TELECOM INFRA PROJECT

OOPT-CANDI Whitepaper

Optical Proof of Concept 2021

Date July 16, 2021

Document version: v1.0

Authors

Anders Lindgren

Network Architect, Telia Company

anders.x.lindgren@teliacompany.com

Ramon Casellas

Senior Researcher, CTTC/CERCA

ramon.casellas@cttc.es

Esther Le Rouzic

Research Engineer, Orange

esther.lerouzic@orange.com

Emmanuelle Delfour

Software expert, Orange

Emmanuelle.delfour@orange.com

Arturo Mayoral López de Lerma

Technical Program Manager, Facebook

adelerma@fb.com

Raghavan Subramanian

Product Manager, Fujitsu Network Communications

Raghavan.Subramanian@fujitsu.com

Roshan Joyce

Software Architect, Fujitsu Network Communications

Roshan.Joyce@fujitsu.com

Stephan Neidlinger

BD NMS/SDN/Alliances, ADVA

SNeidlinger@adva.com

Paweł Kaczmarek

Software Manager, ADVA

PKaczmarek@adva.com

Katherine McDowell

Program Manager, IP infusion

katherine.mcdowell@ipinfusion.com

Erdem Cavusoglu

Network Engineer, Facebook

ecavusoglu@fb.com

Contributors

Juan Pedro Fernández-Palacios Giménez
Head of Unit Transport Networks Telefonica CTIO
juanpedro.fernandez-palaciosgimenez@telefonica.com
Alfredo González Muñiz
Technology Expert, Telefónica CTIO
alfredo.gonzalezmuniz.ext@telefonica.com

Table of Contents

List of Figures

1. Introduction

This whitepaper is describing the findings and conclusions of the OOPT-CANDI (Open Optical and Packet Transport – Converged Architectures for Network Disaggregation and integration) optical PoC performed during May 2021.

OOPT-CANDI focuses on defining operator use cases in open and converged packet and optical networks. OOPT-CANDI also proves that use cases can be met with architectures based on open technologies and finally leverages the opportunity provided by TIP to involve different players to accelerate technical development and help operators in real-world scenarios. This is done through Proof of Concept (PoC) trials.

Partial disaggregation consists in separating optical line system (OLS) from terminals (transponders). This separation is motivated by several reasons: for example, the life cycle of transponders is shorter than the optical line system's life cycle and transponders represent most of the cost of the WDM network. The capability to support multiple generations of transponders is expected to help introducing new vendors, favouring competition, and protecting against industrial hazard. Although limited to the transponders, the partial disaggregation raises challenges because of the analogue transmission behavior of optical systems, which requires coordination between OLS and transponders, usually provided by the same controller entity as a turn-key solution. One of the main issues in such scenario is to discover relevant information on the third-party transponders, to be capable of configuring them, to perform multi-vendor path computation, to extract topology and inventory across vendors, to make diagnosis and health assurance across vendors. Most of the challenges lies in the exposed models and their level of openness.

The CANDI Optical PoC demonstrated: i) a unified SDN control and management of disaggregated, multi-vendor components within an open optical network, including

Open optical line system (O-OLS), Open Optical Terminals (O-OTs), Optical SDN controller, O-OLS domain controller and Optical Planning tool and ii) multi-vendor integration and service operations through open standard models and APIs supported by the Optical SDN Controller, including OpenConfig, ONF Transport-API (T-API) and Open interfaces such as the experimental GNPy interface, which is based on REST principles. Integration was then validated across several functionalities: network topology discovery of the open OLS and OTs into the Optical SDN controller; on-line provisioning of optical circuits including dynamic on-demand path computation using an external vendor-neutral planning tool; and the establishment of optical circuits between transponders and line interfaces from different vendors over partially disaggregated OLS.

The goals for this trial were:

1. To explore the state of the art and equipment capability to support partial disaggregation and assess the applicability of open data models to implement the following features:
    a. Use a third-party (external) tool for multi-vendor path computation, by means of a dedicated interface.
    b. Configure third-party O-OT (that are not provided by the same vendor that provides the O-OLS).
    c. Extract topology and inventory information from multiple sources, such as SDN controllers or devices provided by different vendors.
    d. Assess interoperability at the data plane level between the O-OTs and the O-OLS.

2. Test and Validate implementations of OpenConfig and T-API data models and perform a detailed gap analysis in view of the aforementioned features towards standardization.

3. Demonstrate on-line provisioning with GNPy QoT validation and dynamic transceiver mode selection. The QoT validation shall: i) test the new GNPy

experimental API and ii) retrieve exposed physical layer, amplifier, and fiber plant data via T-API interface and iii) map such data to physical layer parameters including dynamically measured values from OpenConfig and T-API.

