# Operational Mode and Slicing Adaptation in OpenConfig Disaggregated Optical Networks

**Davide Scano[1], Alessio Giorgetti[1], Silvia Fichera[1], Andrea Sgambelluri[1], Filippo Cugini[2]**
*1: Scuola Superiore Sant'Anna, Pisa, Italy*
*2: CNIT, Pisa, Italy*
*alessio.giorgetti@santannapisa.it*

**Abstract:** This paper proposes and experimentally validates a workflow to handle network failures implying the change of the operational mode on optical OpenConfig transponders. An SDN control plane is considered with a real packet-optical data plane.
**OCIS codes:** 060.0060 Fiber optics and optical communications, 060.4250 Networks.

## 1. Introduction

Partially disaggregated optical networks are gaining consensus since they avoid vendor lock-in of transponder solutions [1]. In partial disaggregation, the optical transport infrastructure (e.g., optical line systems – OLS – including ROADMs and amplified links) is provided by a single vendor, while transponders can be provided by multiple/different vendors. However, since transponders are provisioned in pairs, each optical connection is typically managed by a single vendor. This way, the most advanced, even proprietary, transmission solutions can be adopted, so not compromising the transmission performance. In partial disaggregation, the OpenConfig YANG provides a standard way to configure, through NETCONF, the common transponder parameters [2-4]. Specifically, in OpenConfig, transponders are described by the terminal-device YANG model, relying on the concept of logical channels to define the mapping among client ports and line ports. At the line side, the model specifies OTN config and state parameters (e.g., pre/post Forward Error Correction - FEC). The OpenConfig terminal device model is augmented with a platform component defining config parameters such as (i) frequency, (ii) target output power, and (iii) operational modes (OP modes). No other transmission parameters are defined (e.g., modulation format, FEC type, constellation shaping, etc). Indeed, all these parameters have to be defined within the OP modes, which are vendor-specific. This way the model remains stable despite of technological evolutions and capable of including proprietary advanced transmission solutions maximizing throughput performance. As a drawback, this approach requires definition and implementation of specific workflows to manage the OP modes that go beyond the model itself.

In [5], a telemetry-driven procedure was presented enabling the SDN Controller to enforce testing and validation of all supported OP modes for a new optical connection to be setup, so to identify the most efficient OP to be actually configured. So far, both the adaptation of OP modes at line side and the related implication and procedures at the client side have not been discussed yet. The latter aspect is particularly relevant if QoS-guaranteed network slicing is applied end-to-end across the whole network [6,7], in particular, for recovery upon failures.

In this work, we design and implement a novel comprehensive workflow providing OpenConfig OP mode adaptation at the transponder line side together with the required actions at the client side on the interconnected packet-based network adopting QoS-guaranteed slicing.

## 2. Problem description and proposed workflow for OpenConfig operational mode and slicing adaptation

Fig.1 shows the reference network scenario encompassing a sliceable two-layer (i.e., packet over optical) metro infrastructure based on SDN control. The packet-based layer is implemented using devices controlled with the OpenFlow protocol, while the optical layer is a disaggregated optical transport network, whose devices are controlled using the NETCONF protocol. QoS-guaranteed connectivity is provided end-to-end across the whole network employing a number of network slices with differentiated Service Level Agreements (SLAs).

The metro network portion shown in Fig.1 includes four packet-based nodes (N1, N2, N3 and N4) connected to the optical infrastructure as client tributaries. In this work, we assume transponders of type muxponder, i.e. equipped with N client ports and one optical line port enabling transmission across the disaggregated optical network, a pair of muxponders is included in Fig. 1 (i.e., T1 and T2). Each muxponder is equipped with eight client ports (i.e., P1, P2, P8), and one optical line port (i.e., OL1). The OLS includes three ROADMs (R1, R2, and R3). Three network slices are established on the network, where each slice is configured over a number of packet-based switches, and all three slices are sharing the optical line port OL1 of the two muxponders (see Fig.1). Specifically, slice S1 is served by nodes N1 and N3 using port P1 of muxponder T1 and T2. Slice S1 carries high class bandwidth guaranteed services, which have to be completely preserved also in case of network problems (e.g., failures). Slice S2 traverses packet nodes N1, N3 and N4 using port P1of muxponder T1 and T2, but it carries low class bandwidth guaranteed services, i.e. its SLA

allows 50% bandwidth reduction to be temporarily applied in case of network failures. Slice S3 is served by packet nodes N2 and N4 respectively connected to port P2 of muxponder T1 and T2. It also accepts 50% bandwidth reduction.

