIFIS, University of Western Macedonia, Greece
CHRISTOS DALAMAGKAS, Public Power Corporastion, Greece
PANAGIOTIS RADOGLOU-GRAMMATIKIS, University of Western Macedonia, Greece
THOMAS LAGKAS, International Hellenic University, Greece
EVANGELOS MARKAKIS, Hellenic Mediterranean University, Greece
VALERI MLADENOV, Technical University of Sofia, Bulgaria
PANAGIOTIS SARIGIANNIDIS, University of Western Macedonia, Greece

The rapid growth of computer networks in various sectors has led to new services previously hard or impossible to implement . Internet of Things has also assisted in this evolution offering easy access to data but at the same time imposing constraints on both security and quality of service. In this paper, an SDN fault tolerant and resilient SDN controller design approach is presented. The proposed solution is suitable for a wide range of environments. Benefits stemming from actual scenarios are presented and discussed among other solutions.

CCS Concepts: • **Networks** → **Network monitoring**; **Network management**; **Network security**.

Additional Key Words and Phrases: Smart Grid, Cybersecurity, Software Defined Networking

ACM Reference Format:
Athanasios Liatifis, Christos Dalamagkas, Panagiotis Radoglou-Grammatikis, Thomas Lagkas, Evangelos Markakis, Valeri Mladenov, and Panagiotis Sarigiannidis. 2022. Fault-Tolerant SDN Solution for Cybersecurity Applications. In *The 17th International Conference on Availability, Reliability and Security (ARES 2022), August 23–26, 2022, Vienna, Austria.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3538969.3544479

1 INTRODUCTION

Networks have evolved from simple systems that enable communication between peers to complex ones that perform complex tasks. This has lead to a new architectural design, the Software Defined Networking (SDN) [13]. Moreover, the advent of the Internet of Things (IoT) and emerging paradigms like smart grid has also evolved and adopted complex network designs [5, 19]

Integrating SDN to these systems can offer many benefits and challenges as well. In this paper we present a fault tolerant SDN controller combining multiple Ryu instances [21] and coordinate them via Zookeeper framework [10]. Moreover a user friendly front-end dashboard is presented allowing the user (i.e. the administrator) to monitor and interact with the network. Based on the aforementioned remarks the contribution of this work are:

- **High availability SDN controller solution:** A fault tolerant SDN controller consisting of multiple Ryu SDN controllers coordinated by the Zookeeper framework that offers a high availability control plane.

- **Resource efficient synchronisation service:** A synchronisation and coordination service that coordinate the Ryu instances to elect a master controller.
- **Lightweight front-end interface:** A user friendly dashboard to visualise the network and assist the administrator in enforcing policies to data plane devices.

The remainder of this paper is structured as follows. Section 2 discusses related work related to SDN applications in cybersecurity. In Section 3 an overview of the SDN paradigm is discussed briefly. Section 4 present the proposed architecture that was designed. Section 5 is dedicated to applications and other security services that make use of SDN capabilities to enforce a desired pocily, or behaviour, to the data plane. Finally, Section 6 concluded the work presented in this paper and provides future directions.

2 RELATED WORK

Although SDN offers many benefits compared to traditional approaches it introduces a single point of failure, namely, the SDN controller [11]. In the past decade, the research community has proposed several solutions often adopting distributed system approaches [2]. Moreover various works focus on the security applications offered by SDN, such as [17, 18, 22, 23, 26]

Zhao and Wu [24] proposed a scalable SDN architecture that can handle WAN traffic without overloading the control plane. The authors propose a joint design methods that take into consideration multiple parameters by formulating an Integer Linear Programming (ILP) problem. The computed result is a set of the minimal controller set instances to achieve load balancing and ensure proper scalability. To further minimize the computation time the authors propose a heuristic approach. The proposed system achieves lower controller instances at the cost of more computational time to identify the optimal placement.