Tests were performed in the TIP community lab in London, UK, checking the interfaces compliance to relevant and applicable standard(s) (T-API, OpenConfig and GNPy) and validating the end-to-end provisioning results, cross-checking measured and estimated performances.

In conclusion, most tests were successful and while specific small deviations to the standards were reported, they should be addressed in subsequent software upgrades. Performance cross-check enabled to identify some inconsistencies between the link design and the actual configurations. This was due to a missing provisioning/validation step in the agreed workflow of the PoC and it is to be noted that such step is commonly present in production networks.

1.1 Terminology

In this document the following terms are employed with the definitions specified below:

- **Open:** Generally, when this term is prefixed to a target (e.g., an Open Line System, O-OLS), it implies that the target (system, piece of equipment, software (SW), network function, etc.) exposes interfaces, often Application Programming Interfaces (API), with a well-known and standard protocol, model, and rules for use for the integration by the Operators in their networks and OSS/BSS systems.
- **Network Element (NE):** in this context, an NE is a piece of equipment, housing homogenous network functions, possibly made by several shelves or blades, but seen by Management & Control systems (i.e., SDN Controllers), as a single management entity through a suitable API, often termed South Bound Interface (SBI).

- **Optical Terminal (OT):** in the context of this document, the term designates a category of NEs in an Optical Transport Network, including the network functions of Transponders (1:1 mapping of clients to line side interfaces); Muxponders (N:1 mapping and multiplexing); Switchponders (N:M mapping, digital switching, and multiplexing). Their role is to adapt digital clients of the Optical Transport Network over DWDM channels.
- **Open-Optical Terminal (O-OT):** this term as defined in this paper, extends the definition of Optical Terminal to all the categories of devices housing Open Optical Line Interfaces (O-OLIs), which fulfils the set of requirements, also included in this document. In summary, these are: the support of Open and Standard management interfaces; the standalone deployment and management, independently from the Open Optical Line System (O-OLS). An O-OT adapts digital clients and generates one or more "alien wavelength" optical DWDM channels to be transparently transported by one, or a chain of O-OLS.
- **Optical Line System (OLS):** the optical transport network segment that exclusively supports optical analogical DWDM channels as clients.
- **Open-Optical Line System (O-OLS**): a complete and autonomously managed optical transport network also supporting (together with digital clients), or exclusively supporting optical analogical DWDM channels as clients. In the context of this document, the term "open" refers to the fact that an O-OLS allows, as analogical clients, any signal which follows a given behaviour, specified by the O-OLI definition in [MUST OON]. It exposes a programmable NBI API complying with specification herein.

2 PoC Overview

The CANDI optical PoC involved, from a macroscopic perspective, the know-how, expertise, and hardware/software elements provided by multiple actors, including network operators, infrastructure providers/vendors and research centers. In particular, the PoC included: i) a generic planning tool - GNPy - developed by the TIP OOPT-PSE working group, ii) the Virtuora Optical SDN controller and T600 O-OTs provided by Fujitsu, iii) the Ensemble Controller for the O-OLS domain and an O-OLS composed of three FSP 3000 ROADMs provided by ADVA, iv) Cassini O-OTs hardware boxes provided by Edgecore with the Network Operating system (NOS) provided by IP Infusion. Telco operators (Telia, Orange, Telefonica) provided the initial set of requirements and specifications, and CTTC contributed with the test specification(s) and protocol implementation validation.

The APIs used in the CANDI optical PoC where Open GNPy API (based on REST principles), used between GNPy and Virtuora; OpenConfig/Netconf between Virtuora and the respective O-OTs; T-API/RESTCONF between Virtuora and the Ensemble Controller, and a proprietary ADVA API between the Ensemble Controller and the O-OLS ROADMs. The full architecture is shown in Figure 1.

