GRPC-BASED SDN CONTROL AND TELEMETRY FOR SOFT-FAILURE DETECTION OF SPECTRAL/SPACIAL SUPERCHANNELS

Ricard Vilalta¹, Noboru Yoshikane², Ramon Casellas¹, Ricardo Martínez¹, Shohei Beppu², Daiki Soma², Seiya Sumita², Takehiro Tsuritani², Itsuro Morita², Raul Muñoz¹

¹Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Spain
²KDDI Research, Inc., Saitama, Japan
*ricard.vilalta@cttc.es

Keywords: Control, orchestration, and management of optical networks; Integration of optical transmission and networks with higher layer network services

Abstract

This paper presents the first experimental demonstration of gRPC-based SDN control and telemetry architecture for spectral/spatial super-channels with SDM/WDM transceivers that monitor the BER to detect soft-failures over an 11-km 6-mode 19-core fiber. An evaluation of possible SDN protocols is provided for the discussed scenario.

1 Introduction

Spatial division multiplexing (SDM) and wavelength division multiplexing (WDM) combination allows to exploit the spatial and spectral dimensions of the fibers (i.e. wavelengths, modes and cores) together and to provide spectral-spatial super-channels (SChs). A spectral/spatial SCh is defined as an association of flexi-grid WDM channels, which are allocated together in different cores and/or modes in order to create a (logical) optical channel with the desired capacity [1].

Novel control and telemetry mechanisms for fully network programmability and real-time monitoring are needed in SDM/WDM networks. In order to evaluate these mechanisms, it is important to keep in mind that there are some metrics of success. On one hand, the industry adoption of these mechanisms will determine their success. On the other hand, multi-vendor interoperability will also be important.

During last years, several protocols have been identified as possible candidates for control and management of software defined networking (SDN), being the first OpenFlow. With the need of covering more network elements, NETCONF and YANG [2] have been demonstrated as valid solutions that bring more flexibility and innovation to optical software networks. With an attempt to simplify and adapt NETCONF to HTTP REST interfaces, RESTconf [3] has been introduced successfully and adopted in control and management of optical networks, being the selected protocol for several standards, such as ONF Transport API [4] and IETF TEAS ACTN [5].

gRPC Remote Procedure Calls (RPC) [6] is a cloud native high-performance RPC framework open sourced by Google. It uses HTTP/2 as a transport protocol and uses protocol buffers encodings for transported messages. gRPC has demonstrated its applicability in flexi-grid networks in [7], with focus on telemetry reception.

In this paper, the authors present the first experimental demonstration of an SDN-enabled sliceable spectral-spatial transceiver (S-SST) with SDN control and telemetry based on a protocol buffer data model and gRPC protocol. A proof of concept is developed in a joint testbed between CTTC and KDDI Research. We evaluate multiple plugins developed for the SDN controller and SDN agent, which allow to compare performance of the novel gRPC protocol, with implementations of the same scenario using NETCONF and RESTconf protocols.

2. SDN-enabled control and monitoring architecture for SDM/WDM transceivers

Fig. 1 SDN-enabled control and monitoring architecture

Fig. 1. shows the considered SDN-enabled control and monitoring architecture for the SDM/WDM transceivers presented by the authors in [8]. The SDM/WDM transceivers are based on a highly modular solution that is composed of arrays of sliceable spectral transceivers (S-STS), which provide multiple flexi-grid DWDM channels with bandwidth adaptability. The arrays of S-STS are connected to mode muxes/demuxes and multicore fiber fan-in/fan-out devices. Joint Tx and Rx DSP modules for all modes within a core are available (e.g., MIMO equalization), but not among cores, since it is assumed that inter-core crosstalk can be reduced to negligible values.
The SDN control and monitoring solution deploys an SDN controller that includes a monitoring manager, and SDN node agents with monitoring agents at the transmitter and receiver. In this paper we extend the SDN controller to support two new plugins for RESTCONF and gRPC, in addition to NETCONF that has been previously validated. This logically centralized architecture allows real-time data collection and its analysis. The monitoring agents are responsible of configuring the Rx DSPs modules by interfacing with proprietary protocols, and obtaining the monitoring data (i.e., BER from the Rx DSPs).

In the next subsection, the different used protocols between the SDN controller and SDN node agent are presented.

2.1 NETCONF protocol

NETCONF is a protocol based on the exchange of XML-encoded RPC messages over a secure (commonly Secure Shell, SSH) connection [2]. It offers primitives to view and manipulate data, providing a suitable encoding as defined by the data-model. As it uses YANG models, data is arranged into one or multiple configuration datastores and provides the set of rules by which multiple clients may access and modify a datastore within a NETCONF server. During last years, adoption of YANG as a data modelling language across standard defining organizations (SDO) and open source software (OSS) projects has been notable. IETF, ETSI, ONF, MEF, 3GPP, among others have adopted this data modelling language in order to provide standard data models.

2.2 RESTconf protocol

The need for a simpler and faster protocol based on HTTP, in order to properly modify and configure the YANG datastores, was detected. With this in mind, RESTconf was defined as a Representational State Transfer (REST)-like protocol that provides a HTTP-based API to access the data, modelled by YANG [3]. The REST-like operations are used to access the hierarchical data within a datastore.

2.3 gRPC protocol

This protocol is based on HTTP/2 transport protocol and uses byte-oriented encoding [6]. It has been presented to be used for high performance in telemetry streaming [7] in elastic optical networks. Protocol buffers are a flexible, efficient, automated mechanism for serializing structured data in a byte-oriented protocol, allowing faster and reduced transmission.

gRPC has been open sourced by Google, and it provides open source tools to validate protocol buffers data models. Moreover, services can be described using protocol buffers. Python bindings are easily generated, and a server stub allows ease of integration with the SDN agents.