At the line side, the muxponders flexibly support multiple combinations of transmission parameters such as modulation formats, bit rates, and FEC types, which are defined as OP modes in the OpenConfig model. For example, they support as OP mode 1 ($OP_1$) a transmission at bitrate $R_1$=200 Gb/s with modulation format $MF_1$ corresponding to Polarization Multiplexed 16 Quadrature Amplitude Modulation (PM-16QAM) and FEC type $F_1$. With $OP_1$, all the N client ports of the muxponder are fully exploited (e.g., N=8 at 25Gb/s each), i.e. both port P1 and P2, with reference to Fig.1. The line side also supports as OP mode 2 ($OP_2$) the combination, at the same baud rate of $OP_1$, of bitrate $R_2$=100 Gb/s, with modulation format $MF_2$ as PM Quadrature Phase Shift Keying (PM-QPSK) and FEC type $F_2$. In $OP_2$, only N/2 muxponder client ports are exploited (i.e., $P_1$ and not $P_2$ with reference to Fig.1). Indeed, in muxponders, the tributary flows are rigidly multiplexed within the OTN line framing without performing packet switching (as in switch-ponders, not largely deployed yet). Under working conditions, $OP_1$ is configured along an optical connectivity using a path traversing the ROADMs R1 and R3. Connectivity is thus provided at 200 Gb/s by fully exploiting both client ports P1 and P2. When a failure occurs at link R1-R3, an alternative optical path is used, e.g., traversing ROADM R2. However, this path suffers from additional impairments which make 200 Gb/s $OP_1$ transmission unpractical. Thus 100 Gb/s $OP_2$ transmission needs to be enforced at the line side of the muxponder.

However, so far, the adaptation from different OP modes has never been considered and experimented. Moreover, such adaptation at the line side of the muxponder imposes the tear down client port P2 which implies the disconnection of slice S3 traffic arriving from node N2, thus violating the SLA of S3. Thus, a comprehensive procedure is needed accounting for OP mode adaptation at the line side as well as metro reconfiguration at the client side.

Fig.2 shows the proposed workflow implemented within the ONOS SDN controller to handle network failures requiring a change of the OP mode on the OpenConfig transponders. Specifically, we implemented two separate applications on the ONOS NBI and we have adapted the drivers on the ONOS SBI; all the interactions depicted with red-solid lines in Fig.2 have been developed for this work, while the interactions depicted with blue-dashed lines are provided by the ONOS core. The *Operational Modes* app uses the NETCONF protocol to monitor the state of the optical devices using specific NETCONF subscriptions compliant with the OpenROADM model. This way, for instance, when a ROADM detects a loss of light event (LoL), it sends a NETCONF notification to the controller. The *Slicing QoS* app exposes a set of CLI commands to create/manage/delete network slices interacting directly with Path Computation, Flow Rule and the Meter services provided by the ONOS core; moreover it can be invoked by the Operational Modes app to dynamically handle existing slices in case of network events affecting the bandwidth availability on the existing optical connectivity.

We consider a starting situation where S1, S2 and S3 are sharing a single optical connection (i.e., OpticalConnectivity intent) between the optical line ports OL1 of the two transponders. Thus, upon failure between R1 and R2, the Operational Modes app receives a NETCONF notification advertising the LoL event (step 1 in Fig.2). The vision of network topology at the controller is consequently updated through the Topology service (step 2 in Fig.2) removing the failed link; this action triggers the ONOS intent framework that reroutes the disrupted OpticalConnectivity intent along the path R1, R2, R3 (steps 2a, 2b, 2c in Fig.2).