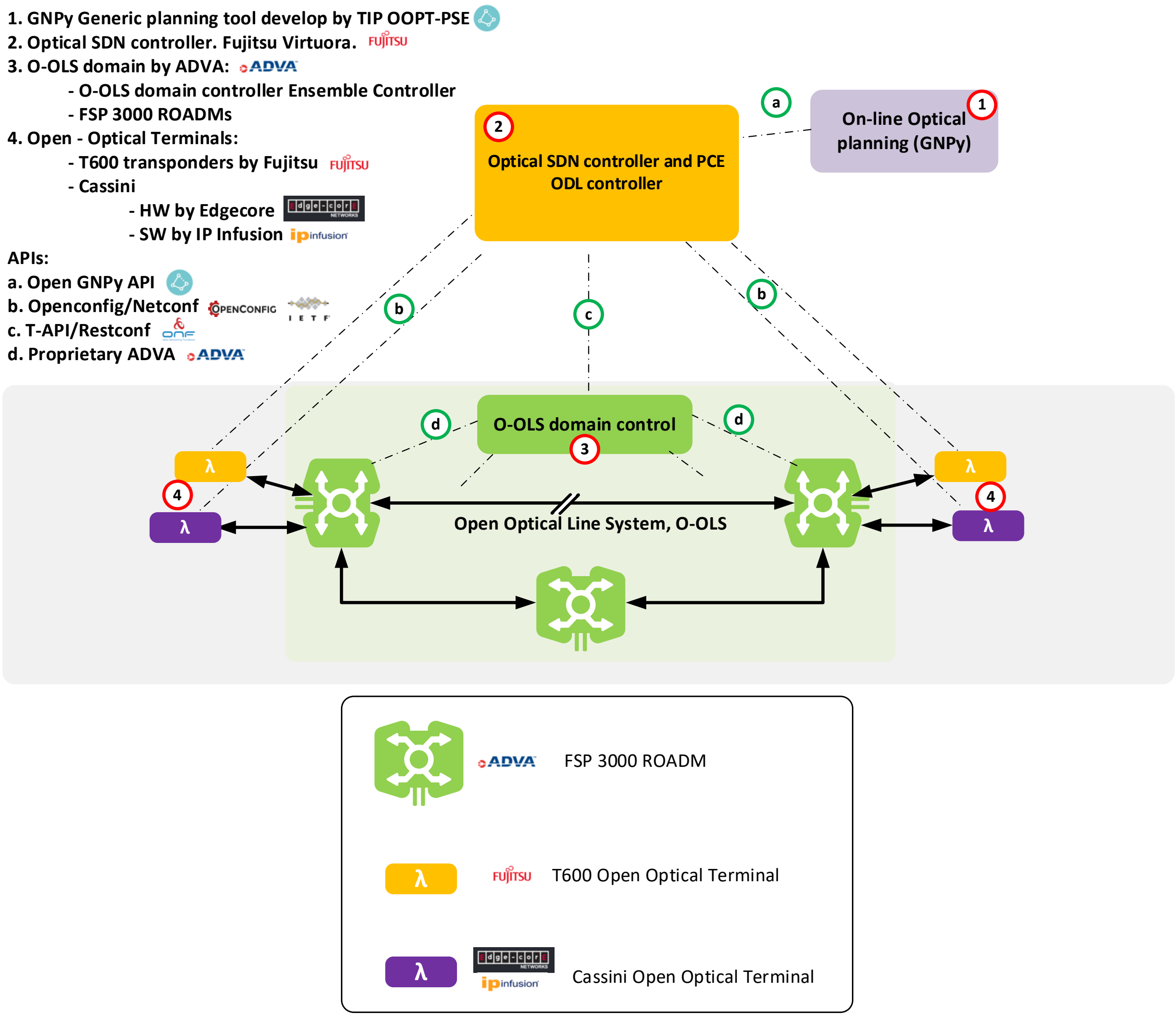

Figure 1 The architecture of the CANDI PoC showing the different functional elements, the involved APIs and interfaces and the underlying topology.

3 Demonstration Workflow

The workflow of the Demonstration of the on-line provisioning with GNPy QoT validation included the following steps, as can be seen in Figure 2:

1. The end user defines an equipment library, off-line, in the GNPy software planning tool.
2. The user uploads, via a GUI, non-equipment information, notably:
   a. Fiber type(s) for the fibers in the optical mesh (e.g., SSMF)
   b. Fiber length(s)
3. The user, via GUI:
   a. Instantiates the links between the O-OLS and the O-OTs line interfaces since such links cannot be automatically discovered.
   b. Provides the required information (IP address, port, credentials, etc.) to add the O-OT
4. The Optical SDN controller:
   a. Discovers the O-OTs main properties using a Netconf session and OpenConfig data models. This includes the component hierarchy, names, and status of the optical channel(s).
   b. Discovers the O-OLS topology via RESTCONF using the T-API models from the O-OLS domain controller. In particular, the different T-API Service-Interface Points (SIPs) nodes and links, along their properties.
   c. Builds a network topology including the O-OTs, fiber(s) and the O-OLS elements. Such topology can, in turn, be exported towards the end user and displayed in the GUI.
5. The user, via the GUI, requests a light path in the optical topology, specifying the endpoints and the required attributes.
6. The Optical SDN controller:

   a. Computes a light path in the optical topology

   b. Requests feasibility for the candidate light path from GNPy

7. GNPy performs impairment validation calculation and sends response to optical SDN controller if the candidate light path is feasible.