Molina et al [15] highlight the diverse needs of the IEC-61850 protocol stack in terms of communication and design an SDN-based framework to perform Traffic Engineering named Smart Grid Architecture Model (SGAM). The proposed scheme integrates IEC-61850 Substation Configuration Language (SCL) models used to specify substation behaviour. By parsing these configuration files the control plane can allocate networking resources, create virtual paths between devices and prioritize traffic. Specifically for each pair of devices, the control plane construct a Layer-2 VLAN tag to distinguish traffic from other device pairs. To evaluate SGAM the authors design a prototype using OpenDaylight controller, OpenVswitch (OVS) software switch and Mininet emulator. Results of use case scenarios demonstrated enhanced security, monitoring and routing of IEC-61850 traffic.

Ravana [12] is a fault-tolerant SDN controller that processes control messages in a transactional manner ensuring a multi-stage action is completed entirely or reverts the steps and rollback to the previous state. To achieve this level of fault-tolerance Ravana uses many well-known practices of distributed application design. Specifically, Ravana introduces a two-stage replication protocol that ensures event messages are delivered and their processing is completed successfully storing all information in a shared in-memory log. The authors further optimize the proposed design through parallelization of event logging and transaction processing. Evaluation results show that Ravana's fault-tolerance introduces little overhead in terms of throughput if some inconsistencies can be tolerated.

Görkemli et al. [8, 9] propose a network control mechanism of switches where switches communicate with SDN controller instances over an in-band overlay network. The proposed system architecture includes three tiers, namely, the controller cloud, the out-of-band switches fabric and the in-band controlled switches. A key component of the proposed architecture is the Control Plane Manager (CPM) responsible to monitor the network, collecting statistics, and performance metrics to efficiently load-balance the controller instance load. The purpose of the out-of-band tier is to quickly load-balance traffic or reroute using OpenFlow Flow tables as the control flow table for data plane to control plane traffic. The evaluation results of scenarios run in a complex network demonstrate lower overall CPU utilization and lower communication overhead.

3 OVERVIEW OF SOFTWARE DEFINED NETWORKING

Software-Defined Networking (SDN) [13] is an emerging paradigm that automates network monitoring and management processes resulting in lower costs and complexity by decoupling data plane and control plane mechanisms. The separation of these two planes is characterised by establishing well-defined communication interfaces. Figure 1 illustrates the architectural design of an software defined network.

In an SDN environment, all networking devices (e.g. routers, firewalls, etc.) are responsible for packet forwarding only. Moreover, networking devices now are programmable in the sense that they can be programmed and enforce the desired behaviour through programming interfaces. This approach simplifies the management of these devices and enables rapid prototyping and innovation, granting network operators the ability to create custom solutions.

A key entity in the SDN paradigm is the SDN controller, a logically centralised software stack responsible to manage the networking infrastructure and ensure overall network stability [25]. The SDN controller exposes two interfaces, the Southbound Interface (SBI) and the Northbound Interface (NBI). The SBI defines the communication protocols utilised between the SDN Controller and the data plane device.

Multiple SBIs have been defined over the years [14] with OpenFlow being the predominant one. NBIs on the other hand expose an interface to external services. Through an NBI other services can collect statistics from the controller, or instruct it to enforce a high-level policy. Most modern SDN controllers support two NBIs, REST and RPC.

