Scalable and Efficient Pipeline for ML-based Optical Network Monitoring

Carlos Natalino,[1,*] Lluis Gifre[2], Raul Muñoz[2], Ricard Vilalta[2], Marija Furdek[1], Paolo Monti[1]

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

Abstract: We demonstrate a scalable processing of OPM data using ML to detect anomalies in optical services at run time. A dashboard will show operational SDN controller metrics, raw OPM data, and the ML assessment results. © 2023 The Author(s)

1. Overview

Optical networks are evolving towards disaggregated infrastructures with frequently executed closed-loop automation processes, often relying on Machine Learning (ML) models. Software-Defined Networking (SDN) and ML are two main enablers of this evolution, allowing for a logically centralized control of the network and for data-driven models to aid decisions, respectively. Autonomous operations allow for low(er) safety margins and quick(er) reaction to potential service degradation and malfunctioning equipment. However, scalable and efficient realization of this concepts brings forth several challenges. One challenge relates to the scalability features of the monitoring data collection process in the presence of an increasing number of devices. Network telemetry is one viable solution to this problem [1]. Another challenge relates to building suitable pipelines that process the monitoring data in real time, possibly taking advantage of ML models/ensembles whose composition may change over time. Currently, solutions taking advantage of open-source software and doing exactly this have been demonstrated, e.g., [2]. However, the currently available pipelines are monolithic in nature, which makes their scaling resource inefficient and time consuming (unless redundant resources are pre-allocated).

In this demonstration, we showcase a scalable, efficient, and flexible microservice-based pipeline that tackles the limitations of conventional monolithic monitoring solutions. The pipeline is implemented over ETSI TeraFlowSDN [3], an open-source, microservice-based SDN controller. In the envisioned use case, TeraFlowSDN deploys and monitors a variable number of optical services monitored through the periodical collection of Optical Performance Monitoring (OPM) data.The objective is to detect potential anomalies affecting the running optical services, with the help of a few, pre-selected ML models. The audience will be able to interact with the demo using a dashboard that visualizes the pipeline inputs/outputs, and allows for selection of the following options: *(i)* the ML model to be applied on the OPM data, i.e., Supervised Learning (SL) and/or Unsupervised Learning (UL); and *(ii)* the number of active optical services in the network. The results visualized via the dashboard will highlight how, thanks its microservice-based nature, the TeraFlowSDN monitoring pipeline replicates only the functionalities that need scaling up, thus improving the resource efficiency and scalability of the pipeline.

2. Innovation

This demonstration showcases a proof-of-concept of a scalable, efficient, and flexible pipeline capable of monitoring a large number of optical services with a frequent monitoring cycle (e.g., every 30 seconds). We focus on the processing of OPM data by ML models that aims at detecting anomalies in the optical services. Scalability of the solution is demonstrated by varying the number of optical services from a few to several hundreds while maintaining the targeted monitoring cycle. Efficiency of the solution is demonstrated by the ability of the pipeline to dynamically self-adjust the amount of committed computational resources. Finally, to illustrate the flexibility of the solution, UL and SL models can be dynamically activated/deactivated, providing the ability to not only detect an anomaly, but to classify it as well if needed. Through a dashboard, the audience is able to select how many optical services are active in the network at any time, as well as which ML model(s) to be deployed, and they are able to see how these choices impact the final solution of the anomaly detection process.

3. Components and Workflow

The architecture of the demonstrator is illustrated in Fig. 1. The TeraFlowSDN controller manages the optical network. A transparent Open Line System (OLS) exposes the management configurations of all optical devices. The OPM data from the devices is sent to the monitoring component of TeraFlowSDN through the South-Bound

Interface (SBI), while requests for optical services are received by TeraFlowSDN via the North-Bound Interface (NBI). The demonstrated pipeline is composed of four components: *manager*, *service monitor*, *ML inference*, and *mitigator*. Except for the *manager*, all other components are stateless and can be replicated when their load increases. A dashboard based on Grafana provides a visualization and configuration interface through which the audience can configure the use case and observe how performance varies in real time.

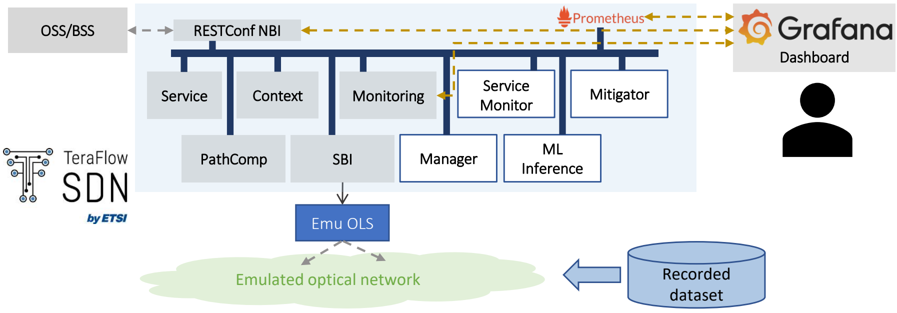

Fig. 1. Architecture of the demonstrator.

