Open Source Implementation of OpenConfig Telemetry-Enabled NETCONF Agent

A. Sgambelluri[1*], A. Giorgetti[1], F. Paolucci[1], P. Castoldi[1], and F. Cugini[2]

[1] Scuola Superiore Sant'Anna, Pisa, Italy

[2] CNIT, Pisa, Italy

**e-mail: a.sgambelluri@santannapisa.it*

ABSTRACT

Next generation networks are evolving towards disaggregated schemes, where the structure of the optical nodes presents multiple components to be combined and built into a complete solution (i.e., white-box concept). In such a scenario, operators and service providers are supporting vendor neutrality to enhance devices inter-operability, simplify the control-plane operation and reduce network costs. OpenConfig is a working group of network operators focused on the definition of a set of vendor-neutral data models allowing dynamicity and programmability of the infrastructure by means of software-defined networking paradigm. Moreover, the telemetry functionality is a novel solution for network monitoring in which real time data and statistics are efficiently streamed from devices.

In this paper, we propose an open-source implementation of an OpenConfig NETCONF agent enhanced with telemetry. The adoption of the gRPC-based telemetry allows the streaming of monitoring parameters from the transponders. In fact, the controller is able to request the streaming of one or more selected parameter(s), on demand and with proper granularity, in order to detect and localize the presence of a fault or identify hardware degradation. The implemented telemetry-enabled NETCONF agent for transponders has been experimentally validated over an Elastic Optical Networks (EON) scenario, highlighting the main functionalities and the effectiveness of the proposed solution.

Keywords: telemetry, optical network, disaggregation, white-box.

1. INTRODUCTION

Software Defined Network (SDN) paradigm drastically changed the way to manage and control telecommunication networks. The SDN Controller is an entity in charge of control all the network devices in the network. In the case of EONs, the adoption of NETCONF protocol for the communication among controller and devices has enabled new configuration and monitoring functionalities.

Each device vendor, other than providing their own SDN controller, is extending "standard" configuration and monitoring features providing advanced and proprietary functionalities.

In order to overcome proprietary and vendor-specific transmission solutions, the optical network disaggregation is emerging as an important network scenario [1-4]. In fact, the white box concept is envisaging multi-vendor scenarios where different vendors provide the building blocks required for the realization of an optical node, including the optical transport infrastructure (e.g., ROADMs, amplifiers and links) and optical termination (e.g., transponders/muxponders). In such a heterogeneous scenario, the support of vendor neutrality is a key requirement, in order to overcome compatibility issues and enhance the device control/management in a vendor agnostic way. However, even if transponders of different vendors can now coexist in the same shelf, it is a good practice to couple the adopted transponders within the same vendor, in order to optimize the performance of the transponders, rely on proprietary advanced solutions and avoid compatibility problems.

OpenConfig [7] is a working group of network operators, lead by Google, devoted to the definition of vendor-neutral data models to be adopted in order to simplify and standardize the control and management of the optical infrastructure by mean of SDN scheme. More specifically, with reference to optical networks and in particular to the transponders, a set of YANG models have been introduced to provide a common way to configure, through NETCONF, the main transponder parameters [5, 6]. More specifically, the root of the tree (<terminal-device>) represents an xPonders node in a modular structure composed by logical channels associated to physical ports. By exploiting the exposed <config> parameters (i.e., target-output-power, central-frequency and operational-mode) the SDN Controller is able to enforce a specific configuration to the underlying network devices, without relying on proprietary functionalities. Moreover, <state> fields are adopted in the data-model in order to store both the key OTN parameters to be monitored in order to check the status of an active connection (i.e., pre-FEC-BER, post-FEC-BER, Q factor, ESNR) and the physical parameters at hardware level (i.e., output-power, input-power, chromatic dispersion).

In this work, we provide the detailed description of an open-source implementation of an SDN agent that includes the OpenConfig NETCONF agent [8] enhanced with a gRPC-based telemetry service [9]. The three main functional blocks of the SDN agent are highlighted and the overall functionalities have been experimentally validated in a testbed scenario with two xPonders nodes. After the configuration of a connection among a couple of commercial 100Gbps cards and periodically updating the selected monitoring parameters, the telemetry

978-1-7281-2779-8/19/$31.00 ©2019 IEEE

service is adopted in order to identify the degradation of the connection performance in case of soft failure on the traversed link.

2. PROPOSED DESIGN AND IMPLEMENTATION

In this section the proposed design of the open-source SDN agent is described [10]. In Fig. 1 the considered optical node architecture is shown. In the bottom part of the figure, the data plane optical node is represented, where xPonders (i.e., several transponders and/or muxponders), possibly of different vendors, are installed. On the top of the figure the control plane software architecture of the SDN agent is shown. Three are the main functional blocks:

1. the NETCONF agent,
2. the device driver.
3. the telemetry server.

