
# Scalable Physical Layer Security Components for Microservice-Based Optical SDN Controllers

# Scalable Physical Layer Security Components for Microservice-Based Optical SDN Controllers

Carlos Natalino[(1)], Carlos Manso[(2)], Ricard Vilalta[(2)], Paolo Monti[(1)], Raul Munõz[(2)], Marija Furdek[(1)]

[(1)] Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. carlos.natalino@chalmers.se
[(2)] Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Spain. ricard.vilalta@cttc.es

***Abstract*** *We propose and demonstrate a set of microservice-based security components able to perform physical layer security assessment and mitigation in optical networks. Results illustrate the scalability of the attack detection mechanism and the agility in mitigating attacks.*

## Introduction

Scalability of Software-Defined Networking (SDN) controllers has always been a concern. Traditionally, (optical) SDN controllers are developed as a software monolith, i.e., a single software entity deployed and scaled as a whole[1],[2]. However, network growth in terms of the number of devices and services to monitor and control poses important challenges related to, among other, synchronization between multiple controller instances and their efficient scaling with the number of tasks to perform.

While most of the (optical) SDN controller operations relate to the set-up, reconfiguration, and/or tear down of a Connectivity Service (CS), there are other complex tasks that require continuous monitoring of all running CSs. An example are physical layer security assessment and mitigation procedures, where the SDN controller processes the Optical Performance Monitoring (OPM) data collected periodically from the optical devices, and, in case an attack is detected, the affected CSs are torn down or reconfigured, depending on the mitigation procedure. With the increasing number of CSs running in a network, efficient scaling of such operations is imperative.

In this work, we propose and demonstrate a set of microservice-based components that perform physical layer security assessment and mitigation operations in an optical transport network infrastructure. With this purpose in mind, we developed three components able to perform attack inference, detection, and mitigation, and integrated them into uABNO[3], a microservice-based optical SDN controller. We tested the scalability performance of these Physical Layer Security Components (PLSCs) by measuring their response time under different network load conditions. Results show that the proposed microservice-based implementation approach takes only 2 seconds from attack detection to mitigation, and can scale from 10 to 100,000 CSs without any noticeable impact on the attack detection response time.

978-1-6654-3868-1/21/$31.00 ©2021 IEEE

## The uABNO Architecture

Compared to a monolith, microservice architecture brings forth several advantages such as ease of including new functionalities and possibility to efficiently scale independent components. These benefits propelled microservices as the *de-facto* architecture currently used by the industry to develop the so-called *cloud-native applications*.

Fig. 1 shows the architecture of uABNO[3] that implements a cloud-native optical SDN controller. uABNO exposes network topology and CS to external applications (e.g., Operations Support Systems/Business Support Systems - OSS/BSS). CS requests are received through the North-Bound Interface (NBI) microservice, which translates Open Networking Foundation (ONF) Transport API (TAPI) requests to internal protocol buffers and forwards it to the connectivity microservice. Path computation, context, among other components, communicate through standardized protocols. More information is provided in[3].

