D 2.2 OVERALL NETWORK ARCHITECTURE AND CONTROL ASPECTS DEFINITION

Project title	**Photonics technologies for ProgrAmmable transmission and switching modular systems based on Scalable Spectrum/space aggregation for future aglle high capacity metrO Networks**
Project acronym	**PASSION**
Grant number	**780326**
Funding scheme	**Research and Innovation Action – RIA**
Project call	**H2020-ICT-30-2017 Photonics KET 2017 Scope i. Application driven core photonic technology developments**
Work Package	**WP02**
Lead Partner	**CTTC**
Contributing Partner(s)	**CTTC, SMO, TID, TUE**
Nature	**R(report)**
Dissemination level	**PU (Public)**
Contractual delivery date	**31/01/2020**
Actual delivery date	**31/01/2020**
Version	**1.0**

History of changes

Version	Date	Comments	Main Authors
0.0	18/11/2019	ToC	Ricardo Martínez, Michela Svaluto Moreolo
0.1	08/01/2020	Inputs in Section 2 and 3	Ricardo Martínez, Michela Svaluto Moreolo
0.2	21/01/2020	Providing Executive Summary, Introduction and Conclusions	Ricardo Martínez, Michela Svaluto Moreolo

0.3	23/01/2020	First integrated version	Ricardo Martínez, Michela Svaluto Moreolo, Javier Vilchez, Raul Muñoz, Josep Mangues, Paolo Dini
0.4	27/01/2020	Added contribution on HL4-HL2/HL1 path characterization. Full section 2 update.	Gabriel Otero, David Larrabeiti, Pedro Reviriego, Jose A. Hernández, Juan P. Fernandez-Palacios, Victor Lopez, Netsanet Tessema, Patty Stabile, Nicola Calabretta
0.5	27/01/2020	Integrating reviews and comments from SMO	Germano Gasparini, Giorgio Parladori
0.6	27/01/2020	Editorial changes of 1st full integrated version	Ricardo Martínez, Michela Svaluto Moreolo
0.7	30/01/2020	Quality Check	Netsanet Tessema, Patty Stabile, Nicola Calabretta
1.0	31/01/2020	Compiled final version	Ricardo Martínez

Disclaimer

This document contains confidential information in the form of the PASSION project findings, work and products and its use is strictly regulated by the PASSION Consortium Agreement and by Contract no. 780326.

Neither the PASSION Consortium nor any of its officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

The contents of this document are the sole responsibility of the PASSION consortium and can in no way be taken to reflect the views of the European Union.

***This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 780326**.*

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This report corresponds to the deliverable D2.2 of the PASSION project and aims at providing the overall networking framework with the focus on the control plane related aspects. In such a macroscopic objective, this document first overviews the defined use cases being tackled within the project along with their derived requirements. Two use cases are specially investigated: the “pay-as-you-grow” and the “on-line restoration”. The “pay-as-you-grow” use case leverages the modular design of both the S-BVT transmitter (Tx) and receiver (Rx). According to the capacity needs at each metro network location, the activated devices within the S-BVT Tx and Rx are tailored/adjusted to the envisaged volume of data traffic. The on-line restoration, on the other hand, targets the regular network operator needs to provide resilient and cost-effective network services. Both use cases constitute the pillars to build up and define the control functions and operations tailored to the technological solutions tackled within the project. The rest of the use cases herein presented are addressed leveraging technological basis and solutions from both control and data plane devised in the first two (“pay-as-you-grow” and the “on-line restoration”) scenarios.

From a high-level perspective, it is overviewed the most relevant features and capabilities of the key network elements and devices explored within the PASSION project, to deploy a cost- and energy-efficient optical metro network supporting high transport capacity (in the order of Pb/s): These network elements and devices are: the S-BVT transmitter and receiver as well as the optical switches. Herein, from the control plane perspective, it is studied the abstracted information model to accomplish the automatic programmability of the whole metro network resources at the time of setting up and removing optical connection demands. The abstraction model is discussed for each network element and it is referred to as the Hardware Abstraction Layer (HAL). The resulting HAL model for the S-BVT Tx / Rx and optical switches provides a scalable management done by a centralized software defined networking (SDN) controller. The definition of the HAL models leads to design the set of control interfaces (APIs) enabling the communication between the controller and the agents locally governing each network element and device. Such APIs cover two purposes: i) enabling an external application requesting optical connections to the SDN controller; ii) enabling the SDN controller to retrieve information and perform the configuration of the network elements and devices. The former is referred to as the Northbound Interface (NBI). The latter is named as the Southbound Interface (SBI). For both NBI and SBI interfaces, it is described the supported operations along with the adopted protocol messages and payload encoding.

The building blocks and function within the designed SDN controller are also detailed. It is described the basic control operations and interactions (via NBI and SBI) with external entities such as the agents of the network elements and devices as well as the application demanding optical connection services. Bound to that, it is discussed the workflow which describes the functions to be handled within the SDN controller as well as the required (NBI and SBI) messages to be exchanged at the time of setting up a new optical connection. The validation of the SDN controller, the NBI and SBI APIs and the designed workflow is being carried out within the WP5 activities.

Finally, and to deal with the defined use cases, two devised Routing and Spectrum Assignment (RSA) algorithms at the controller are described. Both algorithms are executed to dynamically compute and select the route and resources (i.e., S-BVT and optical spectrum) to accommodate optical connections requests. It is assumed that connection demands may request heterogenous data rates. The purpose of both algorithms is to accommodate such data connection demands attaining an efficient use of the overall network resources. The performance evaluation of these RSA algorithms is planned to be carried out experimentally at the control plane level within the context of the WP5.

1 INTRODUCTION

Metro networks are crucial to cope with the huge transport and highly dynamic traffic scenarios to deploy 5G (and beyond) network services. Thereby, notable efforts are being invested to design metro networks offering enhanced flexibility and agility along with high capacity, while attaining reduced energy and power consumption, low footprint and cost. The dynamicity and agility capabilities of the metro network scenario are mainly accomplished thanks to the programmability offered by a centralized SDN controller. The present deliverable focuses on describing the networking control solutions adopted within the PASSION project. The goal is to detail all the network and control aspects (e.g., network topology, interfaces, routing algorithms, etc.) tailored to both the targeted use cases as well as addressing the requirements, capabilities and limitations of the optical technological solutions tackled in PASSION.

In a nutshell, the targeted control framework takes over of the automatic processing, computation, resource selection and configuration of dynamically arriving (and departing) end-to-end optical connection requests within a metro network. The configuration of the metro infrastructure embraces three main network elements and devices: i) S-BVT Tx made up of a modular set of vertical cavity surface emitting lasers (VCSELs); ii) S-BVT Rx formed by a modular pool of Coherent Receivers (CO-Rxs); iii) optical switches with heterogeneous architectures (e.g., different filtering features) supporting basic operations of add, drop and bypass/express of optical signals. A local controller, referred to as *agent*, handles each of these network elements and devices. The agent allows mapping the control commands sent by the centralized SDN controller and the actual configuration of the hardware inventory. To this end, a client-server relationship between the SDN controller and each agent is created which interact among them through a defined interface (API). From the SDN controller perspective, this API is the SouthBound Interface (SBI). In general, the SBI supports retrieving the minimum (abstracted) information about the network element and device and its configurability (e.g. employing a specific VCSEL or Co-Rx in an S-BVT, deploying an optical switch cross-connection, etc.). The definition of the abstracted information model for the different SBIs is essential and is herein termed as the Hardware Abstraction Layer (HAL). Consequently, the design of the HAL for each network element and device paves the way for defining the SBI (operations / commands, used protocol and message encoding), which eventually do support the targeted automatic programmability issued by the SDN controller.

The dynamic arrival of the optical connection requests to the SDN controller are assumed to be generated by an application running on top of the controller. This application, called On-Demand Bandwidth Application, communicates with the controller via a defined interface. Through this interface, the application can request the SDN controller optical connections specifying the endpoints (source and destination) and the required bandwidth (in Gb/s). The interface between the application and the SDN controller is the so-called NorthBound Interface (NBI). The design of this NBI is also tackled in this deliverable describing the set of operations (i.e., Create and Delete an optical connection) as well as the protocol and encoded payload on each defined control message.

The creation and deletion of any connection within the SDN-controlled metro network entails many interactions among the application, the controller, the network elements, and the devices agents. Control interactions are, as said, provided by the defined NBI and SBI APIs. However, there are important functions that are internally handled by the SDN controller, such as the computation and selection of the network resources. The combination of both the controller functions along with the NBI and SBI interactions determine a sequence of commands and operations (i.e., workflow) to be executed to achieve the automatic network management. The design of the SDN controller building

blocks (i.e., application front-end processer, path computation, topology and provisioning managers) and the whole required workflow are thoroughly presented and discussed.

Finally, when a new connection needs to be set up, an on-line routing algorithm is triggered. The objective of the routing algorithm is to seek for available resources to accommodate the optical connection guaranteeing its requirements (e.g., bandwidth). In the framework of optical flexi-grid networks, those routing algorithms are typically called Routing and Spectrum Assignment (RSA). The RSA algorithms use information about the network connectivity, resource availability, technological limitations of the underlying data plane elements and other constraints (e.g., spectrum continuity and contiguity) to find feasible paths for the received connection requests. Two on-line RSA algorithms tailored to the PASSION metro network scenario are devised and discussed. Their exhaustive experimental evaluation and benchmarking under dynamic traffic using different figures of merit are planned to be done in the WP5.

This deliverable addresses the following sections:

- In section 2, the details of the defined PASSION optical metro network architecture are discussed. This is addressed from different perspectives: i) revisiting the hierarchical model of the optical switches forming the targeted metro network infrastructure; ii) recalling the use cases to be investigated with specific focus on deriving the implications for the control functions and data plane aspects; iii) summarizing from WP3 and WP4, those architectural designs for the three key network elements and devices within the PASSION metro network, namely: S-BVT Transmitter (relying on VCSEL technology), S-BVT Receiver (using Co-Rx solution) and optical switches.
- Section 3 constitutes the core part of this deliverable. Specifically, this section addresses the definition of the HAL information used by the SDN controller to enable the programmability of the network elements and devices. By doing so, it is designed the SBI API for supporting the control interactions between the centralized controller and the network elements and devices agents. The APIs are defined to provide low-level information in terms of the protocol messages and the payload encoding. Similarly, it is also described the NBI used by an external application requesting (dynamically) bandwidth connection services to the SDN controller. The architecture of the SDN controller is also presented paying attention on the required control workflow to automatize the establishment of new optical connections. Finally, two proposed on-line RSA algorithms are provided addressing the intrinsic features of the PASSION data plane solutions. To this end, the pseudo-code for each RSA algorithm is given.
- Section 4 concludes this deliverable highlighting that all the control solutions discussed in this deliverable are planned to be validated and evaluated experimentally in the WP5.
- Annex 1 and Annex 2 provide relevant description using the Swagger tool for implementing and deploying the skeleton of the client and server codes of both defined NBI and SBI APIs.

2 PASSION NETWORK ARCHITECTURE

This section is devoted to reviewing briefly the MAN network architecture targeted by PASSION, initially described in [D2.1], and to refining several aspects of this architecture of interest for the use cases that have been developed since its publication.

The target PASSION network architecture is made up of a reference topology synthesized from real network data from Telefonica, and the suite of network elements being developed in the project, namely two types of nodes (ROADMs) and the S-BVTs. These elements implement the optical transport layer for a hierarchy of MAN routers, with the aim of supporting a series of use cases.

This section reviews the PASSION network architecture as follows. The hierarchical structure of this huge MAN topology is reviewed in subsection 2.1. The objective of the target reference topology is setting a challenging scenario for the Pb/s-capable cost-efficient, flexible and dynamic optical transmission, multiplexing and switching technology developed by PASSION. A summary of the requirements derived from the selected use cases of [D2.1] (plus a fifth use case recently added) and additional HL4-HL2/1 path characterisation from statistical analysis of the reference topology are the main architectural aspects considered in subsections 2.2 and 2.3. Then, in order to make the document self-contained, subsection 2.4 overviews the essential internal structure and functionality of the network elements and devices that compose the architecture. This way the document includes schematic definition of the resources whose control is dealt with in the rest of the document. More detailed descriptions of the implementation and progress of the network elements and devices, which are the central research focus of PASSION, can be found in WP3 and WP4 deliverables.

2.1 PASSION REFERENCE MAN TOPOLOGY

One of the activities conducted within the T2.1 and reported in [D2.1] was to elaborate the PASSION reference network structure based on a real metropolitan area topology, where the performance and applicability of PASSION solutions could be analyzed. The motivation and methodology employed to build such a reference network along with an exhaustive topological characterization can be found in [D2.1]. Nevertheless, herein, for the sake of completeness, we provide a quick overview of such reference network.

At the optical layer a MAN network is seen as a layered composition of ring and star topologies (usually ring within the same layer and a star when aggregating). This basic scheme is progressively evolving to meshed topologies as ROADMs with degrees greater than 2 are deployed. In general ROADMs are placed with higher layer network elements such as IP routers, MPLS switches, etc. Add and drop ports provide the physical connectivity between the high layer nodes and the ROADMs.

At the IP layer, the logical topology mainly hides the physical layer topology and the aggregation/distribution hierarchy becomes more evident as shown in Figure 1. In general, an IP level *n* node is connected to a pair of nodes of level (*n*-1) (for protection purposes) by means of physically disjoint optical paths. In general, it can be assumed that a HL*n* node consists of a packet switch attached to a ROADM controlled with an SDN control plane, except in the lower level of the hierarchy (in our case HL5), where, for cost reasons, the topology usually consists of multiple rings of packet switches made up of point-to-point links.

Figure 1 shows a scheme of the logical layer view of PASSION's reference metro network. As it can be seen, five hierarchical levels (HL) are defined, with differentiated functionalities. The average

nodal degree mostly grows from the lowest HL (HL5 degree is 2) to the highest one (HL2 degree is 6).

HL1 and HL2, connected in a mesh topology, constitute the top level in the hierarchy. Both HL1 and HL2 are treated indeed as a single level providing the traffic to be steered towards the core. Specifically the carried traffic is routed: i) to/from either service gateways (e.g. IPTV or CDN caches at HL2, carrying 40% of the core traffic); ii) to the appropriate WAN routers (at HL1) for Internet and other global connectivity services (estimated as 60% of the core traffic).

At the next level, HL3 are fast aggregation and transit nodes. HL5 consists of Base Stations and small COs (Central Office) hosting Optical Line Terminals (OLTs) and Digital Subscriber Line Access Multiplexers (DSLAMs), whereas HL4 are bigger COs that aggregate/distribute HL5 traffic, deal with traffic conditioning and have subscribers attached as well.

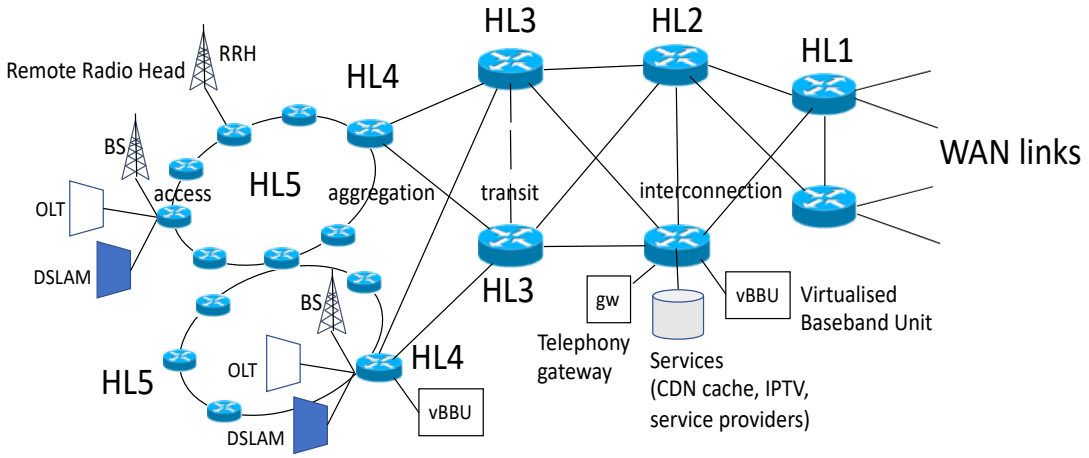


Figure 1. Schematic IP Layer hierarchy of routers as defined in [D2.1]

Network size

As a KET project, the network size, switching and transmission rates need to be a years-ahead forecast of the needs of future metro networks. PASSION project investigates solutions to support the envisaged transport capacity and use cases to be supported within 10 years. PASSION system design is guided by a huge metro network topology synthesized from a real topology reflecting a very large metropolitan infrastructure fed by a huge access network infrastructure.

In this summary, we show two relevant statistics to provide an idea of the sizes of the different segments in which the MAN is split. Table 1 shows the link length.

Table 1 Link lengths at the same level

Link distances [Km]	Mean	Max	Min	Std (σ)
Access Network	1.60	5.88	0	1.16
L1 Aggregation	8.69	22	0.50	5.98
L2 Aggregation	13.10	36.07	1.61	8.84
Core Network	40.17	65.67	26.10	13.96

More importantly, understanding the characteristics of paths across Hierarchical Levels (HL) allow to estimate to what extent all-optical end-to-end interconnection of MAN nodes is feasible.

The characterization of path lengths in terms of hops and span distances across the different HLs is essential to devise the control functions (e.g., routing and wavelength assignment algorithms) aiming at ensuring feasible end-to-end optical connections. Table 2 updates the path lengths considered in [D2.1] with the definition of path between levels as *reaching the closest node of the target level* instead of *any-to-any*.

Table 2 Path lengths between a node and its closest (in fiber distance) node of a higher HL

Connected HLs	Mean (Km)	Mean (hops)	Max (Km)	Max (hops)	Min (Km)	Min (hops)
HL5 - HL4	10.01	5.38	37.56	20	0.69	0
HL5 - HL3	21.00	9.35	44.47	23	7.61	2
HL5 -HL2/1	13.83	8.35	35.35	22	1.30	1
HL4 - HL3	19.73	6	33.07	13	6.81	1
HL4 -HL2/1	12.44	5	23.95	12	0.5	0
HL3 -HL2/1	30.91	2.42	138.74	10	5.09	0

Regarding the links capacities, [D2.1] included data traffic projections accommodating expected future new services (e.g., 5G and bidirectional 1Gb/s connections) which are summarized in Table 4. Given the traffic expected at each type of node, [D2.1] concluded that all-optical connection from HL4 to HL2/1 was the most interesting target for PASSION (use case #2 reviewed in the next subsections).

2.2 SUMMARY OF ARCHITECTURAL REQUIREMENTS DERIVED FROM USE CASES

[D2.1] defined a set of preliminary use cases justifying and motivating the planned design and implementation of the PASSION technological solutions. Those use cases are constantly revisited as the project activities progress because PASSION's design approach is use-case-driven. In other words, the purpose of the use cases is: i) to anticipate/identify key / relevant capabilities required by the targeted future metro network infrastructure; ii) to make sure that the design, technologies and solutions developed PASSION do support them; iii) to analyze the theoretical economic impact of PASSION technology. The following table enumerates the target use cases and identifies their requirements.

Table 3 Considered use cases and their derived main requirements

Use Case #	Use Case Descriptor	Main requirements
#1	Cost-effective ultra-broadband transport and expansion in a large MAN. Key pursued objective is to	• High transport capacities • High modularity featuring "**pay as you grow**": 2Tb/s – 16Tb/s

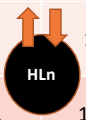
	provide a feasible **'pay as you grow'** scheme	• Granularity of data rate allocation at 50Gb/s, supporting heterogenous data rate demands: 50, 100, 150 and 200 Gb/s • Automatic SDN programmability of network elements and devices • HL1/2 nodes equipped with interfaces supporting aggregate capabilities of tens of Tb/s • Cost-effective upgradeable optical transceivers for HL4 starting at 1 or 2 Tb/s. • Sliceability of transceivers to support simultaneous multiple connections stemming from a single device. • Fast on-line RSA algorithms to attain efficient resource utilization.
#2	Embedded support of **on-line restoration** mechanisms to fast recover disrupted services caused by link and node failures (this is a compulsory reliability-by-design feature included use case #1)	• Ability to compute and program of backup (link and node disjoint) paths restoring "on the fly" disrupted services. • Backup paths may allocate the same VCSELs (i.e., optical carriers / central frequencies) as the primary paths at the endpoints if so required. • Make-before-break and Break-before-make approaches are to be studied determining the implications within the SDN control interfaces and functions • Evaluation under both dynamic traffic scenario along with link failure generation. • Connection requests demanding heterogeneous data rates with 50Gb/s granularity • Multiple performance indicators: blocked bandwidth ratio, restorability, average setup time, etc.
#3	Cost-effective ultra-broadband transport and expansion in a large MAN: Dynamic capacity adaptation and HL3 IP off-loading	• Support of all-optical HL4-HL2/1 paths. • Ability to perform packet (IP) off-loading of HL3 traffic onto the optical layer, leveraging S-BVT and control plane capability to set up direct HL4-HL1 optical channels. • Low-penalty optical switching and multiplexing capability of the PASSION HL4 and HL3 switches to enable long-distance multi-hop all optical circuits without electronic regeneration. • Fast multi-channel provisioning and reconfiguration (in the order of minutes in final production devices) to better adapt to traffic changes variability during a day. • Devising routing and wavelength/spectrum assignment RWA/RSA algorithms to attain efficient resource selection leading to low blocking probabilities even for high traffic loads. • Programmability of traffic trunks and optical circuits to daily patterns.

		• Sliceability should make it possible to simultaneously connect HL4 to HL2/1 nodes and locally connect neighbouring HL4 nodes.
#4	Interconnection for distributed computation sites (e.g., CDN) within the MAN: efficient protection schemes	• Ability to automatically migrate a full edge CDN node to another backup location (2 Tb/s) • Agile switch-over and switch-back capabilities handled by a centralized SDN controller • Fast multi-channel (re-)configuration (in the order of seconds in production products) when applying a switch-over/switch-back or over-flow of traffic to another data center • Full control of end-to-end latencies to keep the requirements of the services being accommodated
#5	Support of massive events: drastic dynamic re-allocation of capacity near the access	• Capability to dynamically allocate massive transport capacity at an HL4 node to support eventual concentration of people demanding high consuming data traffic services and applications such as augmented reality (2Tb/s) • Network elasticity to re-use that capacity after the event in the aggregation segment.
#6	C-RAN support: Midhaul/fronthaul traffic transport over the MAN	• Low latency and jitter optical circuit provisioning on demand for the rate requirements of 5G new radio DRoF. • Elasticity to adapt capacity to load in cellular areas under an HL4 node. • Integration with C-RAN control plane to direct traffic to CU (Central Unit) and DU (Distributed Unit) from the RU (remote Unit) and vice versa.