The NETCONF agent has been implemented using ConfD software [11], adopting the OpenConfig YANG modules. It exposes a North Bound Interface (NBI) to the SDN controller based on NETCONF protocol, allowing the configuration and the monitoring of the main key transmission parameters of the underlying transponders, exploiting the edit-config rpc and the get-config rpc. Those parameters are maintained in a database (DB) organized according to the OpenConfig YANG modules. The NETCONF agent has been equipped with two ad-hoc defined sockets (i.e., config socket and monitoring socket) in order enable the communication with the device driver module. The syntax adopted over the sockets is custom and text-based.

The device driver, developed in python, has two main functions: enforcing the configuration received on the config socket to the physical transponder(s) and periodically collecting from the physical device and pushing to the NETCONF agent through the monitoring socket the required key parameters. The communication with physical device is performed adopting proprietary APIs provided for the control of the devices (e.g., the typical proprietary APIs are not NEYCONF based).

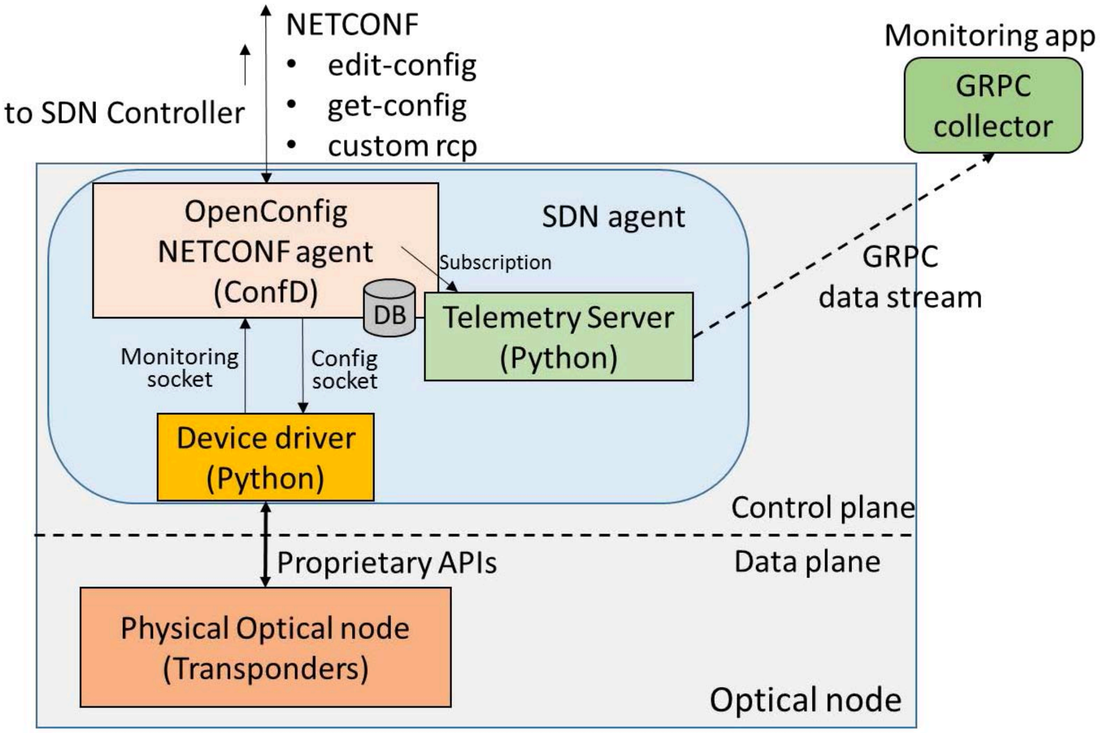

Figure 1. NETCONF agent architecture.

The telemetry server is a novel module devoted to handle the telemetry requests and enable the stream of telemetry information. In our implementation the telemetry has been implemented using the open source python implementation of the gRPC with protobuf protocol. According to the OpenConfig schema, the telemetry subscriptions are received via NBI by the NETCONF agent. The NETCONF agent submits the proper gRPC subscription to the Telemetry server, including in the request the remote gRPC collector (i.e., IP address and port), the list of parameters to be monitored (more parameters can be included in the same data stream), the period and the duration of the stream. Then, the telemetry server enables the telemetry stream to the collector, retrieving the data directly from the NETCONF database.

3. EXPERIMENTAL VALIDATION

In order to validate our solution, we considered the network scenario shown in Fig. 2 where two xPonders nodes were equipped with the SDN agent described in the previous section. Each node presented three muxponders: one commercial 100Gb/s with coherent reception, two commercial 10Gbps with direct detection reception. On the link between the 2 ROADMs we connected a Variable Optical Attenuator (VOA), in order to perform the degradation of the connection. We focused our validation on the xPonders nodes, managed through OpenConfig NETCONF agents, while the ROADMs devices were managed using a different NETCONF agents.

