Experimental Assessment of Automatic Optical Metro Edge Computing Network for Beyond 5G Applications and Network Service Composition

Bitao Pan, Fulong Yan, Xiaotao Guo, and Nicola Calabretta

***Abstract*—The upcoming 5G and beyond 5G heterogeneous applications with different quality of service (QoS) will impose strict latency, bandwidth, and flexibility requirements on optical metro access networks. Conventional cloud computing is gradually unable to fulfill the application requirements, especially on latency due to the distance causing propagation and networking delay. Therefore, the edge computing that distributed in metro access networks is promising to serve the applications with the requirements of ultra-low latency. As the resources of edge computing nodes are restricted and light compared with cloud data centers (DC), it is significant to manage across multiple edge computing nodes to enable joint allocation of the distributed resources. To address this issue, the optical metro network infrastructure should be flexible on the data plane and able to interact with the control and orchestration plane to automatically adapt to the communication requirements of multiple edge computing nodes. Related works have been focused on the simulation and numerical study. In this paper, an experimental testbed of a flexible optical metro access network including hardware and software components is built, and the performance is validated with real server traffic. The presented network system is based on the field-programmable gate array (FPGA), and hardware adapted open source network management and telemetry tools. Different from the commercial electrical switches, FPGA is fully programmable making it able to flexibly forward and monitor the traffic, in the meantime, to dynamically control the optical devices according to the feedback from the control plane. By exploiting dynamic software defined networking (SDN) control and network service orchestration, the proposed network is able to establish capacity adapted network slices for edge computing connections. Successful telemetry-assisted dynamic network service chain (NSC) generation, automatic bandwidth resources assignment, and QoS protection are demonstrated.**

***Index Terms*—Network automation, network function virtualization, optical metropolitan area networks, software defined networking.**

I. Introduction

THE upcoming 5G and the beyond 5G systems are expected to operate in a flexible and dynamic environment with the existence of multiple types of applications (i.e., IoTs, Industry 4.0, Artificial Intelligence) with different QoS requirements like latency, bandwidth, and reliability [1]–[5]. In addition, new applications working on sensors, robotics, and autonomous driving cars will generate a huge amount of data to be processed at the computing nodes [6]. However, data centers based cloud computing is facing difficulties in analyzing the generated data from these massively distributed end devices. Specifically, the huge amount of computation and management tasks need to be delivered to and served by the cloud, which imposes significant challenges on communicating and computing power of cloud computing. Moreover, many new types of applications (i.e., Autonomous Driving, Virtual Reality) are extremely latency-sensitive, which makes cloud computing would have difficulties to accommodate due to its location far away from the users [7]. As a promising technology, edge computing is potentially able to address the concerns of the restricted latency requirements, cloud computing alleviation, and bandwidth cost-saving [8]–[9]. Meanwhile, edge computing nodes distributed in optical metro networks will need to be properly served and managed by the networking system. Therefore, optical metro networks, as the infrastructure to support the upcoming new applications and edge computing interconnection, needs to be re-designed to provide the flexibility and ability to dynamically adapt to the requirements of the new use cases. First, the metro access network should be reconfigurable on the data layer and physical layer with the cooperation of the SDN controller to adapt to the diverse requirements. Second, a centralized orchestrator is necessary to comprehensively manage the system with respect to the network services and IT resources [10]–[11]. Finally, the network system should be monitored in real-time to make the orchestrator aware of the network status and the resource utilization for automatically optimizing the network performance and resource allocation.

Recent studies on optical metro access network have been focused on the control plane for flexible network and IT resources assignment [12]–[15], and on the data plane for the traffic flow programmability based on P4 [16]–[20]. We use FPGA as the multifunctional optical control and traffic processing (aggregation, classifier) & monitoring interface in this work. FPGAs provide a fully programmable data plane with the capability of per bit packet processing, making it able to flexibly aggregate, classify, forward, and monitor the traffic and in the meantime to control the optical devices accordingly. Moreover, the FPGA

Manuscript received November 22, 2020; revised February 10, 2021; accepted March 5, 2021. Date of publication March 9, 2021; date of current version May 16, 2021. This work was supported in part by the EU-H2020 project METRO-HAUL under Grant G.A. 761727 and in part by PASSION project under Grant GA 780326. *(Corresponding author: Bitao Pan.)*

The authors are with the Institute for Photonic Integration, Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: b.pan@tue.nl; f.yan@tue.nl; x.guo@tue.nl; n.calabretta@tue.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JLT.2021.3064800.

