Slice Management in SDN PON Supporting Low-Latency Services

Carlo Centofanti(1), Andrea Marotta(1), Dajana Cassioli(1), Fabio Graziosi(1), Nicola Sambo(2), Luca Valcarenghi(2), Chris Bernard(3), Hal Roberts(3)

(1) University of L'Aquila, 67100 L'Aquila, Italy carlo.centofanti1@graduate.univaq.it
(2) Scuola Superiore Sant'Anna, 56127 Pisa, Italy
(3) Calix, Inc, 2777 Orchard Parkway, San Jose, CA 95134

Abstract *We study possible slice management strategies in software defined passive optical networks for low latency services. Our results show that reactive slice deployment is able to enforce latency requirements requiring a minimal setup time while increasing network efficiency compared to proactive strategies. ©2022 The Author(s)*

Introduction

Software-Defined Networking (SDN) is a control architecture used to manage network operations in a centralized, elastic and efficient way[1]. In Passive Optical Network (PON), SDN can be used for network configuration and resource allocation based on requirements coming from Service Level Agreement (SLA). Moreover, network slicing has emerged as a paradigm to guarantee targeted performance to a variety of services sharing the same physical infrastructure[2].

Due to their high capillarity, PONs represent an interesting supporting infrastructure for a variety of services with heterogeneous requirements, e.g. multimedia[3], IoT, mobile networking, and critical services. Thus, the adoption of network slicing in PONs may be beneficial from both business and network efficiency viewpoint.

Among the protocols that have been proposed for the control of physical network infrastructures, NETCONF[4],[5] has found wide application due to its capability to aid automated configuration of heterogeneous network devices, e.g. optical network devices, packet switching nodes, and virtual and physical machines. Such wide range of applicability allows to enforce end-to-end slice performance[6],[7] and enables network automation.

It is worth considering that service requirements and traffic conditions may vary over time leading to inefficient usage of network resources or to violation of SLA. Thus, network automation mechanisms for slice management become crucial in PONs. While static configuration of network settings, such as optical access bandwidth allocation policies,[8]–[10] is a well known problem in the literature, there is a gap on studying how to automate slice management in SDN-enabled PONs and evaluating the impact of slice management when slice requirements change over time or are dynamically activated/deactivated. This is a common scenario in Low Latency Services (LLS) where ad-hoc slice resource reservation may be triggered by either the occurrence of a critical event (e.g., a seismic wave in earthquake early warning or other alert conditions) or by the initiation of a Extended/Mixed Reality (XR/MR) transmission requiring high throughput and low latency.

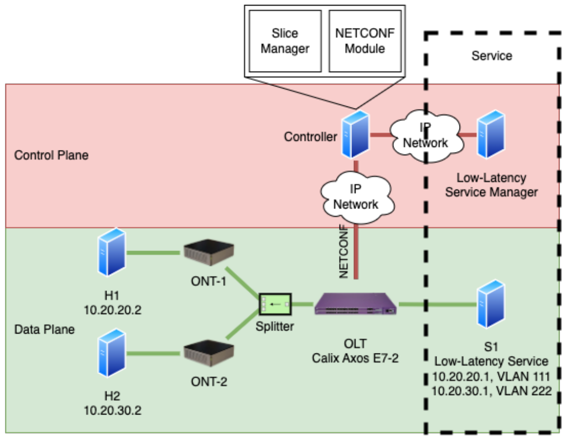

Fig. 1: System architecture

In this work we provide an overview of different slice management mechanisms and perform an experimental evaluation of the impact in terms of slice deployment time and user latency of deploying an on-demand slice in a real PON in automated manner via NETCONF.

System Description

We consider the system architecture shown in Fig. 1 where multiple users are connected to a data network through a Time Division Multiplexed PON (TDM-PON) at the data plane while the control plane is based on SDN and NETCONF. The

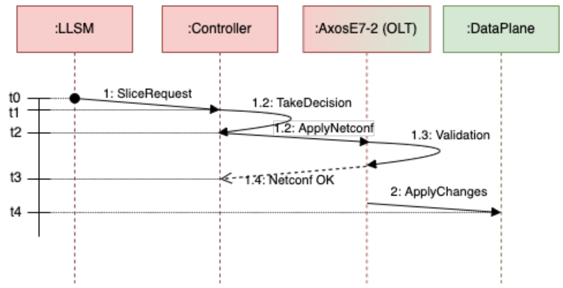

Fig. 2: Communications between different elements. On the left a timeline shows references for time measurements.

