

D3.2: Design and Evaluation of the Adaptive Network Manager and Functional Protocol Extensions

Status and Version:	Version 1, Final	
Date of issue:	12.11.2014	
Distribution:	Public	
Author(s):	Name	Partner
	Victor Lopez	TID
	Oscar Gonzalez de Dios	TID
	Juan Pedro Fernandez Palacios	TID
	Filippo Cugini	CNIT
	Ramon Casellas	CTTC
	Ricardo Martínez	CTTC
	Raul Muñoz	CTTC
	Ricard Vilalta	CTTC
	Daniel King	ODC
	Adrian Farrel	ODC
	Antonio D'Errico	Ericsson
	Jorge Lopez de Vergara	NAUDIT
	Javier Aracil	NAUDIT
	Sergio Lopez Buedo	NAUDIT
	Giorgio Parladori	ALU-I
Checked by:	Daniel King	ODC

Table of Contents

1	Introduction	5
1.1	Executive Summary	5
1.2	Reference Material	8
1.2.1	Reference Documents	8
1.2.2	Acronyms	11
1.3	Document History	13
1.4	Document Overview	13
2	GMPLS / PCE Control Plane Functional description	14
2.1	Control Plane Architecture	14
2.1.1	Resource Discovery and Topology Dissemination	14
2.1.2	Stateful Path Computation	14
2.1.3	Signalling Aspects	15
2.2	Advanced Functionalities	15
2.2.1	Filter Control	15
2.2.1.1	Differentiated Filter Configuration (DFC)	15
2.2.1.2	Super-filters	17
2.2.2	Transceiver Control	19
2.2.2.1	Elastic Operation on Code Adaptation	19
2.2.2.2	GMPLS/PCE Control and Configuration of MF OTPs	21
2.3	Control Plane in a Multi Domain Network	27
2.3.1	Architecture	28
2.3.1.1	Inter-connected Traffic Engineered Networks	29
2.3.1.2	Multi-domain Topology Management and Inter-domain Routing	30
2.3.1.3	Hierarchical Stateful PCE	31
2.3.1.4	Signalling Aspects	31
2.3.1.5	Control Plane Procedures	32
2.3.2	Selected Implementation Scenarios	32
2.3.2.1	Scenario 1: Architecture, Interfaces and Protocol Functions	32
2.3.2.2	Scenario 2: Architecture, Interfaces and Protocol Functions	33
3	GMPLS/PCE Protocol Extensions	35
3.1	Label format	35
3.1.1	Single Label Component	35
3.1.2	Multiple Label Components	36
3.2	PCEP	37
3.2.1	Path Computation	37
3.2.2	PCEP-based Inter-AS TE Link Update	39
3.2.3	Label and Label Set Extensions	39
3.2.4	LSPDB Synchronization	40
3.2.5	LSP Provisioning	41
3.2.6	LSP Concatenation (ERO format)	43



3.2.7	LSP Modification	44
3.3	RSVP-TE Signalling Protocol	44
3.3.1	Signalling Scenarios	44
3.3.2	Signalling Procedures	47
3.3.2.1	Connection Establishment	47
3.3.2.2	Connection release	48
3.3.2.3	Refresh Overhead Reduction	48
3.3.3	RSVP Objects Structure	48
3.3.4	Switching Capability	49
3.3.5	Explicit Label Control	49
3.3.6	Traffic Parameters	50
3.3.7	OSPF-TE	50
3.3.8	Available Label Set Sub-TLV	51
3.3.8.1	Inclusive Label Lists	51
3.3.8.2	Bitmap Label Set	51
3.3.9	Inter-AS TE Link Update	54
3.4	BGP-LS	55
3.4.1	BGP-LS session establishment	55
3.4.2	Topology exchange	56
3.4.3	Describing Elastic Optical Nodes and Links in BGP-LS	56
3.4.1	Describing a Flexi-Grid Link	58
4	Carrier SDN Orchestrator Architecture: Adaptive Network Manager	60
4.1	Motivation	60
4.2	Requirements	60
4.4	Modules within the architecture	63
4.5	Interfaces Definition	65
4.6	Summary of the Implementation Advances in the ANM	66
4.7	Implementation feedback	67
5	Interface between network and applications	69
5.1	North Bound Interface for the Adaptive Network Manager	69
5.1.1	North Bound Interface for Layer Zero Provisioning	69
5.1.2	North Bound Interface for IP Link Provisioning	69
5.1.3	North Bound Interface for MPLS Service Provisioning	70
5.1.4	North Bound Interface for Re-optimization Process	71
5.1.5	North Bound Interface for dynamic bandwidth allocation	71
5.1.6	ABNO REST API Summary	72
5.2	YANG Data Model for a Traffic Engineering Database of Optical Networks with Flex-grid	73
5.2.1	Optical Network Topology Model Overview	73
5.2.2	Main Building Blocks	75
5.2.2.1	Optical-TED.yang	75
5.2.2.2	Media-channel.yang	78



5.2.3	Example of Use	80
6	Traffic Monitoring and OAM	81
6.1	Monitoring Architecture	81
6.2	Bandwidth Estimation Algorithms	82
6.2.1	Correlation between flows and bits per second	82
6.3	Example of Use and Implementation Feedback	84
7	Conclusions	86
I.	Appendix YANG.1: optical-ted.yang file	87
II.	Appendix YANG.2: media-channel.yang file	96

1 Introduction

1.1 Executive Summary

IDEALIST project is focused around the design of high capacity elastic optical networks. While WP2 works on data plane technologies, that is the flexible and sliceable transponders, switching devices, nodes, etc, WP3 provides the software that controls and manages the data plane and is key to providing the elasticity.