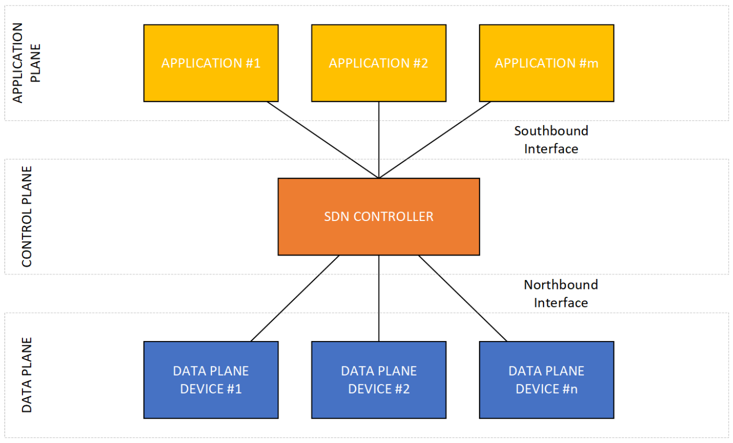


Fig. 1. SDN architecture

4 PROPOSED SYSTEM ARCHITECTURE

The proposed system architecture can operate on multiple environments like EPES, Edge or LAN. In particular the proposed system can collect statistics from data plane devices, monitor network flows, mitigate malicious actions and enforced desired policies. It includes a three-tier approach to achieve this, namely, (a) Risk assessment, (b) intrusion detection and finally, (c) self-healing [7]. The SDN controller plays a crucial roles in the design architecture as depicted in Figure 2. In the following subsections each component is analysed and discussed.

4.1 SDN Controller

The SDN Controller (SDNC) is based on the Ryu framework [21] and is enhanced with additional features to cover the needs of diverse environments like Local Area Networks (LANs), EPES and Edge. Unlike the publicly available Ryu, SDNC offers multiple enhancements. Firstly link delay estimation and expose these measurements to other applications through its REST API. Secondly, with the help of the Synchronisation and Coordination Service (SCS) multiple Ryu instances can be coordinated and elect a master controller. Finally, existing applications have been enhanced to prevent ARP broadcast storms and communicate with the data plane when elected.

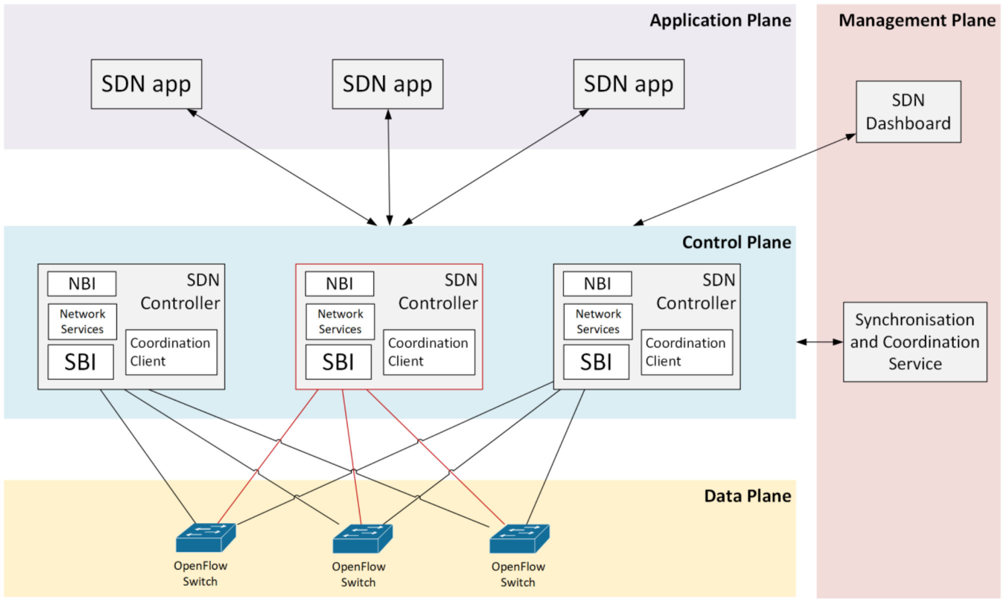


Fig. 2. High Level View of the SDN Controller

4.2 Synchronisation and Coordination Service

The SCS is a background service that supports the master controller elections process. SCS is based on the Zookeeper framework [10], a popular framework for distributed services coordination. Multiple SDNC instances register themselves in a distributed tree structure and run the same master election process to elect a master. Moreover, the SCS exposes a simple REST API endpoint for other services to identify the master controller instance. The master election process is based on the Zookeeper Election recipe. Each data plane device is added as a node to a tree-like structure. If the corresponding controller is connected to that device then it is added as a lead-node under that data plane ID and assigned a unique number. The election process chooses the lowest leaf-node as the master controller. In case an SDNC node is disconnected their corresponding leaf-nodes are deleted and the election process is started anew. It is important to set proper timeout limits to avoid unnecessary node deletions.