The most demanding use case in terms of transport capacity is #1. [D2.1] proposes a method for estimating mid-term traffic demands in the considered hierarchical MAN. A summary of the envisaged data traffic on each HL node is provided in Table 4.

Table 4 Hierarchical levels in the target topology and traffic offered from/to each level to/from the core HL1/HL2

Data Rate per node [Tb/s] Node \ Traffic to/from higher level	Uplink Traffic UPS	Uplink Traffic DOWNS	To / From	Avg. # of nodes Aggregated toward Uplink	Degree (phy layer) Min/Avg/Max
HL5 (Access Network)	0.15	0.15	HL4 nodes	2432/380=6.4	1/2/4
HL4 (L1 Aggregation)	1	1	HL3 nodes	380/33=11.51	2/4/10
HL3 (L2 Aggregation)	11	12	HL1 or HL2 nodes	33/6=5.5	3/3/5
HL1 HL2 (MAN Core)	55 (40%) 73 (60%)	60 (40%) 80 (60%)	WAN (Internet, etc) Services (CDN caches, etc)	2/1 4/1	4/6/7



Only hierarchical traffic is shown as it is estimated that only 10% of the traffic remains in each level. Sliceability is paramount, as a single transceiver must be enough to simultaneously:

- Connect a node to their directly attached neighbours to exchange HL-local traffic that is routed at IP layer.
- Connect a node to any node of another HL to create direct optical circuits to exchange hierarchical traffic.

The table gives also relevant information about the nodal degree at the physical layer (rightmost column) and the average number of HL(*n*-1) nodes served by HL*n*.

2.3 WORST CASE HL4-HL1/2 PRIMARY AND SECONDARY PATH CHARACTERIZATION

To determine the target node characteristics at the physical layer and estimate the applicability of the PASSION technology to the targeted use cases, it is paramount to characterize the worst-case path in terms of fiber length, number of hops and type of signal processing. This becomes particularly critical in the Use Case #2 since it poses the most challenging network setting in terms of transmission and switching requirements. Indeed, the objective of bypassing HL3 nodes at the optical layer and establishing all-optical paths from an HL1 or HL2 router to an HL4 router transparently without electronic regeneration may not be possible in a very large MAN.

Use Case #2.1: General case: from a HL4 node to a HL2/1 node. The optical channel is assigned at the source HL4 and optical transported to any destination HL2 or HL1 node. This is the most challenging case because optical paths are longer (in distance) as well as it requires a more complex accommodation of the traffic to the available wavelengths depending on the destination HL1/2 node. This is illustrated in Figure 2. The path starting from the HL4 node at the bottom of the figure is made up of 6 bundles of channels that terminate at each one of the 6 nodes HL1/HL2.

The general case requirements for Use Case #2 is given by the statistics in Table 5.

Table 5 Shortest (in fiber distance) path lengths between any pair of nodes of Hierarchical Levels (HL) HL4 and HL2/HL1

Connected HLs	Mean (Km)	Mean (hops)	Max (Km)	Max (hops)	Min (Km)	Min (hops)	Std (σ) (Km)	Std (σ) (hops)
HL4 -HL2/1	43.02	11.11	90.28	22	8.50	3	17.52	4.86

These requirements are hard to achieve without regeneration, especially in **number of hops (11 hops on average and 22 in the worst case)** due to the impairments introduced by also by the switching.

Alternatively, we may try to minimize the number of hops rather than the physical distance. Table 6 shows the statistics in this case.

Table 6 Shortest (in number of hops) path lengths between any pair of nodes of HL: HL4 and HL2/HL1

Connected HLs	Mean (Km)	Mean (hops)	Max (Km)	Max (hops)	Min (Km)	Min (hops)	Std (σ) (Km)	Std (σ) (hops)

HL4 -HL2/1	78.32	5.2	236	10	0.5	1	40.82	1.51

Now, on average the number of hops is halved, and the distance doubled on average. The worst-case number of hops is halved (from 22 to 10 hops) but the worst-case distance is too long. On the other hand, assuming a standard distribution, it can be said that 68% of paths lie under 118 Km and 7 hops.

Figure 2 shows a path HL4-HL1/2 which may be a candidate to be the worst-case in the reference topology. The figure only shows an arbitrary number of links, except at HL1/2, which corresponds to the real topology. In the scenario we employ the target traffic Use Case #1, which considers up to 1Tb/s of traffic offered at each HL4 node. It should be noted that all 1Tb/s-sources are steered, not to a concrete HL1/HL2 node but to a few HL1/HL2 nodes.

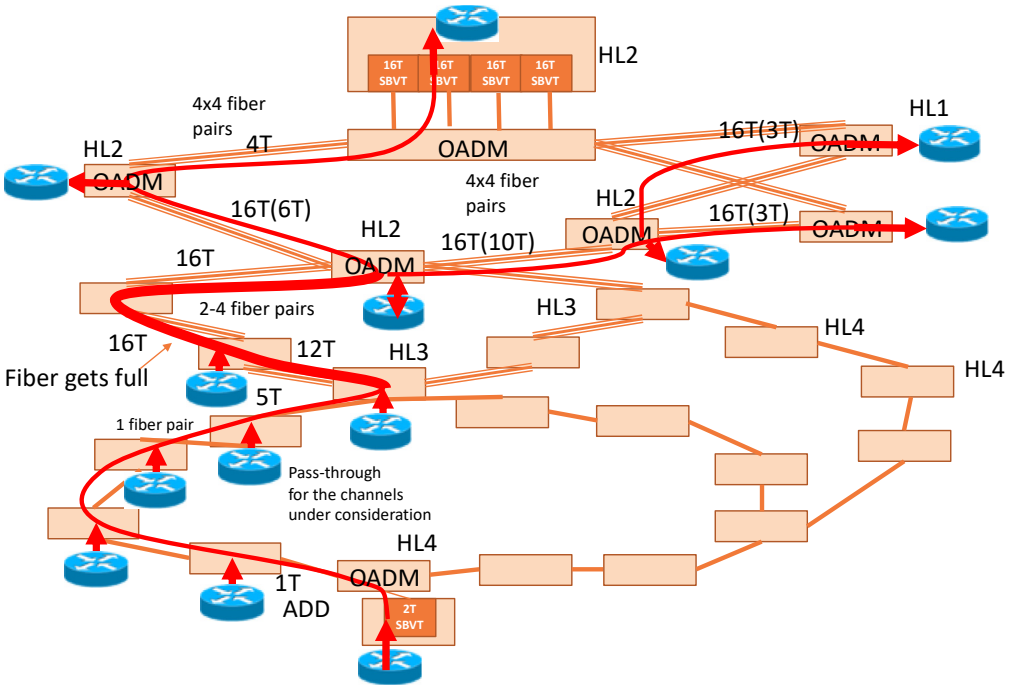


Figure 2. An example illustrating Use Case #2.1 paths HL4-HL1/2.

The figure shows that there is an aggregation of wavelengths (spectral channels) towards the core, and an early saturation of fiber at the second HL3 hop. Then all the wavelengths arrive at the first HL2, some wavelengths are dropped at this node and the rest are forwarded to other HL2/HL1 nodes. This case is rather challenging, and we plan to analyze to what extent it can be covered both as primary paths and through disjoint backup paths if the network operator needs this functionality.

However, let us discuss next another interpretation of HL4-HL2 interconnection that may be more realistic and practical. The sub-use-case #2 target is not reaching a concrete HL2 or HL1 node from a given HL4 node, but just “reaching the hierarchical level HL2”. Indeed, if we constraint the study to the closest two nodes of the target HL, the requirements are relaxed as it was advanced in D2.1 (see Table 7 excerpt from D2.1 Table 4).

Table 7 Shortest (in fiber distance) path lengths between an HL4 node and its closest two nodes of Hierarchical Level HL2/HL1

Connected HLs	Mean (Km)	Mean (hops)	Max (Km)	Max (hops)	Min (Km)	Min (hops)	Std (σ) (Km)	Std (σ) (hops)
HL4 -HL2/1	24.78	6.16	50.33	11	8.50	3	12.20	3.92

In the statistics, it should be noted that a node that plays the role of interface between a HLn and HL(n-1) is supposed to have reached HL(n-1).

Use Case #2.2: from a HL4 node to Hierarchical Level 2/1 i.e. to the closest HL2/1 node or nodes. The optical channel is terminated at the closest HL2/1 node and then de-groomed at the IP layer; after that, the traffic among HL2/1s is exchanged by means of fixed transceivers. The utilization of fixed transceivers at HL2/1 makes full sense given the degree of aggregation of traffic, given that the closer to the core, the least the traffic variation among nodes at a given time of the day and hence the advantage of bandwidth-variable transmission is not so relevant.

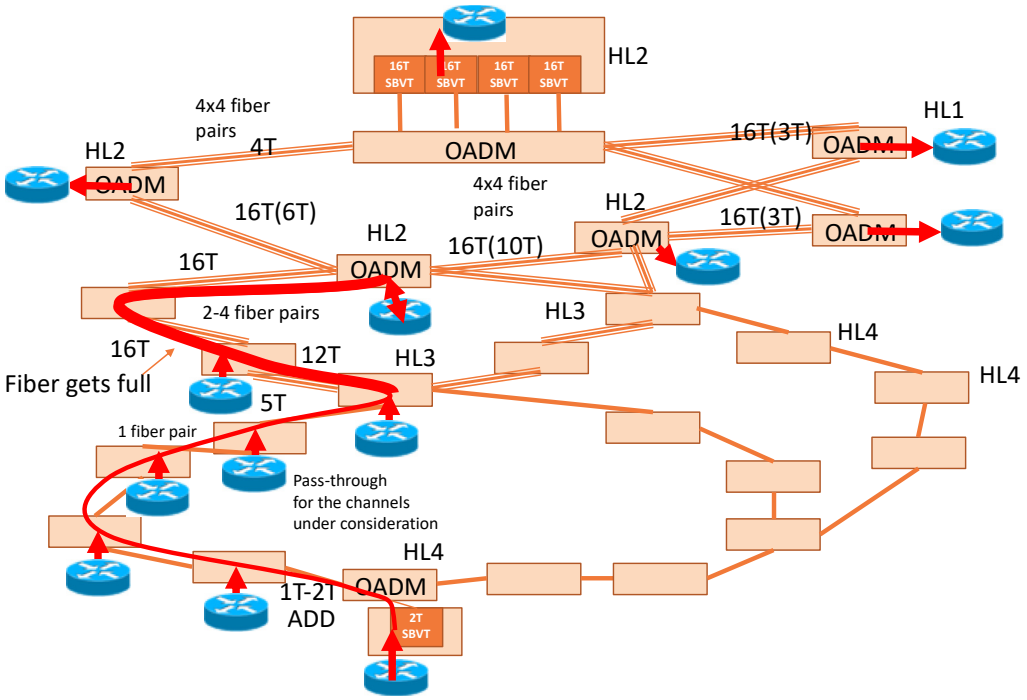


Figure 3. An example illustrating Use Case #2.2: primary path HL4-HL1/2.

In D2.1 [D2.1], we described the procedure to obtain the average, min and max path lengths between different network HLs. Such distances consider that an HL*n* node is logically connected to the two closest HL(*n*-1) nodes for protection purposes. Including in the statistics other (longer) paths do not make sense at the IP layer. Therefore, the statistics only include the shortest two paths.

We call the shortest path to HL2/1, the *primary path*, and the second shortest path to HL2/1, the *secondary path* (a.k.a. protection or backup path). In addition, we impose the constraint that the physical-layer path of the primary and secondary paths must be disjoint (see Figure 4). Normally the primary path is the shortest (in distance or number of hops) attaining the lower physical impairment degradations (i.e., higher optical signal-to-noise ratio, OSNR). Consequently, in this sub-use-case the worst-case path is the secondary path.

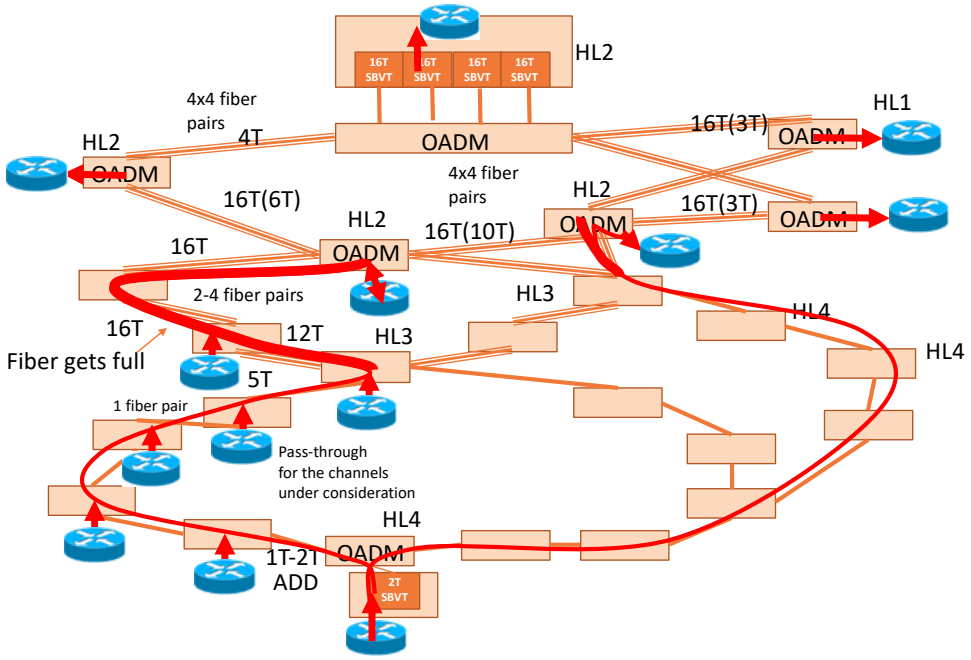


Figure 4. An example illustrating Use Case #2.2: secondary path HL4-HL1/2 (on the right) disjoint to primary path (on the left).

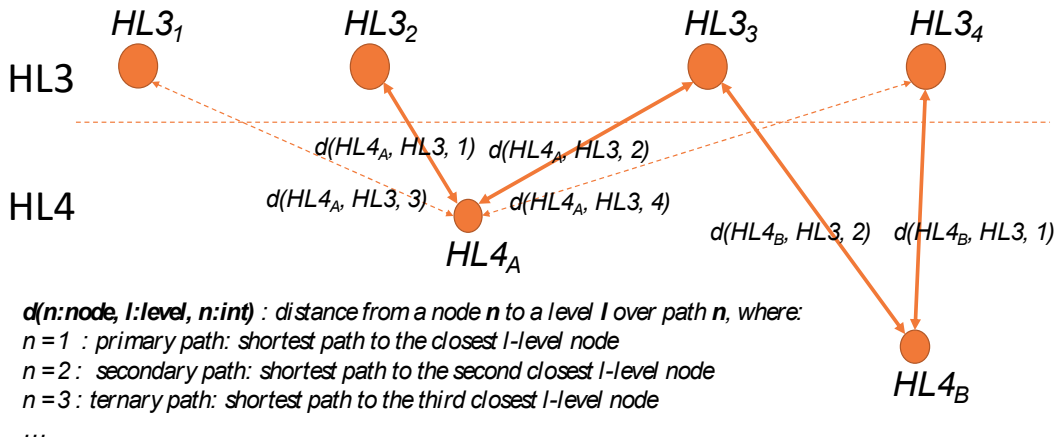


Figure 5. Definition of distance node-level over different paths ordered by topological proximity: example.

Figure 5 shows the definition of distance from a given node to a hierarchical level n. The example shows the distances from node $HL4_A$ to a set of HL3 nodes. Thus, $d(HL4_A, HL3,1)$ is the distance from $HL4_A$ to the closest HL3 node which in the example is $HL3_2$. The path from $HL4_A$ to $HL3_2$. becomes indeed as the primary path from $HL4_A$ to HL3. The secondary path from $HL4_A$ to HL3 is the path towards the second closest HL3 node (in the figure, $HL3_3$) whose distance is denoted by $d(HL4_A, HL3, 2)$. *x* in *d(HLn, HLn-1, x)* gets value 1 or 2 to denote the primary and secondary path respectively.

When the HLs are not contiguous, the top HL reached may or may not be the same. In the following statistics, we consider that the target is reaching a different HL2/1 node to attain node redundancy (sub-use case #2.2 in Figure 6).

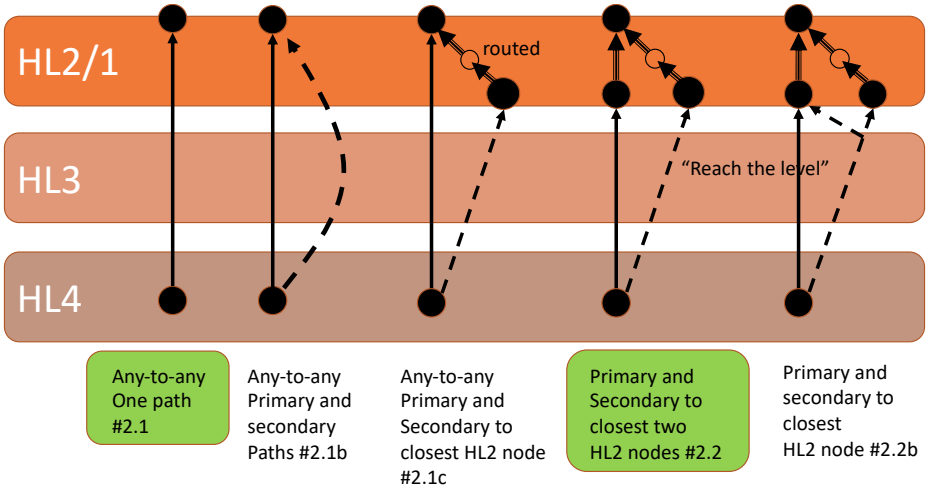


Figure 6. Sub-Use Cases under analysis.

Then, the shortest paths can be calculated based on fiber distance or number of hops. The following tables are an update of the path lengths calculated in [D2.1] and MS4, using the whole topology:

Table 8 Path lengths between a node and its closest (in fiber distance) node of a higher Hierarchical Level (HL): Primary Path

Connected HLs	Mean (Km)	Mean (hops)	Max (Km)	Max (hops)	Min (Km)	Min (hops)	Std (σ) (Km)	Std(σ) (hops)
HL5 - HL4	6.57	5.05	29.66	21	0.55	1	6.04	4.57
HL5 - HL3	14.70	9.48	45.70	24	1.30	2	8.09	4.66
HL5 -HL2/1	31.98	11.38	128.81	29	1.30	2	23.62	4.98
HL4 - HL3	13.23	6.08	33.07	14	0.5	1	7.51	3.54
HL4 -HL2/1	30.20	7.87	121.63	25	0.5	1	22.75	3.94
HL3 -HL2/1	22.63	2.24	104.13	5	2.75	1	23.29	1.30

Table 9 Path lengths between a node and its second closest (in fiber distance) node of a higher Hierarchical Level (HL) : Secondary Path

Connected HLs	Mean (Km)	Mean (hops)	Max (Km)	Max (hops)	Min (Km)	Min (hops)	Std (σ) (Km)	Std(σ) (hops)
HL5 - HL4	6.85	5.17	37.56	20	0.47	1	6.04	4.57
HL5 - HL3	38.04	18.59	44.47	36	9.29	4	8.09	4.66
HL5 -HL2/1	60.23	20.71	35.35	42	7.94	4	23.62	4.98

HL4 - HL3	41.71	15.44	33.07	25	13.91	3	7.51	3.54
HL4 -HL2/1	63.93	17.86	23.95	46	9.19	3	22.75	3.94
HL3 -HL2/1	34.11	5.36	138.74	21	5.09	1	23.29	1.30

When finding a physical layer path to interconnect directly two non-adjacent levels (e.g. HL4 and HL2) that optical path may not necessarily cross the intermediate level (for instance, a minimal HL4-HL2 path may not necessarily cross an HL3 ROADM if the physical layer allows connecting HL4 nodes to HL2 nodes).

Again, the number of hops for HL4-HL2/1 is considerably high (7.1 on average and 12 as maximum) for the secondary path. Therefore, we try the same statistics, minimizing the number of hops, instead of the physical distance next.

Table 10 Path lengths between a node and its closest (in number of hops) node of a higher Hierarchical Level (HL): Primary Path

Connected HLs	Mean (Km)	Mean (hops)	Max (Km)	Max (hops)	Min (Km)	Min (hops)	Std (σ) (Km)	Std(σ) (hops)
HL5 - HL4	6.76	4.97	29.66	21	0.69	1	6.02	4.54
HL5 - HL3	21.12	7.41	71.41	24	1.30	2	14.55	4.51
HL5 -HL2/1	38.37	8.88	141.23	28	1.30	2	26.42	4.64
HL4 - HL3	18.32	2.64	61.45	5	0.5	1	11.69	0.89
HL4 -HL2/1	36.14	4.19	131.27	8	0.5	1	25.19	1.39
HL3 -HL2/1	23.73	1.97	104.13	5	2.75	1	23.09	1.05

In other words, **the worst-case primary path case is 8 hops, total path length: 131Km**.

Table 11 Path lengths between a node and its second closest (in number of hops) node of a higher Hierarchical Level (HL): Secondary Path

Connected HLs	Mean (Km)	Mean (hops)	Max (Km)	Max (hops)	Min (Km)	Min (hops)	Ratio of Non-existing paths
HL5 - HL4	7.01	5.11	29.66	20	0.47	1	526/2546
HL5 - HL3	44.59	11.00	86.72	23	9.11	4	1537/2546

HL5 -HL2/1	77.44	12.92	161.78	26	9.87	5	1548/2546
HL4 - HL3	45.35	5.84	78.7	9	7.70	2	0/380
HL4 -HL2/1	74.70	7.63	148.44	14	16.20	3	8/380
HL3 -HL2/1	43.92	2.5	102.24	5	5.09	1	0/33

In this use case we pay attention to row:

Connected HLs	**Mean (Km)**	**Mean (hops)**	**Max (Km)**	**Max (hops)**	**Min (Km)**	**Min (hops)**	**Ratio of Non-existing paths**
HL4 -HL2/1	74.70	7.63	148.44	14	16.20	3	8/380

which, as a preliminary estimation, could be considered as the target worst case for this use case. In other words, the worst-case secondary path case is 14 hops, total path length: 148.44 Km. PASSION needs to approach this target as much as possible, as it can provide lots of advantages in terms of IP offloading and optical layer multiplexing (i.e., grooming). The purpose is that the average path length must be supported and try to approach as much as possible to the maximum, which may still be an outlier without practical interest. In upcoming deliverables, when a path transmission model is devised it will be possible to estimate what percentile of HL2-HL4 paths can be supported.