8. If the light path is feasible, the optical SDN controller sends the order to the O-OLS domain controller to provision the optical light path in O-OLS network using T-API.

9. The O-OLS domain controller configures the optical circuit.

10. The Optical SDN controller configures the O-OTs modes, spectrum, and power of optical signals.

11. The Light path is activated by optical SDN controller.

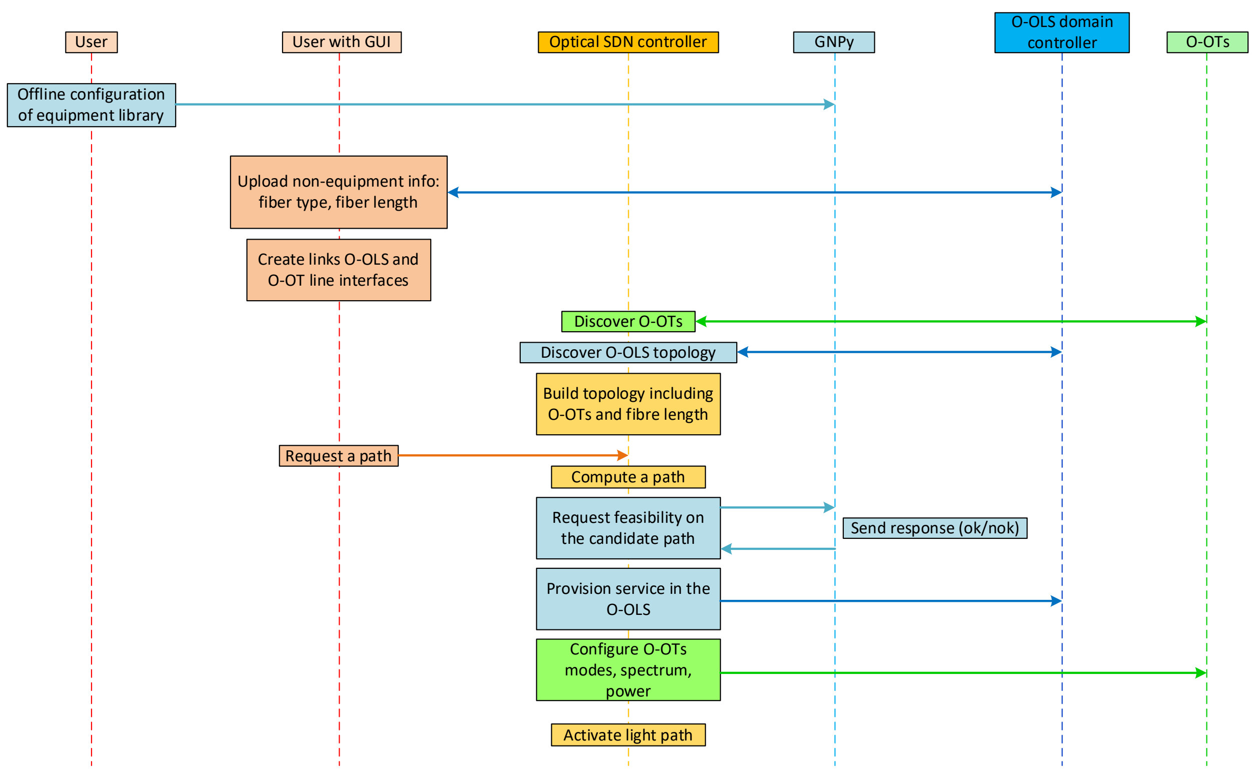

Figure 2 Demonstration workflow

4 Functional Assessment of Interfaces and Tests/Validation

4.1 Open Optical Terminal support with OpenConfig device models

OpenConfig (OC) is an informal working group of network operators sharing the goal of moving our networks toward a more dynamic, programmable infrastructure by adopting software-defined networking principles [OpenConfig]. OC initial focus is on compiling a consistent set of vendor-neutral YANG data models based on actual operational needs from use cases and requirements from multiple network operators. It is commonly understood that such models are used along with the NETCONF protocol as defined by the IETF [RFC 6241]. In this regard, in all tests (isolated or integrated) the SDN controller acts as NETCONF client and the agent in the device acts as NETCONF server. In consequence, the PoC aims at testing not only the support of OC data models but the NETCONF protocol implementation as well.

