SDN Controlled Edge Computing Metro Access Network with Network Slicing and Load-aware end-to-end Service Protection for 5G applications

Bitao Pan, Xuwei Xue, Fulong Yan, Fu Wang, Eduardo Magalhães, Nicola Calabretta
Institute for Photonic Integration, Dept. of Elec. Eng, Eindhoven Univ. of Technology,
PO Box 513, 5600MB Eindhoven, The Netherlands
b.pan@tue.nl

Abstract: We demonstrate SDN reconfigurable edge-computing metro-access network based on low-cost ROADM nodes with edge-computing and programmable FPGA-based interfaces supporting classification and network slicing. Dynamic network operation and QoS protection is validated with live-streaming use case.

1. Introduction

The upcoming 5G systems will operate in a highly dynamic environment characterized by the existence of multiple types of applications (i.e. IoTs, Industry 4.0) with different quality of service (QoS) requirements [1]. In addition, edge computing will support new services that require fast data processing and low latency communication [2]. Therefore, novel optical metro access network nodes, as the infrastructures supporting the 5G traffic and the edge computing, need to be re-designed to provide flexibility and efficiency to dynamically adapt to the different applications' requirements. The new edge-computing nodes in combination with the software defined networking (SDN) control could provide a dynamically distribute computing and bandwidth adaptive all-optical connections according to the traffic requirements. Network slicing allows operators to split the physical network into multiple, virtual, end-to-end (E2E) networks that are logically isolated and dedicated for different types of slices with different characteristics and requirements [3]. For instance, the traffic of live video streaming should be given adaptive and stable bandwidth (tens of Mbps per connection) to keep the synchronization of video packets in each frame and further successfully construct the video. To provision and maintain the end-to-end QoS of critical applications, the network node itself should monitor in real time the traffic load to protect the critical traffic either locally by applying priority scheduling or centrally interacting with the SDN controller to coordinate the wavelength assignment and network slicing. However, an end-to-end traffic protection implementing a real time traffic monitoring and locally (low latency) and/or coordinated dynamic network slicing through interaction with the SDN control has not been experimentally demonstrated. In this paper, we present an experimental investigation of a load-aware SDN reconfigurable metro-access ring network with edge computing. Network slicing and load monitoring are implemented in the FPGA based traffic aggregation and classification network access interfaces and coordinated with the SDN control to implement an end-to-end traffic protection. The operation of the traffic protection is validated with a live streaming video use case.

2. System Operation of the Edge-computing Metro Access Network

The schematic of the metro access ring network is depicted in Fig. 1. Each node consists of a low-cost SOA based 2-degree ROADM with the functions of wavelength blocking (WBL) and amplification, an FPGA based programmable network interface for traffic classification, aggregation and load monitoring, and high-performance servers as edge computing. The metro access network employing low cost 2-degree add-drop wavelength blocker (WBL) switch based on SOAs, which provide power amplification, removing high cost EDFA. The SOA gates inside the ROADM can be turning on and off by the FPGA based O/E/O interface to make each single wavelength to pass or block, or drop and continue. Note that each of the incoming wavelength is dropped to the current node no matter it will be blocked by the node or not. Therefore, the node checks only the data that belong to the node based on the destination information, similar to the mechanism in passive optical network (PON) technology. In addition, each node can add traffic on a wavelength only if the wavelength is free or it was blocked by the node itself. The SDN controller is based on ONOS where specific drivers have been implemented to support the metro-access node through the NETCONF protocol. The SDN agent is PC based OpenROADM agent, which is derived from the OpenROADM YANG model, specifically enhanced and adapted to account for the hardware control. The agent extracts the command information from the NETCONF packets and further converts them to PCIE data flow. The FPGA will then map the PCIE data flow to logical high or low signals to drive the SOA-based WBL and the BVT,