hosts at the ONT side may request multiple services with heterogeneous requirements. We introduce a Low Latency Service Manager (LLSM) as a component of a low latency service architecture which is in charge of requesting network slices to the SDN controller. After receiving a slice request, the controller validates the request and then sends the right configuration to the Optical Line Termination (OLT) through NETCONF, to deploy the slice. In PONs, different bandwidth allocation policies can be selected in order to meet service requirements, such as: (i) Expedited Forwarding which gives to a particular ONT the possibility to utilize a reserved amount of bandwidth in a Grant-free fashion in order to reduce experienced latency; (ii) Request-Grant based access which leverages Dynamic Bandwidth Assignment procedures to exploit statistical multiplexing and increases network efficiency at the price of a larger experienced latency. The steps needed for the slice deployment in the considered architecture can be summarized as:

1. The LLSM sends the slice request to the controller.
2. The controller validates the incoming message and builds the NETCONF message for the OLT.
3. The controller sends the NETCONF message to the OLT through a Remote Procedure Call (RPC) over a pre-established SSH connection.
4. The controller receives an acknowledgment message from the OLT, i.e. a NETCONF OK message.
5. The configuration is applied at the PON data plane.

We consider the OLT to be directly connected to a Server which represents an Edge node able to host service instances to reduce end-to-end latency experienced by service users.

As latency represents a critical aspect, we consider the following events to identify different time contributions to measure slice deployment delay and perform relative comparisons among slice management strategies (a graphical overview being given in Fig.2):

- t_0: The reference starting time, the time at which the slice request is issued, e.g. an alert condition is triggered.
- t_1: The time at the which the slice request message reaches the controller, influenced by network delay.
- t_2: The time at the which the controller has taken decision and sends a NETCONF message to the OLT.
- t_3: The time at which the controller receives a NETCONF OK message from the OLT.
- t_4: The time at the which new configuration is applied to the data-plane.

The overall Slice Deployment Time (T) is defined as $T = t_4 - t_0$.

Slicing Management Techniques

Different deployment strategies can be adopted to enforce slice requirements at the data plane. We consider the three following slicing management policies and show how they perform in real network conditions.

Request-Grant Committed Information Rate (RG-CIR) is a proactive strategy guaranteeing a given bandwidth to the users of the slice. The access to the bandwidth is based on a request-grant mechanism: the slice traffic is buffered and queued at the ONU until a transmission opportunity is granted, so increasing end-to-end latency.

Proactive Low-Latency Reservation (PLLR) uses EF to guarantee low latency and traffic prioritization at the OLT's scheduler. As a proactive strategy, resources are statically reserved before $t0$. This leads to an over-provisioning of resources but guarantees low-latency without waiting any slice deployment time.

Reactive Low-Latency Reservation (RLLR): when the LLSM requests low latency, the SDN controller dynamically changes the OLT configuration from normal request-grant to EF seamlessly. When the LLS wants to release resources, the controller reverts the configuration seamlessly, dismissing the slice and restoring request-grant conditions. This slice management policy can be easily adapted to a pay-as-you-go[11] model so reducing costs for the LLS provider.

Results

To perform experimental evaluation we adopted the architecture shown in Fig.1. The physi-

Tab. 1: Mean time measures to instantiate and destroy a RLLR slice. Values in μs.

	Slice request transmission	Controller Computation	NETCONF communication	SDT
	$t_1 - t_0$	$t_2 - t_1$	$t_3 - t_2$	T
Deployment	93	70	125834	1032081
Decommissioning	82	31	87997	1001638

cal infrastructure is composed by a Calix E7-2 XGS-PON OLT running AXOS platform, two Calix 801XGS ONTs offering connectivity between two client PCs (H1 and H2) and one edge server (S1). Then, we implemented a Python-based NETCONF controller. Fig. 4 shows an excerpt of a NETCONF message sent to the OLT to enforce EF resource reservation.

We implemented a simple service able to perform time measurements at application layer through a time-stamped-based mechanism and deployed an instance at the edge node S1. In order to have a common reference time between the application layer, the LLSM and the SDN controller we deployed all this element on the same physical machine. In the considered scenario H1 generates service data toward the edge instance of the low-latency service under consideration, while H2 is utilized to generate secondary service traffic concurring in optical resource usage in the PON.

First, we perform a comparison of the latency achievable through a request-grant approach (RG-CIR) and expedited forwarding slice reservation (PLLR and RLLR). Fig.3 shows average packet latency and standard deviation experienced by H1 (low latency traffic) and H2 (secondary service traffic) when a RG and EF slice resource reservation are implemented. Results show that while both approaches are able to enforce bandwidth requirements, only EF is able to offer latency level around 0.1 ms which represent more than 85% latency reduction compared to RG baseline.