The PoC has consolidated the use of OC data models for terminal devices (such as transponders or mux-ponders) notably in partial disaggregated scenarios, further contributing to their wide industry adoption across multiple vendors, projects, and demonstrations. Let us note that the OpenConfig data models cover many aspects related to network operation and configuration (including for example, layer 2 and layer 3 devices). In this regard, the PoC has focused on a **well-defined subset of OC models**: the modules related to OC platform, terminal device, and optical transport models (and their dependencies), targeting the configuration of (terminal) optical devices.

In this setting, the trial has covered, mainly**, a set of common operations** - as defined by the OOPT-MUST working group [MUST] use cases -- related to the basic discovery of system properties, platform components and their hierarchy, as well as the configuration of optical channels. Significant effort has been spent in testing the latter, in terms of configuring nominal central frequencies, target launch powers and operational modes (an abstraction that encompasses key transmission properties such as the modulation format, baud rate, or FEC). Additional tests have validated setting the administrative state of a port and the agreed procedures to delete a service. In all cases, the tests involve the use of NETCONF get, get-config and edit-config operations over a secure channel and expected behaviour and potential deviations have been documented.

The PoC has validated the use of OC models and their implementation and support of the agreed use cases in selected devices. Macroscopically, the different test cases have been validated to a large extent. **The main objectives of the test** (that is, the validation of the use of NETCONF/YANG with OC data models) have been completed yet specific issues remain, to scale, especially in medium to large-sized networks.

As a summary of main **takeaway message**s:

- The NETCONF protocol remains quite complex to implement for fully conformant solutions. In the case of the tested implementations, full support of the advanced filtering mechanisms (used by the clients either by using subtree filters or xpath expressions) is still a work in progress. The impact of this limitation at the scale of the PoC is low, given the scale of the infrastructure, for a client may retrieve large volumes of information and perform local filtering.
- Another aspect worth developing is the consolidation of the use of namespaces in the NETCONF/XML messages. Support across implementations varies and temporary workarounds in specific encodings have been required. While this is not a major issue, it needs to be addressed in the long-term goal of having unique implementations regardless of actual vendor devices. As of today, even minor deviations across multiple vendors limit interoperability, especially when the number of supported vendors increases.
- Regarding the OpenConfig data model, the specification itself is not closed and keeps evolving, based on operators' requirements, and following an open process. In this regard, different versions across the YANG modules may mean that specific configuration options (YANG leaves) may or may not be

present. While the effect of this can be mitigated (e.g., by agreeing beforehand the actual version to be used) robust implementations are required, and such implementations should include strict feature and version checking.

- Another important feature of OpenConfig is that it allows the configuration of associations between logical channels (as a flexible abstraction to configure a given connection multiplexing hierarchy in terminal devices with several ports) and down to the optical channel. For example, two client 100G Ethernet signals coming from different ports may be encapsulated using OTN framing and transmitted over a single line port as a 200G optical signal. This feature provides a lot of flexibility in configuring services (e.g., adding and removing client signals dynamically) The support of this capabilities across multiple implementations is not homogeneous and further work is required in this regard.

Despite having identified a few aspects worth of development and improvement, the OC support of Open terminals has been tested and deemed usable in most cases and in deployments close to production networks. That said, it is worth highlighting that some of the PoC elements have used experimental versions of the underlying software and further/extensive validation should be performed as part of a product development process (e.g., within the TIP Test & Validation program) as well as unifying OC implementations across vendor solutions.

4.2 O-OLS controller T-API support

For several years, ONF T-API models [T-API] have been used in multiple interop events to demonstrate their use as an open and standard controller North Bound Interface (NBI), thus allowing clients (such as telcos' business and operations support systems, parent controllers or, orchestrators) to control and operate a transport network. T-API provides a uniform and coherent set of YANG models covering basic aspects such as topology management, service provisioning or path computation. Recent versions of T-API (2.1.2 and later) have largely increased support for the photonic media layer, allowing a new degree of flexibility in the provisioning of digital (ODU/OTU) and optical (OTSi/media channels) connectivity services. The T-API models are assumed to be used in conjunction with a transport protocol, to enable the parent SDN controller to request services to the underlying OLS controller. Such services are understood as the provisioning of Optical Channels over the Open Optical Line System.

T-API already includes mechanisms to constrain the provisioning, notably in terms of route inclusion and exclusions, which has been a key functionality to apply path restrictions as provided by the external planning tool. The set of T-API models are complex, covering many aspects of the control and management of transport networks. In this regard, the availability of companion documents such as the Reference Implementation Agreement (RIA) within [ONF TR-547] has proven to be very valuable. The RIA is focused on a set of key use cases and provides guidelines on the use of the models and examples of actual information objects.