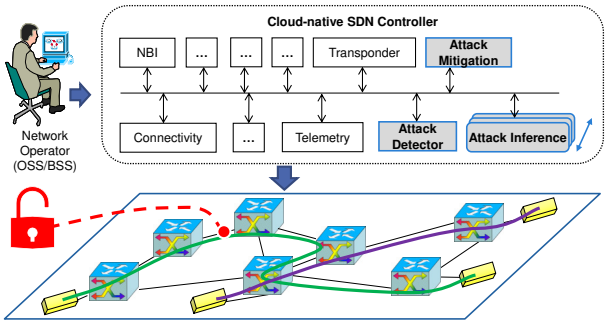


**Fig. 1:** Architecture of the uABNO SDN controller including the optical physical layer security components.

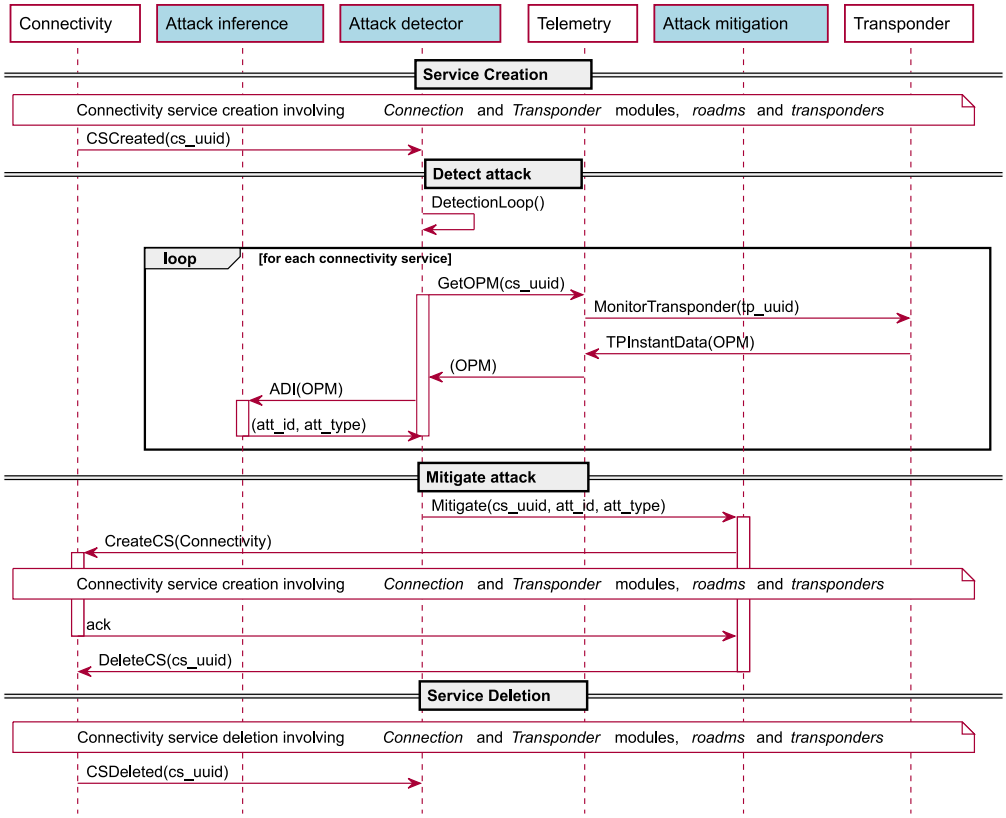


**Fig. 2:** Interactions among the PLSCs and the other uABNO components. The processes used for the creation of CSs are simplified since they are not relevant for the use case under exam.

The uABNO controller in[3] did not have any physical layer security management capabilities. Here, we extend it by including an attack detector, an attack inference, and attack mitigation components (dark gray in Fig. 1) The attack inference component contains a Machine Learning (ML) model that performs Attack Detection and Identification (ADI) based on OPM data[4]. The attack detector processes the output of the attack inference component to get the correct security assessment for each CS in the network, and triggers the attack mitigation component when an attack is detected. The attack mitigation component is responsible for counteracting the detected attack, e.g., by re-routing a vulnerable CS[5], or by computing and allocating a backup protection path[6].

### Security Assessment and Mitigation

Fig. 2 details the interaction among the uABNO components when performing security assessment. During the creation/deletion of CSs, the connectivity component notifies the attack detector about the change administered to a particular CS. This enables the attack detector to maintain a list of all the active CSs in the network.

Attack detection operations are periodically performed by a detection loop shown in Fig. 2. The attack detector obtains the latest OPM data from the active transponders in the networks. These data are then sent to the attack inference component for ADI purposes. This instance of the attack inference component uses a supervised learning model that only needs the latest optical performance data point. It would also be possible to use an unsupervised learning model, but in this case a sequence of OPM data (and not only the latest points) might be necessary[4]. Since the three PLSCs are implemented separately they can also be scaled independently, depending on their needs. For example, when the number of CSs increases but no attacks are detected, only the attack inference and detector components need scaling, while the attack mitigation does not.

Once an attack is detected, the attack detector component informs the attack mitigation component about the affected CSs and the attack ID (i.e., the attack type). The attack mitigation component is responsible for computing ways to circumvent the vulnerability of the CS. In this specific implementation the mitigation component implements a simple yet efficient solution to counteract power jamming attacks. It first tries to establish a new CS between the same source and destination of the vulnerable CS.If a new CS is found, the vulnerable CS is deleted. If not, the

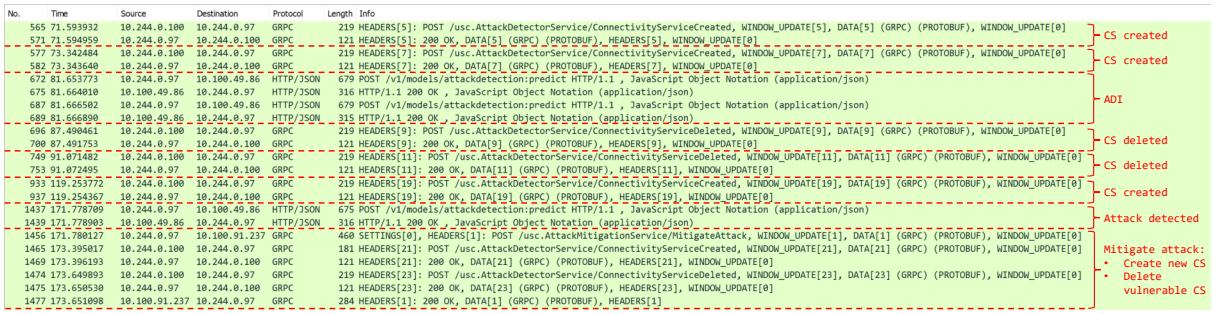


**Fig. 3:** Messages exchanged by the attack detector component. The attack detector has IP=10.244.0.97. The connectivity component has IP=10.244.0.100, the attack inference has IP=10.100.49.86, and the attack mitigation has IP=10.244.0.237.

affected CS is torn down nonetheless.The attack detector is notified about the creation of the new CS and the deletion of CS affected by the attack.