Digital Object Identifier 10.1109/JLT.2021.3064800

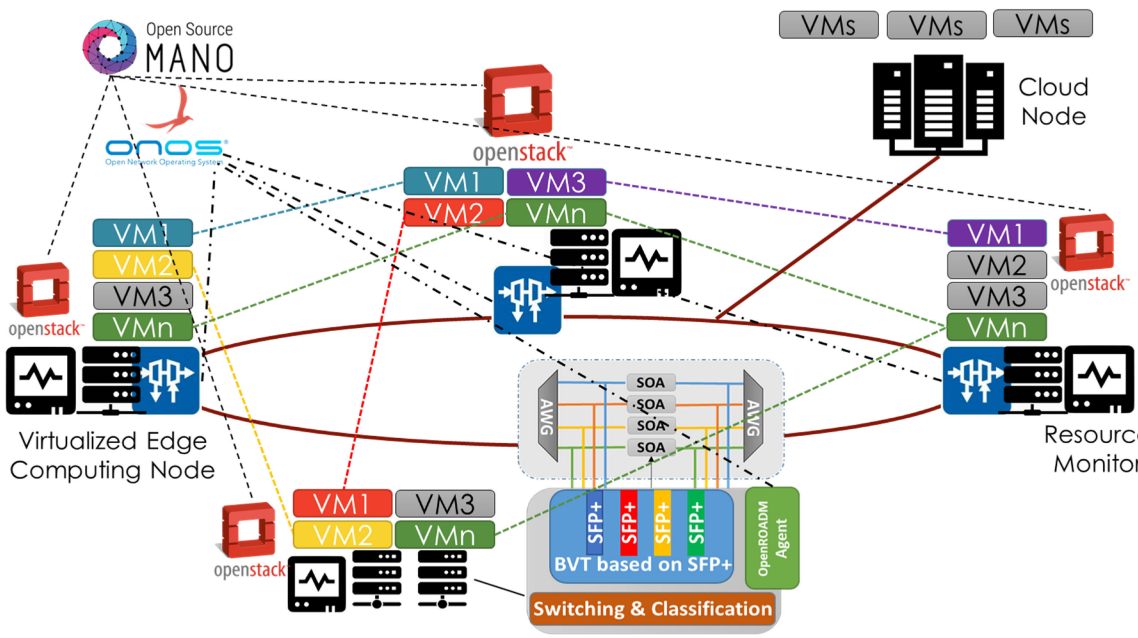

Fig. 1. Architecture of the proposed disaggregated and sliceable optical metro access network.

brings extra efficiency of traffic forwarding since the traffic from different ports can be parallel processed. The mentioned advantages make that the FPGA is more powerful than the other electrical switches like P4 based ones. Several other research efforts have been proposed to provide simulation and numerical optimization for the network service function chaining [21]–[24]. However, the automatic network reconfiguration, based on the programmable hardware with resource monitoring and application requirements to provide efficient edge computing services chaining has not been experimentally investigated and demonstrated.

Our previous work [11] addresses the challenges on the network efficiency by experimentally investigating a flex-reconfigurable optical network infrastructure. However, the proposed network is only investigated on the data plane without considering the efficiency of the precious edge IT resources. In this paper, we propose and demonstrate a metro-access network architecture with a programmable and flexible data plane, edge IT resource orchestrator, and telemetry for monitoring the network & IT resource usage. Open source SDN controller, edge computing virtualized infrastructure manager (VIM), and network function virtualization (NFV) orchestrator have been employed to automatically manage the edge IT resources, the network operation, and the service provision. To validate the proposed network architecture, a testbed of the proposed metro ring network with four nodes has been implemented. The performance has been assessed under the study case of dynamic NSC deployment. Results show that the optical network is dynamically reconfigurable with the collaboration of the SDN controller and FPGA traffic monitor. In addition, successful telemetry assisted and NFV orchestrated NSC deployment has been demonstrated. More than 50% bandwidth improvement has been achieved for the virtualized network function (VNF) connections with high bandwidth requirements.

II. Architecture and System Operation of the Metro Access Edge Computing Network

To efficiently and automatically manage the multiple-edge computing nodes to enable joint allocation of the distributed resources, a flexible and reconfigurable optical data plane and a telemetry assisted centralized control plane are necessary. The schematic of the proposed metro access ring network with the hardware and software building blocks is depicted in Fig. 1. In the data plane, each node consists of a disaggregated semiconductor optical amplifier (SOA) based reconfigurable optical add-drop multiplexer (ROADM) and an FPGA based flexible multi-function traffic interface. The SOA gates work as the wavelength blocker and can be turned on and off by the FPGA based traffic interface & ROADM controller to pass or block each single wavelength. In the meantime, the SOAs can provide optical amplification, replacing the high-cost EDFAs in the optical switching system [25]. Each node can add traffic only on the wavelength that is free or was blocked by the node itself. It is worth noting that the optical power of all the incoming wavelengths are splitted in order to achieve an optical bypass, or block and drop, or bypass and drop (for drop and continue operation) based on the destinations of each wavelength. In this operation, the node has to check the dropped data that only belong to itself based on the packet destinations, which is similar to the mechanism used in the passive optical network (PON) technology [26]. The wavelength allocation of the network is controlled by the FPGAs at each node via the SDN assigned network configuration. The SDN controller is based on the ONOS platform [27] where specific hardware drivers have been implemented to support and control the optical components through the NETCONF protocol [28]. The platform provides application interfaces with several high-level abstractions, through which the upper layer applications can learn about the status of