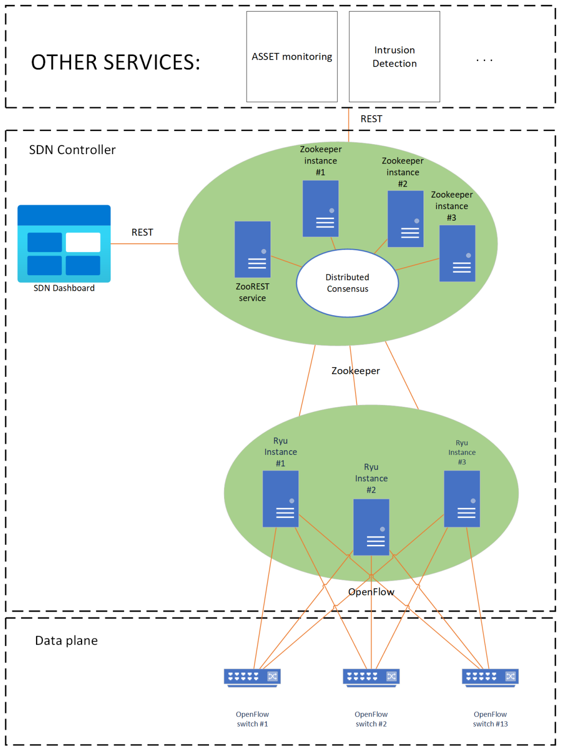


Fig. 3. Technical overview of the SDN Controller

4.3 SDN Dashboard

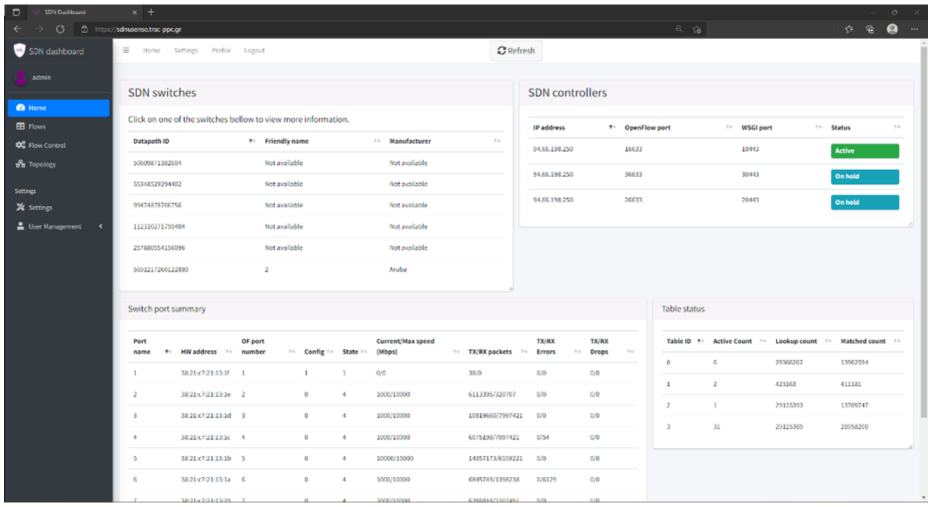
The SDN Dashboard (SDND) is a user-friendly Graphical User Interface that eases the management and monitoring of the networking infrastructure. Using the SDNC REST API it can collect statistics, push custom flow commands and visualize the topology. To realise the SDND the Django framework was used [4]. Figure 4 depicts the home page of the SDN Dashboard whereas Figure 5.