IDEALIST control architecture is a combination of distributed protocols, which provide a real-time response and let the network survive against failures, and a centralized intelligence (Adaptive Network Manager) that, on the one hand, provides a point for optimization (e.g. interfacing with the planning tool), and on the other hand, interfaces with the applications, like Cloud or Video services, as shown in the figure below. The distributed functions are based on the well-known GMPLS architecture, while the centralized intelligence and interface with applications follows a SDN approach. Thus, the Adaptive Network Manager is the IDEALIST SDN Network Controller, that considers not only the Flexi-grid Network (the main focus of IDEALIST), but a wider scope, a multi-layer IP/MPLS over optical Network.

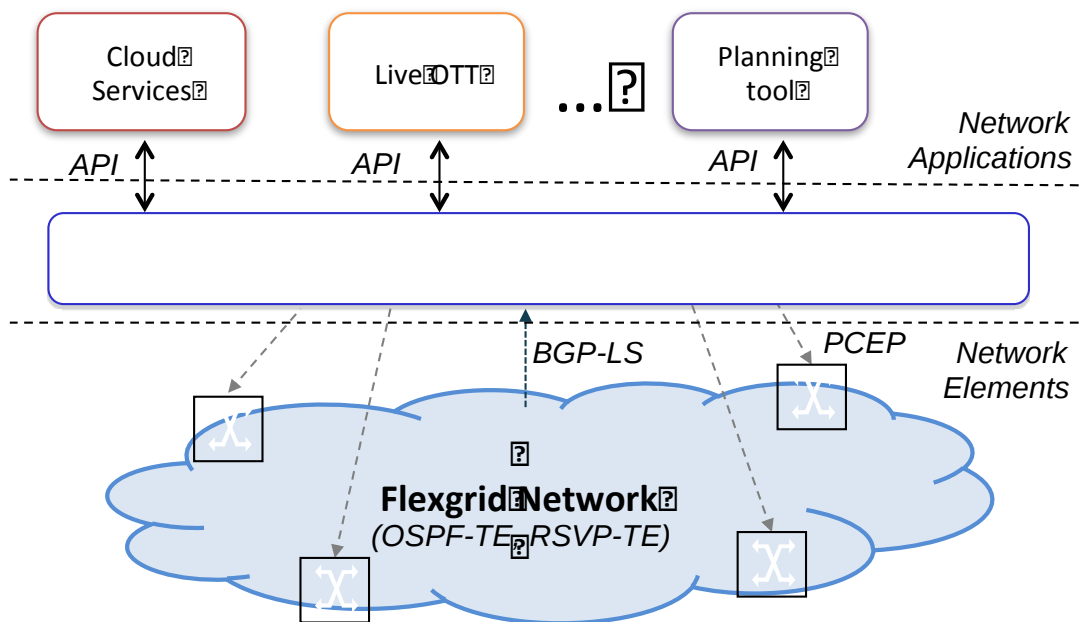


Figure 1 IDEALIST Control Plane Architecture

IDEALIST GMPLS / PCE Control Plane

The distributed control plane has two main functions, signaling, and routing. These functions in the IDEALIST approach are provided with the RSVP-TE and OSPF-TE protocols, which are enhanced to support the control of advanced functionalities developed in WP2, starting from the support of the flexible grid, filters, differential filter configuration and transceiver control (including code adaptation and multi-flow transponders).

Protocol Extensions for the Flexi-grid GMPLS / PCE Control Plane

The routing protocol (OSPF-TE), for example, is extended to support the dissemination of the availability of the spectrum, by modifying the available Label Set Sub-TLV.

As mentioned, one of the goals of the control plane software designed in WP3 is to provide an optimal use of the resources. In order to achieve such optimality, it is needed a mean to export the topology from the elastic optical network to an external entity with enough processing power and a holistic view of the network able to compute the optimal situation of the network. BGP-LS, a recent proposal in the IETF to export Link-State Traffic Engineering information, has been chosen as the protocol to export the traffic engineering information to the central SDN control. BGP-LS has been extended to support the TE capabilities of the flexi-grid network.

Another important protocol to achieve the optimality is the Path Computation Element Protocol, which serves various purposes. On the one hand, it facilitates optimum Path computation, using an Stateful PCE approach, and on the other hand is a tool for supporting Multi-Domain flexi-grid networks, through the Hierarchical PCE model.

