Hierarchical Control of SONiC-based Packet-Optical Nodes encompassing Coherent Pluggable Modules

Davide Scano(1), Alessio Giorgetti(2), Andrea Sgambelluri (1), Emilio Riccardi (3), Roberto Morro (3), Francesco Paolucci(4), Piero Castoldi (1), Filippo Cugini (4)

(1) Scuola Superiore Sant'Anna, Pisa, Italy, davide.scano@santannapisa.it
(2) CNR IEIIT, Pisa, Italy, alessio.giorgetti@ieiit.cnr.it
(3) TIM, Turin, Italy
(4) CNIT, Pisa, Italy

Abstract: a *hierarchical structure of SDN controllers is demonstrated to enable the configuration of hybrid packet-optical nodes. The proposed workflow leverages on extension of the ONOS controller and T-API interface to coordinate the configuration steps required by the packet and the optical controllers.*

Introduction

The availability of cost-effective coherent pluggable transceivers is driving the removal of transponders as standalone network elements in transport optical networks. Indeed, the utilization of pluggable modules equipped within packet switching devices guarantees several benefits[1,2]. This way, packet nodes are directly connected to the optical transport network (e.g., to reconfigurable optical add drop multiplexers – ROADMs) thus guaranteeing significant benefits in terms of CAPEX, power consumption and space occupation in central offices. Examples of such pluggable modules include Digital Coherent Optics (DCO) transceivers with CFP2 or QSFP-DD form factors which are already commercially available with rates up to 400 Gbps and configurable transmission parameters.

A further benefit of packet nodes equipped with coherent pluggables is the tight integration between packet-based aggregation networks and optical transport networks, guaranteeing effective traffic engineering solutions while simplifying the network management. A typical use case exploits a single packet switch for both intra-datacentre traffic aggregation and, relying on coherent pluggables, inter-datacentre communication.

However, the control of integrated packet-optical nodes requires the evolution of the currently available operating systems for packet nodes (e.g., Software for Open Networking in the Cloud - SONiC) to also support configuration, state information retrieving, and management of coherent pluggable modules. Indeed, following the traditional control plane architectures, packet-related configurations should be enforced by a an SDN Controller dedicated to the packet domain (i.e., PckC) while optical parameters need to be configured by another SDN controller dedicated to the optical domain (i.e., OptC). So far, the problem of coordinated control of packet-optical nodes by two SDN controllers has been addressed only in[3], relying on the Network Configuration Access Control Model (NCACM) solution detailed in RFC 8341. In particular, in[3], the packet-optical node participates in the coordination process by applying restricted rights to access the pluggables. Specifically, OptC is provided with writing rights on the optical parameters and read-only rights on packet parameters. Vice versa, the PckC is provided with writing rights on packet parameters and read-only rights on optical parameters. Using this approach, no inter-Controller communication is needed, however the packet-optical node takes an active role in the configuration workflow and has to interact with two controllers. Such solution may introduce significant maintenance problems especially in case of firmware and software updates at the node and at the controller level.

In this demo, we show an alternative approach based on inter-Controller communication. With this solution, packet-optical nodes only interact with the PckC. In turn, the PckC is enabled to configure optical parameters by proper interaction with the OptC. The demo leverages on several innovative components: i) Intent-based connectivity between pairs of ROADMs ports managed by ONOS SDN Controller; ii) Extension of the T-API connectivity service request to include the pluggables relevant parameters; iii) Utilization of a double communication channel from the PckC to the packet-optical devices.

Coordinated control of packet-optical nodes

Fig. 1 shows the proposed workflow to guarantee coordinated control of packet nodes, hybrid packet-optical nodes using pluggables, and optical nodes. The figure illustrates both the network initialization procedure (steps A-C), and the procedure used to activate a multi-layer connectivity service (steps 1-8).