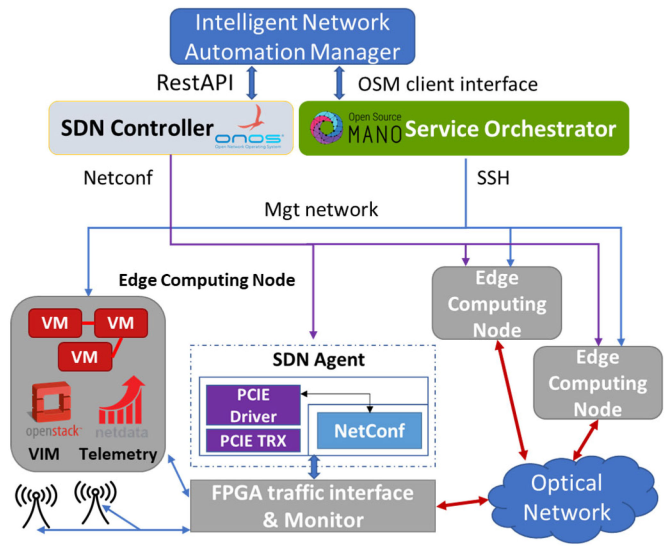

Fig. 2. Management system of the automatic metro access network.

the network and through which they can control the resources of the data plane (wavelength allocation and L2 forwarding) through the network. This simplifies the network management and optimization by hiding the underlayer hardware devices that need to be manually operated. Network configurations can be loaded and unloaded dynamically, via REST API or GUI, and without the need to restart the nodes. As shown in Fig. 2, the hardware SDN agent at the ROADM side to drive the optical components is PC based OpenROADM agent. It is based on the OpenROADM YANG model [29], specifically enhanced and adapted for the hardware control. In each conversation, the agent first extracts the SDN commands from the NETCONF packets, and further converts them to PCIe data flow. The FPGA will then translate the PCIe data to logical high or low signals to drive the SOA based ROADM. In addition, the controller monitors the traffic load in each port and the latency on a specific optical path by the uploaded information from FPGA. It allows the network management applications to optimize the network performance and resource utilization according to the real time status of the network. Specifically, the interface is real time monitoring the status of the FIFOs at each TX and RX (see Fig. 4). If the FIFO at a TX is almost full (2 cells to full), then an alarm signal will be immediately sent to the SDN, which will try to allocate more link capacity to the node or re-forward the latency insensitive traffic to the cloud or other possible edge computing nodes. Thus, the SDN controller is aware of the traffic load of each node and able to automatically re-configure or slice the network in an optimized way. In addition, the round-trip time (RTT) of a packet can be recorded by the timestamp unit (TSU) that generates and captures a timestamp from each processed packet. As a part of link QoS, the RTT is also sent to the SDN controller for better managing the network system.

Besides SDN, VIM is responsible for making the flexibility in the IT resource of the edge computing by virtualizing the hardware resource to a resource pool for the users. By the VIM, VNFs can be flexibly and efficiently deployed in the edge computing nodes and to further compose a VNF chain. At the edge computing side, an OpenStack [30] based VIM (see Fig. 2) is running for virtualized hardware management. The main function of the edge VIM is lifecycle management of the virtual machine (VM) instances through the edge computing network. Specifically, the VIM is in charge of creating, migrating, and deleting the VM instances, and managing the VM network interfaces. With the VIM, each edge computing node can be treated as a resource pool providing the required resource (computing, storage, memory) as a VM instance for the upper layer applications. As the resources of edge computing nodes are restricted and limited due to the economic consideration, it is significant to compose service chains across multiple-VIMs. Thus, OSM [31] is employed as the network service orchestrator for orchestrating the NSCs/VNFs placement in multiple edge computing nodes that are distributed in different locations. Each VIM is registered to the OSM via Secure Shell (SSH) link with a valid tenant name and identification information. The OSM is commanded by the upper layer program (see Fig. 2) by its specific client interface to manage the lifecycle of VNFs in each VIM. The OSM receives the requests as an NSC with a set of VNFs, virtual connections and requirements, and generates as output in which edge computing node each VNF should be instantiated. This requires having the visibility of both the available resources in the edge computing nodes and in the optical network that interconnects them. Therefore, the telemetry function of the network system is implemented in both optical data plane and edge computing nodes. The telemetry information of the optical data plane is implemented in FPGA and includes the network load (buffer fill ratio) in a specific port and the RTT in a specific connection. For the resource utilization and hardware performance monitoring of the edge servers, a Netdata [32] based opensource performance and resource monitoring system is employed. Netdata has been selected because it has a highly efficient database that stores the real time metrics of the resource usage for CPU, memory, hard disk, and network interfaces at 1-second granularity. Moreover, it is able to work in a master-slave mode, which helps the telemetry function implementation also with distributed multiple nodes. By the collaboration with ONOS, OpenStack, OSM, and the telemetry, the network is able to efficiently deploy and compose NSCs in the edge computing optical metro network. Thus, the clients can be served efficiently by the jointly allocated resource in the metro edge computing network.