Lastly, an interface with the network devices to send the provisioning orders is needed. Several protocols are suited for this work. IDEALIST has selected PCEP due to its mature syntax and low overhead.

Function	Protocol	Distributed/centralized
Signaling	RSVP-TE	Distributed (GMPLS)
Routing	OSPF-TE	Distributed (GMPLS)
Traffic Engineering Export	BGP-LS	Centralized
Path Computation	PCEP	Centralized
Instantiation (southbound)	PCEP	Centralized
Service Provisioning	NBI (REST API)	Centralized (SDN)
Topology Export	YANG / NETCONF	Centralized

Table 1 IDEALIST Control Plane Functions

Protocol	Extensions
RSVP-TE	New label defined. Multi-flow transponder support. Super-filter and Differentiated Filter Configuration Support.
OSPF-TE	Flexi-grid support (Available Label Set Sub-TLV extended)
BGP-LS	Flexi-grid support
PCEP	Flexi-grid support. Multi-domain support. ERO Format for Elastic Optical Path.
NBI (REST API)	NBI for L0 provisioning, IP Link provisioning, MPLS service provisioning, Re-optimization trigger, Dynamic Bandwidth

	allocation
YANG / NETCONF	Optical Network Topology Model (Optical TED yang model + media channel yang model)

Table 2 IDEALIST Protocol Extensions Summary

Carrier SDN Orchestration: Adaptive Network Manager

The SDN IDEALIST approach is complementing the distributed control plane and aims at orchestrating with network applications (such as cloud services), Operation Support Systems (OSSs) and Specialized offline Planning Tools (developed in IDEALIST in WP1, as described in D1.4). The IDEALIST proposed Architecture is called the Adaptive Network Manager, and has been taken to the standards as the “ABNO Architecture” (Application – Based Network Operation) by IDEALIST partners.

The architecture, as shown below, is based on the interaction of standard components, such as PCE, Provisioning Manager, VNTM, OAM Handler, and defines well-known interfaces among them. IDEALIST contributes by specifying in detail some of these interfaces.

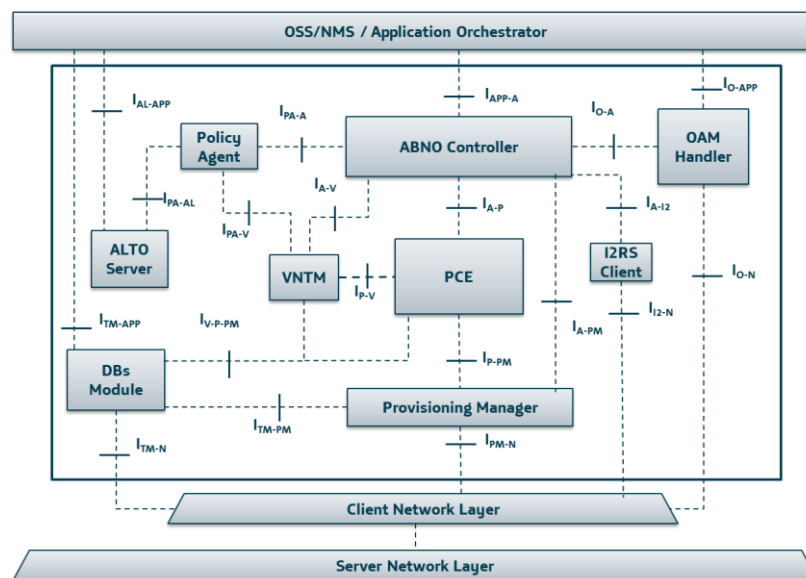


Figure 2. ANM / ABNO Architecture

The interfaces involved in SDN orchestration do not need to interact directly with Network Elements and thus, bit-level optimization and having the syntax attached to the protocol is not needed. Indeed, human readability and easiness to program are the qualities needed for this interfaces. ONF (Open Network Foundation) uses the term NBI (North-Bound Interface) for those interfaces with applications or OSSs that run on top of the SDN controller. IDEALIST has developed a set of NBI, a NBI for L0 provisioning, for IP Link provisioning, MPLS service provisioning, Re-optimization trigger and Dynamic Bandwidth allocation, which cover the use cases defined in D3.1.

With regards to topology interfaces, IDEALIST SDN approach is to define a detailed YANG model that describes the elastic optical network and can be exported to applications. This interface allows topology and inventory data to be modified, summarized, etc before being

consumed. E.g. a cloud based orchestration does not need the full flexi-grid details, just connectivity possibilities.

Traffic Monitoring

One of the key advances in IDEALIST is the use of traffic measurement tools to give feedback and take real-time decisions. In this document the architecture of the Traffic Monitoring module is presented

The monitoring architecture is shown in Figure 39. Monitoring probes will be connected to the span ports of the edge routers in order to examine incoming traffic. The monitoring probes will be in charge of generating network flows, that will be exported to the OAM. Inside the OAM, the DBA (dynamic bandwidth allocation) Monitor will execute bandwidth estimation algorithms and will send updates to the ABNO Controller.

