# A Live Demonstration of In-Band Telemetry in OSM-Orchestrated Core Networks

Thomas Dreibholz*, Mah-Rukh Fida‡*, Azza H. Ahmed*, Andrés F. Ocampo*, Foivos I. Michelinakis*

* SimulaMet – Simula Metropolitan Center for Digital Engineering
Center for Resilient Networks and Applications
Pilestredet 52, 0167 Oslo, Norway

‡ University of Gloucestershire – School of Computing and Engineering
The Park, GL50 2RH, Cheltenham, United Kingdom

Email: {dreibh,mahrukhfida,azza,andres,foivos}@simula.no

***Abstract*—Network Function Virtualization is a key enabler to building future mobile networks in a flexible and cost-efficient way. Such a network is expected to manage and maintain itself with minimum human intervention. With early deployments of the fifth generation of mobile technologies – 5G – around the world, setting up 4G/5G experimental infrastructure is necessary to optimally design Self-Organising Networks (SON). In this demo, we present a custom small-scale 4G/5G testbed. As a step towards self-healing, the testbed integrates Programming Protocol-independent Packet Processors (P4) virtual switches, that are placed along interfaces between different components of transport and core network. This demo not only shows the administration and monitoring of the Evolved Packet Core VNF components, using OPEN SOURCE MANO, but also serves as a proof of concept for the potential of P4-based telemetry in detecting anomalous behaviour of the mobile network, such as a congestion in the transport part.**

***Index Terms*—Network Function Virtualisation (NFV), Open Source MANO (OSM), P4, Telemetry, Anomaly Detection**

## I. INTRODUCTION

On the journey towards 5G networks, aim is not only to cater for a wide spectrum of services with diverse requirements, but also to automate service assurance by performing self-healing and scaling. Software-Defined Networks (SDN) and Network Function Virtualisation (NFV) are key architectures that provide an infrastructure for a 5G network, such that to orchestrate and control its resources and enable efficient, flexible and scalable provision of its network services. For smooth fulfillment of Service Level Agreements (SLA) and self-management, the infrastructure needs fault avoidance, detection and correction across multiple layers of the network stack.

Building 4G/5G experimental setups that integrate the so-called 5G key enablers including self-healing solutions, is essential for future mobile networks design. In this demo, we present a custom, small-scale 4G/5G testbed based on the open source LTE implementation from OPENAIRINTERFACE[1] [1], [2]. The testbed integrates P4 software switches at the Radio Access Network (RAN) and Evolved Packet Core (EPC), with the EPC including the P4 switches deployed as Virtual Network Functions (VNF). For self-healing, continuous monitoring of the network is necessary, so that to detect, diagnose and respond faster to an unusual or unwanted network behaviour.

In this demo, we demonstrate flagging of congestion at the RAN, with in-band network telemetry based on Programming Protocol-independent Packet Processors (P4). As a part of the self-healing system, P4 virtual switches are deployed at different network interfaces that regularly track behaviour of the ongoing traffic flows, as well as switch-queue status. We demonstrate that the P4-based telemetry can signal out degraded end-to-end performance with the same accuracy as an end-user can. The demonstration hints that if a self-learning process from AI/ML is incorporated in the network, it may help in an early signalling of an unwanted behaviour, which in turn can assist in ensuring a seamlessly good Quality of Experience (QoE) of the end-users. In addition, we also demonstrate the usage of OPEN SOURCE MANO to orchestrate our setup, i.e. managing the VNF instances of EPC and P4 switches in a virtualisation infrastructure, including the handling of basic instance telemetry data.

## II. BACKGROUND

### A. VNF-based EPC

The SIMULAMET EPC[2] [1]–[3] VNF is our open source EPC implementation, based on OPENAIRINTERFACE. For the NFV orchestration of Network Services (NS) with this VNF, we are using OPEN SOURCE MANO[3] (OSM). Basically, OSM takes care of [4, Chapter 1]:

- Composition of VNFs into Network Services (NS),
- Instatiation of NSs and their VNFs in an underlying Network Function Virtualisation Infrastructure (NFVI) as so-called Virtual Deployment Units (VDU), which are virtual machines and/or containers;
- Run-time configuration of the VDUs;
- Monitoring of the VDUs (details in Subsection II-A);
- Scaling and removal of VDUs.