III. Experiment Setup & Results Analysis

A. Testbed Setup

The experiment setup is shown in Fig. 3. It is composed of a ring network including four metro access nodes connected by 10 Km fiber between each node. At the optical metro network side, each node is equipped with an SOA-based 2-degree ROADM that controlled by the Xilinx UltraScale FPGA. The same FPGA is equipped with four SFP+ transceivers (ITU Ch21, 23, 25, 27) to emulate the BVT by dynamically turning ON/OFF each TRX. Moreover, the FPGA is employed also as a client interface at the access side performing an electronic Ethernet switch for aggregation, classification, switching and

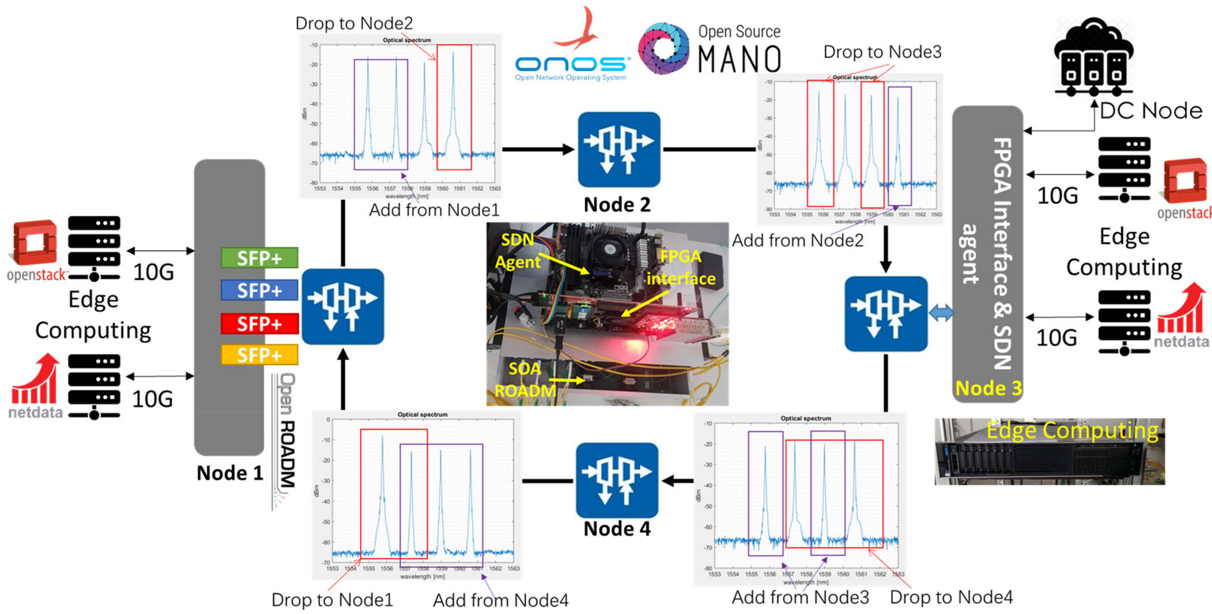

Fig. 3. Experiment setup of the proposed metro access network.

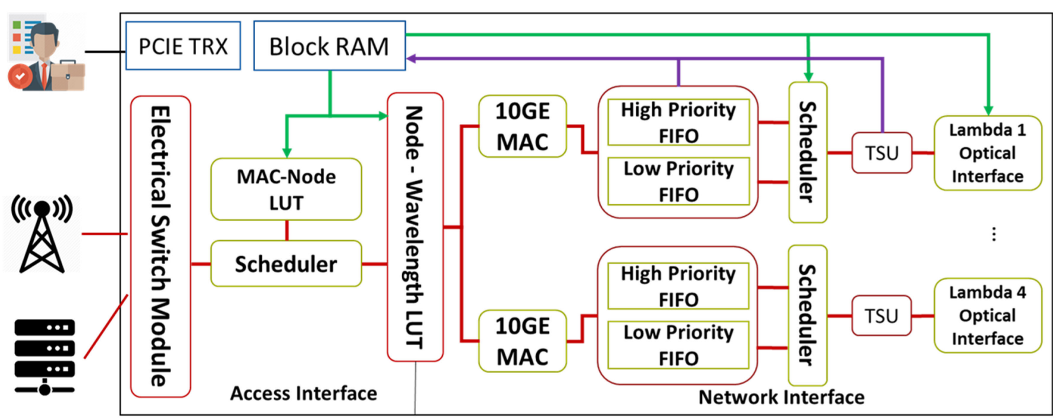

Fig. 4. FPGA based flexible network slicing interface & ROADM controller.

monitoring of the access traffics. The details of the FPGA implementation are depicted in Fig. 4, where the red, green and purple lines show respectively the data, control and monitor flows. The interface processes the Media Access Control (MAC) address of the incoming Ethernet packets in order to forward the packets to the correct destination ports. If the destination belongs to local network, then the packet will be directly forwarded to the related port at the access interface, otherwise, it will be aggregated to the priority FIFOs at the network interface according to its target node. The FIFOs in the network interface store XGMII data flow from the 10GbE MAC module. Before the data flow is sent out, there is a TSU that adds to each packet a 12 Bytes timestamp for network latency statistics and monitoring. The relation between packet MACs and target nodes is stored in a look up table and it can be modified by the SDN controller. Moreover, the SDN controller decides which wavelength can be used for sending packets to which destination node. The communication between the FPGA interface and SDN agent is made via its PCIe interface. A two ports block memory module is employed for storing the commands from the SDN agent and the collected statistics information from the FPGA system. Both FPGA and SDN agent can read & write the block memory in Direct Memory Access (DMA) mode for the information exchange. The monitored traffic statistics are elaborated and provided to the SDN controller via OpenROADM SDN agent interfaces implemented by a PC motherboard attached to the FPGA. The I/O FIFOs inside FPGA for both RXs and TXs paths are set to 8192Bytes (1024 x 8Bytes). Each node includes also a server with a 4X10Gbps network interface card (NIC) serving as edge computing. A data center node with 16 servers is connected to the setup for emulating the realistic network operation from the collaboration between edge computing nodes and DC node. Fig. 3 also shows the spectrum of the four channels in the ring network. All the four channels that dropped and added at the four nodes are operating error free by adjusting the injection current of the SOAs of the ROADMs to guarantee the required OSNR and received power at the SFP+ receivers. The SOAs have <1 dB polarization dependent gain and compensate the