2.4 Network Elements and Devices

The PASSION metro network architecture encompasses the PASSION programmable modular system architecture composed of three main network elements:

I. the optical switching and aggregation node,

II. the photonic S-BVT Transmitter (S-BVT Tx) using the cost-effective VCSEL technology,

III. the receiver (S-BVT Rx) relying on a Coherent Receiver (CO-Rx) module (CRM).

The design and implementation of such elements derive from a set of technological capabilities and features to be considered at the time of managing end-to-end optical connections. In the following, it is provided and overview (at a very high level) of the solutions proposed for such network elements within the respective PASSION WPs, namely, WP3 and WP4. Further details on the PASSION modular system architecture, including node and transceiver, can be found in MS6 and [Sva19A]. The consolidated node and transceiver architecture design, including advances on the work developed within WP2, WP3 and WP4, and further analysis of the proposed solution will be addressed in D2.3.

2.4.1 Switching and aggregation Network Nodes

Figure 7 depicts the optical node architecture proposed in the context of the WP4. Detailed characteristics of the optical network architecture are provided in [D41], [D2.1] and MS6. The basic components are:

- Photonic switch module (PSM) providing the optical fiber cross-connection between express in and out as well as add and drop functionalities;
- Aggregate and disaggregate switch;
- Add switch;
- Multi-cast switch (MCS).

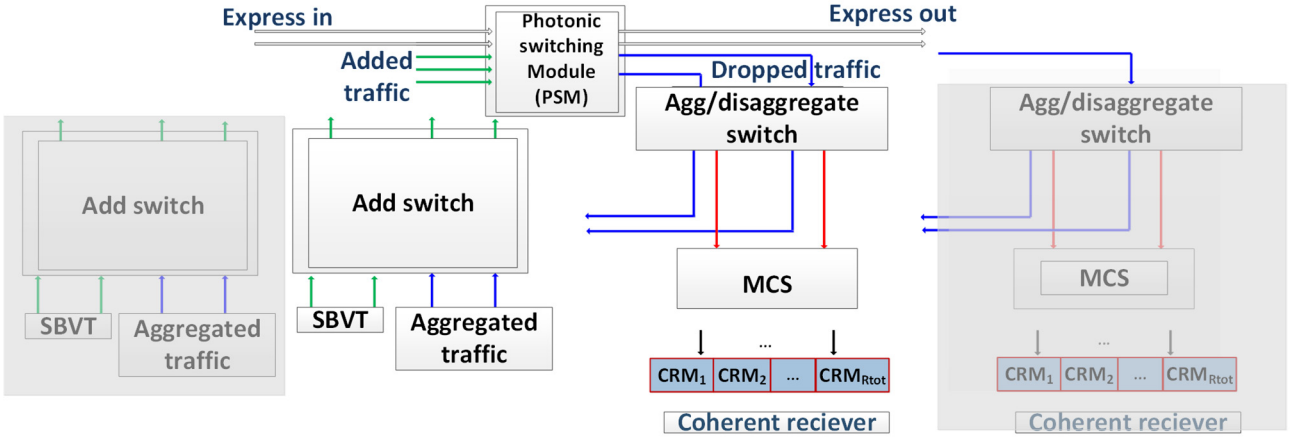


Figure 7. PASSION Optical Node Architecture (HL1/2).

The above optical node architecture is planned to constitute the modular architecture of both HL1/2 nodes and HL3 in the reference metro network (see section 2.1). The difference is that HL3 are not supposed to be equipped with terminating devices such as S-BVTs. In the optical switch architecture, the planned filtering capabilities are provided by WSS of 25 GHz.

On the other hand, the HL4 node architecture is shown in Figure 8. In this optical switch, the filtering capabilities use AWG of 50 GHz.

Depending on the hierarchical level and “pay-as-you-grow” approach, the node is equipped with specific S-BVT modules [MS6].

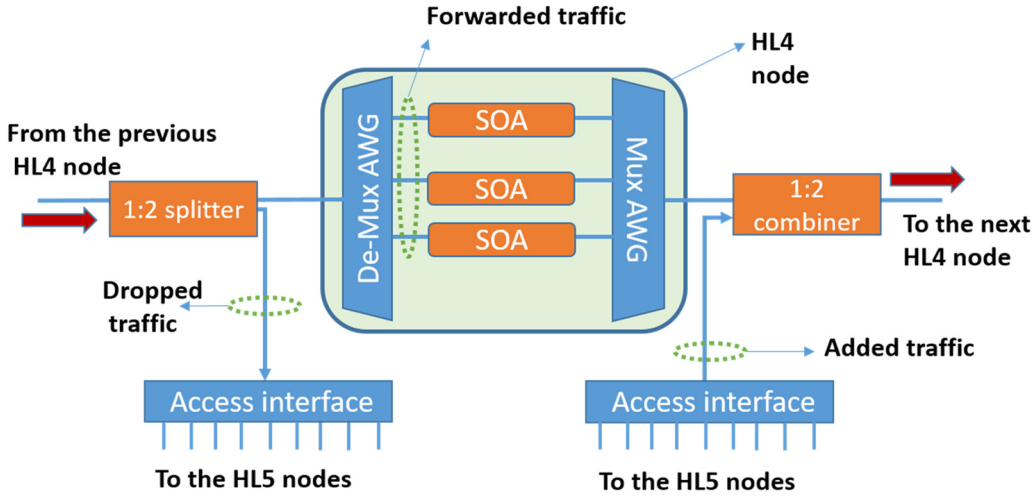


Figure 8. PASSION Optical Node Architecture (HL4).

2.4.2 S-BVT Transmitter Device

The S-BVT Transmitter (Figure 9) architecture provides a modular approach capable of targeting capacities higher than Tb/s per spatial channel, higher than 100 Tb/s per link and Pb/s per node by means of spectral and spatial aggregation [D3.2]. The modular and scalable capabilities of the S-BVT are very appealing to fulfill the requirements of the “pay-as-you-grow” use case.

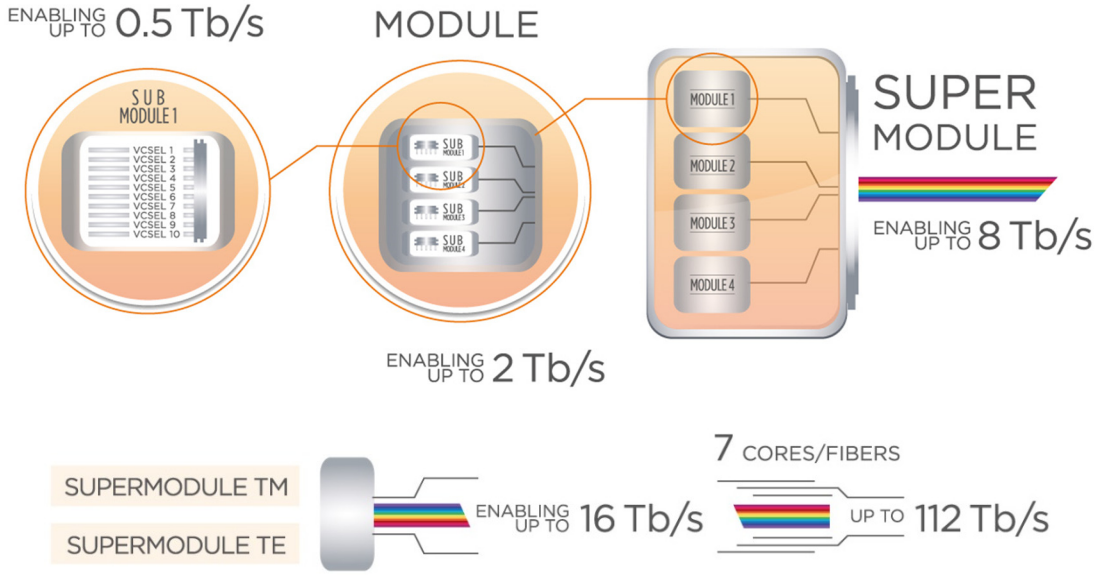


Figure 9. PASSION Modular S-BVT Tx architecture.

The principal element of the S-BVT is the Module. This Module integrates in a silicon-on-insulator (SOI) chip up to 4 SubModules. Each SubModule contains up to 10 VCSELs, where every VCSEL operates up to 50Gb/s. Thus, if all the VCSELS are licensed (i.e., equipped and activated) in all the 4 SubModules within a Module (single) SOI chip, 40 VCSELs operating at different optical carriers (within the C-band in the range 191.900 and 195.875 THz) can transport an aggregated data rate up to 2 Tb/s. It is worth noting that the capability of licensing (activating) VCSELs within a Module does attain higher granularity on the supported data rates offered in a node (e.g., HL4). For instance, a Module could activate only the VCSELs of two SubModules. This results on 20 licensed VCSELs

that provide an aggregated transport capacity up to 1Tb/s. Additionally, the modular design not only allows scaling down (within a single Module) but also scaling up when multiple Modules are combined. For the latter, few Modules (with all the 40 VCSELs being activated) can be used to accomplish larger aggregated capacities (e.g., 8 Tb/s in a SuperModule formed by 4 Modules). This may result very convenient at locations requiring large transport capacity such as HL2/1 nodes. Furthermore, the modular approach can also include the polarization and space dimensions to further increase the capacity of a factor of 2 or equal to the number of available fibers (or multi-core fiber cores), respectively [Sva19B].

2.4.3 S-BVT Receiver Device

Figure 10 depicts the design of the CRM to be considered within the PASSION for receiving coherently the optical flows. The CRM follows the same principles of the S-BVT Tx, that is attaining a modular approach. Specifically, the CRM is made of up to M subModules (integrated as individual PICs) each equipped with a local oscillator (LO). The LO allows detecting and receiving a wavelength in the range between 1528 nm and 1568 nm. The control aspects for the programmability of the S-BVT Rx are reported also in [M4.4].

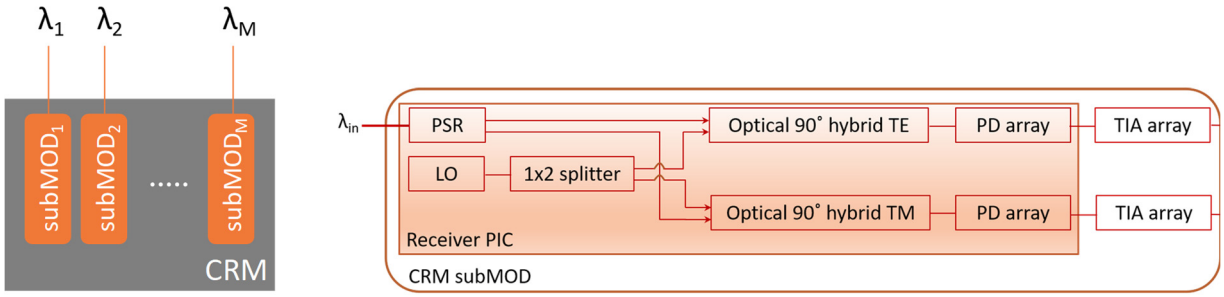


Figure 10. PASSION Coherent Receiver Module.

3 SDN NETWORK PROGRAMMABILITY

One of the key objectives tackled in the PASSION is to enable the automatic management of optical connections dynamically arriving and departing from the network. The establishment and removal of such connections are achieved via the programmability handled by a centralized SDN controller. That is, the SDN controller takes over of the configuration of most of the relevant parameters and features from the underlying network elements and devices forming the metro network.

Nevertheless, the SDN controller does not require having a detailed view of every network element and device characteristics entailing very low-level details and internal architectural aspects. Rather the SDN controller operates at a higher level (i.e., abstract view) where the minimal required configurable parameters of every network element and device are known. This of course does accomplish a more scalable approach from the SDN control perspective. Consequently, for each transport network element and device, it is needed to define its Hardware Abstraction Layer (HAL). Roughly, the HAL provides a high-level vision about the network element and device represented by a specific information model that the SDN controller consumes to conduct the control functions and operations.

In the following, it is discussed the designed HAL for each PASSION network element and device. After this, it is detailed the centralized SDN controller architecture and the supported functions and control interfaces (APIs) with the underlying agents. A description of the control workflow for setting up a new optical connection is also presented. Finally, a description of two devised RSA algorithms to dynamically compute routes is made, considering the features of the PASSION technological solutions.

3.1 HARDWARE ABSTRACTION LAYER (HAL)

The HAL is discussed for the three key transport network elements and devices addressed in section 0. That is, the optical nodes for the different targeted HL (i.e., HL4, HL3 and HL2/HL1), the S-BVT Tx and S-BVT RX. The devised HAL for each of them is done via defining the information model to be used by the SDN controller. In other words, such information model constitutes the basis for designing the protocol and encoding enabling the communication between the SDN controller and the specific agents controlling every network element and device.

3.1.1 HAL for PASSION switching and aggregation network nodes

Figure 11 represents the HAL information for a PASSION network node / optical switch, that is HL1/2, HL3, and HL4. As mentioned above, low-level and internal aspects (e.g., circuitry, mux/demux, etc.) are hidden and not exposed to the SDN controller.

The network node HAL provides basic information about the switch connectivity encompassing the set of ingress and egress physical interfaces / ports. Three types of ports (portType) are defined:

i) Add (labeled as Tx in Figure 11). These ports are attached to the transmission client device (S-BVT Tx) to enter the optical signal towards the network.

ii) Drop (labeled as Rx in Figure 11). These ports are connected to the receiver client device (S-BVT Rx) to receive optical signal arriving from the network.

iii) Express (labeled as I_x and O_x in Figure 11). These ports provide the physical connectivity to other optical switches. In general, both directions are supported for the express interfaces.

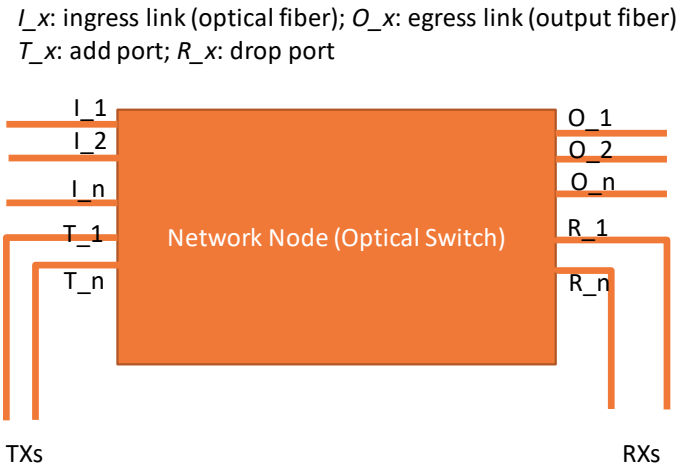


Figure 11. PASSION abstraction view of the Optical Switch Node.

Regardless of the interface/port type, there is a unique identifier per port. This allows unambiguously identifying a specific port when performing any operation (e.g., allocation of optical spectrum, cross-connection, etc.). The identifier is referred to as *portId* and is implemented via an unsigned 32-bit integer (uint32). Additionally, each port may be associated to a name (i.e., *portName*) implemented via a string.

The direction attribute is relevant for each port since it determines whether the specific port is used for Tx (departing from the node), Rx (entering the node) or both. It is implemented via an integer.

For the pool of physical express and add/drop ports besides parameters describing their identifiers, name, etc., there are other attributes related to how optical spectral resources are handled through them. For instance, the filtering capabilities associated to a specific port do impose restrictions to the optical spectrum to be allocated by the optical flows in order to be then adequately filtered and switched. Prior to detail the information model used to describe the optical spectral support per port basis, it is convenient to briefly describe the reference used to specify an optical flow within the flexi-grid optical network [ITU-TG694.1].

An optical flow occupies a frequency slot (FS) which is determined by its central frequency (*n*) and the slot width (*m*). The center frequency granularity (6.25 GHz in flexi-grid) determines the separation between two neighboring nominal central frequencies (NCFs). Therefore, a FS for an optical flow is computed by:

Central Frequency (THz) = 193.1 THz + n * CenterFreqGranurality (THz),

where n is a positive or negative integer and 193.1 THz is the anchor frequency defined in [ITU-TG694.1].

On the other hand, the slot width of the FS for an optical flow is computed using the SlotWidthGranularity . For instance, an optical switch using WSS at 25 GHz (e.g., HL2/1 and HL3) requires SlotWidthGranularity of 25 GHz, whilst a node with AWG at 50 GHz (HL4) uses a SlotWidthGranularity of 50 GHz. The *Slot Width* of a FS for an optical flow is thus determined by:

Slot Width (GHz) = m * SlotWidthGranularity (THz),

where m is positive integer.

For each port it is thus needed to provide information related to the supported NCF and slot width granularities. This information is then considered by the SDN controller to determine an optical flow's FS. Accordingly, the following information per port is provided:

- *total_n*: specifies the number of supported NCFs. For instance, in PASSION the targeted range is 191.900 – 195.900 THz. Therefore, in flexi-grid (6.25GHz CenterFreqGranurality), a total number of 645 NCFs are supported;
- *min_n*: specifies the minimum *n*. Then, for lowest frequency in the range, i.e. 191.900 THz, *min_n* = -192;
- *max_n*: specifies the maximum *n*. Thereby, for the highest frequency in the targeted range, i.e., 195.900 THz, *max_n* = 448;
- *centerFreqGranurality*: specifies the separation between two consecutive NCFs. In PASSION it is set to 6.25 GHz;
- *slotWidthGranurality*: specifies the minimum size of the FS. It depends on the filtering capabilities of each port in an optical switch (i.e., 25 or 50 GHz).

The above information model for the HAL of the optical switch nodes has been encoded (see Figure 12) using JSON format. This encoding is used via a defined REST API enabling the communication between the SDN controller and the agent governing locally each optical switch.

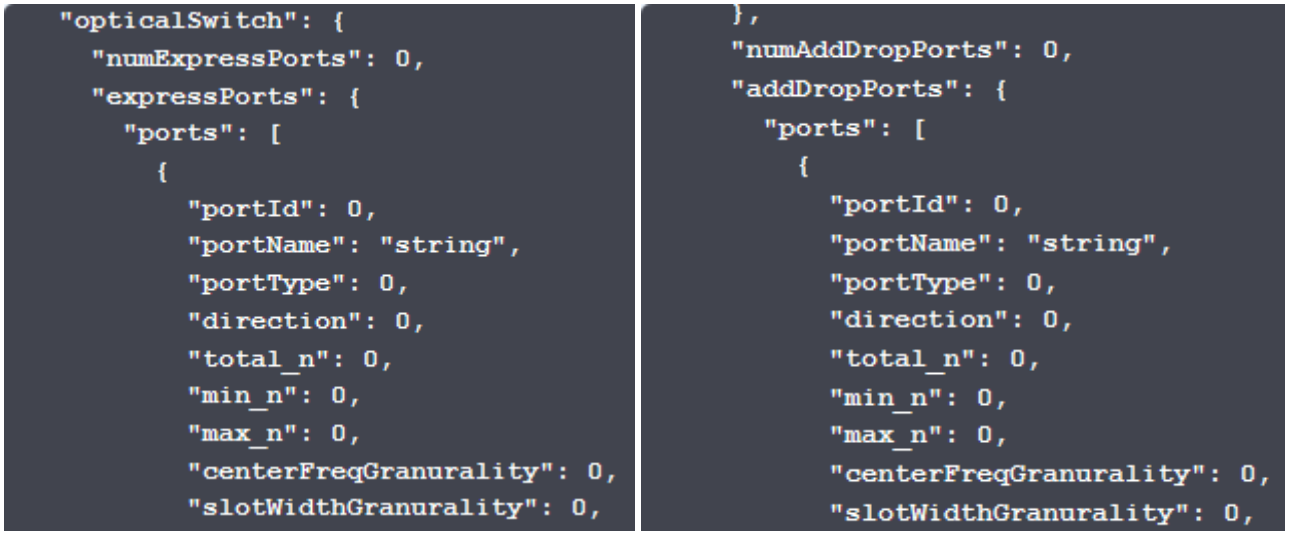


Figure 12. Optical Switch JSON encoding describing the Express Ports (left) and Add/Drop (right).

3.1.2 HAL for PASSION Transmitter

In Figure 13 it is depicted a schematic view of a PASSION SBVT device (involving both Tx and Rx) attached to a network node (i.e., optical switch such as HL4 and HL2/1). Observe that the links between the S-BVT and the optical switch correspond to the add/drop ports discussed in the previous section. As addressed in section 2.4.2, the S-BVT transmitter adopted in PASSION follows a modular approach. That is, a set of Modules (SOI) are arranged into a single S-BVT element. Each S-BVT, depending on the associated HLx node, may have different number of modules: 1, 2, 3 or 4. This provides different amounts of transmission capacities which depend on the activated / licensed VCSELs. The number of equipped Modules in a single S-BVT is determined by *numModulesTx*.

Regardless of the number of Modules per S-BVT Tx, each Module (SOI) has 4 SubModules which integrates 10 VCSELs each. The HAL describing the information model for the S-BVT Tx provides the appropriate identifiers to the SDN controller. By doing so, the controller can program a specific VCSEL or pool of VCSELs at the time of accommodating or releasing optical connections. Therefore, within a S-BVT Tx device, each Module has a unique Id, named as *moduleTxId*. Within each Module, each SubModule has also its own Id, termed as *subModuleTxId*. Finally, within a given SubModule, each licensed VCSEL has its own Id as well, named as *vcselId*. Thus, the tuple formed by *moduleTxId / subModuleTxIdId / vcselId* allows determining a VCSEL to be (de-) allocated within the S-BVT Tx device. In the designed information model, all the identifiers are implemented using unsigned 32-bit integers.