### B. Network Monitoring and Telemetry

*1) Monitoring NFV:* OSM already provides two ways of monitoring the deployed NSs: (1) by using features of the

[1] OPENAIRINTERFACE: https://www.openairinterface.org.

[2] SIMULAMET EPC: https://github.com/simula/5gvinni-oai-ns.

[3] OPEN SOURCE MANO: https://osm.etsi.org.

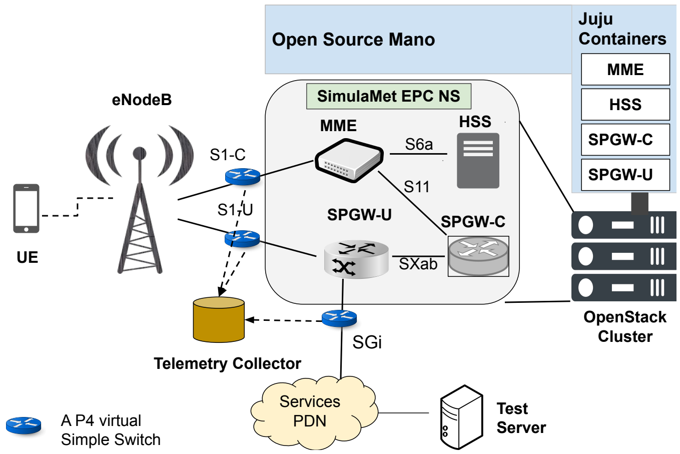


Fig. 1. SIMULAMET EPC with OPENSTACK as NVFI and with integrated virtual P4 simple switches.

NFVI, (2) By JUJU Charms, that run customised monitoring code as part of the configuration service managing the VDUs. However, this monitoring only covers coarse metrics [3] – like CPU utilisation, packet counters, etc. – and does not represent the actual *user QoE perspective* of the system performance. For this, a user-centric in-band network monitoring is useful.

*2) In-band Network Monitoring:* In-band network telemetry [5] is a type of passive network monitoring strategy that is widely used in data centres [6], [7] to track the resource status of switches and the characteristics of the data flow across their interfaces. It uses the data plane to directly drive the network measurement process, by inserting a telemetry header in passing-by packets. The data packets are modified/appended with telemetry information, such as the identifier of the switch or relay router, using the P4 programming language [8] via match-action tables that are configurable by the network controller. The version of the data plane packets with embedded telemetry data are then mirrored to the network monitoring engine.

In-band network monitoring helps in collecting latency observed by data packets, events of congestion, micro-bursts, elephant flows, queue status of the switches and lossy links, to name a few. Unlike previous studies, we deploy P4 switches to monitor the data-plane status of a mobile network.

## III. System Setup

For our setup, we consider the network scenario depicted in Figure 1. This network scenario is composed of User Equipment (UE), an eNodeB, the EPC, P4 virtual switches, a telemetry collector that tracks in-band network telemetry from P4 switches, and a server residing outside the mobile network in a packet data network (PDN; here: the Internet). The EPC, including the P4 switches, is realised as NS based on the SIMULAMET EPC VNF and P4 switch VNF instances in OPENSTACK as NFVI, orchestrated by OSM. OSM uses JUJU Charms for configuration (i.e. there is one Charm container for each VDU, each managing its assigned VDU). These Charms also collect basic performance metrics (like CPU and memory utilisation, packet counts, etc.) within their corresponding VDUs.

UEs, i.e. smartphones and computers with LTE USB modem, communicate via LTE radio with the eNodeB. The eNodeB is running OPENAIRINTERFACE eNodeB software, and equipped with a software-defined radio board (in our case: ETTUS B210). Over the EPC, the UEs can communicate with the Internet and particularly also the test server.

The P4 switches forward the traffic on the EPC interfaces, i.e. S1-C, S1-U and SGi. Particularly, due to our custom P4 programming, they can attach telemetry information to packets, and remove it later, as well as forward relevant telemetry information to our Telemetry Collector server. The P4 switching is implemented using the Behavioral Model Version 2 (BMv2) Simple Switch software implementation[4]. A future version of our setup may also use off-the-shelf P4 hardware switches, when available.