5 SDN-BASED SECURITY APPLICATIONS

SDN has a lot to offer to SG communications. Having full control of the data plane where CPS devices operate can timely detect misbehaviours and minimize potential side effects. This section discusses SDN applications in SG environments.

5.1 SDN-enabled Devices

Most IoT devices support protocols like Modbus or MQTT that were designed decades ago. When initially released, these protocols assumed a close secure environment with no internet access to limited personnel. IoT though assumes these devices will be connected to the internet and communicate with other services. SDN technology can help stakeholders integrate these devices minimizing the security risks and assisting their proper functionality without disrupting their functionality. Connected IoT devices in an SDN-based environment can assist administrators closely monitor such interactions and if needed restrict them instantaneously.



SDN switches

Click on one of the switches bellow to view more information.

Datapath ID	Friendly name	Manufacturer
50609871382604	Not available	Not available
55348529294402	Not available	Not available
99474878706756	Not available	Not available
112310271790404	Not available	Not available
257880554156096	Not available	Not available
5691217260122880	2	Aruba

SDN controllers

IP address	OpenFlow port	WSGI port	Status
94.66.198.250	16633	10443	Active
94.66.198.250	36633	30443	On hold
94.66.198.250	26633	20443	On hold

Switch port summary

Port name	HW address	OF port number	Config	State	Current/Max speed (Mbps)	TX/RX packets	TX/RX Errors	TX/RX Drops
1	38:21:c7:21:13:1f	1	1	1	0/0	38/0	0/0	0/0
2	38:21:c7:21:13:1e	2	0	4	1000/10000	6113395/320707	0/0	0/0
3	38:21:c7:21:13:1d	3	0	4	1000/10000	10819660/7997421	0/0	0/0
4	38:21:c7:21:13:1c	4	0	4	1000/10000	6075198/7997421	0/54	0/0
5	38:21:c7:21:13:1b	5	0	4	10000/10000	14357173/6559221	0/0	0/0
6	38:21:c7:21:13:1a	6	0	4	1000/10000	6845749/1398238	0/6129	0/0
7	38:21:c7:21:13:19	7	0	4	1000/10000	6295018/2102451	0/0	0/0

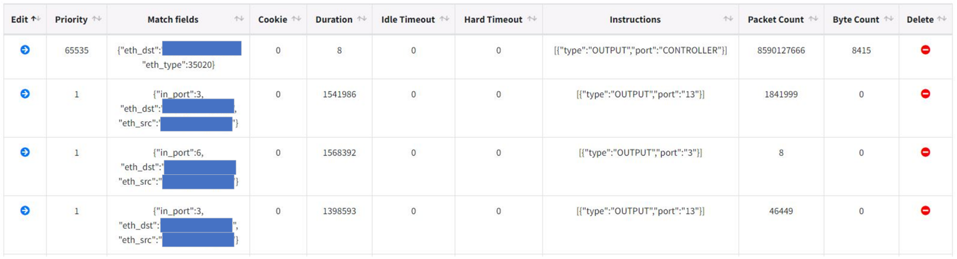
Table status

Table ID	Active Count	Lookup count	Matched count
0	6	29366262	13962594
1	2	423163	411181
2	1	29125393	13709747
3	31	29125389	28958200

Fig. 4. Home page of the SDN Dashboard

5.2 Risk Assessment

Risk assessment involves proper estimation of the negative impact and actions that may impose and proper mitigation actions to minimize these negative effects in the most suitable manner [20]. The SDN technology could provide fine-grained statistics collected directly from data plane devices. Moreover, an SDN-enabled data plane can accurately and timely enforce risk mitigation policies. Depending on the type of statistics requested they can be port-based, flow-based or switch-based. Port-based statistics include measurements related to number of bytes that traversed through this port, errors in frames redundancy checks and number of drop packets. Flow-based statistics include number of packets that triggered a particular rules and number of total bytes this rule is applied on. Finally, switch-based statistics include measurements like total number of flows installed in that device and metrics like table popularity. Figure 5 depicts statistics collected from all three aforementioned statistics categories.