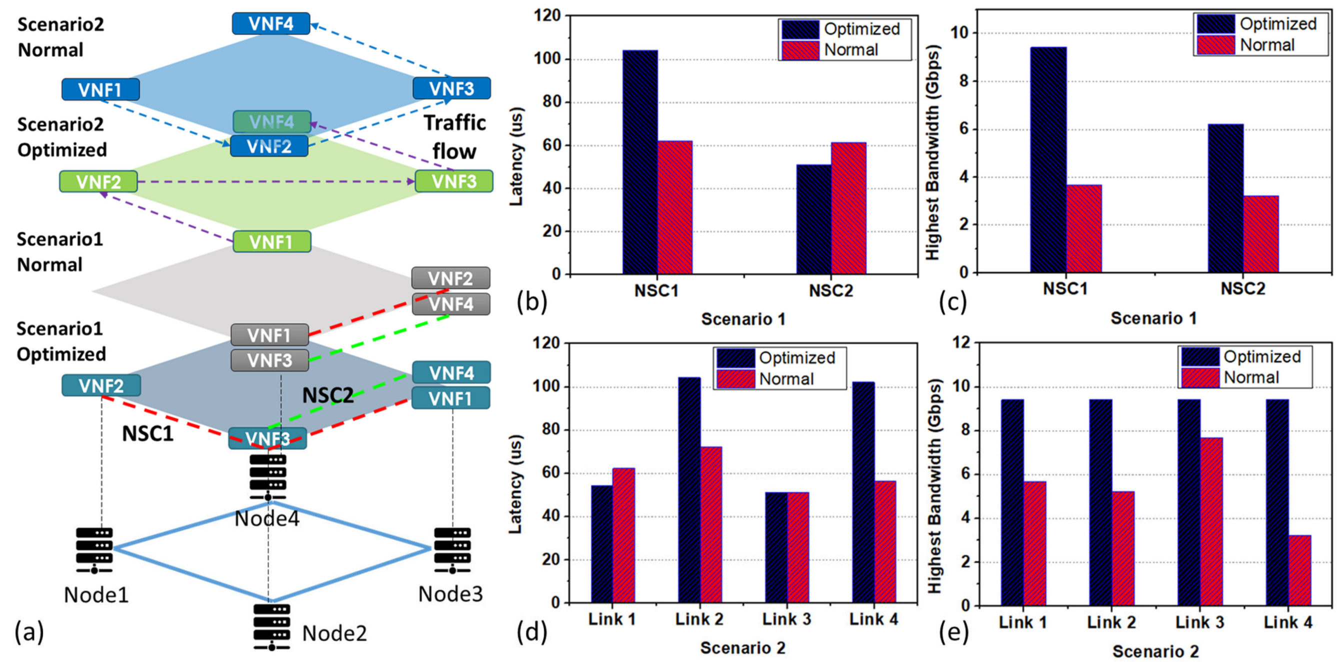

Fig. 5. (a) Optimized and normal NSC compositions of the investigated scenarios; (b), (c) Performance compare between optimized and normal NSC allocation for the scenario 1; (d), (e) Performance compare between optimized and normal NSC allocation for the scenario 2.

power loss from 10 Km link span and the arrayed waveguide gratings (AWG) in the ROADMs.

B. Investigation of the Telemetry Assisted, Orchestrated Cross Edge Computing Nodes Network Services Deployment

The performance investigation of the implemented network setup in terms of the packet loss ratio and latency under the Internet of Things (IoT) and urban macro traffics has been reported in our previous work [11]. This work investigates the overall network performance with the collaboration of optical data plane and telemetry assisted network orchestration tools. There are two objectives in this study. The first one is the optimized deployment of NSCs by VNFs allocated in different edge computing nodes. The second one is the optimization of the service quality of the NSCs (bandwidth and connectivity) by dynamically adjusting the data plane parameters such as traffic priority and path selection. In the experiment, a top-level network management program is controlling the network system including the SDN optical network controller and network service orchestrator by their North Bound Interface (NBI) API as shown in Fig. 2. Moreover, the SDN controller and network service orchestrator manage the optical network and VIM of the edge computing nodes by a 1GE management network. The telemetry information is updated every second as the finest granularity supported by the monitoring tool. For every VNF, the Enhanced Platform Awareness (EPA) characteristics are defined as the specific hardware requirements (CPU, RAM, Storage, External volume) to the VIM. The objective of on-demand deployment and composition of the NSCs is to deploy the VNFs while satisfying their EPA requirements, and chain the VNFs while fulfilling the QoS of the network connection between them. In the experimental scenarios, we use four VNFs (see Fig. 5(a)) to be deployed in the network and we assume that the VNFs in each NSC are not allowed to be deployed at the same edge computing site in order to investigate the co-operation of the nodes and network operation. The required network QoS are listed in Table I [33], the considered applications for this study are Tactile Internet for NSC 1 in scenario 1 and scenario 2, and Robotics for NSC 2 in scenario 1. We investigate and compare the performance of the network latency and the highest bandwidth of the VNF connections in two cases, with and without NSC optimization. In the normal case (without optimization), only the hardware requirements of the VNF allocation is considered. All the VNFs choose the first server (from edge 1 to edge 4) that fulfills the hardware requirements. The VNFs are emulated by the Ubuntu based VM, the hardware requirements of each VNF instance are set to 4 GB RAM, 2 virtual CPUs, and 10 GB storage. In the optimized case, the deployment and composition consider also the network QoS. The optimization strategy is to deploy the NSC with high bandwidth requirement to the edge computing nodes with the higher available bandwidth according to the telemetry. In addition, the traffic of VNF connections can be given high priority by the SDN controller to protect it from the other applications traffic that competes for the shared resources. Therefore, the NSC can approach the maximum transmission bandwidth of the TRX since it can be prioritized over other traffics competing for the available bandwidth. For the latency sensitive NSC, the network path with the lowest latency will