Figure 2 illustrates the communication among the pipeline components, and between the pipeline components and the SDN controller. The demonstration comprises two stages: optical service setup and the monitoring loop. During service setup, a *customer* (Fig. 2) requests an optical service. The SDN controller triggers its internal processes to set the service up. At this stage, the *manager* needs to *(i)* be notified of the creation/deletion of services and *(ii)* create a list of Key Performance Indicators (KPIs) for the ML assessment of the newly created service. Thanks to *(i)*, the *manager* maintains a list of optical services currently active in the network.

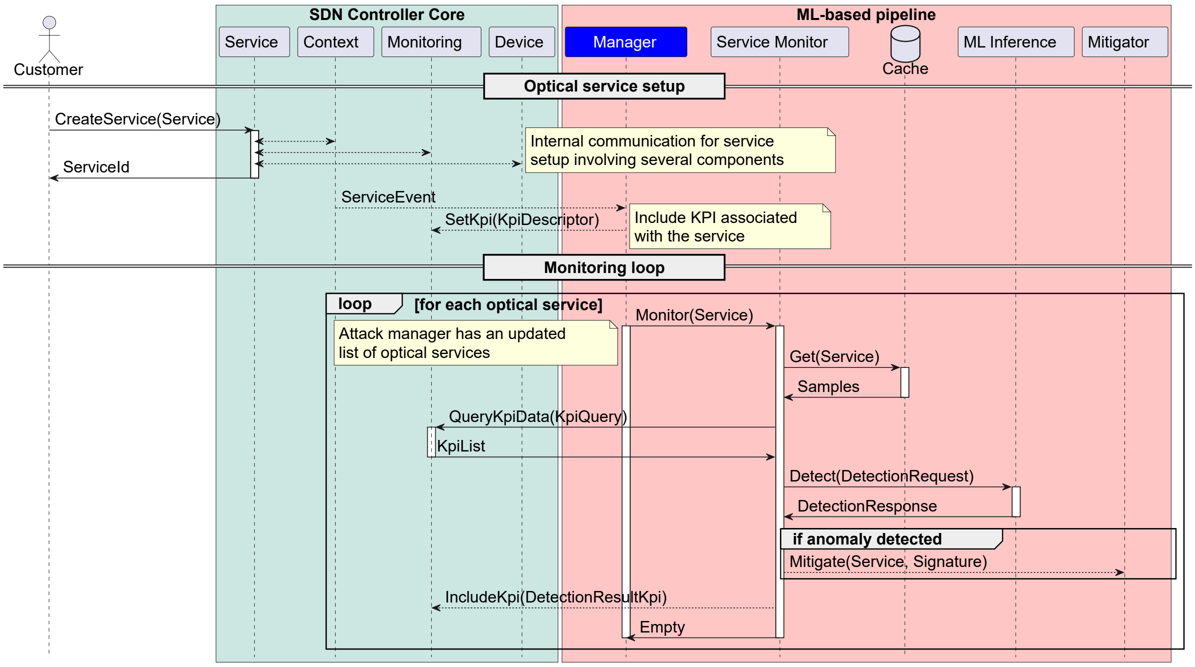

Fig. 2. Communication between the different components in the demonstrator.

In the second stage, the *manager* periodically triggers the monitoring loop according to a pre-defined frequency. Using its internal list of active optical services, the *manager* asks the *service monitor* to check them one by one. The requests are sent concurrently, each with a service ID, allowing the replicas of the *service monitor* to handle the pipeline of each service independently and balance the load among them. For each optical service, the monitoring pipeline works as follows. First, the latest OPM data are retrieved from the *monitor*. To alleviate the load on the *monitor*, a copy of latest OPM samples of all services can be saved in the *cache* (e.g., when using UL [4]). Once the OPM data is retrieved, the *service monitor* calls the appropriate ML model(s) within the *ML inference* component. The results from the *ML inference* are included in the *monitoring* database with the same timestamp of the monitoring sample that generated it, thus enabling visualization and correlation afterwards. If an anomaly is detected, the *service monitor* notifies the *mitigator* of the affected service ID and the anomaly signature. The *mitigator* is responsible for defining a solution to mitigate the anomaly. Mitigation strategies are outside the scope of this demonstration. We adopt a make-before-break strategy where a new optical service is established over a node-disjoint path, the traffic is moved to the new service, and the service affected by the anomaly is dropped.