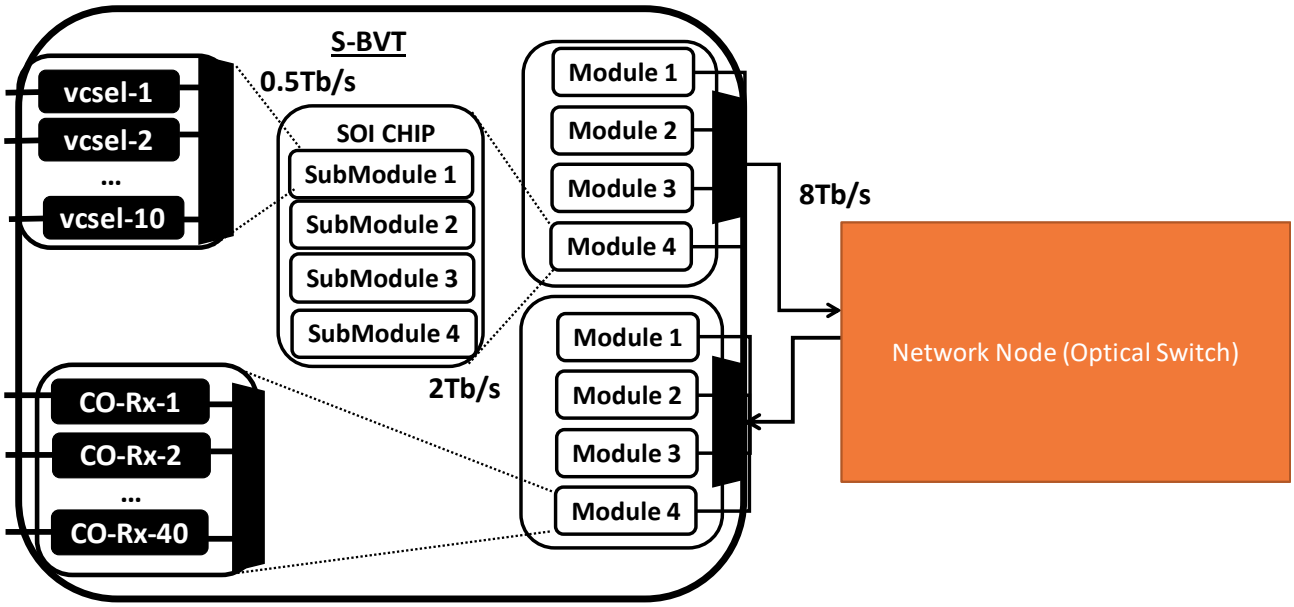


Figure 13. PASSION abstraction view of the SBVT Tx and Rx devices.

The S-BVT Tx device needs to be identified unambiguously within the context of the entire metro network. To do that, the SDN controller refers to a given S-BVT Tx device relying on the *nodeId* of the optical switch attached to the S-BVT and the add port link identifier (see section 3.1.1).

Other relevant attributes of the S_BVT Tx devices known by the SDN controller are:

- *usedState*: it is a Boolean parameter where *True* means that the VCSEL is occupied, whereas *False* determines that the VCSEL is unused.
- *bandwidth*: it is a float type parameter used for specifying the optical spectrum occupied by each VCSEL. In PASSION, this is set to 20 GHz.
- *central-frequency* it is a float value determining the central frequency (optical carrier) associated to a given VCSEL within the PASSION spectral range. Such a central frequency is associated to a given *n* using the expression in section 3.1.1. For instance, a VCSEL with optical carrier at 192.050 THz corresponds to *n*= -168.
- *modulation-format*: it is an integer specifying the modulation format.
- *fec*: it is an integer detailing the FEC type being considered for configuring.

The HAL contents are encoded via JSON format (see Figure 14) which then constitutes the payload of the REST API messages between the SDN controller and the S-NBVT Tx agent.

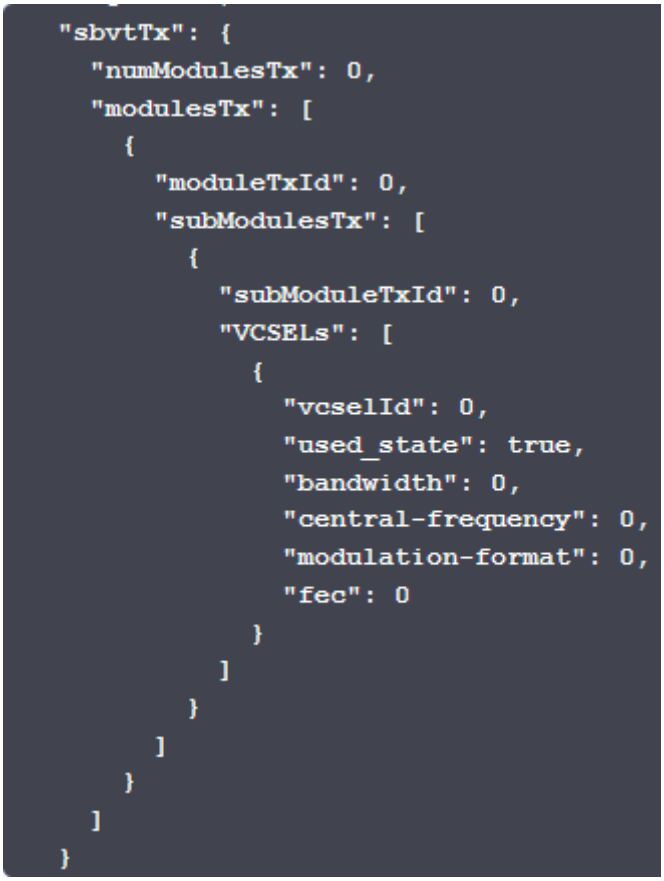


Figure 14. JSON encoding describing the designed SBVT Tx HAL.

3.1.3 HAL for PASSION Receiver

The architecture of the PASSION S-BVT receiver is also modular. A single Module is equipped with a set of CO-Rxs (see section 3.1.3). Figure 13 reflects the view at the SDN controller about the S-BVT Rx device. Observe that an S-BVT Rx may embrace a set of Modules. This number of Modules is determined by *numModulesRx*. Every S-BVT Rx Module has a single identifier (*moduleRxId*). Each Module also has a pool of CO-Rxs determined by *numOpticalReceivers*. A CO-Rx within a Module has its own Id, named as *optReceiverId*. The SDN controller refers to the a given S-BVT Rx using both the *nodeId* of the optical switch attached to the receiver device along with the drop port identifier (i.e., *portId*). Other relevant HAL attributes are:

- *usedState*: it is a Boolean parameter determining whether a CO-Rx is occupied or available.
- *freqLocalOscillator*: it is a float value used to specify whether the central frequency of the LO needs to be tuned. The central frequency is determined using the ITU-T flexi-grid format, that is through the *n* value.

The HAL for the S-BVT Rx is encoded with JSON format (see Figure 15) that constitutes the information to be carried in the REST API. Again, this API is used to communicate the SDN controller with the S-BVT Rx agent for the programmability of the device.

```
"sbvtRx": {
  "numModulesRx": 0,
  "modulesRx": [
    {
      "moduleRxId": 0,
      "numOpticalReceivers": 0,
      "opticalReceivers": [
        {
          "optReceiverId": 0,
          "used_state": true,
          "freqLocalOscillator": 0
        }
      ]
    }
  ]
}
```

Figure 15. JSON encoding describing the designed SBVT Rx HAL.

3.2 Architecture of the SDN Controller: Interfaces and Workflow

The functional representation of the SDN controller architecture used in PASSION project is depicted in Figure 16. The controller is responsible for coordinating the set of automatic operations to configure the underlying network elements and devices. These operations are:

I. Process and serving incoming connection demands arriving from an external application (herein it is called "*On-demand bandwidth Application*"). This may be considered as a business application running in the operations support system (OSS) and/or business support system (BSS) of the operator owning the network infrastructure.
II. Computing end-to-end paths and selecting resources to be configured on the transport network. This entails triggering an on-line path computation process.

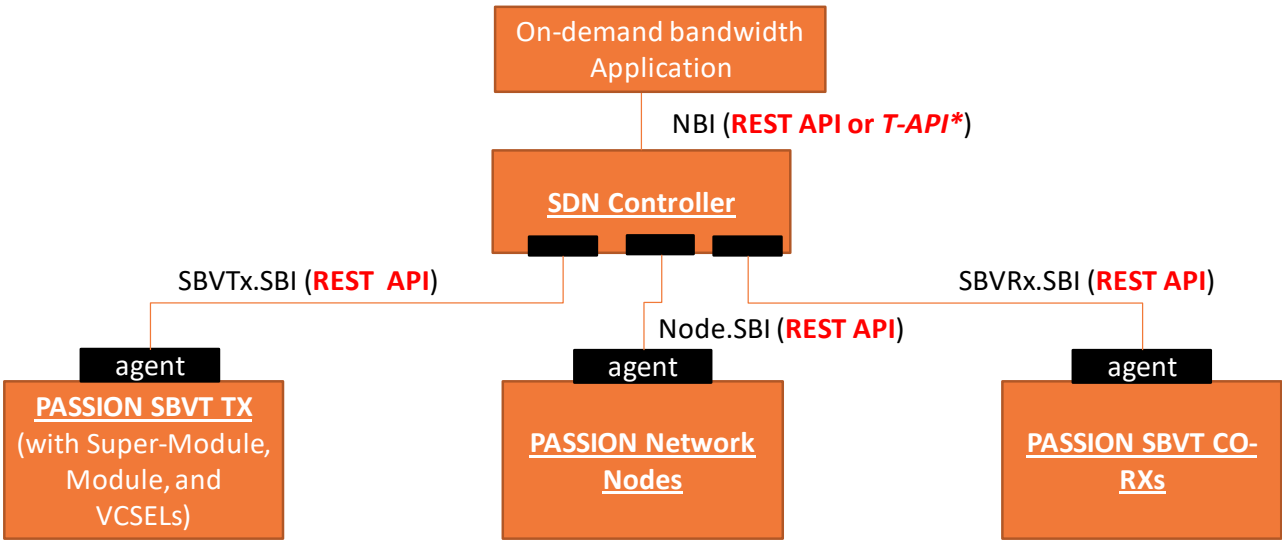


Figure 16. Schematic view of the PASSION SDN control architecture.

To conduct these operations, the SDN controller is made up of a set of building blocks taking over of the control functionalities. A representation of these pool of functions is depicted in Figure 17.

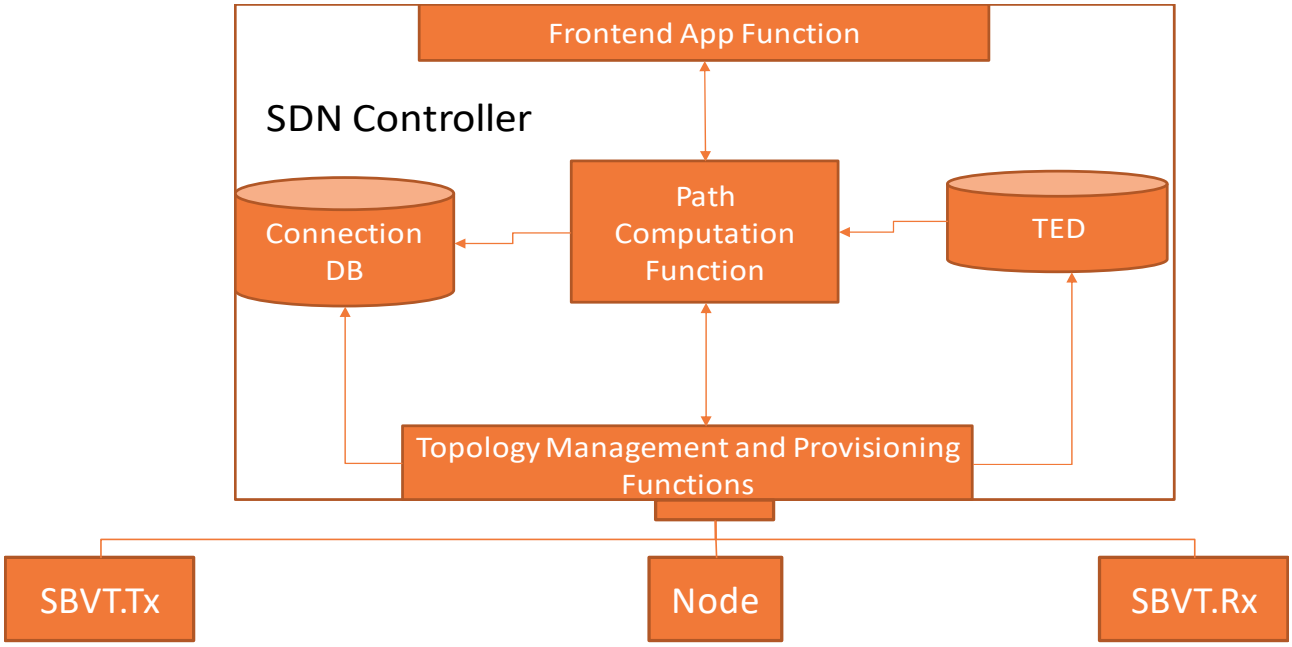


Figure 17. SDN Controller key building blocks and functions.

In a nutshell, there is a *Frontend App Function* which receives and processes the incoming requests (e.g., create and delete an end-to-end optical connection) arriving from an application. The *Path Computation Function* is the element coordinating most of the actions (e.g., triggering the end-to-end path computation) and operations (i.e., instantiating a new connection to be set up, etc.). To do that two repositories contain relevant information enabling the execution of the above Path Computation Function actions and operations. One repository stores information about the network connectivity (i.e., topology) along with the status of the available resources (e.g., optical link spectrum, available VCSELs at the SBVT Tx and Co-Rxs at the S-BVT Rx, etc.). Such a database is referred to as the Traffic Engineering Database (TED). The second repository keeps track of the optical connections existing within the network and is called as Connection DB. The connection DB stores information relevant to the connections such as its name (to identify the connection), the endpoints, as well as the set of optical flows. It is worth noting, as it will be discussed in section 3.3, that a connection may demand larger bandwidth than the supported data rate by a single VCSEL (set to up to 50 Gb/s). Thus, if an optical connection requests a bandwidth > 50 Gb/s, multiple optical flows related to the same connection are set up. Each optical flow is deployed allocating its own VCSEL and CO-Rx at the endpoint SBVTs.

Finally, the Topology Management and Provisioning Function provide the means to communicate with the underlying network elements and devices (i.e., agents). This allows, on the one hand, retrieving updated status of each network element and device (then it is stored in the TED), and, on the other hand, executing the management (programmability) of the end-to-end optical connections.

In addition to the SDN controller key building blocks, there are two controller interfaces: NBI and SBI. These provide the communication with the On-Demand Bandwidth Application (via NBI) and the network elements' and devices' agents (via SBI). The selected implementation of both interfaces is based on REST APIs using JSON-encoded payload. In the following, it is provided a description of each of these interfaces (in terms of protocol and encoding) which will be validated in the WP5.

3.2.1 NorthBound Interface (NBI)

The NBI is the API supporting the interworking between the On-Demand Bandwidth Application and the SDN controller. It deploys a client-server relationship via a TCP connection based on a REST interface. Such an API defines:

i) the request of a new end-to-end unidirectional optical connection between SBVTs at HL4 and HL2/1 nodes (and vice versa);
ii) the removal of an existing optical connection.

The design of the NBI API has been done leveraging an on-line (web) editor called Swagger [swagger]. This application allows creating the whole skeleton of a RESTful API. That is, it provides the deployment of both the client and the server source code as well as defining the message payloads (encoded in JSON format).

In the devised targeted PASSION NBI, there are two defined messages/methods (shown in Figure 18):

- POST (URI: /rest/api/v1/lsp): This message is sent by the On-Demand Bandwidth Application to the SDN controller whenever a new optical connection is requested. The connection is also referred to as label switched path (LSP). As said before the payload of the message body are encoded via JSON format and are shown in Figure 19. These contents are:
    - *id*: It is a string parameter specifying the name of the connection being requested.
    - *src*: a string associated to the *nodeId* of the optical connection ingress (i.e., HL4 or HL2/1).
    - *dst*: a string associated to the *nodeId* of the optical connection egress (i.e., HL4 or HL2/1).
    - *bw*: it specifies via a string the amount of bandwidth being requested.
    - *bw_unit*: it provides the units of the amount of bandwidth (*bw*) required by the optical connection, for instance Gb/s.
    - *of*: it provides the Objective Function value (string) identifying a specific RSA algorithm among a pool of candidate ones within the SDN controller. By doing so, the On-Demand Bandwidth Application informs the controller about the specific requirements / objectives to be guaranteed for setting up the optical connection such as simple provisioning, pre-computed restoration path, etc.

  If the connection is successfully computed and established, the SDN controller returns to the On-Demand Bandwidth Application an HTTP response with the code set to 201 (Created). Otherwise, if the connection cannot be set up (e.g., due to the lack of resources), the response code is 404 (Not found).

- DELETE (URI: /rest/api/v1/lsp/{id}): This message informs the SDN controller that an existing connection is to be removed de-allocating the associated resources such as S-BVT Tx VCSELs, S-BVT Rx CO-Rx and optical spectrum over the traversed optical switches. The *id* parameter in the URI refers to the specified connection name.

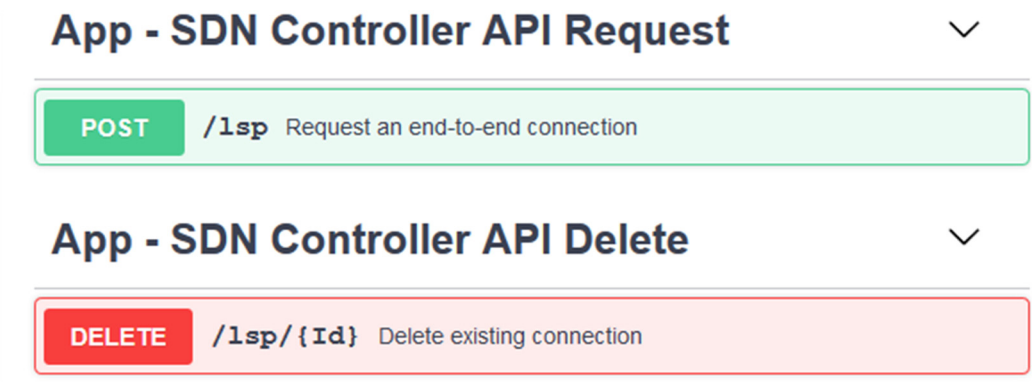


Figure 18. Swagger Editor: NBI API methods (POST and Delete) between On-Demand Bandwidth Application and SDN controller.

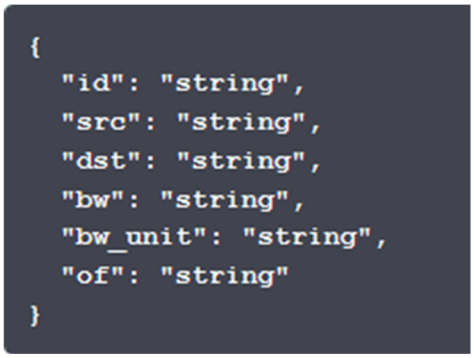


Figure 19. JSON payload of the NBI POST method requesting an optical connection.

The NBI design will be validated in the WP5. That said, at the time of writing this deliverable, another solution is being explored relying on the standard Transport API (TAPI) [TAPI]. For the latter, specific extensions are being investigated to accommodate the particularities of the PASSION technology. Finally, the definition of the NBI using the RESTful interface using swagger is provided in Annex 1.

3.2.2 SouthBound Interface (SBI)

The SBI allows the SDN controller to program the underlying network resources into the different elements and devices. As shown in Figure 16, three different defined SBIs between the SDN controller and the agents governing the SBVT Tx, SBVT Rx and optical switches (HL4, HL3 and HL2/1) exist. The SBI creates client server relationships relying on RESTful API where the message payload is encoded using JSON format. Next, it is provided a description of each of the designed SBIs. For the sake of completeness, the definition of the RESTful-based SBIs using swagger tool is provided in Annex 2.

3.2.2.1 SBI for the SBVT Tx

This specific SBI addresses the configuration operations to the S-BVT Tx device. To this end, it is defined a set of methods for:

i) retrieving the status of the Modules/SubModules/VCSELs;
ii) allocating a selected set of VCSELs to accommodate a specific connection request;
iii) releasing occupied VCSELs by a given active optical connection.

Using the swagger editor, the defined methods for this SBI are depicted in Figure 20.

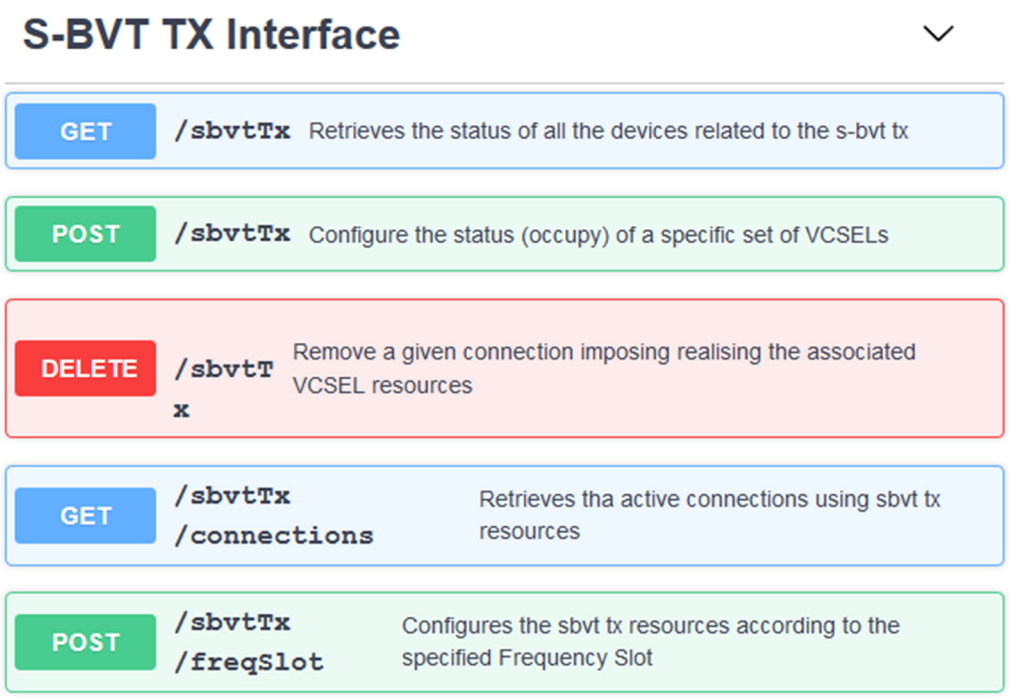


Figure 20. Swagger Editor: SBI API methods (GET, POST and Delete) between SDN controller and the agent of the SBVT Tx.

The following provides the features of the supported methods:

- GET (URI: /passion/sbi/sbvtTx): This method is used by the SDN controller to gather information about deployed Modules / subModules / VCSELs by a SBVT Tx. The contents (formatted in JSON) are those discussed in section 3.1.2, which are returned to the SDN controller by the agent within the payload of the HTTP response. The information is then stored in the SDN controller's TED repository and used whenever a new connection request is processed.