3 Experimental setup and results

3.1 Experimental setup

The experimental setup consists of an SDN controller that has been deployed at CTTC premises in Castelldefels (Spain), and an SDN-enabled SDM/WDM transceiver deployed at KDDI Research in Saitama (Japan). These have been connected using OpenVPN connections through internet [1].

The hardware setup is the same as described in [8]. For SDN control, YANG data models are the same as provided in [8]. Protocol buffers have been generated from the YANG data models using [9]. The resulting protocol buffers can be obtained in [10].

3.2 Provisioning results

In order to measure the setup delay for the provisioning of the requested slice, which consisted on 6 modes (LP01, LP11a, LP11b, LP21a, LP21b, LP02) and 16 channels (50GHz space, 193.20-193.95THz in C band), three different plugins have been developed in SDN controller based on the provided NETCONF, RESTconf and gRPC clients [9]. Moreover, three daemon servers have been located on top of SDN agents.

Using Wireshark in the SDN controller, the setup delay for each protocol has been measured 10 times and the average is provided (see Fig.2.a). It can be observed that the contribution to protocol setup delay (without considering latency between CTTC and KDDI) for NETCONF protocol is of 1219ms. This is due to the necessity of establishment of an SSH session. As RESTconf is evaluated over TLS, a shorter setup delay is measured of 8ms. Finally, gRPC provides a setup delay of 7.6ms.

![Fig. 2 (a) Provisioning protocol setup delay (ms), (b) Provisioning protocol transmitted bytes](image-url)
Fig. 2.b shows the measured amount of protocol bytes necessary for the setup of the detailed slice. It can be observed that NETCONF, which is using XML, requires almost the double of necessary bytes than RESTconf (143KB and 87KB). gRPC required 5KB, due to its optimized byte encoding.

3.3 Telemetry results

The received electrical signals are digitized at 80 GSa/s using three synchronized real-time oscilloscopes (OSC). Offline processing is performed over the stored samples and bit errors are counted. The obtained BER is notified from the SDN agent towards the SDN controller using the gRPC protocol and protocol buffers [9]. In the case of NETCONF and RESTconf, the SDN controller reads the BER from the SDN agent, which utilize the previously presented data model [8].

Fig. 3.a shows the protocol delay for recovering the BER measurements. It can be observed that NETCONF adds 443ms, while RESTconf needs 6.9ms. gRPC only needs 2ms for using the notification stream between SDN agent and SDN controller.

![BER delay (ms)](#)

Another significant feature that is needed in multiple scenarios is the clear distinction between startup, candidate and running datastores. NETCONF requires the existence of the running configuration datastore on a device but no others. Candidate datastore includes the requested configuration, before it is executed and brought in running datastore. Meanwhile, startup datastore includes the configuration pre-loaded by the device.

Finally, in multi-tenant scenarios, which are typical in operational deployments, a mechanism for locking the access to the datastore is necessary. Only NETCONF implements such mechanism among the explored protocols.

4 Conclusion

We have presented the first implementation of a gRPC-based control and telemetry architecture for spectral/spatial superchannels with SDM/WDM transceivers. The presented results demonstrate the enormous advantages of using gRPC for telemetry recovery, but also show that although gRPC might provide better delay and byte usage, it might not be ready for operational scenarios where other significant features such as capability exchange, multiple datastore management and locking are needed. The authors suggest to explore the extension of gRPC with these mechanisms.

Acknowledgements

Work supported by EC H2020 BLUESPACE (762055) and Spanish AURORAS (RTI2018-099178) projects, and "Massive Parallel and Sliced Optical Network", NICT, JAPAN.
References

[1] R. Muñoz, et al., SDN control and monitoring system for soft-failure detection and optical restoration of spectral/spatial superchannels, European Conference on Optical Communication (ECOC), 2018.

[2] R. Enns, Ed., Network Configuration Protocol (NETCONF), IETF RFC 6241, 2011.

[3] A. Bierman, M. Bjorklund, K. Watsen, RESTCONF protocol, IETF RFC 8040, 2017.

[4] ONF Transport API Software Development Kit release 2.0.3 (SDK),
https://github.com/OpenNetworkingFoundation/TAPI

[5] Y. Lee, S. Belotti, D. Dhody, D. Ceccarelli, and B. Yoon, Information Model for Abstraction and Control of TE Networks (ACTN) (No. RFC 8454), 2018.

[6] gRPC. https://grpc.io/Accessed: 2019-02-14.

[7] F. Paolucci, A. Sgambelluri, M. Dallaglio, F. Cugini, and P. Castoldi, Demonstration of gRPC telemetry for soft failure detection in elastic optical networks, in Proc. of 2017 European Conference on Optical Communication (ECOC) (pp. 1-3). IEEE, 2017.

[8] R. Muñoz, N. Yoshikane, R. Vilalta, et al., SDN Control of Sliceable Multi-dimensional (Spectral and Spatial) Transceivers with YANG/NETCONF, Journal of Optical Communications and Networking (JOCN), Vol. 11, No. 2, pp. A123-A133, February 2019.

[9] Yang2proto,
https://github.com/opencord/voltha/blob/master/experiments/netconf/yang2proto/yang2proto.py , Accessed: 2019-02-14.

[10] https://github.com/CTTC-ONS/SDM, Accessed: 2019-04-14.