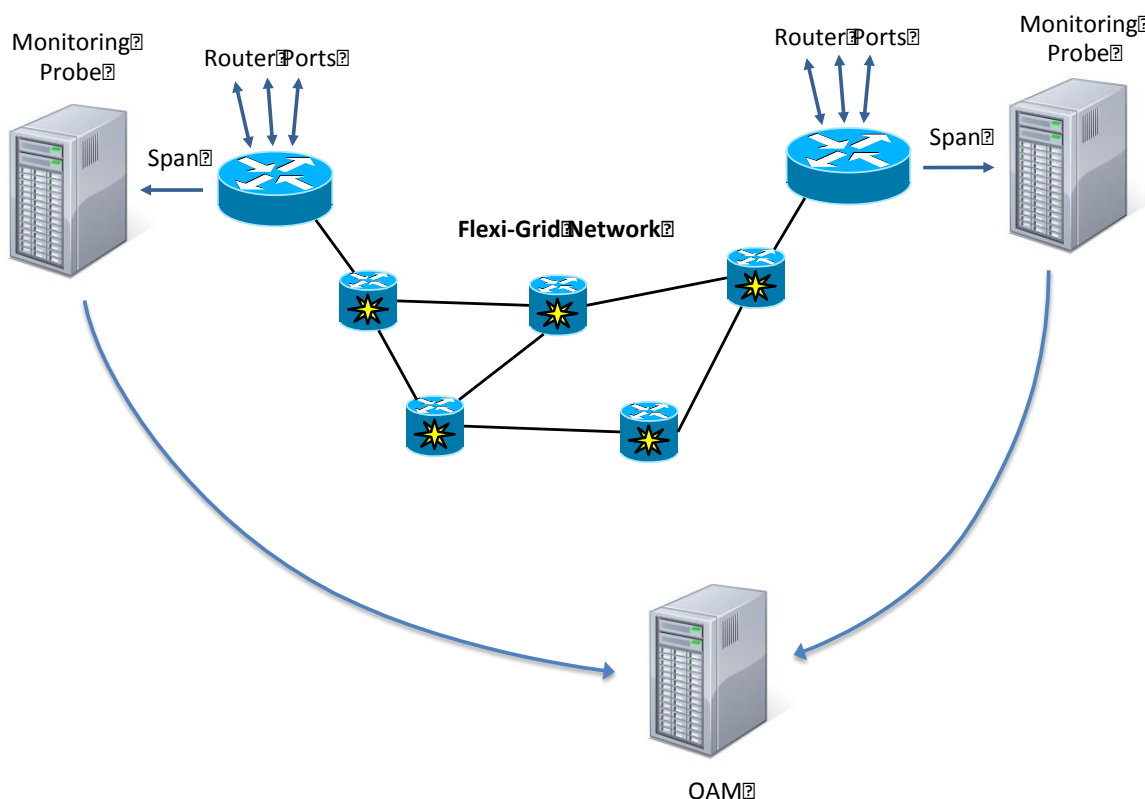


Figure 2. Monitoring architecture

1.2 Reference Material

1.2.1 Reference Documents

- [1] International Telecommunications Union, "ITU-T Recommendation G.694.1, Spectral grids for WDM applications: DWDM frequency grid, draft v1.6 2011/12", 2011.
- [2] International Telecommunications Union, "ITU-T Recommendation G.872, Architecture of optical transport networks, draft v0.16 2012/09 (for discussion)",

2012.