The Telemetry Collector server stores all telemetry data and correlates the information received from the different P4 switches in the EPC. It particularly can also perform analysis, based on a self-learning process from AI/ML. For instance, an anomaly detection model can be used to identify unexpected behaviour in the network, e.g. a bottleneck. Moreover, by leveraging the supervised learning paradigm, we can identify the cause and the location of the bottleneck.

## IV. Live Demonstration

In our live demonstration, we would like to show the following features of our setup:

### A. *Orchestrating the VNFs with* OPEN SOURCE MANO

From the perspective of OSM, we would like to highlight its NS and VNF management features. Particularly, we would first like to show to the audience how we solved the challenges of managing the complex VNF setup, with some details on the build procedures, configuration options, and the management of the instance configurations in OSM. This includes some details on the OSM-based telemetry [3] of our setup as well.

### B. *Monitoring of End-to-End Network Performance*

In a 5G network, a main challenge is to deal with tracking the end-to-end performance when multiple network slices share the network, compute and memory resources. End-users are the entities, to which the network has to provide their desired QoE. To track QoE, the most reliable monitoring point is the end-user device. It is however both, expensive in terms of data capacity capabilities, and is not preferred by users due to privacy and security concerns [9]. Secondly, frequent transfer of monitored performance metrics to a central monitoring engine of the network is costly both, in terms of time and bandwidth.

An alternative method is to perform monitoring at the junction points of the mobile networks, such as virtual P4 switches. The switches not only track the characteristics of the data flow from end-users. They also monitor the status of the link, where it is deployed.

[4]See https://github.com/nsg-ethz/p4-learning/wiki/BMv2-Simple-Switch.

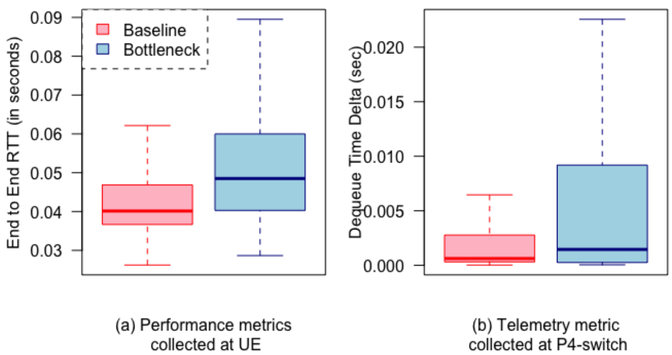


Fig. 2. Monitoring parameters showing the difference between baseline and a bottleneck scenario at two different components of the mobile network.

In our demo, we will demonstrate that telemetry from a P4 switch can be used, as an alternative to end-user monitoring, in detecting end-to-end performance issues. The performance issue we introduce is a bottleneck at the transport part of the mobile network.

### C. Test Scenarios

To show the feasibility of using virtual P4-switch-based telemetry in detecting an end-to-end performance drop, we take two test scenarios. The first one represents a "baseline" scenario, where the UE receives downlink TCP `iperf` data from the server node at a rate set to the maximum bandwidth of the end-to-end path. To track end-to-end performance, we collect the Round-Trip Time (RTT) by sending UDP `ping` messages from the UE to the server, every second. The second scenario is that of a "bottleneck", where along with the data transfer of the "baseline", congestion is created on S1-U by sending `iperf` TCP traffic, at the maximum available bandwidth of the S1-U link, from the SPGW-U VDU to the eNodeB machine. Note that in the second scenario, there are two concurrent TCP `iperf` downlink data transfers on the S1-U link.

As discussed above, in both the scenarios, the UE is tracking end-to-end performance with a delay measure. Simultaneously, the P4 switch at the S1-U clones the incoming packets. It then adds a telemetry information of `dequeue time delta` to the cloned packets, via an IP Option header, and mirrors it to the telemetry collector. The parameter `dequeue time delta` represents the time that a packet spends in the P4 switch's queue. Note, to reduce the processing burden on the switch, the P4 program can be designed to only add telemetry at every $n^{th}$ packet of a flow.