Edit	Priority	Match fields	Cookie	Duration	Idle Timeout	Hard Timeout	Instructions	Packet Count	Byte Count	Delete
➔	65535	{"eth_dst":[redacted] "eth_type":35020}	0	8	0	0	[{"type":"OUTPUT","port":"CONTROLLER"}]	8590127666	8415	⛔
➔	1	{"in_port":3, "eth_dst":[redacted], "eth_src":"[redacted]"}	0	1541986	0	0	[{"type":"OUTPUT","port":"13"}]	1841999	0	⛔
➔	1	{"in_port":6, "eth_dst":[redacted] "eth_src":"[redacted]}	0	1568392	0	0	[{"type":"OUTPUT","port":"3"}]	8	0	⛔
➔	1	{"in_port":3, "eth_dst":[redacted]", "eth_src":"[redacted]}	0	1398593	0	0	[{"type":"OUTPUT","port":"13"}]	46449	0	⛔

Fig. 5. SDN dashboard controller statistics

5.3 SDN-based Honeypots

Honeypots are software components that emulate the behaviour of other services and assets [16]. Their aim is to distract any attacker from targeting a real asset or at least delay them for mitigation actions to take effect. The SDN technology can assist honeypots in blending into the network in many fashions. Taking full advantage of SDN capabilities security solutions can instruct the network to isolate attackers in honeypot networks (honeynets) [1] or redirect traffic to honeypots [3] with zero disruptions. Traffic redirection is enforced at the source and destination switches ensuring minimal flow disruption, whereas the priority of these rules is higher to ensure default behaviour override. Depending on the scope of the attack the redirection rule can be MAC-based or IP-based.

5.4 Host isolation

Unlike traditional security solutions (e.g., firewalls, Access Control Lists - ACLs), an SDN-based environment can isolate malicious hosts from critical services in a transparent manner without disrupting the normal operation of benign hosts [18]. When a malicious host is identified proper mitigation actions are taken. Firstly, they are blocked from communicating with any real asset. This step ensures protection of the remaining assets. Secondly, the malicious host is redirected to honeypot replicas of real assets to further inspect their behaviour and motivations.

5.5 High accuracy Quality of Service

Many SG applications and protocols impose strict Quality of Service (QoS) requirements on the network. Failing to meet those requirements can lead to a negative impact on operators and stakeholders. SDN can easily prioritise traffic of mission-critical applications or redirect traffic with little-to-no disruptions for other services [6]. The SDN controller can assist other services in decision making processes by offering statistics to them.

6 CONCLUSION

The inevitable evolution of electrical grids into smart grids offers many benefits and introduces new security threats as well. In this paper an in-depth discussion of the benefits SDN technology has to offer in SG communications was presented. In particular, an actual SDN solution deployed in the context of a European funded project was discussed and examples stemming from that project were presented in addition to other possible application scenarios. In the future we intend to evolve the existing architecture taking into consideration lessons learnt and limitations they came across. Also, the authors consider adopting this solution to edge and IoT environments.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101021936 (ELECTRON).

REFERENCES

[1] Daniele Antonioli, Anand Agrawal, and Nils Ole Tippenhauer. 2016. Towards High-Interaction Virtual ICS Honeypots-in-a-Box. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy* (Vienna, Austria) *(CPS-SPC '16)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/2994487.2994493

[2] Kostas Choumas and Thanasis Korakis. 2020. When Raft Meets SDN: How to Elect a Leader over a Network. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, Ghent, Belgium, 140–144. https://doi.org/10.1109/NetSoft48620.2020.9165534