However, the intent framework is completely unaware of needed reconfiguration of the muxponders OP modes and on the slices SLAs to be enforced. Therefore, the app evaluates if the new optical path does require an OP modification

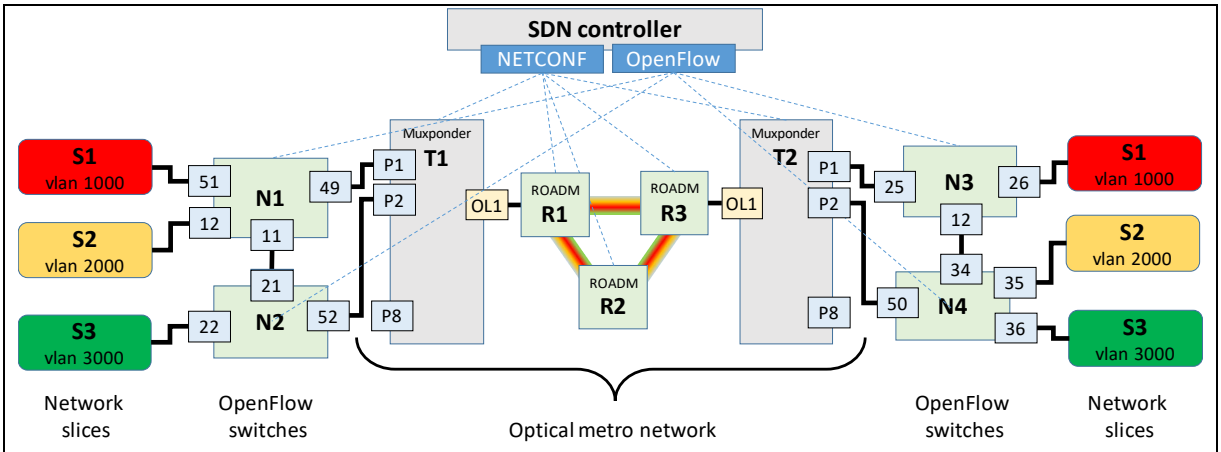


*Fig.1: Reference multi-layer packet over disaggregated optical network: scenario and experimental setup.*

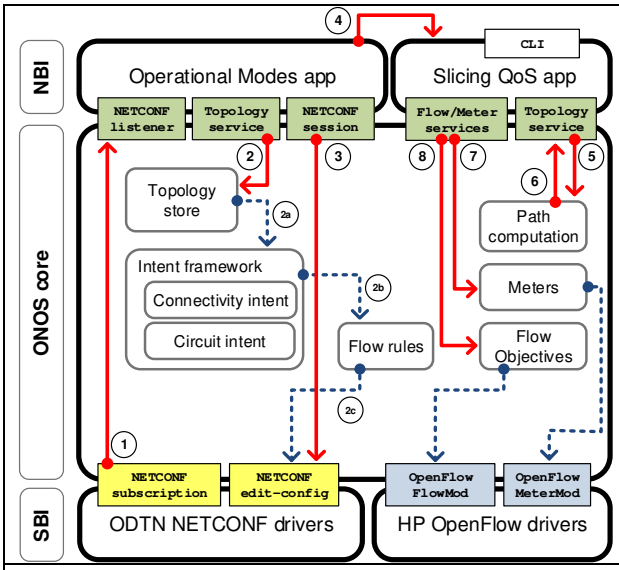


*Fig.2: Proposed ONOS workflow; the two applications and all the interactions depicted with red-solid lines have been implemented for this work.*