TABLE I
NETWORK QOS REQUIREMENTS OF THE TWO STUDY SCENARIOS [33]

Scenario	NSC	VNF	Bandwidth	Latency
1	1	1, 2	1Gbps per link	5-10ms
	2	3, 4	100Mbps per link	1ms
2	1	1, 2, 3, 4	1Gbps per link	5-10ms

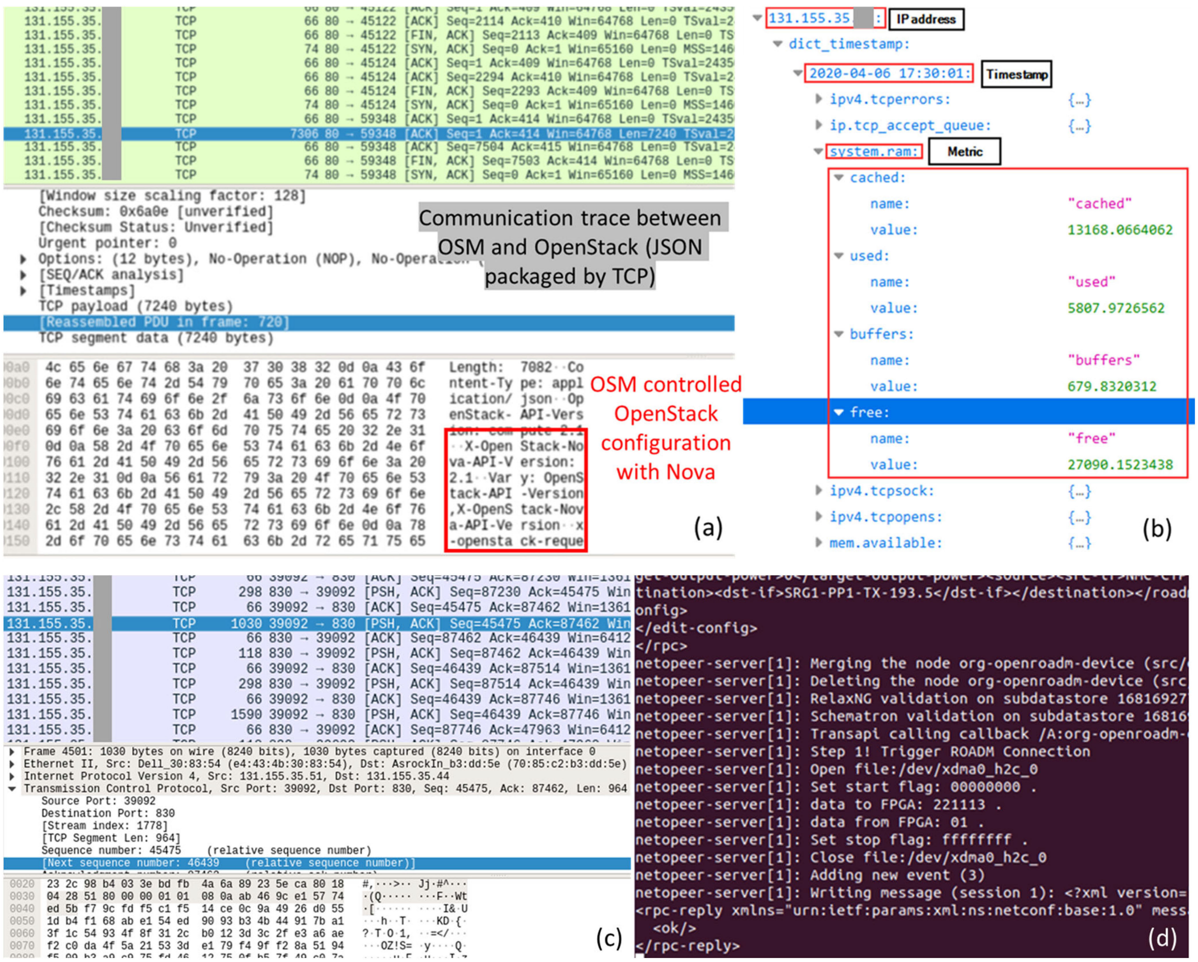

Fig. 6 (a) Captured communication trace of the OSM controlled OpenStack operation. (b) Monitored JSON data from the Netdata; (c) Communication trace between the ONOS and OpenROADM agent; (d) Response of the OpenROADM agent.