[3] Marcos V. O. De Assis, Anderson H. Hamamoto, Taufik Abrão, and Mario Lemes Proença. 2017. A Game Theoretical Based System Using Holt-Winters and Genetic Algorithm With Fuzzy Logic for DoS/DDoS Mitigation on SDN Networks. *IEEE Access* 5 (2017), 9485–9496. https://doi.org/10.1109/ACCESS.2017.2702341

[4] Django Software Foundation. 2022. Django. https://djangoproject.com

[5] Xinshu Dong, Hui Lin, Rui Tan, Ravishankar K. Iyer, and Zbigniew Kalbarczyk. 2015. Software-Defined Networking for Smart Grid Resilience: Opportunities and Challenges. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security* (Singapore, Republic of Singapore) *(CPSS '15)*. Association for Computing Machinery, New York, NY, USA, 61–68. https://doi.org/10.1145/2732198.2732203

[6] Nils Dorsch, Fabian Kurtz, Hanno Georg, Christian Hägerling, and Christian Wietfeld. 2014. Software-defined networking for Smart Grid communications: Applications, challenges and advantages. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, Venice, Italy, 422–427. https://doi.org/10.1109/SmartGridComm.2014.7007683

[7] Panagiotis Radoglou Grammatikis, Panagiotis Sarigiannidis, Christos Dalamagkas, Yannis Spyridis, Thomas Lagkas, Georgios Efstathopoulos, Achilleas Sesis, Ignacio Labrador Pavon, Ruben Trapero Burgos, Rodrigo Diaz, Antonios Sarigiannidis, Dimitris Papamartzivanos, Sofia Anna Menesidou, Giannis Ledakis, Achilleas Pasias, Thanasis Kotsiopoulos, Anastasios Drosou, Orestis Mavropoulos, Alba Colet Subirachs, Pol Paradell Sola, José Luis Domínguez-García, Marisa Escalante, Molinuevo Martin Alberto, Benito Caracuel, Francisco Ramos, Vasileios Gkioulos, Sokratis Katsikas, Hans Christian Bolstad, Dan-Eric Archer, Nikola Paunovic, Ramon Gallart, Theodoros Rokkas, and Alicia Arce. 2021. SDN-Based Resilient Smart Grid: The SDN-microSENSE Architecture. *Digital* 1, 4 (2021), 173–187. https://doi.org/10.3390/digital1040013

[8] Burak Görkemli, A. Murat Parlakışık, Seyhan Civanlar, Aydın Ulaş, and A. Murat Tekalp. 2016. Dynamic management of control plane performance in software-defined networks. In *2016 IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE, Seoul, Korea (South), 68–72. https://doi.org/10.1109/NETSOFT.2016.7502445

[9] Burak Görkemli, Sinan Tatlıcıoğlu, A. Murat Tekalp, Seyhan Civanlar, and Erhan Lokman. 2018. Dynamic Control Plane for SDN at Scale. *IEEE Journal on Selected Areas in Communications* 36, 12 (2018), 2688–2701. https://doi.org/10.1109/JSAC.2018.2871308

[10] Patrick Hunt, Mahadev Konar, Flavio P. Junqueira, and Benjamin Reed. 2010. ZooKeeper: Wait-Free Coordination for Internet-Scale Systems. In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference* (Boston, MA) *(USENIXATC'10)*. USENIX Association, USA, 11.

[11] Murat Karakus and Arjan Durresi. 2017. A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN). *Computer Networks* 112 (2017), 279–293. https://doi.org/10.1016/j.comnet.2016.11.017

[12] Naga Katta, Haoyu Zhang, Michael Freedman, and Jennifer Rexford. 2015. Ravana: Controller Fault-Tolerance in Software-Defined Networking. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research* (Santa Clara, California) *(SOSR '15)*. Association for Computing Machinery, New York, NY, USA, Article 4, 12 pages. https://doi.org/10.1145/2774993.2774996

[13] Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* 103, 1 (2015), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