T-API models are commonly used over the RESTCONF [RFC 8040] transport protocol, in this case, between the optical SDN controller (acting as a client) and the OLS controller (acting as a server). RESTCONF support in the tested implementations is solid, with only a minor issue related to the error return codes. This is expected and has helped improve the T-API RIA and to clarify expected behaviours.

Macroscopically, the PoC has validated the use of T-API as a controller NBI, allowing the provisioning of constrained connectivity services and the operation of an open optical line system, extracting inventory and topology data that is required by planning tools. However, further work is needed to be able to fully use the planning and path computation capabilities of GNPy as an external tool. For the PoC, the T-API models have been augmented with a (limited) set of data to enable the use case targeted in the PoC, where the parent SDN controller is retrieving optical fiber characteristics, such as fiber type or attenuation. Power values captured at the levels of OMS/OTS connections are also reported. Detailed information about the amplifier variant types is managed using temporary encodings and solutions, pending further standardization by the ONF T-API WG.

During the tests, several (sub)cases have been tested. This includes aspects such as: i) the ability to retrieve the set of Service Interface Points (SIPs) along the tunability constraints associated to each client port; ii) the ability to retrieve the topology of the network in terms of links and nodes, node edge points (NEPs, a common term for port) and iii) the creation and deletion of optical channel connectivity services.

As a main conclusion, the tested T-API implementations -- both in terms of client and server -- are conformant to the predefined use cases and scenarios.

It is worth mentioning that, upon completion of the PoC, the ONF T-API working group has started a gap-analysis and extension(s) proposal, covering not only the basic aspects of capability, resource and topology discovery but also missing features such as direct connectivity service provisioning based on GNPy outputs (to reflect GNPy outputs into T-API-enabled network / device configuration including the configuration of amplifiers) as well as the dynamic connectivity service modification and adjustment based on network state changes and continuous operation.

4.3 On-line impairment validation with Open GNPy API

As for GNPy (Gaussian Noise Model in Python), the objective of the CANDI optical demo was to check the possibility to use a third-party open tool to check path feasibility and to experiment the proposed workflow. GNPy inputs drove the requirements on the different data to extract from the O-OLS and from the O-OTs to expose the data to the GNPy. GNPy uses 3 types of inputs: a topology (with equipment settings), a library (with simulation settings and equipment specifications) and a service (listing the requests with their constraints). We identified two classes of data: *on-line*, to be retrieved from controllers or devices and feeding topology and service GNPy input, and *off-line*, to be put into the GNPy library. On-line data was limited to amplifiers' type (preamp or booster in our case) and

gain setting, fibers' type, length and span loss and terminals' type, and supported operational mode identifiers. The span loss was reported as the difference of power measured at the input and output of fiber spans. The detailed specification of terminals' operational modes (including, notably, bit rate, baud rate, modulation format, FEC and minimum required OSNR), amplifiers (max power and noise figure) as well as line settings (per channel target output power out of ROADMs) were recorded in the GNPy library.
The GNPy server was installed on a virtual machine in the TIP community lab premises. GNPy path validation functions are available to the SDN controller via an experimental API using YANG models that correspond to the current JSON encoding of the inputs of the program. The current implementation is available at the GNPy github repository (Telecominfraproject/oopt-gnpy at experimental/2021-candi). The tests consisted in two verifications:

- i) Interfacing tests: Check the correct format of GNPy validation requests.
- ii) Consistency tests: Check the consistency of the simulated performance with respect to the experimental set up

For the interfacing tests, a set of path validation requests was built by Virtuora including several cases addressing potential lightpath requests between either T600 or Cassini O-OTs, optionally including a terminal operational mode. If an operational mode was not provided, GNPy suggested one based on the preloaded library of operational modes. All these interfacing tests were successful.

For the consistency checks, we obtained the simulated performance of the network. During this check we identified some inconsistency between the simulated scenario based on the reported span loss and gains, and the expected design values. We found that some of the computed amplifiers' gains -- from the initial off-line design -- as well as the target output power from ROADM, were not the actually configured ones, which had led to incorrect evaluation of the per channel powers on the propagation for the concerned spans. This difference comes from some fine tuning that was necessary during the system installation but due to a missing step was not properly reflected and not reported to the controller or to the library.

In addition, let us note that the overall workflow relies on the GNPy library, which is currently maintained manually by the operator, based on vendors specifications and other inputs. This proved to be complex to maintain up to date and error prone, even in this limited scope. These results led to some main learnings:

- GNPy was extremely useful to make the consistency check(s).
- Having actual gain is a must but is not enough, having measured input and output power of amplifiers out of the O-OLS appears as a critical requirement (as a minimum, aggregated power values, although per channel power is highly desirable).
- The actual workflow followed by the PoC missed a commissioning step to update library or amplifier settings, making sure that the reported values via e.g., T-API extensions actually correspond to configured / running operational state. It is expected that in production ready implementations the O-OLS domain controller is responsible for assessing the consistencies of values.
- Relying on the operator as a mediator to build the library from different sources, and having to maintain it in case of evolution, new releases, etc. is not advisable in a complex operational context. In this regard, having all (a maximum of) data directly reported by devices and collected/ computed/aggregated through the controller minimizes error and provides increased robustness.