be allocated according to the telemetry information of the optical network. Fig. 5(a) shows the optimized and normal NSC compositions of each scenario. In scenario 1, the two NSCs are composed of VNF1 and VNF2 (NSC1), and VNF3 and VNF4 (NSC2), respectively. In scenario 2, all the four VNFs are composed of one chain. In scenario 1, NSC1 requires higher bandwidth and it is less sensitive on latency, so VNF1 and VNF2 are deployed in node3 and node1, respectively, where the highest bandwidth is available according to the information from the telemetry measurements. Fig. 6(a) shows the communication trace between the OSM and OpenStack based VIM. The trace shows that the Nova service of OpenStack is responding to the OSM for instantiating the VNFs at the node. The data exchanged between the network management program and Netdata server monitor is shown in Fig. 6(b), which is a JSON type data that contains the critical information of the hardware that is currently running. In addition, the status of the optical network is being monitored by the FPGA in each node, and the monitor information is real time delivered to the controller. Fig. 6(c) and (d) show the NETCONF communication trace between the ONOS and OpenROADM agents for the status updating, and the response of an OpenROADM agent. The NSC2 is more sensitive on latency and VNF3 and VNF4 are deployed in node2 and node3, respectively, since the network connection between the two nodes has the lowest latency. In the normal case, both of NSCs are deployed in node2 and node3, which are the first match of the hardware requirements of the VNFs. Fig. 5(b) and (c) show the performance difference between optimized and normal NSC allocation for scenario 1. The results show that in the optimized case the highest bandwidth of NSC1 can reach the maximum bandwidth of the TRX, while the latency of NSC2 is 50.9 us, which is the lowest latency achievable. However, in the normal case, the highest bandwidth of NSC1 is only about 40% of the maximum bandwidth of the TRX, and the latency of NSC2 is about 10 us higher than the optimized case. Those results prove that the telemetry assisted NSC deployment can optimize the QoS of the VNF connections according to the requirements. In scenario 2, the NSC1 is composed of all the four VNFs, and the bandwidth of the network connections between the VNFs should be optimized according to the requirements shown in Table I. In addition, the traffic flows sequentially from VNF1 to VNF4 are shown in Fig. 5(a). In the normal case, the VNF1 and VNF2 are deployed in node1 and node2. However, the wavelength from node1 to node2 is shared by node2 and node4, which results in a lower bandwidth to serve the applications. Based on the monitoring data, in the optimized case the VNF1 and VNF2 are automatically re-configured in node2 and node1 to improve the communication bandwidth. As a result, the optimized deployment outperforms the normal one since node2 now uses the bandwidth of a dedicated wavelength for sending traffic to node1. Fig. 5(d) and (e) show the performance difference between the optimized and normal NSC allocation for scenario 2. The results show that each connection in the NSC can reach the maximum bandwidth of the TRX in the optimized case although the network latency is even higher than the normal case. That is reasonable since the strategy is to find the best solution with the optimized bandwidth, and latency in this case is not the main concern.

IV. Conclusion

We have proposed and experimentally demonstrated a telemetry assisted SDN reconfigurable and NSC orchestrated metro-access network to address the challenges of the future network system with edge computing interconnection for 5G and beyond. The proposed optical metro access network is based on a fully programmable and flexible hardware platform for the data plane, and adapted open source network management and telemetry tools for the control and orchestration plane. An experiment testbed of a metro ring network with four complete edge computing nodes is built for validating the proposed network architecture and operation. Results show that the optical network is automatically and dynamically reconfigurable with the co-operation of the SDN controller, the FPGA data plane and IT resources telemetry. Different use cases have been investigated to validate the system operation. Successful telemetry assisted, NFV orchestrated cross edge computing nodes network services deployment and optimization have been demonstrated. More than 50% bandwidth improvement is achieved for the VNF connections with high bandwidth requirements.

References

[1] N. Calabretta, W. Miao, K. Mekonnen, and K. Prifti, “SOA based photonic integrated WDM cross-connects for optical metro-access networks,” *Appl. Sci.*, vol. 7, no. 9, 2017, Art. no. 865.

[2] M. Michal, A. Salah“5G PPP use cases and performance evaluation models.” 2016. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-use-cases-and-performance-evaluation-modeling_v1.0.pdf

[3] M. A. Lema *et al.*, “Business case and technology analysis for 5G low latency applications,” *IEEE Access*, vol. 5, pp. 5917–5935, Apr. 2017.

[4] K. Lee, J. Kim, Y. Park, H. Wang, and D. Hong, “Latency of cellular-based V2X: Perspectives on TTI-proportional latency and TTI-independent latency,” *IEEE Access*, vol. 5, pp. 15800–15809, 2017, doi:10.1109/ACCESS.2017.2731777.

[5] Q. Zhang, J. Liu, and G. Zhao, “Towards 5G enabled tactile robotic telesurgery,” Mar. 2018, *arXiv:1803.03586*.

[6] H. Li, K. Ota, and M. Dong, “Learning IoT in edge: Deep learning for the Internet of Things with edge computing,” *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan./Feb. 2018.