- [3] International Telecommunications Union, "ITU-T Recommendation G.800: Unified functional architecture of transport networks.", February 2012
- [4] O. Gonzalez de Dios, R. Casellas, editors, "Framework and Requirements for GMPLS based control of Flexi-grid DWDM networks", IETF draft-ietf-ccamp-flexi-grid-fwk, work in progress.
- [5] C. Margaria, O. Gonzalez de Dios, and F. Zhang, "PCEP extensions for GMPLS," IETF draft-ietf-pce-gmpls-pcep-extensions (work in progress).
- [6] Gredler et al., "North-Bound Distribution of Link-State and TE Information using BGP", IETF draft-ietf-idr-ls-distribution (work in progress).
- [7] L. Berger Ed., "Generalized multi-protocol label switching (GMPLS) resource reservation protocol-traffic engineering (RSVP-TE) extensions", IETF RFC 3473, January 2003.
- [8] K. Kompella and Y. Rekhter, "OSPF extensions in support of generalized multi-protocol label switching GMPLS", IETF RFC 4203, October 2005.
- [9] J. P. Vasseur Ed. and JL. Le Roux Ed., "Path computation element (PCE) communication protocol (PCEP)", IETF RFC 5440, March 2009.
- [10] G. Bernstein et al., "General Network Element Constraint Encoding for GMPLS Controlled Networks", IETF draft-ietf-ccamp-general-constraint-encode, work in progress.
- [11] A. Farrel, D. King, Y. Li, F. Zhang, "Generalized Labels for the Flexi-Grid in Lambda Switch Capable (LSC) Label Switching Routers", IETF draft-ietf-ccamp-flexigrid-lambda-label, work in progress.
- [12] M. Jinno, et. al., "Multiflow Optical Transponder for Efficient Multilayer Optical Networking", IEEE Communications Magazine, vol. 50, May 2012.
- [13] N. Sambo, et. al., "Sliceable Transponder Architecture Including Multiwavelength Source" IEEE/OSA Journal Optical Communications Networks, vol. 6, July 2014
- [14] X. Zhang, et. al., "GMPLS OSPF-TE Extensions in Support of Flexible Grid", draft IETF, draft-ietf-ccamp-flexible-grid-ospf-ext, June 2014.
- [15] R. Casellas, et. al., "Design and Experimental Validation of a GMPLS/PCE Control Plane for Elastic CO-OFDM Optical Networks", IEEE Journal on Selected Areas of Communications, vol. 31, January 2013.
- [16] D. King, A. Farrel, editors, "The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in MPLS and GMPLS", IETF RFC6805 [online: <http://tools.ietf.org/html/rfc6805>]
- [17] IDEALIST project, "D3.1: Reference GMPLS control plane, PCE and Adaptive Network Manager architectures" July 2013.
- [18] A. Farrel, editor, "Problem Statement and Architecture for Information Exchange Between Interconnected Traffic Engineered Networks", IETF internet Draft, work in progress, [online <https://datatracker.ietf.org/doc/draft-farrel-interconnected-te-info-exchange>]
- [19] M. Chen, R. Zhang, X. Duan, "OSPF Extensions in Support of Inter-Autonomous System (AS) MPLS and GMPLS Traffic Engineering", IETF RFC5392 [online:

<http://tools.ietf.org/html/rfc5392>]

- [20] A. Farrel, J.-P. Vassie, A. Ayyangar, “A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering”, RFC4726, November 2006
- [21] T. Otani, editor, “Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers”, RFC6205, March 2011.
- [22] E. Mannie, editor, “Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control”, RFC4606, August 2006 [online: <http://tools.ietf.org/html/rfc4606>]
- [23] Greg Bernstein, editor, “Operating Virtual Concatenation (VCAT) and the Link Capacity Adjustment Scheme (LCAS) with Generalized Multi-Protocol Label Switching (GMPLS)”, IETF RFC 6344, August 2011 [online: <http://tools.ietf.org/html/rfc6344>]
- [24] F. Zhang, editor, “GMPLS Signaling Extensions for Control of Evolving G.709 Optical Transport Networks”, IETF RFC7139, March 2014, [online: <http://tools.ietf.org/html/rfc7139>]
- [25] Y. Lee, editor, “Framework for GMPLS and Path Computation Element (PCE) Control of Wavelength Switched Optical Networks (WSOs)”, IETF RFC6163, April 2011, [online: <http://tools.ietf.org/html/rfc6163>]
- [26] R. Enns, M. Bjorklund, J. Schönwälder, A. Bierman, Eds. “Network Configuration Protocol (NETCONF)”, IETF Request for Comments 6241, June 2011.
- [27] M. Bjorklund, Ed., “YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)”, IETF Request for Comments 6020, October 2010.
- [28] J. Medved, S. Previdi, V. López, S. Amante, “Topology API Use Cases,” IETF internet draft draft-amante-i2rs-topology-use-cases-01, October 2013.
- [29] D. King, A. Farrel, “A PCE-based Architecture for Application-based Network Operations”, IETF internet draft draft-farrkingel-pce-abno-architecture-11.txt, August 2014.
- [30] A. Aguado, V. López J. Marhuenda, Ó. González de Dios and J. P. Fernández-Palacios, “ABNO: a feasible SDN approach for multi-vendor IP and optical networks,” Proc. Optical Fiber Conference (OFC), Th31.5, Mar 2014.
- [31] J. Schönwälder, “Network Configuration Management with NETCONF and YANG,” IETF 84 - Vancouver, BC, Canada, July 2012.
- [32] A. Clemm, H. Ananthakrishnan, J. Medved, T. Tkacik, R. Varga, N. Bahadur, “A YANG Data Model for Network Topologies”, IETF Internet draft draft-clemm-i2rs-yang-network-topo-00.txt, February 2014.
- [33] G. Galimberti, R. Kunze, Kam Lam, D. Hiremagalur, G. Grammel, Eds., “A YANG model to manage the optical interface parameters of DWDM applications,” IETF Internet Draft, draft-dharini-netmod-g-698-2-yang-00, July 2014.
- [34] Ó. González de Dios, V. López, C. Haya, C. Liou, P. Pan, G. Grammel, J. Antich, J.P. Fernández-Palacios, “Traffic Engineering Database dissemination for Multi-layer SDN orchestration,” Proc. European Conference on Optical Communication (ECOC), Mo.4.E.2, Sep 2013.