## Experimental results

To validate the effectiveness and the scalability performance of the proposed PLSCs, we use a uABNO deployment (managed by Kubernetes) over an emulated optical network. The transponders' data are collected from a real-world testbed[4] representing $(i)$ normal operating conditions and $(ii)$ two types of jamming attacks: in-band and out-of-band jamming. The neural network developed in[4] is used in the attack inference component. The attack detection loop is executed every 30 seconds.

Fig. 3 illustrates the communication between the PLSCs and the connectivity component. In the first part, the connectivity component notifies the attack detector about the creation of two CSs (at Time = 71 and 73 s). The attack detection loop is triggered and the attack inference component performs the ADI procedure for both CSs (Time = 81 s). In this case, no attack is detected and the attack mitigation component is not called. Shortly after, the CSs are deleted.

At Time = 119 s, a new CS is created. After that, the attack detection loop is executed and an attack is detected affecting the new CS. The attack detector component notifies the attack mitigation component, which starts the process of creating a new CS and deleting the vulnerable one. Note that only 2 seconds are needed to perform the attack mitigation actions (i.e., they start at 171 s and finish at 173 s). Obviously, this time may vary depending on the reconfiguration time of the devices involved.

Finally, Fig. 4 shows the response time and the number of replicas generated by Kubernetes when the attack inference component needs to monitor an increasing number of CSs at the same time. We can observe that the response time

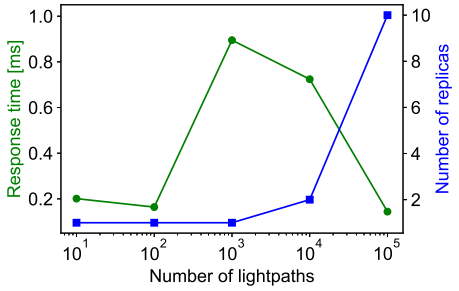


**Fig. 4:** Average response time and number of replicas of the attack inference component.

has a minimal variation in the order of below one millisecond when the number of CSs varies from 10 to 100,000. We also see that 1,000 CSs shows the highest response time. This is because 1,000 CSs is not sufficient to trigger the scaling. However, when the attack inference component is replicated multiple times (i.e., for 10,000 and 100,000 CSs) the response time decreases sharply to the level corresponding to 10 or 100 CSs. Although able to scale themselves too, neither the attack detection nor the attack mitigation component scaled during this experiment. This demonstrates one of the benefits of having PLSCs implemented as separate microservices, i.e., a component is scaled only if needed regardless of the scaling decision taken for the others.

## Conclusions

In this paper we propose and demonstrate the use of microservice-based PLSCs integrated with uABNO. Results indicate that the solution can trigger actions to mitigate power jamming attacks in approximately 2 seconds. We also show that by enabling replication, the solution can maintain a stable response time even with large variations in the number of monitored CSs.


## Acknowledgements

Work partially supported by the EC H2020 TeraFlow (101015857), Spanish AURORAS (RTI2018-099178-I00) and Vetenskapsrådet (2019-05008).

## References


[1] S. Secci *et al.*, “Security and performance comparison of ONOS and ODL controllers”, 2019. [Online]. Available: https://opennetworking.org/wp-content/uploads/2019/09/ONOSvsODL-report-3.pdf.

[2] R. Guerzoni *et al.*, “Analysis of end-to-end multi-domain management and orchestration frameworks for software defined infrastructures: An architectural survey”, *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, e3103, 2017. DOI: https://doi.org/10.1002/ett.3103.

[3] R. Vilalta *et al.*, “uABNO: A cloud-native architecture for optical SDN controllers”, in *Optical Fiber Communication Conference (OFC)*, 2020, T3J.4.

[4] M. Furdek *et al.*, “Machine learning for optical network security monitoring: A practical perspective”, *J. Lightw. Technol.*, vol. 38, no. 11, pp. 2860–2871, 2020. DOI: 10.1109/JLT.2020.2987032.

[5] Y. Li *et al.*, “Fast lightpath hopping enabled by time synchronization for optical network security”, *IEEE Commun. Lett*, vol. 20, no. 1, pp. 101–104, 2016. DOI: 10.1109/LCOMM.2015.2497703.

[6] M. Furdek *et al.*, “Attack-aware dedicated path protection in optical networks”, *J. Lightw. Technol.*, vol. 34, no. 4, pp. 1050–1061, 2016. DOI: 10.1109/JLT.2015.2509161.