[7] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, “Edge computing for autonomous driving: Opportunities and challenges,” *Proc. IEEE*, vol. 107, no. 8, pp. 1697–1716, Jun. 2019.

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[9] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, “Mobile edge computing—A key technology towards 5G,” *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.

[10] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, “Inside the social network’s (datacenter) network,” in *Proc. 2015 ACM Conf. Special Int. Group Data Commun.*, pp. 123–137, 2015.

[11] B. Pan, E. Magalhaes, F. Wang, X. Xue, and N. Calabretta, “Experimental assessment of SDN controlled metro access network with network slicing and edge computing under 5G applications,” in *Proc. 24th IEEE, Optoelectron. Commun. Int. Conf. Photon. Switching Comput.*, 2019, pp. 1–3.

[12] N. Yamanaka, G. Yamamoto, S. Okamoto, T. Muranaka, and A. Fumagalli, “Autonomous driving vehicle controlling network using dynamic migrated edge computer function,” in *Proc. 21st IEEE Int. Conf. Transparent Opt. Netw.*, 2019, pp. 1–4.

[13] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, “Deepcog: Cognitive network management in sliced 5G networks with deep learning,” in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2019, pp. 280–288.

[14] C. Benzaid and T. Taleb, “AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions,” *IEEE Netw.*, vol. 34, no. 2, pp. 186–194, Mar./ Apr. 2020.

[15] A. Mayoral, R. Vilalta, R. Casellas, R. Martínez, and R. Muñoz, “Multi-tenant 5G network slicing architecture with dynamic deployment of virtualized tenant management and orchestration (MANO) instances,” in *Proc. 42nd Eur. Conf. Opt. Commun. VDE*, 2016, pp. 1–3.

[16] B. M. Andrus *et al.*, “Zero-touch provisioning of distributed video analytics in a software-defined metro-haul network with P4 processing,” in *Proc. IEEE, Opt. Fiber Commun. Conf. Exhib.*, 2019, pp. 1–3.

[17] D. Hancock and J. Van der Merwe, “Hyper$_4$: Using P$_4$ to virtualize the programmable data plane,” in *Proc. 12th Int. Conf. Emerg. Netw. Exp. Technol.*, 2016, pp. 35–49.

[18] R. Kundel *et al.*, “P4-BNG: Central office network functions on programmable packet pipelines,” in *Proc. 15th, Int. Conf. Netw. Service Manage.*, 2019, pp. 1–9.

[19] B. Turkovic, F. Kuipers, N. van Adrichem, and K. Langendoen, “Fast network congestion detection and avoidance using P$_4$,” in *Proc. Workshop Netw. Emerg. Appl. Technol.*, 2018, pp. 45–51.

[20] R. Ricart-Sanchez, P. Malagon, J. M. Alcaraz-Calero, and Q. Wang, “P4-NetFPGA-based network slicing solution for 5G MEC architectures,” in *Proc. ACM/IEEE Symp. Architectures Netw. Commun. Syst.*, 2019, pp. 1–2.

[21] W. Rankothge, F. Le, A. Russo, and J. Lobo, “Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms,” *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 343–356, Jun. 2017.

[22] Q. Zhang, Y. Xiao, F. Liu, J. C. Lui, J. Guo, and T. Wang, “Joint optimization of chain placement and request scheduling for network function virtualization,” in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 731–741.

[23] X. Chen, B. Li, R. Proietti, Z. Zhu, and S. B. Yoo, “Multi-agent deep reinforcement learning in cognitive inter-domain networking with multi-broker orchestration,” in *Proc. Opt. Fiber Commun. Conf. Opt. Soc. Amer.*, 2019, Art. no. M2A. 2.

[24] B. Li, W. Lu, and Z. Zhu, “Deep-NFVOrch: Leveraging deep reinforcement learning to achieve adaptive vNF service chaining in DCI-EONs,” *J. Opt. Commun. Netw.*, vol. 12, no. 1, pp. A18–A27, 2020.

[25] W. Miao *et al.*, “Low latency optical label switched add-drop node for multi-Tb/s data center interconnect metro networks,” in *Proc. 42nd Eur. Conf. Opt. Commun. VDE*, 2016, pp. 1–3.

[26] S. J. Park, C. H. Lee, K. T. Jeong, H. J. Park, J. G. Ahn, and K. H. Song, “Fiber-to-the-home services based on wavelength-division-multiplexing passive optical network,” *J. Lightw. Technol.*, vol. 22, no. 11, 2004, Art. no. 2582.

[27] ONOS “Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions,” [Online]. Available: https://www.opennetworking.org/projects/onos/

[28] Netopeer. [Online]. Available: https://github.com/CESNET/netopeer/

[29] OpenROADM. [Online]. Available: https://0201.nccdn.net/4_2/000/000/072/2aa/open-roadm-msa-release-7-device-white-paper-v1.2.pdf, 2020

[30] OpenStack. “OSM Architecture and Functions,” [Online]. Available: https://wiki.openstack.org/wiki/Main_Page

[31] OSM. [Online]. Available: https://osm.etsi.org/docs/user-guide/02-osm-architecture-and-functions.html

[32] Netdata. [Online]. Available: https://github.com/netdata/netdata

[33] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A survey on low latency towards 5G: RAN, core network and caching solutions,” *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 3098–3130, May 2018.