4.4 End-to-end control

As outlined above, Virtuora played the central role of the Optical SDN controller in providing E2E control and management of multivendor scenarios with southbound O-OLS domain(s) and O-OTs through standardized APIs such as OpenConfig (device model) and T-API (network/service model), automatically resynchronizing local data stores in case of southbound loss of communications. In verifying E2E control and visibility across the network, the following test cases were executed within Virtuora:

- Discovery of the entire network topology including nodes and links.
- Service operations across multi-vendor O-OTs / O-OLS domains.
- Online optical reach verification and path validation via GNPy.

The use of OpenConfig/NETCONF based policies for O-OTs as well as T-API/RESTConf based policies for the O-OLS domain enabled the discovery of the network within the optical SDN controller and provided complete observability. Although there remains some manual pre-configuration needed to establish connections between the controller and the OTs and O–OLS domain controller, the use of the policies allows for expansion to additional O-OTs and O-OLS domains as the network continues to scale.

Optical service activation/deletion scenarios considered various path constraints as would be applicable in a production deployment. It was observed that the T-API interface also needed some augmentations to factor in additional service connectivity configurations (e.g., amplifier settings) sent from the optical SDN controller to the O-OLS domain controller. Controller integration with GNPy also tested for the feasibility of various operational modes supported by the O-OTs by reconciling against the predefined equipment library and topology information maintained within GNPy, thereby verifying optical path reach.

In the CANDI PoC, the optical SDN controller performed complex functions and tasks such as topology, path computation, and end-to-end service control by configuring optical parameters including modulation format, output power level, and central frequency on the OTs and establishing optical circuits across the O-OLS domains. Note that the path computation function can potentially be offloaded to GNPy in subsequent scenarios. This structure allows a centralized network operations team to perform extended administration and operations tasks like policy development and intent-based networking, while using O-OLS domain controllers to perform sub-tasks based on pre-configured services that normalize service delivery concurrently across vendor domains.

5 Summary and Conclusions

The **CANDI Optical PoC** successfully demonstrated the use of standards based open interfaces in providing unified SDN control and management **across the disaggregated**, multi-vendor components within an open optical network: O-OLS, O-OTs and Optical Planning tools.

In the overall targets for the PoC we set out to give an **overview of the possibilities** to retrieve data which could be both static and dynamic. The data would be extracted from multiple sources from different suppliers including both devices and SDN controllers. To minimise the manual/human intervention our target was to have as much as possible retrieved via on-line APIs such as OpenConfig, T-API and the open GNPy interface.

OpenConfig and T-API models have proven to provide a strong foundation for multivendor service management over partially disaggregated optical networks. As standards continue to evolve, **supplementing** these models with specific use cases and implementation agreements (E.g., through the TIP OOPT-MUST group) will enable suppliers and operators further operationalize open optical networks.

In general, any suggested API enhancements to OpenConfig / T-API will need to be **backwards compatible,** but in some cases where this is not possible, interoperability will require supplier collaboration to agree on version support and negotiation strategies in getting closer to deployment. As observed over the course of the trial, the optical SDN controller must also accommodate for variations across vendor implementations and provide for augmentations of these models across O-OT, O-OLS suppliers.

The **integration of the optical SDN controller with GNPy** for online optical provisioning and reach verification further validated the feasibility of deploying and automating optical services across multi-vendor O-OTs and O-OLS network domains.

The PoC relied on GNPy currently defined inputs to identify required parameters in this multi-vendor context and having them available from the O-OT and the O-OLS via their control interfaces. Based on this, some **experimental augmentations** to the T-API were adapted, such as the inclusion of span loss, fiber length, amplifier types or operational configuration settings. Let us note that the CANDI group is planning to work on these augmentations and propose concrete improvements to the standard via OOPT-MUST and a dedicated task within the ONF T-API group has been started.

Currently, in multi-vendor partially disaggregated networks, there is no solution on how to retrieve data regarding the link between O-OTs and the O-OLS domain (the connection between O-OT line interface and the add-drop port of the ROADM filter). O-OT device data is retrieved via OpenConfig/Netconf whilst the data from the O-OLS domain is retrieved via T-API/RESTConf. In consequence, the connections had to be created manually in the optical SDN controller, which is not a viable solution for the future.