- POST (URI: /passion/sbi/sbvtTx): This message allows the SDN controller allocating a specific VCSEL (or set of VCSELs) within a S-BVT Tx device. To this end, the tuple Module / SubModule / VCSEL Ids introduced in section 3.1.2 need to be specified. Additionally, it is also indicated the name of the connection occupying that VCSEL (or set of VCSELs) resources. The name of the connection is carried into the JSON-encoded payload of the POST method via the string referred to as *connectionId*. The successful response (i.e., resources are allocated) is notified by the agent via the HTTP response code 201 (Created). Otherwise, if the selected resources are not available or not found, the response code is 403 (Forbidden) or 404 (Not found).

- DELETE (URI: /passion/sbi/sbvtTx): An optical connection being removed entails de-allocating their resources. This message allows SDN controller releasing S-BVT Tx (VCSELs) associated to a specific *connectionId*. Thus, the payload of the message only contains this connection identifier which allows the agent to determine the VCSEL (or set of VCSELs) being used. The HTTP reply message indicates successful removal operation via a response code set to 200 (Ok). Otherwise the response code is 404.

- GET (URI: /passion/sbi/sbvtTx/connections): This message is used by the SDN controller to gather all the connections allocating S-BVT Tx resources. In other words, it is retrieved the set of existing connections (i.e., *connectionId*s) using VCSELs at the specific S-BVT Tx. The

set of connections are returned in the payload of the HTTP response message with an array having the *connectionId*s. In Figure 21 it is depicted the JSON-encoded contents where *numActiveConnections* is the integer determining the number of existing connections, whilst the *setActiveConnections* is the array of strings providing the *connectionId*s.

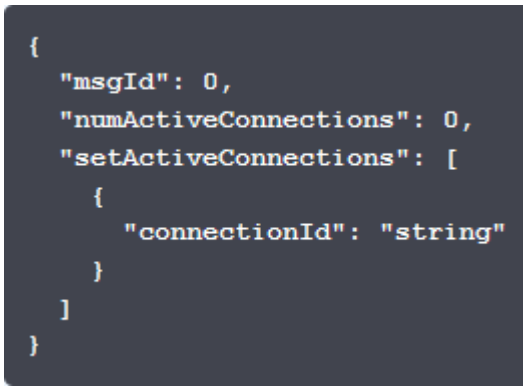


Figure 21. SBI SBVT Tx: JSON payload listing the active connections allocating SBVT Tx resources.

- POST (URI: /passion/sbi/sbvtTx/freqSlot): This message allows the SDN controller allocating a selected VCSEL (or set of VCSELs) without specifying the tuple Module /submodule / VCSEL Ids. Indeed, since VCSELs in a S-BVT Tx are bound to an individual optical carrier, this message provides the center frequency/ies (i.e., *n*) of the VCSEL/s to be occupied. The agent upon receiving the message can unambiguously determine the VCSEL/s associated to a *n* value. The JSON contents carried by this POST message are shown in Figure 22.
    - *connectionId*: it is a string specifying the connection allocating the VCSEL/s resources;
    - *sbvtTxFreqSlot*: an array of JSON objects for each VCSEL being allocated. The parameters forming that object are:
        - *centerFreq_n*: it is a (positive or negative) 16-bit integer specifying the center frequency (i.e., *n*) according to ITU-T flexi-grid recommendation. Such a value is indeed bound to the optical carrier of a single VCSEL within the S-BVT Tx device.
        - *slotWidth_m*: it is a positive 16-bit integer determining the slot width of the frequency slot being allocated following the ITU-T flexi-grid recommendation. For the VCSEL allocation in a S-BVT Tx this information is not considered for the time being.
        - *used_state*: It is a Boolean parameter which is always set to True indicating that the VCSEL associated to *n* must be occupied.
        - The other parameters, i.e. bandwidth, modulation-format and fec, have the same meaning as discussed in section 3.1.2

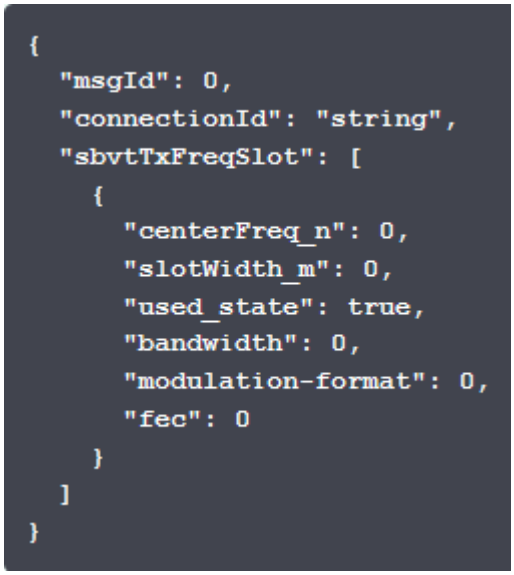


Figure 22. SBI SBVT Tx: JSON payload allocating SBVT Tx VCSEL using the center frequency.

3.2.2.2 SBI for the SBVT Rx

Like the SBI for the S-BVT Tx, the SDN controller uses a specific SBI for managing the resources within the S-BVT Rx device. The supported operations are:

i) To retrieve the status of the Modules and of their optical receivers (CO-Rxs);
ii) To allocate and configure a selected set of CO-Rxs to accommodate a connection demand;
iii) To release occupied CO-Rxs by a given active optical connection.

Using the swagger editor, the defined methods for this SBI are depicted in Figure 23. The following provides features of the supported methods:

- GET (URI: /passion/sbi/sbvtRx): This method is used by the SDN controller to gather information about deployed set of Modules and the set of *opticalReceivers* within the S-BVT Rx. The response message to the GET method contains, into the JSON-encoded payload, the information as described in section 3.1.3. This information allows the SDN controller being aware of not only the available CO-Rxs in each individual Module, but also the central frequencies programmed on every occupied CO-Rx. The latter is essential to avoid double-booking of an already in-used optical carrier within the S-BVT Rx device. In other words, the LO of every CO-Rx can be tuned, but it cannot be configured with a central frequency being programmed by another CO-Rx within the same S-BVT Rx device.

- POST (URI: /passion/sbi/sbvtRx): This message allows the SDN controller allocating a specific CO-Rx (or set of CO-Rxs) within a S-BVT Rx device. This is done indicating the tuple made up of *ModuleRxId* / *optReceiverId* / *freqLocalOscillator* according to the JSON encoding described in section 3.1.3. Furthermore, it is also added the *connectionId* specifying the connection occupying the resources. The successful response (i.e., resources are allocated) is notified by the agent via the HTTP response code 201 (Created). Otherwise, if the selected resources are not available our found, the response code is 403 (Forbidden) or 404 (Not found).

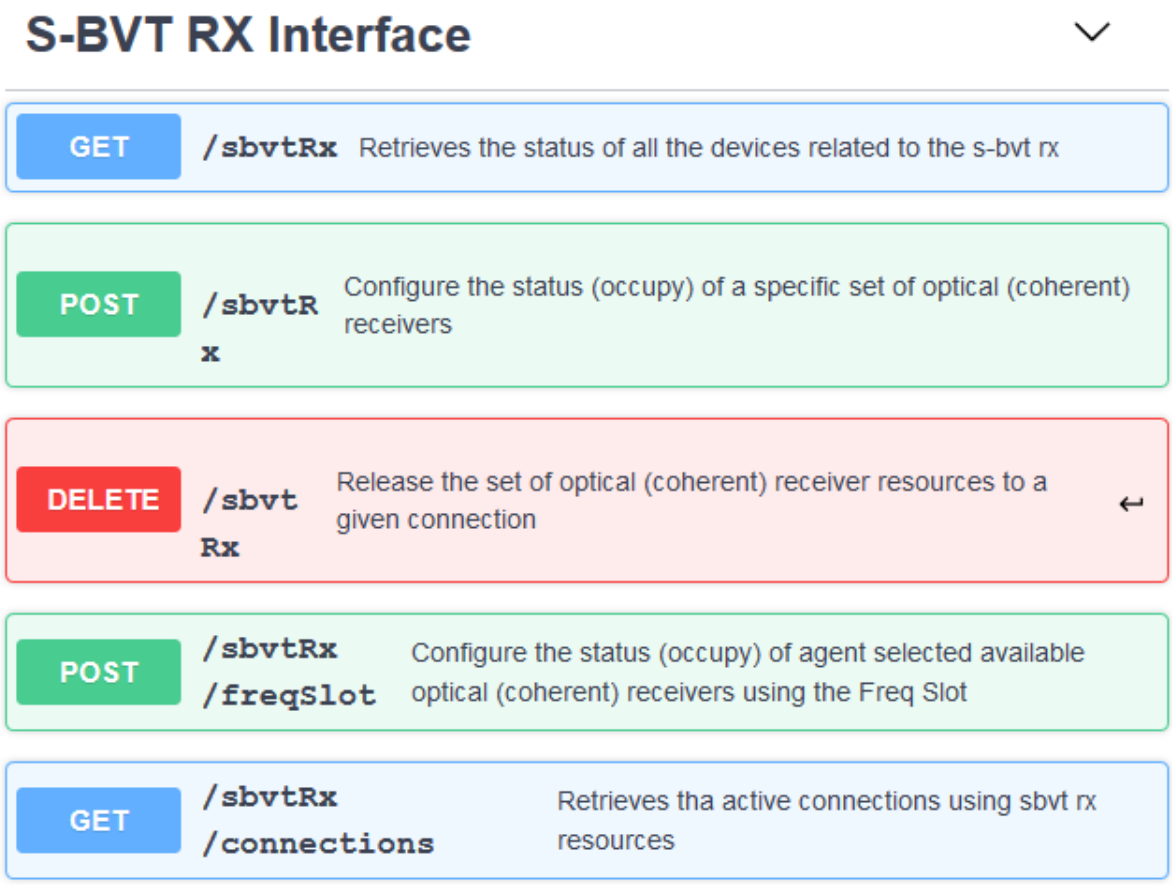
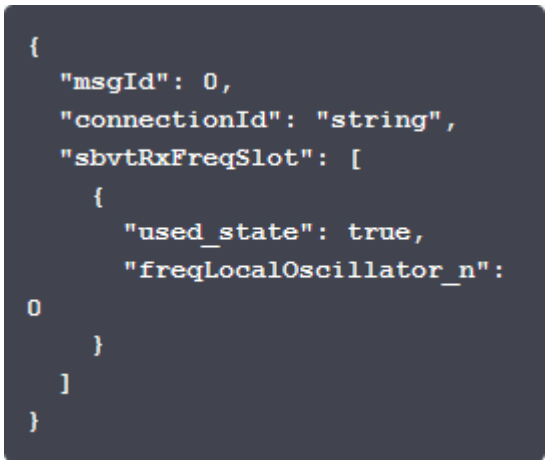


Figure 23. Swagger Editor: SBI API methods (GET, POST and Delete) between SDN controller and the agent of the SBVT Rx.

- DELETE (URI: /passion/sbi/sbvtRx): This method allows the SDN controller releasing S-BVT Rx (CO-Rxs) associated to a specific *connectionId*. Thus, the payload of the message only contains this identifier enabling the agent to determine internally the CO-Rx (or set of CO-Rxs) used by such a connection. The HTTP reply message notifies successful operation with response code set to 200 (Ok). Otherwise the response code is 404.
- GET (URI: /passion/sbi/sbvtRx/connections): This message is used by the SDN controller to gather all the connections allocating S-BVT Rx resources. This set of connections is returned in the payload of the HTTP response message with an array having the *connectionId*s as shown in Figure 21.
- POST (URI: /passion/sbi/sbvtRx/freqSlot): This message allows the SDN controller allocating a selected CO-Rx (or set of VCSELs) regardless of the Module they are associated to. The idea is that the agent locally selects available CO-Rxs to be assigned to the *connectionId*. Nevertheless, for each of the required CO-Rx, the message informs about the LO frequency relying on the recommendation from the ITU-T flexi-grid (i.e., *n*). The agent, upon receiving the message, chooses the unused CO-Rxs and configures the targeted LO frequencies. The JSON contents carried by this POST message are shown in Figure 22.
    - *connectionId*: it is a string specifying the connection allocating the CO-Rx/s resources.
    - *sbvtRxFreqSlot*: an array of JSON objects for each CO-Rx to be allocated. The parameters forming this object are:

- *used_state*: It is a Boolean parameter which is always set to True indicating that the VCSEL associated to *n* must be occupied.
- *freqLocalOscillator_n*: it is a (positive or negative) 16-bit integer specifying the center frequency (i.e., *n*) according to ITU-T flexi-grid recommendation to be tuned by the local oscillator being occupied.



```
{
  "msgId": 0,
  "connectionId": "string",
  "sbvtRxFreqSlot": [
    {
      "used_state": true,
      "freqLocalOscillator_n":
0
    }
  ]
}
```


Figure 24. SBI SBVT Rx: JSON payload allocating SBVT Rx CO-Rx with specific local oscillator to be tuned.

3.2.2.3 SBI for the Optical Switches

This SBI is designed to configure the optical switching nodes (i.e., HL4, HL3 and HL2/1). The objective is to provide the cross-connection between add/drop and express ports to set up the end-to-end optical flows between the S-BVT Tx and S-BVT RX at the connection endpoints. Specifically, the set of commands being supported by this SBI are:

i) retrieving the inventory of the optical switch in terms of express, add and drop ports along with the supported central frequencies covered by the filtering capabilities;
ii) collecting the set of active optical connections traversing the optical switch;
iii) creating the frequency slot cross-connection for a specific optical connection request;
iv) removing the cross-connection associated to an active optical connection.

Relying on the swagger editor, the defined methods for this SBI (depicted in Figure 25) are:

- GET (URI: /passion/sbi/opticalSwitch): This method is used to gather details of the optical switch in terms of the number of add, drop and express ports. Moreover, for each port, it is provided the optical spectrum support related to the filtering capabilities. This entails information such as the supported granularity of the NCFs and the slot width. The information is carried into the HTTP response message payload and it is formatted (JSON-encoding) as shown in Figure 12.

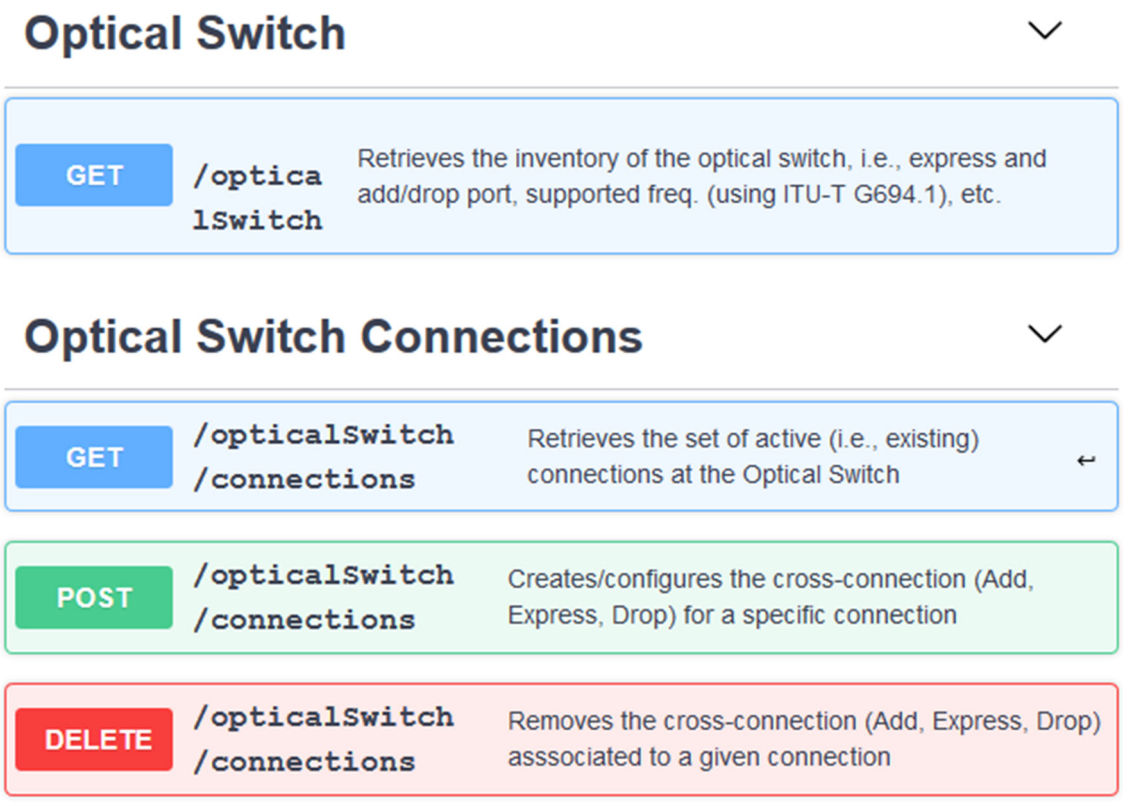


Figure 25. Swagger Editor: SBI API methods (GET, POST and Delete) between SDN controller and the agent of the Optical Switch (HLx).

- GET (URI: /passion/sbi/opticalSwitch/connections): This message allows the SDN controller retrieving for each active connection traversing the optical switch its details about the allocated optical switch resources, i.e. ports and FS. This information is provided in the HTTP response message where the specific JSON-encoded contents are shown in Figure 26. The parameters in such a message are:
    - *numActiveConnections*: it is 32-bit integer determining the total number of active connections traversing the optical switch
    - *setActiveConnections*: it is an array containing on each element an object describing specific attributes of the active connections:
        - *connectionId*: it is string with the name associated to a given connection
        - *crossConnection*: it is a JSON object used to provide details about the optical cross-connection associated to the *connectionId*. Such details / attributes are:
            - *portIn*: It is a 32-bit integer determining the ingress port identifier (i.e., add or express) traversed by the *connectionId*
            - *portOut*: it is a 32-bit integer determining the egress port identifier (i.e., drop or express) traversed by the *connectionId*
            - *centerFreq_n*: it is a 16-bit integer defining the ITU-T flexi-grid encoded central frequency (i.e., *n*) of the frequency slot allocated by the *connectionId*.

- *slotWidth_m*: it is a 16-bit integer defining the ITU-T flexi-grid encoded slot width (i.e., *m*) of the frequency slot allocated by the *connectionId*.

Figure 26. SBI Optical Switch: JSON payload retrieving optical switch allocated resources for all the active connections traversing such as node.

- POST (URI: /passion/sbi/opticalSwitch/connections): This message allows the SDN controller specifying in the request message the spatial (i.e., *portIn* and *portOut*) and spectral (i.e., frequency slot, *n* and *m*) cross-connection to be programmed within the optical switch. The payload of the request message has the same format as shown in Figure 26. The successful configuration of the optical switch is sent via a HTTP reply message with response code set to 201 (Created).
- DELETE (URI: /passion/sbi/opticalSwitch/connections): This message is used by the SDN controller when all the (spatial and spectral) resources occupied by a given connection (i.e., *connectionId*) within a node are released. The payload of the request message simply contains the *connectionId*. The successful removal of the resources as well as removal of connection as an active connection is notified by a HTTP reply having response code to 200 (Ok).

3.2.3 Workflows for Connection Management

The interactions among the SDN controller and the agents governing each network element and device within the PASSION metro network are ruled by a sequence of operations and exchanged control messages (NBI and SBIs). The workflow detailing the messages and the involved SDN control functions and agents are depicted in Figure 27.

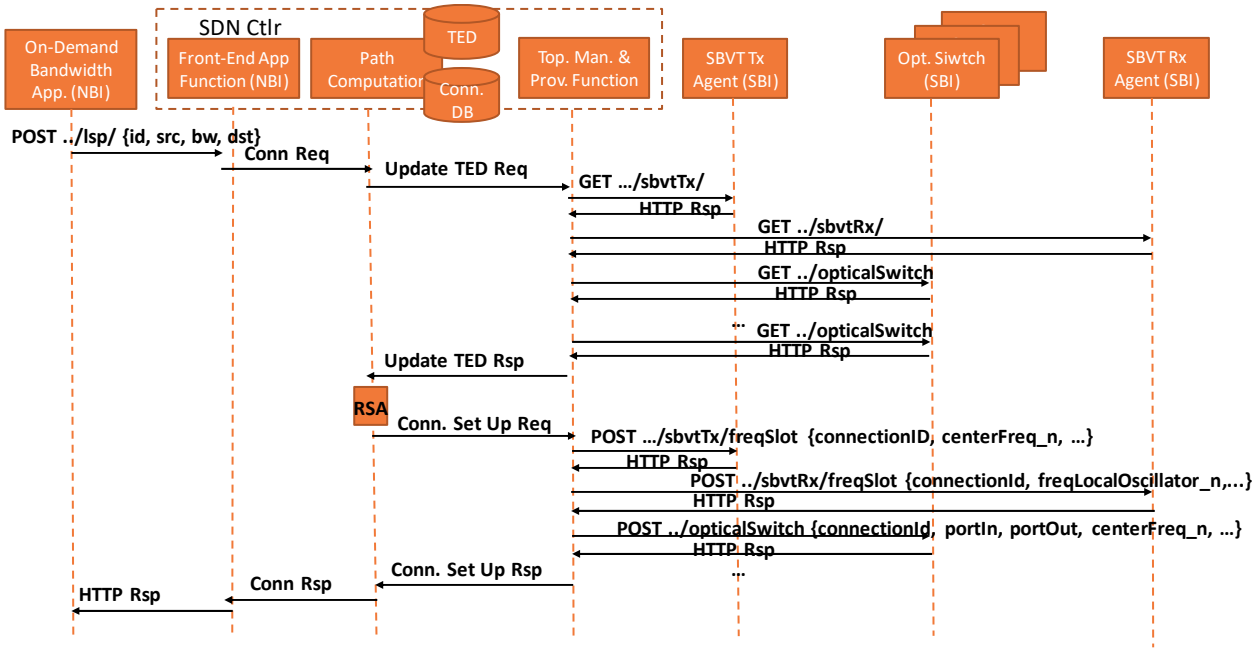


Figure 27. Workflow for setting up a new optical flow using defined NBI and SBI APIs

An optical connection is requested by the On-Demand Bandwidth Application to the SDN controller using the NBI POST message as described in section 3.2.1. This includes the *connectionId*, the source and destination nodes and the requested bandwidth. It is important to recall that the source and destination nodes are bound to both S-BVT devices, i.e. transmitter (Tx) and receiver (Rx).