During network initialization the packet and the

978-0-7381-4679-9/21/$31.00 ©2021 IEEE

optical topologies are pushed into the respective controllers (step A); the hierarchical SDN controller (HrC) loads the topology of the two domains (including the pluggables modules discovered by the PckC) through the controllers REST APIs (step B); finally, the associations between the pluggables modules used in the packet-optical nodes and the ROADM add/drop interfaces are pushed into HrC (step C). All this data is classified as quasi-static information since determined by manual intervention and can be therefore initialized through specific configuration (i.e., POST commands on the REST APIs).
After network initialization, when a layer 2/3 connectivity request arrives at HrC (step 1), it first identifies the pair of pluggable modules to be interconnected through the optical transport network. Note that in OpenROADM such Tx/Rx port pairs are typically part of an add/drop group supported by the same group of circuit packs, and therefore called Shared Risk Group (SRG). At step 2, HrC sends a RESTConf connectivity request (e.g., Open Transport API, T-API) between SRG connection points to the OptC. To effectively perform impairment-aware optical path computation, the OptC must be aware of pluggable supported features (e.g., supported modulation formats, FECs, operational modes). To this goal, two cases can be considered. In the first case, the type and features of the pluggable modules are included in the set of static information related to inter-layer connectivity, that are manually loaded at the OptC. In the second case, the type and features of the pluggable modules are discovered by the PckC. In both case the PckC forwards the pluggable details to the HrC, that, in turn, forwards them as part of the T-API based connectivity request to the OptC[4]. The latter case will be adopted in the demo, it enables automatic discovery of pluggables but requires additional parameters to be exchanged among controllers for each request, even if the information is quasi static.
At step 3 the OptC performs impairment-aware path computation, identifying the suitable configuration for pluggable modules as well as traversed optical path. This step, typically, is not executed inside the SDN controller, but exploits external tools specifically developed with this target, e.g., GNPy[5]. At step 4 the controller enforces the SRG-to-SRG configuration through NETCONF, driving the set-up of all traversed ROADMs. At step 5, once the path is successfully established, the OptC replies to the HrC informing about the available SRG-to-SRG connectivity as well as on the selected configuration of pluggable modules. Indeed, they cannot be directly configured since under the domain of control of the PckC. At step 6, the HrC generates a packet level REST connectivity request to the PckC. The request includes the configuration previously identified by the OptC for the pluggable modules at the line side. At step 7, the PckC enforces the configuration to both involved packet-optical nodes, and other involved packet nodes. At step 8 the PckC informs the HrC about the successful configuration.

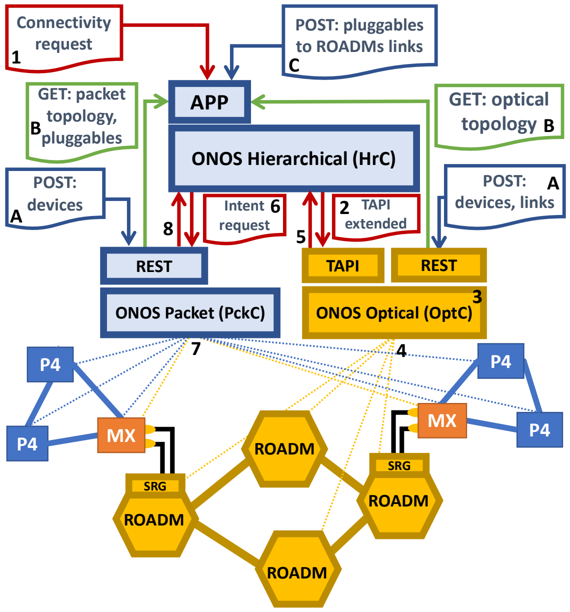

Fig. 1: Control plane architecture and workflows. Letters A-B describe the network initialization workflow; numbers 1-9 describe the connectivity establishment workflow.

Demo setup

This demo shows the above-described workflow implemented in a multi-layer network testbed. As illustrated in Fig. 1, the testbed includes three ONOS-based SDN Controllers, two ROADMs, two packet-optical nodes supporting pluggable transceivers, and some P4-based emulated devices (e.g., based on BMv2 software switch).
The packet-optical node architecture is better detailed in Fig. 2. It consists of a Mellanox/NVIDIA SN2010 Ethernet switch running SONiC operating system over ONIE. Besides the basic components (i.e., sonic-cfggen, syncd, swss, pmon and the redis database) SONiC also includes the P4/P4 Runtime docker container and the NETCONF docker container. Using these two containers two parallel communication channels are established between the packet-optical device and the PckC to enable configuration of packet and optical resources, respectively. Specifically, the P4/P4 runtime docker container is an experimental P4 implementation developed and provided by Mellanox/NVIDIA, allowing the PckC to configure the packet resources using P4 Runtime. The NETCONF docker container has been