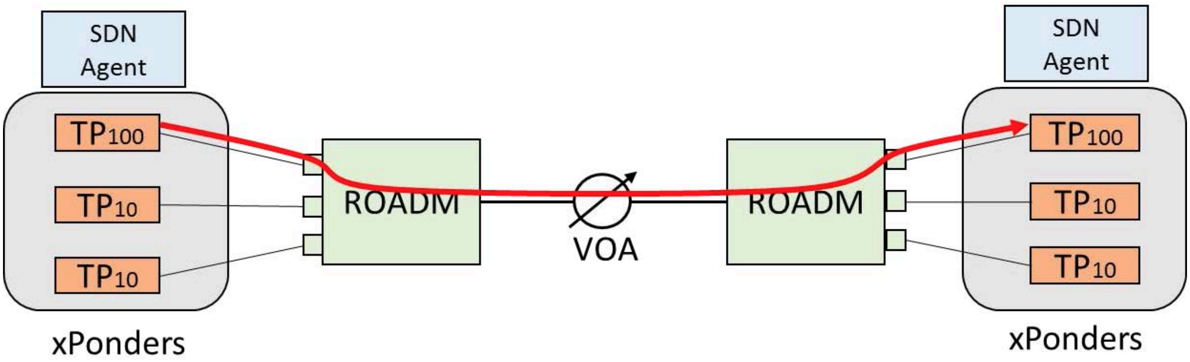

Figure 2. Experimental setup.

Initially, we configured a connection among the pair of 100 Gbps transponders. Figure 3a shows the message sent by the SDN controller to configure the two transponders. More specifically, the transponder with index 10102 (i.e., the TP_{100}) was configured with central frequency 192.5THz (expressed in MHz), output power 0dBm and the related optical-channel enabled.

b)

```
<telemetry-system xmlns="http://openconfig.net/yang/telemetry">
  <subscriptions>
    <dynamic-subscriptions>
     <dynamic-subscription>
       <id>1</id>
       <state>
          <id>1</id>
          <destination-address>10.30.2.24</destination-address>
          <destination-port>50083</destination-port>
          <sample-interval>60000</sample-interval>
          <heartbeat-interval>3000</heartbeat-interval>
          <suppress-redundant>false</suppress-redundant>
          <protocol xmlns:t="http://openconfig.net/yang/
                     telemetry-types">t:STREAM_GRPC</protocol>
          <encoding xmlns:t="http://openconfig.net/yang/
                     telemetry-types">t:ENC_PROTO3</encoding>
       </state>
       <sensor-paths>
         <sensor-path>
           <path>/terminal-device/logical-channels/channel
                [10102]/otn/state/pre-fec-ber/instant</path>
           <state>
            <path>/terminal-device/logical-channels/channel
                  [10102]/otn/state/pre-fec-ber/instant</path>
            </state>
         </sensor-path>
         <sensor-path>
           <path>/terminal-device/logical-channels/channel
                       [10102]/otn/state/esnr/instant</path>
            <state>
              <path>/terminal-device/logical-channels/channel
                         [10102]/otn/state/esnr/instant</path>
            </state>
         </sensor-path>
       </sensor-paths>
     </dynamic-subscription>
    </dynamic-subscriptions>
  </subscriptions>
</telemetry-system>
```

a)

```
<components xmlns="http://openconfig.net/yang/
    platform">
  <component>
    <name>channel-10102</name>
    <optical-channel xmlns="http://openconfig.net/yang/
        terminal-device">
      <config>
        <frequency>192500000</frequency>
        <target-output-power>0</target-output-power>
      </config>
    </optical-channel>
  </component>
</components>
<terminal-device xmlns="http://openconfig.net/yang/terminal-
   device">
  <logical-channels>
    <channel>
       <index>10102</index>
       <config>
          <admin-state>ENABLED</admin-state>
       </config>
    </channel>
  </logical-channels>
</terminal-device>
```

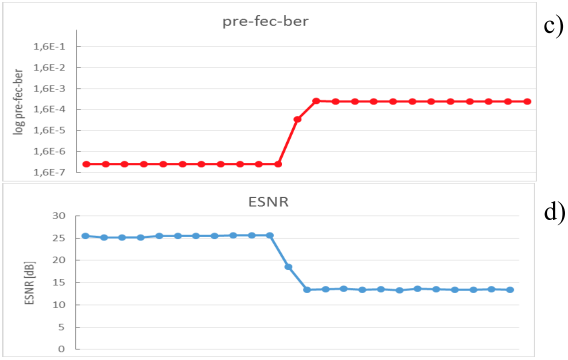

Figure 3. SDN Controller messages and results.