- [35] “Pyang – An extensible YANG validator and converter in python”,
<https://code.google.com/p/pyang/>
- [36] “PlantUML – Open-Source tool in Java to draw UML Diagram”,
<http://plantuml.sourceforge.net/>
- [37] Cugini, F.; Paolucci, F.; Fresi, F.; Meloni, G.; Berrettini, G.; Sambo, N.; Giorgetti, A; Foggi, T.; Poti, L.; Castoldi, P., "Benefits of active stateful PCE for flexgrid networks," Optical Fiber Communications Conference and Exhibition (OFC), 2014 , vol., no., pp.1,3, 9-13 March 2014
- [38] Paolucci, F.; Fresi, F.; Castro, A; Velasco, L.; Cugini, F.; Sambo, N.; Giorgetti, A; Poti, L.; Castoldi, P., "Filter optimization in SDN-based flexgrid networks," Optical Fiber Communications Conference and Exhibition (OFC), 2014 , vol., no., pp.1,3, 9-13 March 2014
- [39] Sambo, N.; Meloni, G.; Paolucci, F.; Cugini, F.; Secondini, M.; Fresi, F.; Poti, L.; Castoldi, P., "Programmable Transponder, Code and Differentiated Filter Configuration in Elastic Optical Networks," Lightwave Technology, Journal of , vol.32, no.11, pp.2079,2086, June1, 1 2014

1.2.2 Acronyms

ABNO	Application-Based Network Operations
ANM	Adaptive Network Manager
ASO	Application Service Orchestrator
BGP-LS	Border Gateway Protocol – Link State
BVT	Bitrate Variable Transponder / Transceiver
BV-OXC	bandwidth variable optical cross connect
CLI	Command Line Interface
CoS	Class of Service
DP	Dual Polarisation
EON	Elastic Optical Network
ERO	Explicit Route Object
FEC	Forward Error Correction
GCO	Global Concurrent Optimization
GMPLS	Generalized Multi-Protocol Label Switching
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IGP	Interior Gateway Protocol
ISIS-TE	Intermediate System to Intermediate System – Traffic Engineering
JSON	JavaScript Object Notation
LSA	Link State Advertisement

LSP	Label Switched Path
LSPDB	LSP Database
LSR	Label Switch Router
MAN	Metropolitan Area Network
MPLS-TE	Multi-Protocol Label Switching – Traffic Engineering
NCF	Nominal Central Frequencies
NMS	Network Management System
OAM	Operation, Administration and Management
OCh-P	Optical Channel Payload
OFDM	Optical Frequency Division Multiplexing
OLA	Optical Line Amplifier
ONF	Open Networking Foundation
OSPF-TE	Open Shortest Path First – Traffic Engineering
OTN	Optical Transport Network
OTS	Optical Tributary Signal
OXC	Optical Cross Connect
PCC	Path Computation Client
PCE	Path Computation Element
PLI	Physical Layer Impairments
PM-16QAM	Polarization Multiplexed 16 Quadrature Amplitude Modulation
PM-QPSK	Polarization Multiplexed Quadrature Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QoT	Quality of Transmission
QPSK	Quadrature Phase Shift Keying
ROADM	Reconfigurable Optical Add Drop Multiplexer
RSA	Routing and Spectrum Assignment
RSVP-TE	ReSerVation Protocol – Traffic Engineering
RWA	Routing and Wavelength Assignment
SA	Spectrum Assignment
SBVT	Sliceable Bitrate Variable Transponder
SDN	Software-Defined Networking
SNMP	Simple Network Management Protocol
SSON	Spectrum Switched Optical Network
SWG	Slot Width Granularity

TED	Traffic Engineering Database
TE-LSA	Traffic Engineering - Link State Advertisement
TLV	Type, Length, Value
UNI	User-Network Interface
VNTM	Virtual Network Topology Manager
WSON	Wavelength Switched Optical Network
WSS	Wavelength Selective Switch

1.3 Document History

Version	Date	Authors	Comment
ToC	08/09/2014	Telefonica/CTTC	Table of contents proposal
0.0	15/09/2014	WP3 team	Feedback to the ToC and new version.
1.0	30/09/2014	WP3 team	First complete version, some version drafts.
1.1	15/10/2014	WP3 team	Finished all versions
1.2	22/10/2014	Telefonica/CTTC	Initial review by editors.
2.0	27/10/2014	Quality Manager	Revised version with comments for final editorship
3.0	12/11/2014	Telefonica	Final version

1.4 Document Overview

D3.2 describes in detail the different building blocks of the IDEALIST control architecture presented in section 1.1. First, Sections 2-3 describe the GMPLS/PCE architecture for the data plane technologies defined in IDEALIST WP2 (i.e S-BVT, Flexgrid nodes, etc) are described in Sections 2-3. Afterwards, the centralized SDN orchestration architecture proposed in IDEALIST (Adaptive Network Manager) is described in Section 4. Finally, Network-Application and monitoring systems are described in sections 4-5..