specifically designed within this work for enabling the acquisition and the configuration of optical resources (i.e., pluggables) by the PckC. Within the NETCONF agent optical pluggables are described using an OpenConfig model[7]. This container collects node status information by directly accessing the SONiC Redis database and enforces the configurations using a custom-built REST APIs toward the swss docker container. Additionally, the REST client of the NETCONF container can be used to configure external devices acting as pluggable modules. This latter case will be adopted in the demo.
More in detail, in the Mellanox/NVIDIA switch, the two ports connected on the packet layer (i.e., port 1, 2) are equipped with 10Gb/s SFP+ transceivers, monitored by SONiC pmon container; on the other side two ports (i.e., port 3,4) are attached to an external 100 Gb/s coherent system (i.e., provided by Ericsson) configured as being a coherent pluggable module, i.e., its driver resorts to the REST client running inside the NETCONF docker container. This way, there is no direct connection between the coherent system and the SDN controller as it would be for a standalone transponder. Thus, port 1-2 act as tributary (/client) interface while port 3-4 provide the 100Gb/s coherent communication across the optical transport network.
When the PckC applies the connectivity configuration (step 7 in Fig. 1) two different configuration messages are received by each packet-optical one. The first is processed by the NETCONF agent that results in the activation/configuration of the Ericsson SPO muxponder; the second is processed by the P4/P4 Runtime agent that configure the P4 pipeline to enable a bidirectional connectivity between port 1 and port 3.
The ROADMs are controlled with OpenROADM model[6]. The OptC is implemented exploiting the ONOS SDN Controller equipped with the Open Disaggregated Transport Network (ODTN) modules that have been extended to support the configuration of SRG-to-SRG intent in the optical domain[7]. Moreover, the T-API interface has been extended to include the pluggable optical description that may have an impact on the physical impairment aware path computation performed by the optical controller[4].
During the demonstration, the testbed and the controllers will be physically located at the CNIT laboratories in Pisa. Remote access to the controller interfaces will be configured from the conference demo zone enabling the real-time demonstration of both phases, i.e., network initialization and connectivity enforcement. Also, the exchange of messages among controllers will be logged in real-time and described during the demonstration.

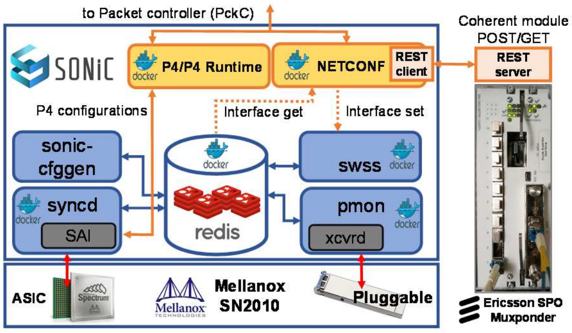

Fig. 2: Hybrid-node architecture including P4-based and NETCONF agents, both connected to the Packet controller. Dashed interactions are implemented but not included in the demonstration.

Conclusions

Coordinated control of SONiC-based packet-optical nodes including pluggable transceivers will be demonstrated by exploiting inter-SDN controller communication between Packet and Optical controllers through the utilization of a hierarchical controller. Coherent pluggable parameters are computed by the optical controller while access to pluggable resources is provided to the packet controller only.

Acknowledgements

This work has been partially supported by TIM under the UniversiTIM project and by the BRAINE Project, funded by ECSEL Joint Undertaking under grant agreement No. 876967.

References

[1] E. Riccardi, et al., “An Operator view on the Introduction of White Boxes into Optical Networks”, JLT 2018.

[2] V. Lopez, et al., “Enabling fully programmable transponder white boxes [Invited]”, JOCN 2020.

[3] A. Sgambelluri, et al, “Coordinating Pluggable Transceiver Control in SONiC-based Disaggregated Packet-Optical Networks”, OFC 2021.

[4] C. Manso et al., "TAPI-enabled SDN control for partially disaggregated multi-domain (OLS) and multi-layer (WDM over SDM) optical networks," JOCN 2020.

[5] A. Ferrari et al.,”GNPy: an open source application for physical layer aware open optical networks”, JOCN 2020.

[6] A. Sgambelluri, et al., “OpenConfig and OpenROADM Automation of Operational Modes in Disaggregated Optical Networks”, IEEE Access 2020.

[7] A. Giorgetti, et al., “Control of open and disaggregated transport networks using the Open Network Operating System (ONOS)”, JOCN 2020.