After the setup of the connection, the SDN Controller performed the telemetry subscription at the receiver transponder by sending the XML message shown in Fig. 3b. The subscription request includes: IP address and port of the remote collector that will receive the data stream (i.e., 10.30.2.24 and 50083, respectively), the duration (i.e., sample-interval) and the period of the samples (i.e., heartbeat-interval) expressed in milliseconds (60 and 3 seconds, respectively), the option to suppress samples equal to the previous one (disabled), protocol and encoding type (gRPC and PROTO3, respectively) and the list of the required parameters (in the example instantaneous ESNR and instantaneous pre-FEC BER of transponder 10102, expressed according to the OpenConfig hierarchy).

During the experiment, we induced the degradation of the link between the two ROADMs by increasing the attenuation at the VOA. This operation resulted in a soft failure of the connection. The telemetry stream, that included the instantaneous values of ESNR and pre-FEC BER, reported updated data to the remote collector. In Fig. 3c and Fig. 3d the trends of both pre-FEC-BER, in logarithmic scale, and the ESNR of the connection are reported, respectively. In normal condition (i.e., without the VOA attenuation) the connection presented a pre-FEC BER around 3.7 E^{-7} and an ESNR of 25.3 dB. Once the soft failure occurred, the performance of the connection degraded, resulting to a pre-FEC BER around 4.3 E^{-4} and an ESNR of 13.5 dB. The telemetry functionality allowed the monitoring system to detect the degradation of the performance of the connection (i.e., soft failure) and perform the required operations in order to recover from the failure or eventually reroute the connection before the connection is break.

4. CONCLUSIONS

This paper presents the description of the open-source implementation of an OpenConfig NETCONF Agent for xPonders nodes. Initially, the design of the proposed NETCONF OpenConfig agent is described, with details on the NETCONF module, the device driver and the novel telemetry server. The overall functionalities of the proposed NETCONF OpenConfig agent have been experimentally validated, showing the details of the configuration performed by the SDN Controller in order to enable a connection and activate the gRPC telemetry. Moreover, we shown how the telemetry information, enabled through gRPC module, can be useful in order deeply monitor the performance of the connections in real time and eventually identify possible soft failures.

ACKNOWLEDGEMENTS

This work was partially supported by the EC through the METRO-HAUL project (G.A. nº 761727).

REFERENCES

[1] L. Velasco, A. Sgambelluri, R. Casellas, L. Gifre, J. L. Izquierdo-Zaragoza, F. Fresi, F. Paolucci, R. Martínez, and E. Riccardi, “Building autonomic optical whitebox-based networks,” *J. Lightwave Technol.*, vol. 36, pp. 3097-3104, 2018.

[2] V. Lopez, O. G. de Dios, and J. P. Fernandez-Palacios, “Whitebox flavors in carrier networks,” in *Proc. Optical Fiber Communication Conference (OFC)*, 2019, paper M1A.5.

[3] A. Sgambelluri, J. L. Izquierdo-Zaragoza, A. Giorgetti, L. Gifre, L. Velasco, F. Paolucci, N. Sambo, F. Fresi, P. Castoldi, A. C. Piat, R. Morro, E. Riccardi, A. D’Errico, and F. Cugini, “Fully disaggregated ROADM white box with NETCONF/YANG control, telemetry, and machine learning-based monitoring,” in *Proc. Optical Fiber Communication Conference (OFC)*, 2018, paper Tu3D.12.

[4] E. Riccardi, P. Gunning, Ó. G. de Dios, M. Quagliotti, V. López and A. Lord, “An operator view on the introduction of white boxes into optical networks,” *J. Lightwave Technol.*, vol. 36, no. 15, pp. 3062-3072, 1 Aug. 2018.

[5] R. Muñoz *et al.*, “SDN control of sliceable multidimensional (spectral and spatial) transceivers with YANG/NETCONF,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 2, pp. A123-A133, Feb. 2019.

[6] M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, “Control and management of transponders with NETCONF and YANG,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 3, pp. B43-B52, 2017.

[7] Open Config, http://www.openconfig.net

[8] F. Paolucci, A. Sgambelluri, R. Emmerich, A. Giorgetti, P. Castoldi, C. Schubert, J. K. Fischer, and F. Cugini, “OpenConfig control of 100G/400G filterless metro networks with configurable modulation format and FEC,” in *Proc. Optical Fiber Communication Conference (OFC)*, 2019, paper Tu3H.4.

[9] F. Paolucci, A. Sgambelluri, F. Cugini, and P. Castoldi, “Network telemetry streaming services in SDN-based disaggregated optical networks,” *J. Lightwave Technol.*, vol. 36, pp. 3142-3149, 2018.

[10] NETCONF Agent https://github.com/asgamb/OpenConfig

[11] Tailf ConfD https://www.tail-f.com/confd-basic/