Figure 2 depicts the distributions of the performance metrics values, collected by the UE and the P4 switch at S1-U. We can observe an obvious difference in the network performance features, of end-to-end `RTT` and `dequeue time delta` during the the two scenarios. The $3^{rd}$ quantile of the end-to-end `RTT` increases by 28%, while that of `dequeue time delta` increases by 231%, pointing out additional delay caused by the congestion in the transport network. To rigorously confirm that the difference between the distributions are statistically significant, we employ the GNU R package $lmPerm$ [10]. It executes a non-parametric permutation test, to compare the distributions. Both, for end-to-end `RTT` and `dequeue time delta`, the difference between the metrics from the two set of experiments, is statistically significant with a $p$-value of $< 2.2 * 10^{-16}$.

This initial study depicts the potential of using P4 switches, both for monitoring network segments and end-to-end performance in 4G/5G mobile networks.

## V. Conclusions

With the steadily increasing ubiquity of mobile networks, and the complexity of the underlying NFV deployments, anomaly detection based on the user QoE becomes a crucial task in maintaining the networks and assuring their services. In our proposed demo, we would like to demonstrate our proof-of-concept setup, realising P4-based in-band telemetry in a 4G/5G testbed, based on open source software. Particularly, we would like to highlight our solutions for orchestration of the testbed and collection of in-band telemetry, as well as demonstrate its potential for tracking end-user QoE and anomaly detection.

## References


[1] T. Dreibholz, "Flexible 4G/5G Testbed for Mobile Edge Computing using OpenAirInterface and Open Source MANO," in *Proceedings of the 2nd International Workshop on Recent Advances for Multi-Clouds and Mobile Edge Computing (M2EC) in conjunction with the 34th International Conference on Advanced Information Networking and Applications (AINA)*, Caserta, Campania/Italy, 2020.

[2] ——, "A 4G/5G Packet Core as VNF with Open Source MANO and OpenAirInterface," in *Proceedings of the 28th IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Hvar, Dalmacija/Croatia, 2020.

[3] M. Xie, J. S. Pujol-Roig, F. I. Michelinakis, T. Dreibholz, C. Guerrero, A. G. Sánchez, W. Y. Poe, Y. Wang, and A. M. Elmokashfi, "AI-Driven Closed-Loop Service Assurance with Service Exposures," in *Proceedings of the 29th IEEE European Conference on Networks and Communications (EuCNC)*, Dubrovnik, Dubrovnik-Neretva/Croatia, 2020.

[4] A. Reid, A. González, A. E. Armengol, G. G. de Blas, M. Xie, P. Grønsund, P. Willis, P. Eardley, and F.-J. R. Salguero, "OSM Scope, Functionality, Operation and Integration Guidelines," ETSI, White Paper, 2019.

[5] L. Tan, W. Su, W. Zhang, J. Lv, Z. Zhang, J. Miao, X. Liu, and N. Li, "In-band Network Telemetry: A Survey," *Computer Networks*, vol. 186, p. 107763, 2021.

[6] B. Arzani, S. Ciraci, L. Chamon, Y. Zhu, H. Liu, J. Padhye, B. T. Loo, and G. Outhred, "007: Democratically Finding the Cause of Packet Drops," in *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2018, Renton, WA, USA, April 9-11, 2018*, S. Banerjee and S. Seshan, Eds. USENIX Association, 2018, pp. 419–435.

[7] Q. Huang, H. Sun, P. P. C. Lee, W. Bai, F. Zhu, and Y. Bao, "OmniMon: Re-architecting Network Telemetry with Resource Efficiency and Full Accuracy," in *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, H. Schulzrinne and V. Misra, Eds. Virtual: ACM, Aug. 2020, pp. 404–421.

[8] M. Budiu and C. Dodd, "The P4-16 Programming Language," *ACM SIGOPS Operating Systems Review*, vol. 51, no. 1, pp. 5–14, 2017.

[9] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and Privacy in Mobile Crowdsourcing Networks: Challenges and Opportunities," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 75–81, 2015.

[10] B. Wheeler and M. Torchiano, *lmPerm: Permutation Tests for Linear Models*, 2016, R package version 2.1.0. [Online]. Available: https://github.com/mtorchiano/lmPerm