[14] Zohaib Latif, Kashif Sharif, Fan Li, Md Monjurul Karim, Sujit Biswas, and Yu Wang. 2020. A comprehensive survey of interface protocols for software defined networks. *Journal of Network and Computer Applications* 156 (2020), 102563. https://doi.org/10.1016/j.jnca.2020.102563

[15] Elias Molina, Eduardo Jacob, Jon Matias, Naiara Moreira, and Armando Astarloa. 2015. Using Software Defined Networking to manage and control IEC 61850-based systems. *Computers & Electrical Engineering* 43 (2015), 142–154. https://doi.org/10.1016/j.compeleceng.2014.10.016

[16] Marcin Nawrocki, Matthias Wählisch, Thomas C. Schmidt, Christian Keil, and Jochen Schönfelder. 2016. A Survey on Honeypot Software and Data Analysis. https://doi.org/10.48550/ARXIV.1608.06249

[17] Achilleas Pasias, Thanasis Kotsiopoulos, Georgios Lazaridis, Anastasios Drosou, Dimitrios Tzovaras, and Panagiotis Sarigiannidis. 2021. Enabling Cyber-attack Mitigation Techniques in a Software Defined Network. In *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, Rhodes, Greece, 497–502. https://doi.org/10.1109/CSR51186.2021.9527932

[18] Panagiotis Radoglou-Grammatikis, Panagiotis Sarigiannidis, George Efstathopoulos, Paris-Alexandros Karypidis, and Antonios Sarigiannidis. 2020. DIDEROT: An Intrusion Detection and Prevention System for DNP3-Based SCADA Systems. In *Proceedings of the 15th International Conference on Availability, Reliability and Security* (Virtual Event, Ireland) *(ARES '20)*. Association for Computing Machinery, New York, NY, USA, Article 115, 8 pages. https://doi.org/10.1145/3407023.3409314

[19] Mubashir Husain Rehmani, Alan Davy, Brendan Jennings, and Chadi Assi. 2019. Software Defined Networks-Based Smart Grid Communication: A Comprehensive Survey. *IEEE Communications Surveys Tutorials* 21, 3 (2019), 2637–2670. https://doi.org/10.1109/COMST.2019.2908266

[20] Erkuden Rios, Angel Rego, Eider Iturbe, Marivi Higuero, and Xabier Larrucea. 2020. Continuous quantitative risk management in smart grids using attack defense trees. *Sensors* 20, 16 (2020), 4404.

[21] Ryu Development Team. 2022. Ryu SDN framework. https://github.com/faucetsdn/ryu.

[22] Raja Majid Ali Ujjan, Zeeshan Pervez, and Keshav Dahal. 2019. Snort Based Collaborative Intrusion Detection System Using Blockchain in SDN. In *2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*. IEEE, Island of Ulkulhas, Maldives, 1–8. https://doi.org/10.1109/SKIMA47702.2019.8982413

[23] Azka Wani, Revathi S, and Rubeena Khaliq. 2021. SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL). *CAAI Transactions on Intelligence Technology* 6, 3 (2021), 281–290. https://doi.org/10.1049/cit2.12003 arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/cit2.12003

[24] Zhipeng Zhao and Bin Wu. 2017. Scalable SDN architecture with distributed placement of controllers for WAN. *Concurrency and Computation: Practice and Experience* 29, 16 (2017), e4030. https://doi.org/10.1002/cpe.4030 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.4030 e4030 CPE-16-0369.

[25] Liehuang Zhu, Md Monjurul Karim, Kashif Sharif, Fan Li, Xiaojiang Du, and Mohsen Guizani. 2019. SDN Controllers: Benchmarking &amp; Performance Evaluation. https://doi.org/10.48550/ARXIV.1902.04491

[26] Skhumbuzo Zwane, Paul Tarwireyi, and Matthew Adigun. 2019. A Flow-based IDS for SDN-enabled Tactical Networks. In *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*. IEEE, Vanderbijlpark, South Africa, 1–6. https://doi.org/10.1109/IMITEC45504.2019.9015900