2 GMPLS / PCE Control Plane Functional description

The distributed control plane being designed within IDEALIST is based on the GMPLS/PCE framework and protocols. This part of the control framework of IDEALIST is focused on the real time control of the elastic optical network. The main requirement of the control plane is the dynamic establishment and release of flexi-grid Label Switched Paths (LSPs), representing a media channel, which are switched in media channel matrixes. This dynamic establishment covers the recovery cases (restoration & protection in the flexi-grid layer).

2.1 Control Plane Architecture

IDEALIST main scope of the distributed control plan is the control of the so called media layer. A network media channel transports a single Optical Tributary Signal, or OTS (a particular example of OTS is the Optical Channel Payload, or OCh-P). The main requirement of the control plane is the dynamic establishment and release of flexi-grid Label Switched Paths (LSPs), representing a media channel, which are switched in media channel matrixes (cfr. Figure 1). GMPLS labels locally represent the media channel and its associated frequency slot. Network media channels are considered a particular case of media channels when the end points are transceivers.

The distributed control plane needs to provide functionality to discover in real time the network resources and disseminate the topology of the optical network, as well as the traffic engineering information. Moreover, an important function to be provided is an optimal path computation of the media channels.

2.1.1 Resource Discovery and Topology Dissemination

The Open Shortest Path First with Traffic Engineering Extensions (OSPF-TE) routing protocol is being extended to support the dissemination, via Link and Node Link State Advertisements (LSAs), of TE attributes that enable the building of a network topology view, commonly referred to as the Traffic Engineering Database (TED). Thus, the control plane needs to have a model of all the switching elements and their restrictions (e.g., devices may have a different minimum slot size or cannot support all sizes). The TED is used as an input in the path computation function. Although such computation can be deployed directly in the GMPLS controllers that constitute the control plane, different considerations such as the specifics of the optical layer technology or the benefits of a stateful Path Computation Elements (PCE) justify the choice for their deployment. This does not preclude the use of source based path computation or hybrid approaches combining PCE-based provisioning and source-based recovery.

2.1.2 Stateful Path Computation

The PCE architecture was proposed to provide effective constraint-based path computations. So far, the PCE has been mainly deployed with a stateless architecture, i.e. the PCE only relies on the TED which includes information on resource utilization. More recently, the PCE architecture has been extended with stateful capabilities, enabling the attributes of the established LSPs (e.g., the route) to be stored and maintained at the LSP State Database (LSPDB) [4]. Furthermore, a stateful PCE may also include the active functionality which enables the PCE to issue recommendations to the network, e.g. to dynamically update LSP parameters through the PCE Communication Protocol (PCEP). In IDEALIST, the (active) stateful architecture has been adopted to enable a number of

advanced traffic engineering functionalities, including elastic LSP operations and global defragmentation in flexi-grid networks. For example, the PCE is able to account for the actual network conditions, run complex re-optimization algorithms, and operate on existing LSPs to reduce the overall network fragmentation. The implementation of the stateful functionality has also to account for some deployment considerations, mainly related to reliability, synchronization (e.g., after restart) and scalability issues. In terms of scalability, the stateful PCE is not designed to be operated over the entire Internet. On the contrary, its domain of visibility has to be adequately dimensioned, considering a sufficiently over-provisioned system.

2.1.3 Signalling Aspects

2.2 Advanced Functionalities

2.2.1 Filter Control

In D2.2, two new data plane techniques have been presented, their aim is to reduce the detrimental filtering cascade effects. In this section, the two techniques are discussed in terms of control plane requirements and implementation solutions.

2.2.1.1 Differentiated Filter Configuration (DFC)

The first technique, called Differentiated Filter Configuration (DFC) [39], applies to a single optical connection (i.e., media channel). According to DFC, the passband of the filters traversed by the same connection can be configured to different values.

Figure 1 reports an example of connection transmission through a cascade of three nodes. In Figure 1a, a frequency slot of $m = 4$ is assigned to the connection. It is assumed that after the first and second node, acceptable Quality of Transmission (QoT) is experienced. However, after the third node, excessive detrimental filtering effects are experienced, preventing the actual setup along the three nodes with $m = 4$. Thus, larger bandwidth should be computed and configured in order to operate filtering within a flat region (i.e., avoiding the filter transition bands), thus limiting adverse filtering effects. Figure 1b shows the frequency slot configured, in all the three nodes, with $m=5$. In this case, adequate QoT is achieved, at the expense of more reserved spectrum, across all traversed nodes. Figure 1c shows the proposed DFC solution. In DFC, m can be configured with different values along the path, i.e. different effective bandwidth is configured in the filters traversed by the same connection. In particular, the first and second node are configured with $m = 4$, while $m = 5$ is applied to last node. This way, no additional detrimental filtering effects are introduced by the third node, and in particular by its filter, which is traversed by the lightpath in its flat region. As a result, spectrum reservation can be minimized on a per node basis according to the expected QoT, thus improving the overall spectrum utilization. Such a technique can be applied both to single carrier and super-channel.