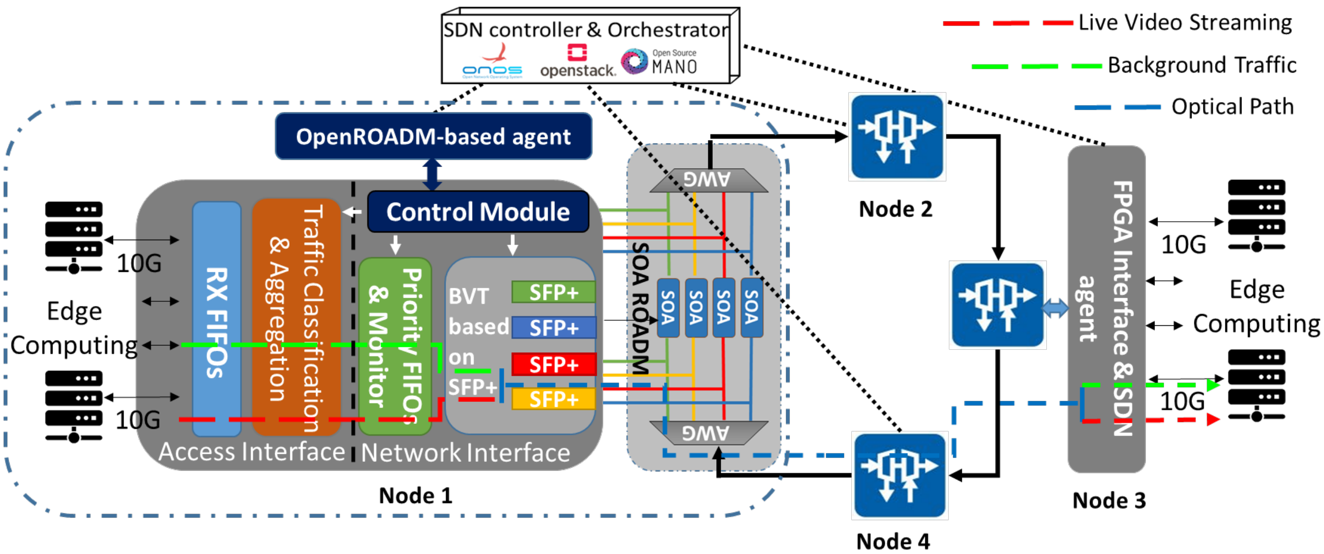

Fig. 1. SDN enabled metro access network with SOA based low cost ROADM and edge computing.

which is implemented here as multiple 10Gb/s transceivers (SFP+) directly plugged on the evaluation board of the FPGA. Besides, the FPGA interface is also working as load monitor and traffic classifier. The FIFOs of each TX (in the network interface) and RX (in the access interface) are real time monitored by the interface itself. When the RX FIFO of the high priority port is detected to be (almost) full, then the incoming traffic from this port will be aggregated in a high priority buffer and sent first when contention happens. By monitoring the RX FIFOs, the node itself detects and protects locally the QoS for critical traffic, avoiding long waiting time of the feedback from the centralized SDN controller. If the FIFO (almost) full has been detected on the TX side, then a trigger signal will be sent to the SDN controller, which could assign more link capacity by controlling the BVT of the node or re-destine the low priority traffic to other possible edge computing nodes. By monitoring the TX FIFOs, the SDN controller is aware of the traffic load of each node and then it can re-configure or slice the network in an optimized way.

3. Experiment Setup & Result Analysis

The experiment setup schematic is shown in fig. 1 and the deployed network elements in fig. 2(a). It is composed by a ring network including four metro access nodes. Each node is equipped with an SOA-based 2-degree ROADM, a Xilinx UltraScale FPGA with O/E/O interfaces, and a powerful server with a 4X10Gbps network interface card (NIC) as the edge computing. Additionally, a PC motherboard is used as the OpenROADM SDN agent. Each FPGA is equipped with four SFP+ transceivers (ITU Ch21, 23, 25, 27) to emulate the BVT by dynamically turning ON/OFF of each TRX. The I/O FIFOs inside FPGA for both RXs and TXs paths are set to 8192Bytes (1024 x 8Bytes). In the experiment, we use the edge server in node one as live video streaming server and the other edge servers in the other three nodes as the clients. At the server side, we use OBS-studio to generate live video streaming and Nginx to provide the Web network services. The output bandwidth of the video streaming is set to 10Mbps per single connection. The video streaming is considered as the high priority traffic in this work and it is flowing through one of the 10Gbps ports. Background traffic is generated by the other ports to load the network as the low priority traffic for demonstrating the network slicing mechanism at the FPGA interface. We use iperf TCP function for generating low priority traffic. The iperf operation is always trying to send as much as traffic to the destination for testing the maximum TCP bandwidth of the connection, so it is suitable for generating the additional traffic load and emulating the worst network condition for video streaming traffic. At the client side, we use VLC for requiring and playing the video from the metro access ring network and recording the statistic information of the video streaming. Netdata (an open source server monitoring application) is used for recording the traffic trace of the server NIC. In this experiment, node one has one wavelength available for each of the other nodes as the initial starting condition.

Fig. 2(b) to 2(I) show the received traffic trace from the server NICs in 5 minutes duration for three study scenarios, which are no traffic protection, local traffic protection and SDN controlled traffic slicing. Fig. 2(b) and (c) show the traffic trace in the case of no traffic protection. The low priority traffic is always transmitted and shares the 10Gbps