The connection request (NBI) message is then received by the SDN controller at the Front-End App Function. Internally within the controller, the Path Computation function is triggered via the Connection Request process. At this point, the Path computation updates its network resource view within the TED repository. This is done by the Topology Manager. Indeed, the Topology Manager requests via the defined SBIs:

i) the resource availability of the S-BVT Tx devices (VCSELs) at the source node using the GET /…/sbvtTx message;
ii) the resource status of the S-BVT Rx elements (CO-Rxs) at the destination node using the GET /…/sbvtRx) message;
iii) the status of the spectral resources on each optical switch port. It means to collect the information of all the optical switch agents within the network. This is supported via the GET /…/opticalSwitch message.

All HTTP responses are processed by the Topology Manager and the information of those messages stored into the TED.

Once the TED is updated, the Path Computation Function triggers the routing algorithm. The routing algorithm uses as inputs the request's attributes (Conn. Req) and the updated TED. The output of the routing algorithm provides:

i) the set of ordered network nodes to be traversed (detailing the input and output physical ports). This is referred to as spatial path;
ii) the selected FS (i.e., optical spectrum) to accommodate the optical flow;
iii) the configurable parameters of both S-BVT Tx and S-BVT Rx such as the selected set of VCSELs and the CO-Rxs and their central frequencies (i.e. LO).

The programmability of each selected network element and device represented in the computed spatial and spectral paths is carried out separately using the defined SBIs. Specifically, the controller's Provisioning Function communicates with the specific network element and device agents. In the S-BVT Tx device, the VCSEL associated to the central frequency *centralFreq_n* is occupied and assigned to the specified *connectionId*. This is done using the POST /…/sbvtTx/freqSlot message. Similarly, for the S-BVT Rx, the POST /…/sbvtRx/freqSlot message indicates to the corresponding agent to first select an available CO-Rx and then program its LO to be tuned at the specified central frequency *freqLocalOscillator_n*. Finally, for each traversed optical switch (i.e., HL4, HL3 and HL2/1) in the spatial path, an individual POST /…/opticalSwitch message is sent indicating the required cross-connection via *portIn*, *portOut*, *centerFreq_n*, etc.

Once the configuration of all the network elements and devices in the computed path is successfully done, the connection is established and thus added to the controller's Connection DB. Additionally, the successful connection establishment is notified to the On-Demand Bandwidth Application via the corresponding NBI-defined HTTP response message.

For the sake of completeness, the workflow along with the protocols and encodings within both designed SBI and NBI are validated in WP5. The deletion workflows of active connections will be discussed in the context of WP5 derived documents.

3.3 On-line Routing Mechanism

This section focuses on describing the requirements and constraints to be considered at the time of executing on-line routing algorithms upon receiving a new optical connection request. To this end, it is needed to explicitly consider the capabilities imposed by the underlying network elements and devices as well as the restrictions / limitations derived from the adopted technological solutions in PASSION. The discussion of the devised routing algorithms within this deliverable is done to exclusively cover the Use Case #1 ("pay-as-you-grow") presented in section 2.2. The algorithms to be applied for covering the other use case "on-line restoration" are not tackled at the time of writing this deliverable. They will be explored in subsequent activities within both WP2 and WP5.

The routing algorithm is executed within the SDN controller's Path Computation Function with the following inputs:

i) Connection request parameters (i.e., source, destination, bandwidth)

ii) Topology and resources information available in the controller's TED repository.

Typically, in optical flexi-grid networks, routing algorithms are referred to as Routing and Spectrum Assignment (RSA) algorithms. Such RSA algorithms target a specific set of performance objectives (e.g., improve the average resource usage) and must satisfy a set of constraints mostly derived from the underlying transport technology. In PASSION, the RSA restrictions to be accounted are:

- VCSEL limited frequency tunability
- Maximum "net" capacity achieved by S-BVT Tx VCSEL devices depending on the path distance (in km) and number of traversed network nodes.
- Filtering capabilities (either 50GHz or 25 GHz) of the -heterogeneous HL4, HL3 and HL2/1 optical switch.
- Ensuring the end-to-end optical spectrum continuity and contiguity through the entire path.

In the explored Use Case #1, the RSA performance objective function is related to dynamically provision optical connections attaining an efficient use of the overall network resources. It is done aiming at maximizing the likelihood of setting up future arriving optical connections requests. Figure

28 sketches the set of requirements (constraints and objective functions) and inputs/outputs addressed for devising the RSA algorithms.

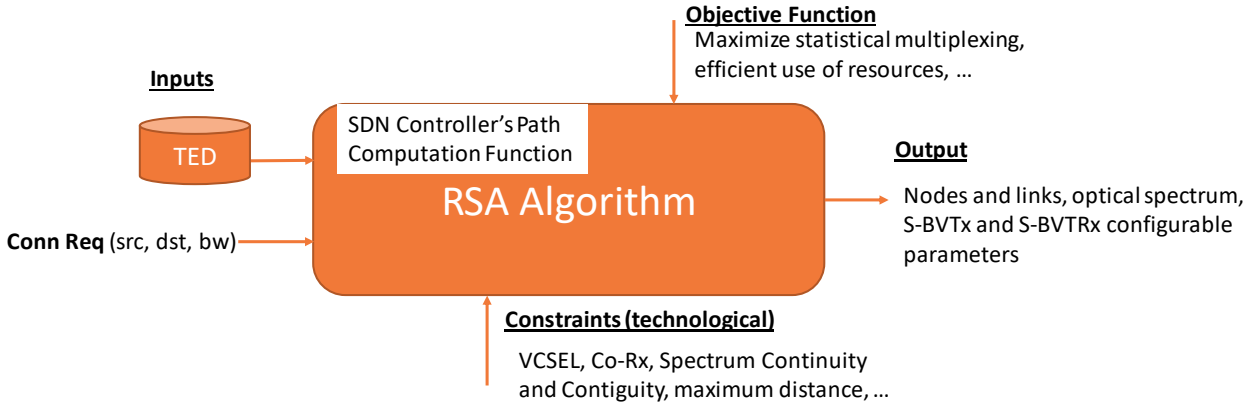


Figure 28. Schematic view of the inputs/outputs of a generic RSA mechanism in PASSION.

The output of the RSA algorithm is made up of a path (nodes and links) connecting both source and destination endpoints as well as the configurable resources at the S-BVT Tx and Rx devices determining the selected FS to be allocated for the requested optical connection.

In the following, it is addressed the framework for devising RSA algorithms. To do that, the RSA algorithms must consider the capabilities and restrictions derived from the adopted PASSION metro network. In this regard, a metro network scenario is formed by heterogenous HLx optical switches. Furthermore, the PASSION modular architectures for the S-BVT Tx and Rx devices are also considered. The goal is to provide on-line RSA algorithms able to compute feasible routes within a PASSION network scenario for dynamic connection requests demanding different data rates.

3.3.1 Optical Flexi-Grid MAN infrastructure

The considered optical flexi-grid metro network is shown in Figure 29. It follows a star-ring topology formed by three identical clusters: A, B and C. Each cluster has a pair of linear networks branches interconnected among themselves through a ring network. The deployed optical switches through the network infrastructure are mapped into the defined HL node types. HL4 nodes (represented in Figure 29 as black squares) provide traffic aggregation at the access network segment. This is done via an attached S-BVT (represented by a circle in Figure 29). These S-BVT devices has only 20 licensed VCSELs providing a maximum aggregated capacity of 1Tb/s. Additionally, 20 CO-Rxs are deployed at every HL4 S-BVT Rx. S-BVT VCSELs operate at a specific optical carrier within the targeted spectral range (i.e., between 191.900 and 195.875 THz). To support all the frequencies over that range within a cluster, 4 S-BVTs are needed per branch whose optical carriers are spaced 50 GHz. As shown in Table 12, clusters' S-BVTs are labeled by 1A, 1B, 2A, ..., 4B. Thus, optical carriers within a single cluster are spectrally spaced to 25GHz. The S-BVT's VCSELs optical carriers for both branches within a single cluster are deployed to avoid spectral collisions considering the filtering restrictions of the HL4 nodes. These nodes are built using Array Wavelength Grating (AWG) of 50GHz. The LO of every S-BVT's CO-Rx is fully tunable.

Table 12 Deployed S-BVT VCSELs (supported central frequencies) and CO-Rxs

S-BVT	#VCSELs & CO-Rxs	Supported Frequencies (THz)
1A	20	191.900, 192.100, 192.300, …, 195.700
1B	20	192.000, 192.200, 192.400, …, 195.800
2A	20	191.925, 192.125, 192.325, …, 195.725
2B	20	192.025, 192.225, 192.425, …, 195.825
3A	20	191.950, 192.150, 193.350, …, 195.750
3B	20	192.050, 192.250, 192.450, …, 195.850
4A	20	191.975, 192.175, 192.375, …, 195.775
4B	20	192.075, 192.275, 192.475, …, 195.875
Full (F)	160	191.900, 191.925, 191.950, …, 195.875

HL3 nodes, represented by red squares in Figure 29, are the transit optical switches with 25GHz filters based on Wavelength Selective Switch (WSS) technology enabling all-optical routing between HL4 and HL2/1 nodes. No S-BVTs are placed at HL3 transit nodes. Finally, HL2/1 nodes (blue squares in Figure 29) also use a 25GHz filtering optical switch. This node has 3-fully equipped S-BVTs with 160 VCSELs and Co-Rxs per each S-BVT (see Table 12). Hence, those S-BVTs enable all the 25GHz-spaced carriers within the spectral range providing up to 8 Tb/s transport capacity towards the HL4 nodes. Finally, physical links between neighboring HLx nodes have a pair of fibers.

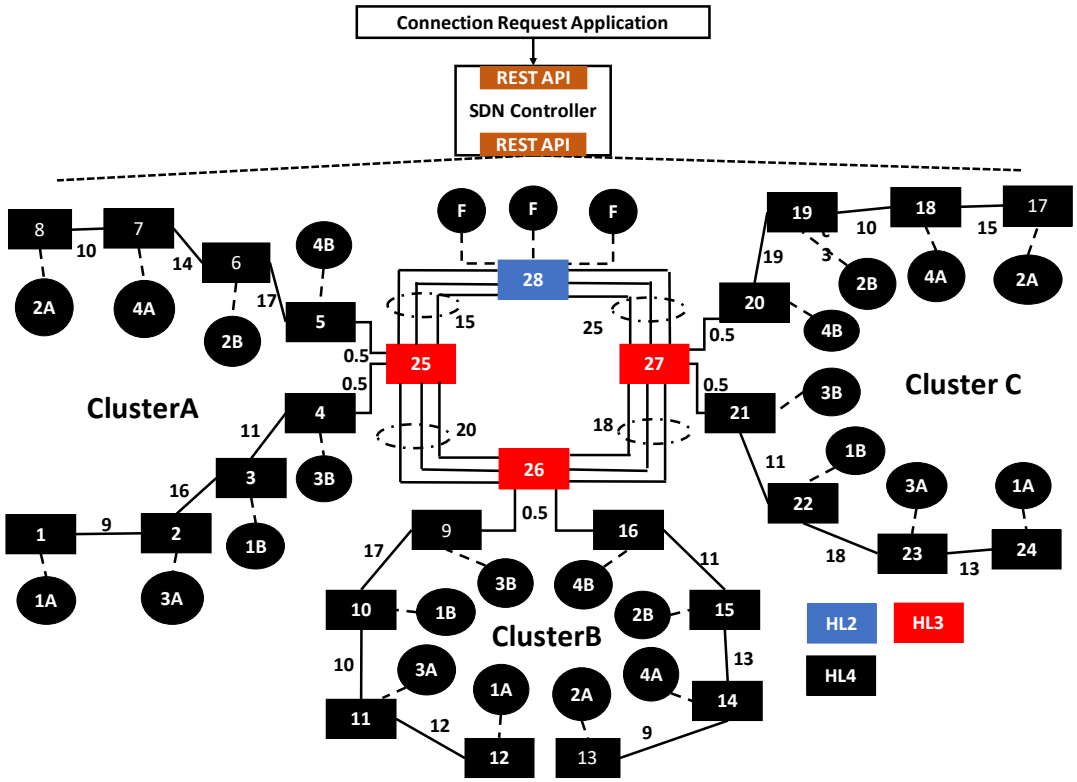


Figure 29. Considered optical flexi-grid metro network scenario.

3.3.2 Dynamic RSA Algorithms: Assumptions

Relying on the above network scenario and targeting the Use Case #1, in the following it is provided the framework to devise on-line RSA algorithms tailored to the requirements and restrictions derived from such a metro infrastructure. Prior to that, it is worth recalling that optical connections requests *req*s are mainly defined by the 3-tuple formed by: the source node (src), the destination node (dst) and the bandwidth bw) in Gb/s. In general, any candidate RSA algorithm seeks for a feasible route entailing: i) the src's S-BVT Tx resources (i.e., VCSELs); ii) the dst's S-BVT Rx resources (i.e., CO-Rxs with the central frequency at each LO); iii) the spatial path; and iv) the spectral path (i.e., frequency slots, FSs).

It is assumed that a *req* may demand a *bw* higher than the maximum capacity supported by a single S-BVT Tx VCSEL (50Gb/s). To deal with that, it is considered that a *req* can be established using a number (i.e., *Num* >= 1) different optical flows. That is, an optical connection request (*req*) is served using *Num* different optical flows. Each of these optical flows is associated to an optical signal transported between a specific pair of S-BVT Tx and Rx devices. Every optical flow allocates an exclusive VCSEL and CO-Rx at the S-BVT endpoints which are not shared among other optical flows even if they belong to the same *req*.

Depending on the spatial path (distance and number of hops), the “net” VCSEL data rate at the dst node could be lower than the maximum transmission rate due to the accumulated physical impairments degrading the optical signal quality. To handle this, it is assumed that S-BVT VCSELs can operate into three deterministic so-called *operational modes* (OM), namely: *high*, *medium* and *low*. Each OM provides different data rates (as shown in Table 13) according to the maximum path distance and number of crossed hops. At the time being, the accurate figures supporting each data rate per OM for both maximum path distance and number of hops is being experimentally evaluated in the context of other project WPs. The outcomes of these experiments will be eventually used to fulfill Table 13 aiming at providing -a scenario reflecting actual features of the technological solutions deployed in the project. This table will be then used for conducting the experimental validation and evaluation of the RSA algorithms within the WP5 activities.

Without loss of generality, when computing the path for a *req*, the RSA algorithm explores iteratively the defined OMs with the objective to select the one attaining the higher VCSEL data rate if the distance and number of hops restrictions are not exceeded. The *Num* (>= 1) of optical flows for a *req* to be allocated are associated to an individual S-BVT VCSEL, S-BVT CO-Rx and FS (i.e., ITU-T flexi-grid *n* and *m* parameters). Observe that the optical flow’s FS is determined by the selected S-BVT Tx VCSEL (optical carrier), the HL node filtering capabilities and the spectrum continuity and contiguity constraints to be ensured at every link forming the end-to-end path.

Table 13 VCSEL Operational Modes (OM)

VCSEL Operation Mode (OM)	Data Rate (Gb/s)	Maximum Path Distance (km)	Maximum Number of Hops
High	50	-	-
Medium	40	-	-
Low	25	-	-

In addition to the *req* parameters, the inputs of any devised RSA algorithm is provided by the information stored in the TED repository, which are: the src’s S-BVT Tx (i.e., VCSELs) availability

and their assigned optical carriers; ii) the dst's S-BVT Rx (CO-Rx) availability; iii) the network topology and the link attributes such as the available optical spectrum (i.e., unused nominal central frequencies, NCFs) and distance. The following notation (Table 14) is used to describe the devised RSA algorithms:

Table 14 Notation for describing devised RSA algorithms

req	Optical connection request
src, dst	Source and destination nodes of req
bw	Data rate (in Gb/s) of req
OM	Set of VCSEL's OM sorted by data rate
OM[i].r	Data rate (in Gb/s) for OM[i]
OM[i].d	Maximum distance (in km) for OM[i]
OM[i].h	Maximum number of hops for OM[i]
Num	Number of VCSELs and CO-Rxs using OM[i]; Computed as upper integer of bw / OM[i].r
$vcsel_n$	Number of unused VCSELs at node n
$vcsel_n$.NCF	Unused NCFs on available VCSELs at node n
$coRx_n$	Number of unused CO-Rxs at node n
$coRx_n$.NCF	Used NCFs by the occupied CO-Rxs at node n.
A	K Shortest Paths (K-SPs) between src and dst nodes sorted by distance and number of hops
A[k].d	Distance (in km) of computed k^{th} SP
A[k].h	Number of hops of computed k^{th} SP
A[k].NCF	Unused and common NCFs (bitmap) over spatial k^{th} SP
A[k].setNCF	Unused and common NCFs (bitmap) considering S-BVT (VCSEL and CO-Rxs) endpoints. Computed intersecting A[k].setNCF & $vcsel_{src}$.NCF & $coRx_{dst}$.NCF
FS	ITU-T flexi-grid Frequency Slot specifying selected n and m
findFS	Function returning first feasible FS over A[k].setNCF; otherwise returns NULL
P	Set of Num paths (p) for each optical flow in req
p_l	Path l for the optical flow in req; $0 \leq l < Num$
P_l.route	Spatial path for p_l
P_l.FS	Frequency Slot (n and m) for pl

Relying on the above notation, two different RSA algorithms have been devised:

- **RSA co-routed optical flows (RSA-CR)**: computes the set of Num optical flows to be accommodated over the same spatial path
- **RSA inverse-multiplexed optical flows (RSA-IM)**: computes the set of Num optical flows without the requirement to be routed over the same spatial path.

In the following, the pseudocodes of the both devised RSA-CR and RSA-IM algorithms are provided relying on the notation in Table 14. For the sake of clarification, the exhaustive performance evaluation conducted over the optical metro network shown in Figure 29 at the control plane is addressed within WP5. Particularly, in T5.2, both RSAs are to be benchmarked considering dynamic traffic demands with heterogeneous data rate requests. Different figures of merit will be used: the *blocked bandwidth ratio* (BBR), the average used of S-BVT resources (VCSELs and Co-Rxs) and the average path computation time. Upcoming reports produced within T5.2 will provide the obtained results of this RSA comparison.

3.3.2.1 RSA for Co-Routed Optical Flows (RSA-CR)

The objective function of the RSA-CR algorithm is to accommodate *reqs* attaining the more efficient use of all network resources (i.e., S-BVT Tx and Rx and optical spectrum), where the resulting *Num* optical flows are routed along the same nodes and links. Figure 30(left) depicts an example of the RSA-CR algorithm. A *req* of 100 Gb/s between S-BVT Tx at src node 9 and S-BVT Rx at dst node 28 arrives. The RSA-CR output describes two VCSELs (operating a high mode, i.e., 50Gb/s) and two CO-Rxs to be allocated. Thus, two optical flows (i.e., X and Y) are set up.

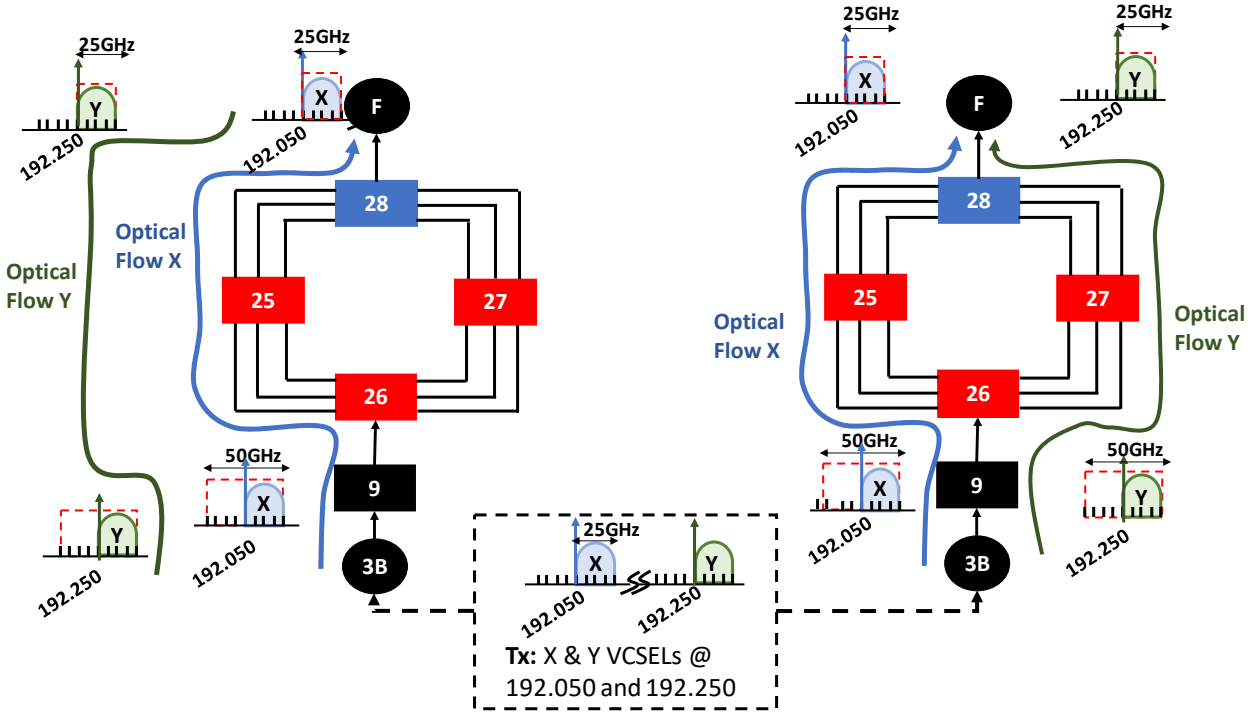


Figure 30. Example of RSA-CR (left) and RSA-IM (-right).

Optical flow X allocates S-BVT's VCSEL with the optical carrier at 192.050THz; whilst optical flow Y allocates the VCSEL with the optical carrier at 192.250THz. At the dst, the respective CO-Rx LOs are programmed to receive such optical carriers. Both X and Y optical flows have their own FS (i.e., n, m). Attribute n describes the VCSEL optical carrier for the optical flow. On the other hand, m describes the slot width resulting from both the VCSEL optical bandwidth (25GHz) and the filtering features at each path node. Since -heterogeneous HL nodes are traversed, this enforces that the FS' n and m parameters over the path vary depending on the HL node filtering capabilities. That is, at node 9 (HL4), the FS of the optical flow X occupies n = -168 (i.e., ITU-T channel for 192.050THz) and m = 4 (slot width of 50GHz). For the optical flow Y at the same node the FS is n = -136 and m = 4. However, at HL3 nodes (26 and 25) and HL2/1 node (28) having filters of 25GHz, the FS for optical flow X uses n = -166 and m = 2. Likewise, for the optical flow Y, the FS in nodes 26, 25 and

28 is determined by n = -134 and m = 2. Albeit optical flow's FS parameters (n and m) may vary along a path, the same optical spectrum carrying effective data (i.e., within 25GHz) remains end-to-end spectrally continuous and contiguous.