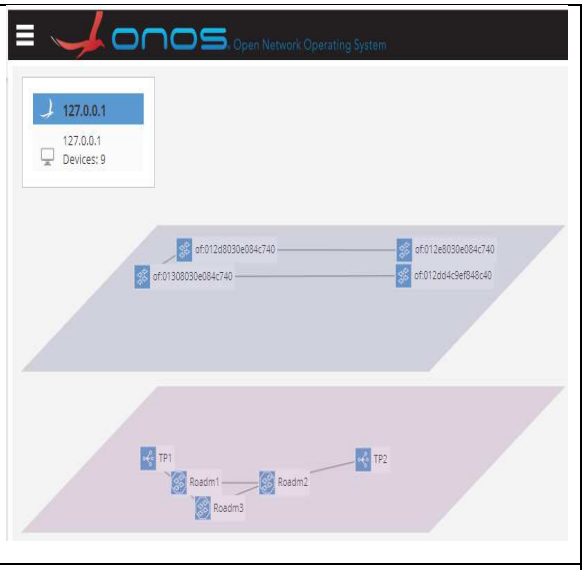


*Fig.3: ONOS screenshot showing the connected devices. The single optical connectivity is mapped in two separate logical links on packet-layer topology.*

and eventually uses a NETCONF edit-config message to modify the OP mode of the muxponders and turn-off the client ports (i.e., P2) that cannot be used with OP2 (step 3 in Fig.2). Then the Slicing QoS app is invoked providing as an input the bandwidth available using OP2 (step 4 in Fig.2). This way, it can re-route S3 that were using port P2 (steps 5 and 6 in Fig.2), and reduce the bitrate associated to each slice utilizing OpenFlow meters, consistently with the enforced SLAs (steps 7 and 8 in Fig.2), thus reducing S2 and S3 birate by 50% while keeping S1 to original bitrate.

## 3. Implementation and experimental demonstration

The data plane testbed includes both real and emulated devices. On the packet-based layer, four commercial HP switches are used. Optical layer devices use a NETCONF agent based on ConfD tool [9]. Specifically, the testbed includes two real OpenConfig compliant transponders (i.e., Ericsson SPO with line cards at 10Gbps and 100 Gbps) and three emulated OpenROADM compliant ROADMs. The failure event has been emulated ten times on the experimental testbed, the sequence of events at the controller has been logged to compute the time required to perform the complete workflow depicted in Fig.2. The performed change of operational mode modifies the FEC type used at the transponders. Obtained results show that the control plane workflow is executed in less than 6 seconds, that is a fully satisfactory result because the change of OP mode implies a physical disconnection of about 40-50 seconds.

## 4. Conclusions

This paper proposed and experimentally validated a workflow to handle failures in SDN multi-layer networks where it is possible to change of the operational mode of optical transponders. Obtained results showed that all the required control plane operations can be executed in few seconds, before the convergence of the transmission layer.

**5. Acknowledgements:** This work has been partially supported by the EU Commission under the project H2020 METRO-HAUL (Grant: 761727).

## 6. References

[1] V. Lopez et al "Whitebox Flavors in Carrier Networks", OFC 2019
[2] B. Andrus et al, “Evaluation and experimental demonstration of SDN-enabled flexi-grid optical domain controller based on NETCONF/YANG, NOMS 2018
[3] M Dallaglio et al, “Control and management of transponders with NETCONF and YANG”, JOCN 2017
[4] F. Paolucci et al. "Network Telemetry Streaming Services in SDN-Based Disaggregated Optical Networks," JLT 2018
[5] R Martínez et al, “Network slicing resource allocation and monitoring over multiple clouds and networks”, OFC 2018
[6] M. Raza et al, “Dynamic Slicing Approach for Multi-Tenant 5G Transport Networks”, JOCN 2018
[7] A. Sgambelluri et al., “Telementry-driven validation of operational modes in Openconfig disaggregated networks”, ECOC 2019
[8] A. Giorgetti et. al., “Control of Open and Disaggregated Transport Networks using Open Network Operating System (ONOS)”, accepted for publication in JOCN 2019.
[9] A. Sgambelluri et. al, “Open source implementation of openconfig telemetry-enabled netconf agent”, ICTON 2019