The trial also showed some weaknesses in the workflow process:

1. The equipment was tuned during installation and diverged from the planned configuration points. The tuning was however not all reported in the OLS T-API topology exposed to Virtuora, which led to inconsistency in the reported design (mixing both measured data for the loss and configuration data for the gain target). GNPy set of tests identified these inconsistencies but further diagnosis needed a complete view of amplifiers' working points, and this was not anticipated. This confirmed the need to have both config and measured data for optical power in amplifiers and will probably drive the discussion on the augmentations' best place in T-API.

2. GNPy relies on an *equipment library* which contains equipment performance data. For the PoC, part of this data was considered confidential (transceivers minimum OSNR for example). Thus, the workflow relied on an additional step where the information was stored in an encrypted way. Even with a limited number of equipment this mediation role / step proved to be error prone. We highlight the need to have this kind of data made available via APIs as much as possible, so that the library can be built out of the devices.

In conclusion, the trial has been a reality check of the partial disaggregation scenario and the learnings have been rich in this respect. The TIP community lab and the will of the partners to contribute to the trial made the demo a success. This shows that international cooperation between SDOs, industry groups and open source communities is key to solve the openness and interoperability challenge.

7 References

[MUST]	"TIP OOPT MUST: Operators Driving SDN for Transport Adoption & Acceleration", online: https://telecominfraproject.com/oopt/
[RFC 8040]	Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.
[RFC 6241]	Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.
[RFC 7950]	Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.
[OpenAPI]	OpenAPI Specification Version 3.0.2, <https://swagger.io/specification/>
[OpenConfig]	OpenConfig Project website and models <http://openconfig.net>
[ONF TR-527]	Functional Requirements for Transport API, June 10, 2016, ONF TR-527, <https://wiki.opennetworking.org/display/OTCC/T-API+Documentation>
[ONF TR-547]	T-API Reference Implementation Agreement, July 2020, ONF TR-547, <https://wiki.opennetworking.org/display/OTCC/T-API+Documentation>
[GNPy]	GNPy: Optical Route Planning Library, https://gnpy.readthedocs.io/en/master/index.html
[MUST OON]	TIP OOPT MUST Optical Whitepaper: “Target Architecture: Disaggregated Open Optical Networks”, online: https://telecominfraproject.com/oopt/

8 Acronyms

API	Application Programming Interface
CANDI	Converged Architectures for Network Disaggregation & Integration (TIP)
DWDM	Dense Wavelength Division Multiplexing
FEC	Forward Error Control
gNMI	gRPC Network Management Interface
GNPy	Gaussian Noise Model in Python (TIP)
gRPC	Remote Procedure Call
GUI	Graphical User Interface
MUST	Mandatory Use Case Requirements for SDN For Transport (TIP)
NBI	North Bound Interface
ODU/OTU	Optical Data Unit/Optical Transport Unit
OLS	Optical Line Systems
ONF	Open Networking Foundation
O-OLS	Open Optical Line Systems
OOPT	Open Optical Packet Transport (TIP)
O-OT	Open Optical Terminal
OSNR	Optical Signal to Noise Ratio
OT	Open Terminal
OTN	Optical Transport Network
PSE	Physical Simulation Environment (TIP)
ROADM	Reconfigurable Optical Add/Drop Multiplexer
SBI	South Bound Interface
SDN	Software-Defined Networking
SDO	Standards Defining Organization
T-API	Transport API (ONF)
TIP	Telecom Infra Project

TIP Document License

By using and/or copying this document, or the TIP document from which this statement is linked, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to copy, display and distribute the contents of this document, or the TIP document from which this statement is linked, in any medium for any purpose and without fee or royalty is hereby granted under the copyrights of TIP and its Contributors, provided that you include the following on ALL copies of the document, or portions thereof, that you use:

1. A link or URL to the original TIP document.

2. The pre-existing copyright notice of the original author, or if it doesn't exist, a notice (hypertext is preferred, but a textual representation is permitted) of the form: "Copyright 2019, TIP and its Contributors. All rights Reserved"

3. When space permits, inclusion of the full text of this License should be provided. We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of TIP documents is granted pursuant to this License. except as follows: To facilitate implementation of software or specifications that may be the subject of this document, anyone may prepare and distribute derivative works and portions of this document in such implementations, in supporting materials accompanying the implementations, PROVIDED that all such materials include the copyright notice above and this License. HOWEVER, the publication of derivative works of this document for any other purpose is expressly prohibited.

For the avoidance of doubt, Software and Specifications, as those terms are defined in

Disclaimers

TELECOM INFRA PROJECT