The pseudocode of the RSA-CR is detailed in Algorithm 1 (see Figure 31). Upon receiving a *req*, the RSA computes the set *A* containing the K shortest paths (K-SPs) between the src and the dst nodes sorted by the distance and traversed number of hops. Then, the RSA iterates through the VCSEL *OM* set starting from the high mode until a feasible *A* path is found. If all *OM*s are checked but no solution is obtained, the *req* is blocked (i.e., *NoPath*). For an explored *OM[i]*, the *Num* of unused VCSELs and CO-Rxs at both S-BVT Tx and Rx is computed. *Num* is obtained as the higher integer above *bw / OM[i].r*. Recall that *Num* also determines the number of optical flows to be established. If either the available number of VCSELs or the CO-Rxs are lower than *Num*, the *req* is blocked. Otherwise, the computed *A* paths are checked. The first *A[k]* path attaining a feasible solution is selected and their computed resources (i.e., VCSELs, CO-Rxs, FSs) are allocated. To do this, for the *A[k]* path, it is checked that both the distance and number of hops do not exceed the maximum permitted by the considered *OM[i]* (i.e., *OM[i].r* and *OM[i].h*). If this occurs, *A[k]* path is discarded. Otherwise, the set of unused and common NCFs (i.e., *A[k]:setNCF*) is computed. This is done by intersecting the available NCFs along the *A[k]* path links, the associated NCFs of each available VCSEL at the source node ($vcsel_{src}.NCF$), and those NCFs not being occupied by the used CO-Rxs at the dst node (i.e., $coRx_{dst}.NCF$). The resulting *A[k].setNCF* becomes the input for the *findFS* function. This seeks for a common and available FS from the S-BVT Tx (VCSELs), spatial path and S-BVT Rx (Co-Rxs). If it succeeds, the FS is associated to a path (i.e., *p*) for one of the *Num* optical flows to be computed. Additionally, *p* is appended to the P set. Those spectral resources (i.e., NCFs) computed for the FS in the previous *p* are removed from *A[k].setNCF*. This does avoid double booking for the subsequent optical flows to be computed. Finally, if *P* reaches the targeted *Num* of optical flows, the RSA-CR successes. Otherwise, *A[k]* path is discarded and the next *A[k+1]* is explored. If none of the A paths provides a feasible solution, next *OM[i+1]* (with lower VCSEL data rate) is considered.

Algorithm 1 RSA-CR; Input: $req\{src, dst, bw\}$

```
1: Compute A; i ← 0
2: while i < OM do
3:     Compute Num
4:     if Num > vcsel_src OR Num > coRx_dst then
5:         return NoPath
6:     end if
7:     while k < K do
8:         if A[k].d > OM[i].d OR A[k].h > OM[i].h then
9:             Continue                          ▷ next k^th SP
10:        end if
11:        P ← {}; l ← 0
12:        Compute A[k].setNCF
13:        while l < Num do
14:            p_l.route ← A[k]
15:            Compute FS ← findFS(A[k].setNCF)
16:            if !FS then                       ▷ No feasible FS
17:                Break                         ▷ next k^th SP
18:            end if
19:            p_l.FS ← FS; P ← P + p_l; l ← l + 1
20:            Update A[k].setNCF ← A[k].setNCF − FS
21:        end while
22:        if l == Num then
23:            return P                          ▷ Based on OM[i]
24:        else
25:            Break                             ▷ next k^th SP
26:        end if
27:    end while
28: end while
29: return NoPath
```

Figure 31. Pseudocode for RSA-CR algorithm.

3.3.2.2 RSA for Inverse-Multiplexed Optical Flows (RSA-IM)

The main difference with respect to the above RSA-CR algorithm is that the RSA-IM allows establishing the *Num* of optical flows for a *req* over different spatial paths. Typically, this is known as *inverse multiplexing* [RMUN13] [GUO14] whose goal is to attain a better utilization of the overall network (spectral) resources. This distributes the set of optical flows over different spatial paths which in turn may relax the complexity of satisfying the spectral continuity and contiguity constraints for each of them.

Figure 30(right) shows the output of the RSA-IM algorithm for setting up the optical flows for the *req*. Note that both optical flows X and Y are deployed over different FSs but also on separated spatial paths. Optical flow X occupies the VCSEL with optical carrier 192.050 THz along path with nodes 9, 26, 25 and 28. Optical flow Y uses the VCSEL with frequency at 192.250THz through the path nodes 9, 26, 27 and 28. The RSA-IM pseudocode is described in the Algorithm 2 (shown in Figure 32). For an explored *OM[i]*, the *Num* of optical flows is computed. Then, it is checked that there is *Num* available S-BVT VCSELs and CO-Rxs at both src and dst nodes. For each targeted optical flow, the pre-computed *A* paths are considered. For an *A[k]* path, both its distance and number of hops must not exceed those imposed by *OM[i]*. If this occurs, next *OM[i+1*] is explored. Otherwise, the *A[k].setNCF* is computed and used as input for the function *findFS*. Again, if *findFS* cannot find a

feasible FS, next path in *A* is checked. Otherwise, the *p* path for an optical flow is made by both the A[k] spatial route and the computed FS. *p* is then added to the targeted *P* set. For the next optical flow (out from *Num*), previously computed FSs (i.e., related NCFs) are removed. If *Num* paths in *P* are computed, the RSA-IM stops and the *req* is successfully computed and the selected resources (S-BVT VCSELs and Co-Rxs and FSs) are allocated. However, if all *A* paths are checked without finding a solution, next *OM[i+1]* is used. Finally, if no *OM[i]* attains a feasible *P*, *NoPath* is returned.

Algorithm 2 RSA-IM; Input: $req\{src, dst, bw\}$

1: Compute A; $i \leftarrow 0$
2: **while** $i < OM$ **do**
3: Compute Num
4: **if** $Num > vcsel_{src}$ OR $Num > coRx_{dst}$ **then**
5: **return** $NoPath$
6: **end if**
7: $P \leftarrow \{\}$; $l \leftarrow 0$
8: **while** $l < Num$ **do**
9: **while** $k < K$ **do**
10: **if** $A[k].d > OM[i].d$ OR $A[k].h > OM[i].h$ **then**
11: **Continue** ▷ next k^{th} SP
12: **end if**
13: Compute $A[k].setNCF$
14: $p_l.route \leftarrow A[k]$
15: Compute $FS \leftarrow findFS(A[k].setNCF)$
16: **if** $!FS$ **then** ▷ No feasible FS
17: **Continue** ▷ next k^th SP
18: **end if**
19: $p_l.FS \leftarrow FS$; $P \leftarrow P + p_l$; $l \leftarrow l + 1$
20: $vcsel_{src}.NCF \leftarrow vcsel_{src}.NCF - FS$
21: $coRx_{dst}.NCF \leftarrow coRx_{dst}.NCF + FS$
22: **Break** ▷ next p_l
23: **end while**
24: **end while**
25: **if** $i == Num$ **then**
26: **return** P ▷ Based on $OM[i]$
27: **end if**
28: **end while**
29: **return** $NoPath$

Figure 32. Pseudocode for RSA-IM algorithm.

4 CONCLUSIONS

This document details the overall metro network infrastructure planned in PASSION including the key technologies being investigated and deployed in WP3 and WP4. In brief, such technological solutions are the optical VCSEL-based S-BVT, the coherent receiver and heterogeneous (HL) optical switch architectures.

First, it is reviewed the targeted optical metro network (reported in D2.1) detailing the hierarchical network topology, i.e., the HL nodes. Features and capabilities (e.g., supported link distances with neighbouring nodes, S-BVT, filtering, etc.) at each HL are described according to the location within the metro network, that is close to the access network segment or deeper within the metro network bordering the core network segment. Additionally, it is also recalled the targeted project use cases (and their requirements) set. In this regard, special attention is put on the “pay-as-you-grow” and “on-line restoration” use cases.

Leveraging the overview of the network elements and devices considered in PASSION, it is derived the information model to be handled by the SDN controller. This is referred to as the HAL information. This indeed provides an abstracted information (hidden low-level details) sufficient to allow the SDN controller computing, selecting and programming the optical resources (i.e., VCSEL elements in the S-BVT Tx, optical switches to be traversed, and CO-Rx in the S-BVT Rx) when serving an incoming connection demand. The outcome of deriving the HAL for each network element and device is to pave the way for designing the SBI API (protocol messages / operations and content encoding). This SBI API is essential to enable the controller automatically programming each transport network element and device.

Moreover, an NBI API is also defined to support an external application demanding the SDN controller specific network connection services. Using both SBI and NBI APIs, the workflow handled by the SDN controller is presented. The workflow poses the basis for deploying the interactions coordinated by the SDN controller to automatically process, compute, and configure the underlying transport infrastructure when a new connection is being set up.

For the path computation, two RSA algorithms are proposed, and their respective pseudo-code presented. These algorithms are referred to as RSA-CR and RSA-IM. The RSA objective is, under dynamic connection demands, to consider the restrictions arising from the technological PASSION solutions and achieve an efficient use of all the network resources.

The validation and experimental evaluation of the control aspects entailing the defined SBI and NBI APIs as well as the RSA algorithms are planned to be carried out within the WP5. To this end, a metro network topology following the PASSION adopted solutions is proposed. This metro network infrastructure becomes the reference scenario to deal with defined use cases.

Finally, this deliverable also provides the implementable solutions of the SDN controller APIs (NBI and SBIs) via the Swagger tool (specified in both enclosed Annexes). Thereby, any interested researcher can leverage such a description and use / adapt it to carry out their own related investigations and/or deployments.

5 REFERENCES

[D2.1] "Definitionof the use cases and requirements for network, systems and subsystems", PASSION D2.1, September 2018.

[MS6] "Preliminary definition of the node and transceiver architecture", PASSION project MS6, Jan. 2019.

[D41] "Circuitry and technology matching to the path functionality in the optical node", PASSION Project D4.1, March 2018.

[Sva19A] M. Svaluto Moreolo, R. Martínez, L. Nadal, J. M. Fabrega, N. Tessema, N. Calabretta, R. Stabile, P. Parolari, A. Gatto, P. Boffi, G. Otero, D. Larrabeiti, J. A. Hernandez, P. Reviriego, J. P. Fernández-Palacios, V. López, G. Delrosso, C. Neumeyr, K. Solis-Trapala, G. Parladori, G. Gasparini, "Spectrum/Space Switching and Multi-Terabit Transmission in Agile Optical Metro Networks," in Proc. 24th OptoElectronics and Communications Conference (OECC/PSC 2019), 7-11 July 2019, Fukuoka (Japan).

[Sva19B] M. Svaluto Moreolo et al., "Programmable VCSEL-based Transceivers for Multi-terabit Capacity Networking," CLEO 2019, San Jose, CA (USA), May 2019.

[D3.2] "First testing of directly modulated VCSELs with selected drivers", PASSION Project D3.2, May 2018.

[M4.4] Realization of a Software Control board for the INP CO-RX, December 2019.

[ITU-TG694.1]Recommendation ITU-T G.694.1, "Spectral grids for WDM applications: DWDM frequency grid", February 2012.

[swagger] https://editor.swagger.io

[TAPI] "Transport API (TAPI) 2.0 Overview", Optical Networking Foundation (ONF), August 2017

[RMUN13] R. Muñoz, et. al., "Design and experimental evaluation of dynamic inverse-multiplexing provisioning in GMPLS-controlled flexi-grid DWDM network with sliceable OTN BVTs", in Proc. of European Conference on Optical Communications (ECOC), Sept. 2013

[GUO14] B. Guo, et. al., "Multicasting based optical inverse multiplexing in elastic optical network", OSA Optics Express, vol. 22, no. 12, 2014.

6 ACRONYMS

API	Applicaiton Programmin Interface
AWG	Array Waveguide Grating
BS	Base Station
BVT	Bandwidth-Variable Transceiver
CDN	Content Delivery Network
CO	Central Office
CO-Rx	Coherent Receiver
CRM	Coherent Receiver Module
CS	Channel Spacing
CWDM	Coarse Wavelength Division Multiplexing
DB	Database
DC	Data Center
DCI	Data Center Interconnection
DSLAM	Digital Subscriber Line Access Multiplexer
FEC	Forwarding Error Correction
HAL	Hardware Abstraction Layer
HLn	Hierarchy Level n
IT	Information Technologies
IP	Internet Protocol
IPTV	IP Television
JSON	JavaScript Object Notation
LSP	Label Switched Path
MAN	Metropolitan Area Network
NBI	NorthBound Interface
NCF	Nominal Central Frequency
OLT	Optical Line Terminal
PCE	Path Conputation Element
REST	Representational State Transfer

RSA	Routing and Spectrum Assignment
URI	Uniform Resource Identifier
WSS	Wavelength Selective Switch

7 ANNEX 1

The following provides the deployed PASSION proprietary NBI to be copied and pasted in the swagger editor. This allows automatically creating the skeletons of both the client and server programs associated to such an API.

```
{
  "swagger": "2.0",
  "info": {
    "description": "REST-API for NBI. Find more at http://www.passion-project.eu/",
    "version": "0.1",
    "title": "EC PASSION PROJECT NBI SPECIFICATION",
    "termsOfService": "/tos",
    "contact": {
      "email": "ricardo.martinez@cttc.es"
    },
    "license": {
      "name": "TBD"
    }
  },
  "basePath": "/rest/api/v1",
  "paths": {
        "/lsp" : {
                "post": {
                        "tags": [
                                "App - SDN Controller API Request"
                        ],
                        "summary": "Request an end-to-end connection",
                        "description": "Request an end-to-end connection for a pair of endpoints specifying
requested bandwidth",
                        "operationId": "connection-request",
                        "produces": [
                                "application/json"
                        ],
                        "parameters": [
                                {
                                        "name": "ConnReqBody",
                                        "schema": {
                                                "type": "object",
                                                "properties": {
```

```
                                                "id": {
                                                        "type": "string",
                                                        "description": "Specifies the identifier / name to refer to such a connection locally in the SDN controller"
                                                },
                                                "src": {
                                                        "$ref": "#/definitions/nodeId",
                                                        "description": "Specifies the source nodeId"
                                                },
                                                "dst": {
                                                        "$ref": "#/definitions/nodeId",
                                                        "description": "Specifies the destination nodeId"
                                                },
                                                "bw": {
                                                        "type": "string",
                                                        "description": "Specifies the amount of bandwidth being requested"
                                                },
                                                "bw_unit": {
                                                        "type": "string",
                                                        "description": "Determines the the units of the requested bandwidth, e.g., Gb/s is Gbps"
                                                },
                                                "of": {
                                                        "type": "string",
                                                        "description": "Determines the objective function (of) code -- i.e., the algorithm to be executed among a potential pool"
                                                }

                                        }
                                },
                                "in": "body",
                                "required": true
                        }
                ],
                "responses": {
                        "200": {
                                "description": "OK"
                        },
                        "201": {
                                "description": "Succesful operation"

                        },
```

```
                        "400": {
                                "description": "Bad request"
                        },
                        "401": {
                                "description": "Unauthorized"
                        },
                        "403": {
                                "description": "Forbidden"
                },
                        "404": {
                                "description": "Not Found"
                        }
                }
        }
},
"/lsp/{Id}": {
        "delete": {
                "tags": [
                        "App - SDN Controller API Delete"
                ],
                "summary": "Delete existing connection",
                "description": "Delete an end-to-end connection by its Id",
                "operationId": "connection-delete",
                "produces": [
                        "application/json"
                ],
                "parameters": [
                        {
                                "name": "Id",
                                "in": "path",
                                "description": "Identifier of the connection to be removed",
                                "required": true,
                                "type": "string"
                        }
                ],
                "responses": {
                        "200": {
                                "description": "OK"
                        },
                        "201": {
```

```
                                        "description": "Succesful operation"

                                },
                                "400": {
                                        "description": "Bad request"
                                },
                                "401": {
                                        "description": "Unauthorized"
                                },
                                "403": {
                                        "description": "Forbidden"
                        },
                                "404": {
                                        "description": "Not Found"
                                }
                        }
                }
        }
  },
  "definitions": {
        "nodeId": {
                "type": "string",
                "description": " (IPv4) using decimal format A.B.C.D"
        }
  }
}
```