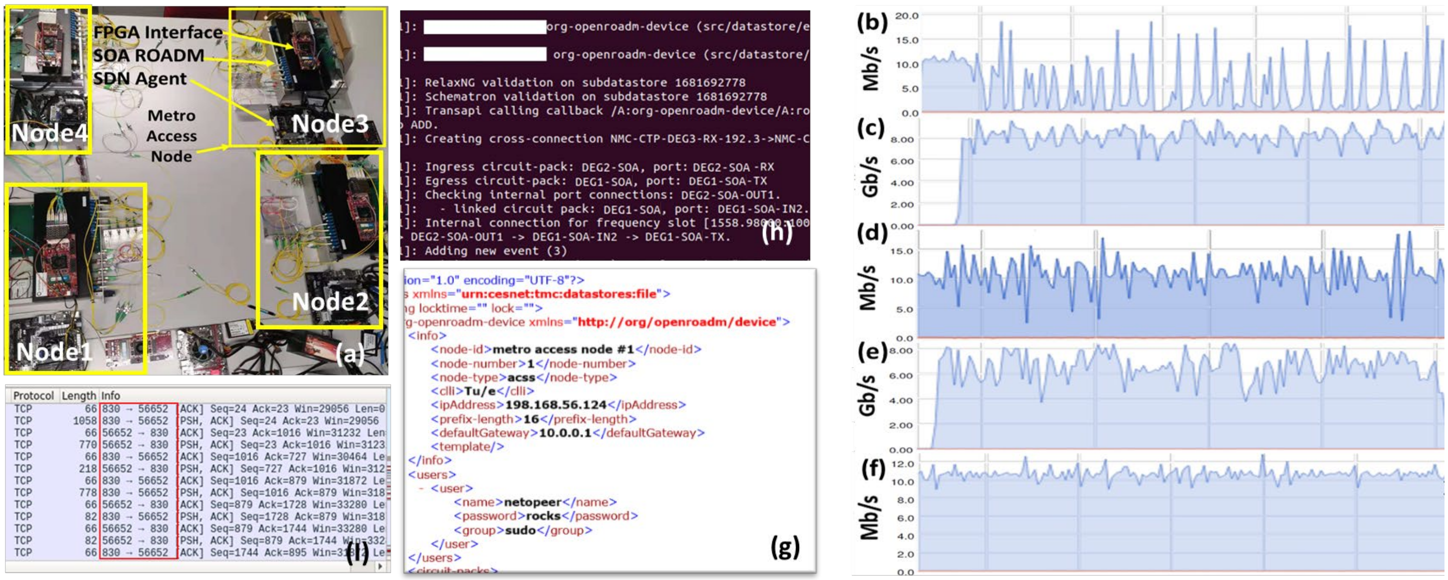

Fig. 2 (a) Experiment setup of the proposed metro access ring network; (b), (c) Traffic flow (video streaming (b) and low priority traffic (c)) with no protection; (d), (e) Traffic flow (video streaming (d) and low priority traffic (e)) with local traffic protection; (f) Traffic flow of video streaming in the case of SDN enabled traffic slicing; (g) OpenROADM YANG model of SOA 2-degree ROADM; (h) OpenROADM agent response to Netconf packets from SDN controller; (I) Netconf packet flow;

optical link with the video traffic. Therefore, there are a lot of collisions between the video streaming and the low priority traffic. It can be seen from Fig. 2(b) that the video streaming is not stable, and there are 132 frames lost in 5 minutes measurement according to the VLC statistics. In this case, the TCP bandwidth of the low priority traffic (see Fig. 2(c)) can reach to 8.05Gbps. Fig. 2(d) and (e) show the traffic trace in the case of local traffic protection. When the RX FIFO is almost full (2 buffer slot threshold), the incoming video traffic will be aggregated in the high priority buffer to be protected. Otherwise the classification will be disabled to avoid the starving of the other traffic. Therefore, the traffic trace of video streaming is relative stable (see Fig.2 (d)) as the protection happens only when the FIFO is almost full. No video frame loss is observed and the TCP bandwidth of the low priority traffic (shown in Fig .2(e)) can reach 6.39Gbps. In addition, the local protection prevents the need of the SDN feedback and thus additional delay. Fig. 2(f) shows the traffic trace when the SDN controls the traffic slicing. In this case, once the TX FIFO almost full has been detected, a trigger signal will be sent to SDN controller. The controller will then slice the traffic by routing the low priority traffic to other destination. The Netconf packets flow between SDN and OpenROADM agent is shown in fig. 2(I), and the response of the agent is shown in fig. 2(h). There is no collision between video traffic and low priority traffic in this case since they have different destinations. We can see from fig. 2(f) that the video streaming is stable at around 10Mbps receiver data rate. Zero frame loss has been measured.

4. CONCLUSIONS

We have experimentally investigated a load-aware and network slicing enabled metro access ring network with edge computing under a live video streaming use case. The results show successful SDN orchestrated network slicing functionalities and load monitoring based local traffic protection for QoS guaranteed traffic transport. Zero frame loss is achieved for the live video streaming under a worse network condition.

5. ACKNOWLEDGMENT

The authors thank the EU-H2020 project METRO-HAUL (G.A. 761727) and PASSION project (GA 780326) for partially supporting this work.

6. REFERENCES

[1] Calabretta N, Miao W et al. SOA based photonic integrated WDM cross-connects for optical metro-access networks [J].Applied Sciences, 2017, 7(9): 865.

[2] Shi, Weisong, et al. "Edge computing: Vision and challenges." IEEE Internet of Things Journal 3.5 (2016): 637-646.

[3] Pan, B., Calabretta, N, et al. Experimental Assessment or SDN Controlled Metro Access Network with Network Slicing and Edge Computing under 5G Applications. In 2019 24th OECC and 2019 PSC (pp. 1-3). IEEE.