RLLR and PLLR differ in the fact that in case

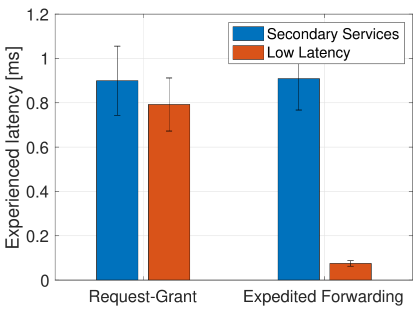

Fig. 3: Latency differences between RG on the left and EF on the right.

```
<profile>
  <pon-cos-profile>
    <name>ont1_ef</name>
    <cos-type>expedited</cos-type>
  </pon-cos-profile>
</profile>
```

Fig. 4: Excerpt of XML configuration to enable EF

of RLLR a time is needed to perform slice deployment and enforce slice requirements. Thus, we study the different contributions to the slice deployment time T. Table 1 defines and shows the different latency contributions. The first row in Table 1 is a measurement of time needed to enable RLLR while the second row is the fallback process from RLLR to no reservation for the slice. It is worth mentioning that the slice request transmission time and NETCONF communication time are strictly related to the physical deployment of the LLSM and the SDN-Controller. The considered co-location scenario allows to perform time synchronization among architectural elements. Results show that RLLR requires around 1s for slice deployment but compared to PLLR avoids to reserve optical resources for the low-latency slice when traffic is not transmitted, by reducing capacity over-provisioning.

Conclusions

In this paper we presented three different Slice Management strategies in SDN PON supporting Low-Latency Services. We implemented a NETCONF-based SDN controller which realizes slice management over a commercial PON infrastructure. We show that it is possible to use reactive strategies in cases where LLS does not request instantaneous low latency communications. The slice deployment time for reactive case may represent a very low time for most human-centric services but may not be not suitable for extreme safety scenarios and other critical applications.

Acknowledgements

This work was supported by the Italian PRIN 2017 (FIRST) and by the H2020-MSCA-RISE OPTIMIST project (GA: 872866).

References

[1] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks", *IEEE Communications Surveys Tutorials*, vol. 16, no. 3, pp. 1617–1634, Third 2014, ISSN: 1553-877X. DOI: 10.1109/SURV.2014.012214.00180.

[2] G. Ishigaki, S. Devic, R. Gour, and J. P. Jue, "Dynamic bandwidth allocation for pon slicing with performance-guaranteed online convex optimization", in *2021 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2021, pp. 1–6.

[3] C. Centofanti, A. Marotta, C. Rinaldi, F. Franchi, D. Cassioli, and F. Graziosi, "Improved dash video streaming performance by mec-enabled optical access", in *Asia Communications and Photonics Conference 2021*, Optica Publishing Group, 2021, T4A.144. DOI: 10.1364/ACPC.2021.T4A.144. [Online]. Available: http://opg.optica.org/abstract.cfm?URI=ACPC-2021-T4A.144.

[4] M. Bjorklund, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*, RFC 6020, Oct. 2010. DOI: 10.17487/RFC6020. [Online]. Available: https://www.rfc-editor.org/info/rfc6020.

[5] R. Enns, M. Bjorklund, A. Bierman, and J. Schonwalder, *Network Configuration Protocol (NETCONF)*, RFC 6241, Jun. 2011. DOI: 10.17487/RFC6241. [Online]. Available: https://www.rfc-editor.org/info/rfc6241.

[6] S. Das and M. Ruffini, "Optimal virtual pon slicing to support ultra-low latency mesh traffic pattern in mec-based cloud-ran", in *2021 International Conference on Optical Network Design and Modeling (ONDM)*, Jun. 2021, pp. 1–5. DOI: 10.23919/ONDM51796.2021.9492428.

[7] S. Das, F. Slyne, and M. Ruffini, "Optimal slicing of virtualised passive optical networks to support dense deployment of cloud-ran and multi-access edge computing", *IEEE Networks*, 2022.

[8] M. P. McGarry and M. Reisslein, "Investigation of the dba algorithm design space for epons", *Journal of Lightwave Technology*, vol. 30, no. 14, pp. 2271–2280, 2012.

[9] M. Zhu, J. Gu, and G. Li, "Pwc-pon: An energy-efficient low-latency dba scheme for time division multiplexed passive optical networks", *IEEE Access*, vol. 8, pp. 206 848–206 865, 2020.

[10] H. Uzawa, K. Honda, H. Nakamura, *et al.*, "Dynamic bandwidth allocation scheme for network-slicing-based tdm-pon toward the beyond-5g era", *Journal of Optical Communications and Networking*, vol. 12, no. 2, A135–A143, 2020.

[11] S. Ibrahim, B. He, and H. Jin, "Towards pay-as-you-consume cloud computing", in *2011 IEEE International Conference on Services Computing*, Jul. 2011, pp. 370–377. DOI: 10.1109/SCC.2011.38.