4. Demo implementation

The architecture of the demonstrator illustrated in Fig. 1 is implemented as follows. We use the ETSI TeraFlowSDN controller deployed over a Kubernetes cluster as the basis of our demonstrator. An emulated Open Line System (eOLS) interacts with the SBI of TeraFlowSDN, mimicking a mesh optical network and replaying OPM data from a recorded dataset. The eOLS allows us to select in real time which specific parts of the dataset (representing the desired anomaly) to replay for specific optical services. The audience will interact with the demonstrator through a custom Grafana dashboard that allows the selection of a few demonstrator parameters and plots the time series of several metrics. The TeraFlowSDN components use gRPC to exchange messages and are deployed as containers. We take advantage of the Kubernetes replication feature to scale the components by increasing/decreasing their number of replicas. The *manager* is the only component that does not replicate.During initialization, it retrieves the list of currently running optical services from the SDN controller. Moreover, the failover features of Kubernetes ensure that the loop is resilient to the failure of this component. Linkerd service mesh is used for load balancing among replicas.

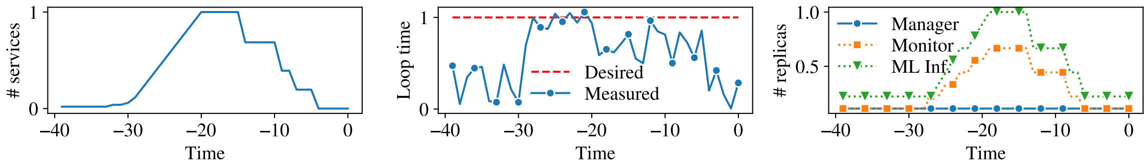

Fig. 3. Mock of the most relevant dashboard plots showing hypothetical normalized values.

To provide diverse anomaly signatures, the emulated optical network replays normal operating conditions and two types of anomalies, i.e., physical layer attacks from [5], and anomalous OPM signatures inspired by the spectrum filtering anomalies reported in [6]. Figure 3 illustrates three of the main KPIs to be shown by the dashboard, in a hypothetical scenario. At the beginning, only a few optical services run on the network, the measured monitoring loop time is well below the desired value, and the number of replicas is small. After we deploy more services in the network, we observe that the loop time increases, even surpassing the desired value in a few instances, but stabilizing below the desired value due to the increase in the number of replicas. In this illustrative example, we focus on the case where an UL model is used, which incurs a high processing time required for the inference calculation [4, 7]. This translates into the fact that the *ML inference* requires more replicas than the *service monitor*. However, during the demonstration, the audience will be able to select whether to use SL, UL, or both. Other relevant operational metrics not illustrated here due to space constraints will also be included, such as the average CPU usage over the replicas of each pipeline component and the response time of each individual component.

5. OFC Relevance

This demonstration will be the first to showcase a proof-of-concept pipeline that meets the scalability requirements of fully disaggregated, fine-granular and ML-based OPM in large-scale optical network deployments. These topics are currently very relevant to OFC's audience. The custom dashboard developed for this demonstration will enable the audience to set parameters and observe in real time their impact on the experiment outcomes, computational load, and monitoring loop time. We expect that the highly interactive setup will foster discussions and further ideas for development in the OFC audience. Scripts for reproducing the demo will be publicly available at the ETSI TeraFlowSDN repository.

Acknowledgment: This work has been supported by EC H2020 TeraFlow (101015857) and Vetenskapsrådet (2019-05008).

References

1. F. Paolucci *et al.*, "Network telemetry streaming services in sdn-based disaggregated optical networks," J. Light. Technol. **36**, 3142–3149 (2018). DOI: 10.1109/JLT.2018.2795345.
2. M. Balanici *et al.*, "Demonstration of a real-time ML pipeline for traffic forecasting in AI-assisted F5G optical access networks," in *Proc. of ECOC,* (2022), p. Tu2.5.
3. R. Vilalta *et al.*, "TeraFlow: Secured autonomic traffic management for a tera of SDN flows," in *Proc. of EuCNC/6G Summit,* (2021), pp. 377–382. DOI: 10.1109/EuCNC/6GSummit51104.2021.9482469.
4. C. Natalino *et al.*, "Microservice-based unsupervised anomaly detection loop for optical networks," in *Proc. of OFC,* (2022), p. Th3D.4.
5. C. Natalino *et al.*, "Experimental study of machine-learning-based detection and identification of physical-layer attacks in optical networks," J. Light. Technol. **37**, 4173–4182 (2019). DOI: 10.1109/JLT.2019.2923558.
6. A. Vela *et al.*, "BER degradation detection and failure identification in elastic optical networks," J Light. Technol. **35**, 4595–4604 (2017). DOI: 10.1109/JLT.2017.2747223.
7. C. Natalino *et al.*, "Scalable physical layer security components for microservice-based optical SDN controllers," in *Proc. of ECOC,* (2021), p. We3E.2. DOI: 10.1109/ECOC52684.2021.9605943.