8 ANNEX 2

The following provides the deployed PASSION proprietary SBIs to be copied and pasted in the swagger editor. This allows automatically creating the skeletons of both the client and server programs associated to such APIs programming the SBVT Tx, SBVT Rx and Optical Switch.

```
{
  "swagger": "2.0",
  "info": {
    "description": "REST-API for the SBI. Find more at http://www.passion-project.eu/",
    "version": "1.0.1",
    "title": "EC PASSION PROJECT SBI SPECIFICATION",
    "termsOfService": "/tos",
    "contact": {
      "email": "ricardo.martinez@cttc.es"
    },
```

```
    "license": {
      "name": "TBD"
    }
  },
  "basePath": "/passion/sbi",
  "paths": {
    "/sbvtTx": {
      "get": {
        "tags": [
          "S-BVT TX Interface"
        ],
        "summary": "Retrieves the status of all the devices related to the s-bvt tx",
        "description": "Retrieves the status of all the devices related to the s-bvt tx",
        "operationId": "get-s-bvt-tx-status",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "SBVTTxStatus",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            },
            "in": "body",
            "required": true
          }
        ],
        "responses": {
          "200": {
            "description": "Succesful operation",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                },
```

```
                "sbvtTx": {
                  "$ref": "#/definitions/sbvtTx",
                  "description": "PASSION S-BVT Tx SuperModule supporting 4 Modules (SOI/Chip) enabling up to
8Tb/s (16 Tb/s using dual polarization"
                }
              }
            }
          },
          "400": {
            "description": "Bad request",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            }
          },
          "401": {
            "description": "Unauthorized",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            }
          },
```

```
          "404": {
            "description": "Not Found",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            }
          }
        }
      },
      "post": {
        "tags": [
          "S-BVT TX Interface"
        ],
        "summary": "Configure the status (occupy) of a specific set of VCSELs",
        "description": "Configure the status (occupy) of a specific set of VCSELs",
        "operationId": "occupy-s-bvt-tx",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "SBVTTxConfBody",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                },
                "connectionId": {
                  "$ref": "#/definitions/connectionId"
                },
                "sbvtTx": {
                  "$ref": "#/definitions/sbvtTx",
                  "description": "PASSION S-BVT Tx SuperModule supporting 4 Modules (SOI/Chip) enabling up to
8Tb/s (16 Tb/s using dual polarization"
                }
              }
```

```
      },
      "in": "body",
      "required": true
    }
  ],
  "responses": {
    "201": {
      "description": "Succesful operation",
      "schema": {
        "type": "object",
        "properties": {
          "msgId": {
            "$ref": "#/definitions/requestId"
          }
        }
      }
    },
    "400": {
      "description": "Bad request",
      "schema": {
        "type": "object",
        "properties": {
          "msgId": {
            "$ref": "#/definitions/requestId"
          }
        }
      }
    },
    "401": {
      "description": "Unauthorized",
      "schema": {
        "type": "object",
        "properties": {
          "msgId": {
            "$ref": "#/definitions/requestId"
          }
        }
      }
    },
    "403": {
      "description": "Forbidden",
```

```
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        }
      }
    },
      "delete": {
            "tags": [
        "S-BVT TX Interface"
      ],
      "summary": "Remove a given connection imposing realising the associated VCSEL resources",
      "description": "Remove a given connection imposing realising the associated VCSEL resources",
      "operationId": "release-s-bvt-tx",
      "produces": [
        "application/json"
      ],
      "parameters": [
        {
          "name": "SBVTTxConfBody",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              },
```

```
            "connectionId": {
              "$ref": "#/definitions/connectionId"
            }
          }
        },
        "in": "body",
        "required": true
      }
    ],
    "responses": {
      "200": {
        "description": "Succesful operation",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
      "400": {
        "description": "Bad request",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
      "401": {
        "description": "Unauthorized",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
```

```
          }
        },
        "403": {
          "description": "Forbidden",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        }
      }
    },
    "/sbvtTx/connections": {
      "get": {
        "tags": [
          "S-BVT TX Interface"
        ],
        "summary": "Retrieves tha active connections using sbvt tx resources",
        "description": "Retrieves tha active connections using sbvt tx resources",
        "operationId": "get-s-bvt-tx-connections",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "SBVTTxConnections",
```

```
      "schema": {
        "type": "object",
        "properties": {
          "msgId": {
            "$ref": "#/definitions/requestId"
          }
        }
      },
      "in": "body",
      "required": true
    }
  ],
  "responses": {
    "200": {
      "description": "Succesful operation",
      "schema": {
        "type": "object",
        "properties": {
          "msgId": {
            "$ref": "#/definitions/requestId"
          },
          "numActiveConnections": {
            "type": "integer",
            "format": "uint32",
            "description": "Total amount of active connections in the SBVTTx"
          },
          "setActiveConnections": {
            "type": "array",
            "items": {
              "required": [
                "connectionId"
              ],
              "properties": {
                "connectionId": {
                  "$ref": "#/definitions/connectionId"
                }
              }
            }
          }
        }
      }
```

```
},
"400": {
  "description": "Bad request",
  "schema": {
    "type": "object",
    "properties": {
      "msgId": {
        "$ref": "#/definitions/requestId"
      }
    }
  }
},
"401": {
  "description": "Unauthorized",
  "schema": {
    "type": "object",
    "properties": {
      "msgId": {
        "$ref": "#/definitions/requestId"
      }
    }
  }
},
"403": {
  "description": "Forbidden",
  "schema": {
    "type": "object",
    "properties": {
      "msgId": {
        "$ref": "#/definitions/requestId"
      }
    }
  }
},
"404": {
  "description": "Not Found",
  "schema": {
    "type": "object",
    "properties": {
      "msgId": {
        "$ref": "#/definitions/requestId"
```

```
                }
              }
            }
          }
        }
      }
    },
        "/sbvtTx/freqSlot": {
                "post": {
        "tags": [
          "S-BVT TX Interface"
        ],
        "summary": "Configures the sbvt tx resources according to the specified Frequency Slot",
        "description": "Programming the sbvt tx resources according to the selected Frequency Slot (n and m)
aligned with ITU-T G694.1 flexi-grid",
        "operationId": "post-s-bvt-tx-freqSlot",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "SBVTTxConfFreqSotBody",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                },
                "connectionId": {
                  "$ref": "#/definitions/connectionId"
                },
                "sbvtTxFreqSlot": {
                  "$ref": "#/definitions/sbvtTxFreqSlot",
                  "description": "PASSION S-BVT Tx configuration of 1..N supported VCSELs according to the
specified Frequency Slots"
                }
              }
            },
            "in": "body",
            "required": true
          }
        ],
```

```
"responses": {
  "201": {
    "description": "Succesful operation",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        }
      }
    }
  },
  "400": {
    "description": "Bad request",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        }
      }
    }
  },
  "401": {
    "description": "Unauthorized",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        }
      }
    }
  },
  "403": {
    "description": "Forbidden",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
```

```
            }
          }
        }
      },
      "404": {
        "description": "Not Found",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      }
    }
  }
  },
"/sbvtRx": {
  "get": {
    "tags": [
      "S-BVT RX Interface"
    ],
    "summary": "Retrieves the status of all the devices related to the s-bvt rx",
    "description": "Retrieves the status of all the devices related to the s-bvt rx",
    "operationId": "get-s-bvt-rx-status",
    "produces": [
      "application/json"
    ],
    "parameters": [
      {
        "name": "SBVTRxStatus",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        },
        "in": "body",
```

```
        "required": true
      }
    ],
    "responses": {
      "200": {
        "description": "Succesful operation",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            },
            "sbvtRx": {
              "$ref": "#/definitions/sbvtRx",
              "description": "PASSION S-BVT Rx supporting an array / pool of coherent receivers"
            }
          }
        }
      },
      "400": {
        "description": "Bad request",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
      "401": {
        "description": "Unauthorized",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
```

```
      "403": {
        "description": "Forbidden",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
      "404": {
        "description": "Not Found",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      }
    },
    "post": {
      "tags": [
        "S-BVT RX Interface"
      ],
      "summary": "Configure the status (occupy) of a specific set of optical (coherent) receivers",
      "description": "Configure the status (occupy) of a specific set of optical (coherent) receivers",
      "operationId": "allocate-s-bvt-rx-status",
      "produces": [
        "application/json"
      ],
      "parameters": [
        {
          "name": "SBVTRxConfBody",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
```

```
            "$ref": "#/definitions/requestId"
          },
          "connectionId": {
            "$ref": "#/definitions/connectionId"
          },
          "sbvtRx": {
            "$ref": "#/definitions/sbvtRx",
            "description": "PASSION S-BVT Rx supporting an array / pool of coherent receivers"
          }
        }
      },
      "in": "body",
      "required": true
    }
  ],
  "responses": {
    "201": {
      "description": "Succesful operation",
      "schema": {
        "type": "object",
        "properties": {
          "msgId": {
            "$ref": "#/definitions/requestId"
          }
        }
      }
    },
    "400": {
      "description": "Bad request",
      "schema": {
        "type": "object",
        "properties": {
          "msgId": {
            "$ref": "#/definitions/requestId"
          }
        }
      }
    },
    "401": {
      "description": "Unauthorized",
      "schema": {
```

```
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "403": {
          "description": "Forbidden",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        }
      }
    },
      "delete": {
      "tags": [
        "S-BVT RX Interface"
      ],
      "summary": "Release the set of optical (coherent) receiver resources to a given connection",
      "description": "Release the set of optical (coherent) receiver resources to a given connection",
      "operationId": "release-s-bvt-rx",
      "produces": [
        "application/json"
```

```
    ],
    "parameters": [
      {
        "name": "SBVTRxConfBody",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            },
            "connectionId": {
              "$ref": "#/definitions/connectionId"
            }
          }
        },
        "in": "body",
        "required": true
      }
    ],
    "responses": {
      "200": {
        "description": "Succesful operation",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
      "400": {
        "description": "Bad request",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
```

```
        },
        "401": {
          "description": "Unauthorized",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "403": {
          "description": "Forbidden",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        }
      }
    }
  },
    "/sbvtRx/freqSlot": {
            "post": {
                    "tags": [
                            "S-BVT RX Interface"
```

```
                        ],
                        "summary": "Configure the status (occupy) of agent selected available optical (coherent) receivers using the Freq Slot",
                        "description": "Configure the status (occupy) of aset of optical (coherent) receivers being available",
                        "operationId": "post-s-bvt-rx-status-freq-slot",
                        "produces": [
                        "application/json"
                        ],
                        "parameters": [
                                {
                                        "name": "SBVTRxConfFreqSlotBody",
                                        "schema": {
                                                "type": "object",
                                                "properties": {
                                                        "msgId": {
                                                                "$ref": "#/definitions/requestId"
                                                        },
                                                        "connectionId": {
                                                                "$ref": "#/definitions/connectionId"
                                                        },
                                                        "sbvtRxFreqSlot": {
                                                                "$ref": "#/definitions/sbvtRxFreqSlot",
                                                                "description": "PASSION S-BVT Rx supporting an array / pool of coherent receivers"
                                                        }
                                                }
                                        },
                                        "in": "body",
                                        "required": true
                                }
                        ],
                        "responses": {
                                "201": {
                                        "description": "Succesful operation",
                                        "schema": {
                                                "type": "object",
                                                "properties": {
                                                        "msgId": {
                                                        "$ref": "#/definitions/requestId"
                                                        }
                                                }
```

```
                }
        },
        "400": {
                "description": "Bad request",
                "schema": {
                        "type": "object",
                        "properties": {
                                "msgId": {
                                        "$ref": "#/definitions/requestId"
                                }
                        }
                }
        },
        "401": {
                "description": "Unauthorized",
                "schema": {
                        "type": "object",
                        "properties": {
                                "msgId": {
                                        "$ref": "#/definitions/requestId"
                                }
                        }
                }
        },
        "403": {
                "description": "Forbidden",
                "schema": {
                        "type": "object",
                        "properties": {
                                "msgId": {
                                        "$ref": "#/definitions/requestId"
                                }
                        }
                }
        },
        "404": {
                "description": "Not Found",
                "schema": {
                        "type": "object",
                        "properties": {
                                "msgId": {
```

```
                                                                "$ref": "#/definitions/requestId"
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    },
  "/sbvtRx/connections": {
    "get": {
      "tags": [
        "S-BVT RX Interface"
      ],
      "summary": "Retrieves tha active connections using sbvt rx resources",
      "description": "Retrieves tha active connections using sbvt rx resources",
      "operationId": "get-s-bvt-rx-connections",
      "produces": [
        "application/json"
      ],
      "parameters": [
        {
          "name": "SBVTRxConnections",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          },
          "in": "body",
          "required": true
        }
      ],
      "responses": {
        "200": {
          "description": "Succesful operation",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
```

```
            "$ref": "#/definitions/requestId"
          },
          "numActiveConnections": {
            "type": "integer",
            "format": "uint32",
            "description": "Total amount of active connections in the SBVTRx"
          },
          "setActiveConnections": {
            "type": "array",
            "items": {
              "required": [
                "connectionId"
              ],
              "properties": {
                "connectionId": {
                  "$ref": "#/definitions/connectionId"
                }
              }
            }
          }
        }
      },
      "400": {
        "description": "Bad request",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
      "401": {
        "description": "Unauthorized",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
```

```
                }
              }
            }
          },
          "403": {
            "description": "Forbidden",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            }
          }
        }
      }
    },
    "/opticalSwitch": {
      "get": {
        "tags": [
          "Optical Switch"
        ],
        "summary": "Retrieves the inventory of the optical switch, i.e., express and add/drop port, supported freq. (using ITU-T G694.1), etc.",
        "description": "Retrieves the inventory of the optical switch, i.e., express and add/drop port, supported freq. (using ITU-T G694.1), etc.",
        "operationId": "get-optical-switch-status",
        "produces": [
          "application/json"
        ],
```

```
"parameters": [
  {
    "name": "OpticalSwitchInfo",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        }
      }
    },
    "in": "body",
    "required": true
  }
],
"responses": {
  "200": {
    "description": "Succesful operation",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        },
        "opticalSwitch": {
          "$ref": "#/definitions/opticalSwitch",
          "description": "PASSION Optical Switch architecture"
        }
      }
    }
  },
  "400": {
    "description": "Bad request",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        }
      }
    }
```

```
        },
        "401": {
          "description": "Unauthorized",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "403": {
          "description": "Forbidden",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        }
      }
    },
  },
  "/opticalSwitch/connections": {
    "get": {
      "tags": [
        "Optical Switch Connections"
```

```
],
"summary": "Retrieves the set of active (i.e., existing) connections at the Optical Switch",
"description": "Retrieves the set of active (i.e., existing) connections at the Optical Switch",
"operationId": "get-optical-switch-connections",
"produces": [
  "application/json"
],
"parameters": [
  {
    "name": "opticalSwitchConnections",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        }
      }
    },
    "in": "body",
    "required": true
  }
],
"responses": {
  "200": {
    "description": "Succesful operation",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        },
        "numActiveConnections": {
          "type": "integer",
          "format": "uint32",
          "description": "Total amount of active connections in the optical switch"
        },
        "setActiveConnections": {
          "type": "array",
          "items": {
            "required": [
              "connectionId",
```

```
                                        "crossConnection"
                    ],
                    "properties": {
                      "connectionId": {
                        "$ref": "#/definitions/connectionId"
                      },
                                        "crossConnection": {
                                              "$ref": "#/definitions/crossConnection",
                                              "description": "PASSION Optical Switch cross-connection
configuration commands"
                                        }
                    }
                  }
                }
              }
            },
            "400": {
              "description": "Bad request",
              "schema": {
                "type": "object",
                "properties": {
                  "msgId": {
                    "$ref": "#/definitions/requestId"
                  }
                }
              }
            },
            "401": {
              "description": "Unauthorized",
              "schema": {
                "type": "object",
                "properties": {
                  "msgId": {
                    "$ref": "#/definitions/requestId"
                  }
                }
              }
            },
            "403": {
              "description": "Forbidden",
```

```
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            }
          },
          "404": {
            "description": "Not Found",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
                }
              }
            }
          }
        }
      },
      "post": {
        "tags": [
          "Optical Switch Connections"
        ],
        "summary": "Creates/configures the cross-connection (Add, Express, Drop) for a specific connection",
        "description": "Creates/configures the cross-connection (Add, Express, Express) for a specific
connection",
        "operationId": "configure-optical-switch",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "OpticalSwitchConfBody",
            "schema": {
              "type": "object",
              "properties": {
                "msgId": {
                  "$ref": "#/definitions/requestId"
```

```
            },
            "connectionId": {
              "$ref": "#/definitions/connectionId"
            },
            "crossConnection": {
              "$ref": "#/definitions/crossConnection",
              "description": "PASSION Optical Switch cross-connection configuration commands"
            }
          }
        },
        "in": "body",
        "required": true
      }
    ],
    "responses": {
      "201": {
        "description": "Succesful operation",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
      "400": {
        "description": "Bad request",
        "schema": {
          "type": "object",
          "properties": {
            "msgId": {
              "$ref": "#/definitions/requestId"
            }
          }
        }
      },
      "401": {
        "description": "Unauthorized",
        "schema": {
          "type": "object",
```

```
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "403": {
          "description": "Forbidden",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        }
      }
    },
    "delete": {
      "tags": [
        "Optical Switch Connections"
      ],
      "summary": "Removes the cross-connection (Add, Express, Drop) asssociated to a given connection",
      "description": "Removes the cross-connection (Add, Express, Drop) asssociated to a given connection",
      "operationId": "remove-optical-switch",
      "produces": [
        "application/json"
      ],
```

```
"parameters": [
  {
    "name": "OpticalSwitchRelease",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        },
        "connectionId": {
          "$ref": "#/definitions/connectionId"
        }
      }
    },
    "in": "body",
    "required": true
  }
],
"responses": {
  "200": {
    "description": "Succesful operation",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        }
      }
    }
  },
  "400": {
    "description": "Bad request",
    "schema": {
      "type": "object",
      "properties": {
        "msgId": {
          "$ref": "#/definitions/requestId"
        }
      }
    }
  },
```

```
        "401": {
          "description": "Unauthorized",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "403": {
          "description": "Forbidden",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        },
        "404": {
          "description": "Not Found",
          "schema": {
            "type": "object",
            "properties": {
              "msgId": {
                "$ref": "#/definitions/requestId"
              }
            }
          }
        }
      }
    }
  },
},
"definitions": {
  "requestId": {
    "type": "integer",
    "format": "uint32",
```

```
      "description": "used to associate request and response message through the RESTful API"
    },
    "connectionId": {
      "type": "string",
      "description": "the conenction Id (string)"
    },
    "bitmapLongWordAvailableNCF": {
      "type": "integer",
      "format": "uint32",
      "description": "used to encod in a 32-bit (long Word) the availability of 32 Nominal Central Frequencies.
Each NCF corresponds to a bit, where set to 1 means used and set to 0 available"
    },
    "VCSELs": {
      "type": "array",
      "items": {
        "required": [
          "vcselId",
          "used_state",
          "bandwidth",
          "central-frequency",
          "modulation-format",
          "fec"
        ],
        "properties": {
          "vcselId": {
            "type": "integer",
            "format": "uint16",
            "description": "Determines the VCSEL Id"
          },
          "used_state": {
            "type": "boolean",
            "description": "true means used; false means unused"
          },
          "bandwidth": {
            "type": "number",
            "format": "float",
            "description": "Specifies the optical bandwidth in MHz"
          },
          "central-frequency": {
            "type": "number",
            "format": "float",
```

```
          "description": "Specifies the central frequency in MHz"
        },
        "modulation-format": {
          "type": "number",
          "format": "integer",
          "description": "Specifies the used modulation format"
        },
        "fec": {
          "type": "number",
          "format": "integer",
          "description": "Specifies the used fec type"
        }
      }
    }
  },
  "subModulesTx": {
    "type": "array",
    "items": {
      "required": [
        "subModuleTxId",
        "VCSELs"
      ],
      "properties": {
        "subModuleTxId": {
          "type": "integer",
          "format": "uint16",
          "description": "Determines the PASSION subModule Id uint16"
        },
        "VCSELs": {
          "$ref": "#/definitions/VCSELs",
          "description": "PASSION S-BVT VCSEL attributes operating up to 50Gb/s"
        }
      }
    }
  },
  "modulesTx": {
    "type": "array",
    "items": {
      "type": "object",
      "required": [
        "moduleTxId",
```

```
        "subModulesTx"
      ],
      "properties": {
        "moduleTxId": {
          "type": "integer",
          "format": "uint16",
          "description": "Determines the Module Id"
        },
        "subModulesTx": {
          "$ref": "#/definitions/subModulesTx",
          "description": "PASSION S-BVT Tx Module (SOI/Chip) supporting 4 subModules enabling up to 2 Tb/s"
        }
      }
    }
  },
  "sbvtTx": {
    "type": "object",
    "required": [
      "numModulesTx",
      "modulesTx"
    ],
    "properties": {
      "numModulesTx": {
        "type": "integer",
        "description": "Specifies the number of total number of equipped (SOI/chip) PASSION S-BVT Tx modules"
      },
      "modulesTx": {
        "$ref": "#/definitions/modulesTx"
      }
    }
  },
      "sbvtTxFreqSlot": {
              "type": "array",
              "items": {
                      "required": [
                              "centerFreq_n",
                              "slotWidth_m",
                              "used_state",
                              "bandwidth",
                              "modulation-format",
                              "fec"
```

```
                        ],
                        "properties": {
                                "centerFreq_n": {
                                        "type": "integer",
                                        "format": "int16",
                                        "description": "Specifies the center frequency wrt the flexi-grid ITU-T
G694.1"
                                },
                                "slotWidth_m": {
                                        "type": "integer",
                                        "format": "uint16",
                                        "description": "Specifies the frequency slot width wrt the flexi-grid
ITU-T G694.1"
                                },
                                "used_state": {
                                        "type": "boolean",
                                        "description": "true means used; false means unused"
                                },
                                "bandwidth": {
                                        "type": "number",
                                        "format": "float",
                                        "description": "Specifies the optical bandwidth in MHz"
                                },
                                "modulation-format": {
                                        "type": "number",
                                        "format": "integer",
                                        "description": "Specifies the used modulation format"
                                },
                                "fec": {
                                        "type": "number",
                                        "format": "integer",
                                        "description": "Specifies the used fec type"
                                }
                        }
                }
        },
    "OpticalReceiver": {
      "type": "array",
      "items": {
        "required": [
          "optReceiverId",
          "used_state",
```

```
          "freqLocalOscillator"
        ],
        "properties": {
          "optReceiverId": {
            "type": "integer",
            "format": "uint16",
            "description": "Determines the optical receiver Id"
          },
          "used_state": {
            "type": "boolean",
            "description": "true means used; false means unused"
          },
          "freqLocalOscillator": {
            "type": "number",
            "format": "float",
            "description": "Specifies the central frequency of the local oscillator of the Coherent Receiver in
MHz"
          }
        }
      }
    },
    "modulesRx": {
      "type": "array",
      "items": {
        "type": "object",
        "required": [
          "moduleIdRx",
          "numDevices",
          "OpticalReceiver"
        ],
        "properties": {
          "moduleRxId": {
            "type": "integer",
            "format": "uint16",
            "description": "Determines the ModuleRx Id"
          },
          "numOpticalReceivers": {
            "type": "integer",
            "description": "Sepecifies the amount of Coherent Receivers being equipped"
          },
          "opticalReceivers": {
```

```
            "$ref": "#/definitions/OpticalReceiver",
            "description": "PASSION S-BVT Rx array containing a pool of Coherence Receivers to be programmed"
          }
        }
      }
    },
    "sbvtRx": {
      "type": "object",
      "required": [
        "numModulesRx",
        "moduleIdRx"
      ],
      "properties": {
        "numModulesRx": {
          "type": "integer",
          "description": "Specifies the number of total number of equipped PASSION S-BVT Rx modules"
        },
        "modulesRx": {
          "$ref": "#/definitions/modulesRx"
        }
      }
    },
        "sbvtRxFreqSlot": {
                "type": "array",
                "items": {
                        "required": [
                                "used_state",
                                "freqLocalOscillator_n"
                        ],
                        "properties": {
                                "used_state": {
                                        "type": "boolean",
                                        "description": "true means used; false means unused"
                                },
                                "freqLocalOscillator_n": {
                                        "type": "integer",
                                        "format": "int16",
                                        "description": "Specifies the central frequency of LO of the CO-Rx via n
as defined in ITU-G691.1"
                                }
                        }
```

```
                }
        },
        "portId": {
          "type": "integer",
          "format": "uint32",
          "description": "The portId is associate to a unsigned int32 bits"
        },
        "centerFreq_n": {
          "type": "integer",
          "format": "int16",
          "description": "Relying on ITU-T G694.1 specifies with n (int 16 bits) the nominal central frequency"
    },
        "slotWidth_m": {
          "type": "integer",
          "format": "uint16",
          "description": "Specifies the slot width of the frequency slot in terms of m (unsigned int 16 bits)
according to ITU-T G694.1"
    },
    "port": {
      "type": "object",
      "required": [
        "portId"
      ],
      "properties": {
        "portId": {
          "$ref": "#/definitions/portId"
        },
        "portName": {
          "type": "string",
          "description": "Specifies the name of the port"
        },
        "portType": {
          "type": "integer",
          "description": "Supported Ports types are: Express (1), Add (2) or Drop (3)"
        },
        "direction": {
          "type": "integer",
          "description": "Supported Port directions are: TXRX (1), TX (2) or RX (3)"
        },
        "total_n": {
          "type": "integer",
```

```
          "description": "Specifies the total number of supported nominal central frequencies according to the
ITU-T G694.1, i.e., the amount of n"

        },

        "min_n": {

          "type": "integer",

          "format": "int16",

          "description": "Minimum n (according to ITU-T G694.1) specifying the minimum nominal central frequency"

        },

        "max_n": {

          "type": "integer",

          "format": "int16",

          "description": "Maximum n (according to ITU-T G694.1) specifying the maximum nominal central frequency"

        },

        "centerFreqGranurality": {

          "type": "number",

          "description": "Determines the center frequency granurality (i.e., Channel Spacing) in GHz"

        },

        "slotWidthGranurality": {

          "type": "number",

          "description": "Determines the minimum slot width of a frequency slot in GHz"

        },

        "txBitmapAvailableNCFs": {

          "type": "array",

          "description": "Array containing a set of 32bits defining the availability of the supported NCFs. The
1st 32bits long word encods the lowest (at the frequency grid) NCFs",

          "items": {

            "type": "object",

            "required": [

              "bitmapLongWordAvailableNCF"

            ],

            "properties": {

              "bitmapLongWordAvailableNCF": {

                "$ref": "#/definitions/bitmapLongWordAvailableNCF"

              }

            }

          }

        },

        "rxBitmapAvailableNCFs": {

          "type": "array",

          "description": "Array containing a set of 32bits defining the availability of the supported NCFs. The
1st 32bits long word encods the lowest (at the frequency grid) NCFs",

          "items": {
```

```
          "type": "object",
          "required": [
            "bitmapLongWordAvailableNCF"
          ],
          "properties": {
            "bitmapLongWordAvailableNCF": {
              "$ref": "#/definitions/bitmapLongWordAvailableNCF"
            }
          }
        }
      }
    },
    "setPorts": {
      "type": "object",
      "properties": {
        "ports": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/port"
          },
          "description": "Set of ports forming the Optical Switch"
        }
      }
    },
    "opticalSwitch": {
      "type": "object",
      "description": "Description of the express, add and drop ports forming the opticalSwitch (e.g., ROADM)",
      "required": [
        "numExpressPorts",
        "expressPorts",
        "numAddDropPorts",
        "addDropPorts"
      ],
      "properties": {
        "numExpressPorts": {
          "type": "integer",
          "description": "Number of total Express Ports (Tx/Rx)"
        },
        "expressPorts": {
          "$ref": "#/definitions/setPorts"
```

```
        },
        "numAddDropPorts": {
          "type": "integer",
          "description": "Number of total Add and Drop Ports "
        },
        "addDropPorts": {
          "$ref": "#/definitions/setPorts"
        }
      }
    },
    "crossConnection": {
      "type": "object",
      "description": "Configuration of a specific cross-connection: portIn to portOut with a specific Freq. Slot (i.e., n and m)",
      "required": [
        "portIn",
        "portOut"
      ],
      "properties": {
        "portIn": {
                    "$ref": "#/definitions/portId"
        },
        "portOut": {
          "$ref": "#/definitions/portId"
        },
                  "centerFreq_n": {
                    "$ref": "#/definitions/centerFreq_n"
                  },
                  "slotWidth_m": {
                    "$ref": "#/definitions/slotWidth_m"
                  }
      }
    }
  }
}
```