DFC is then effective in the presence of detrimental filtering effects or in the case of networks where filters based on different technologies (e.g., BV-WSS and WSS) are present.

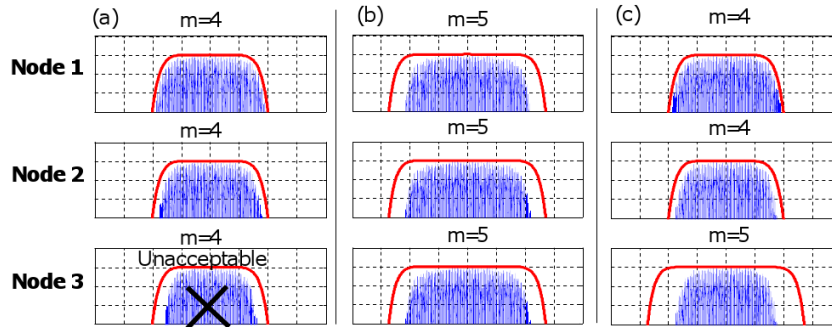


Figure 2. Filtering cascade with: (a) $m=4$, unacceptable filtering effects; (b) $m=5$, acceptable filtering effects; (c) Differentiated Filter Configuration, achieving spectrum saving and acceptable filtering effects [D2.2].

In GMPLS-controlled optical networks, each node configures the filter bandwidth according to the central frequency n and the slot width m described by the tuple (n, m) included within the RSVP-TE protocol (see D3.1 for details).

Network nodes can derive the value of m from the requested effective frequency slot included within the RSVP-TE TSPEC. The reserved effective frequency slot is then mapped into the RSVP-TE FlowSpec.

In addition, mainly to deal with possible hardware constraints in filter configurations, the standard GMPLS protocol encompasses the possibility to specify different values along the path. In particular, the explicit routing with Explicit Label Control (ELC) procedure allows the ingress node to specify for each traversed outgoing link the related (n, m) configuration.

The DFC technique can be successfully applied by relying on ELC. In this case, however, the value specified by the effective frequency slot has to be ignored. Indeed, with reference to Figure 1, a value of $m = 4$ would not guarantee adequate quality of transmission while a value of $m = 5$ would imply the reservation of additional spectrum resources.

The control plane implementation of the proposed DFC is also suitable for software defined networking (SDN). Indeed, SDN implementations, e.g. based on OpenFlow, allows to directly and independently control, besides the ingress and egress nodes, also any intermediate node through the specific frequency slot and filtering parameters. That is, DFC can be implemented in SDN without additional specific extension.

In [39], a DFC control plane implementation is presented and validated in the context of an SDN-based architecture. In particular, DFC is applied to a Terabit/s super-channel connection, where two different types of filter configurations are applied along the path (i.e., 225GHz and 200GHz).

Figure 2 shows the two different OFPT_FLOW_MOD messages sent to the ingress node and to an intermediate node. In this specific example, the two nodes also support different technologies (fixed grid and flexible grid), demonstrating that DFC can be successfully considered also in the context of network upgrades and mixed solutions (see [39] for additional details).

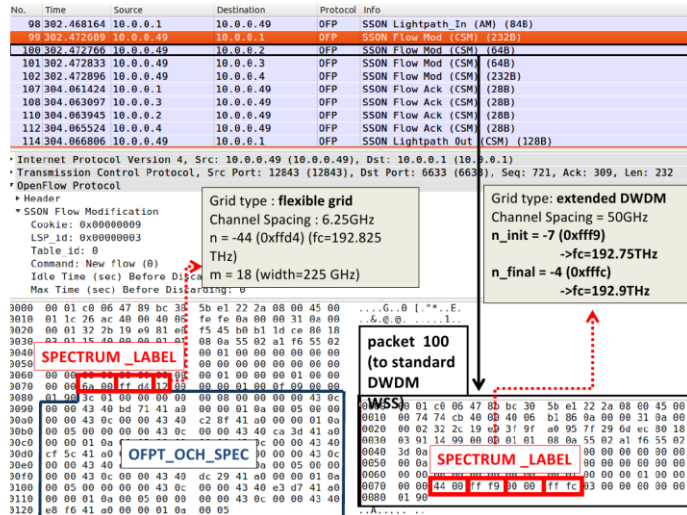


Figure 3: Control plane messages (SDN) implementing the differentiated filter configuration (DFC) technique

2.2.1.2 Super-filters

The second technique proposed within the IDEALIST project to reduce detrimental filtering effects and improve the overall network spectrum efficiency is called *super-filter* [38]. The technique applies to multiple optical connections.

A super-filter consists in the aggregation of multiple independent filter configurations related to different connections that flow through common node output ports.

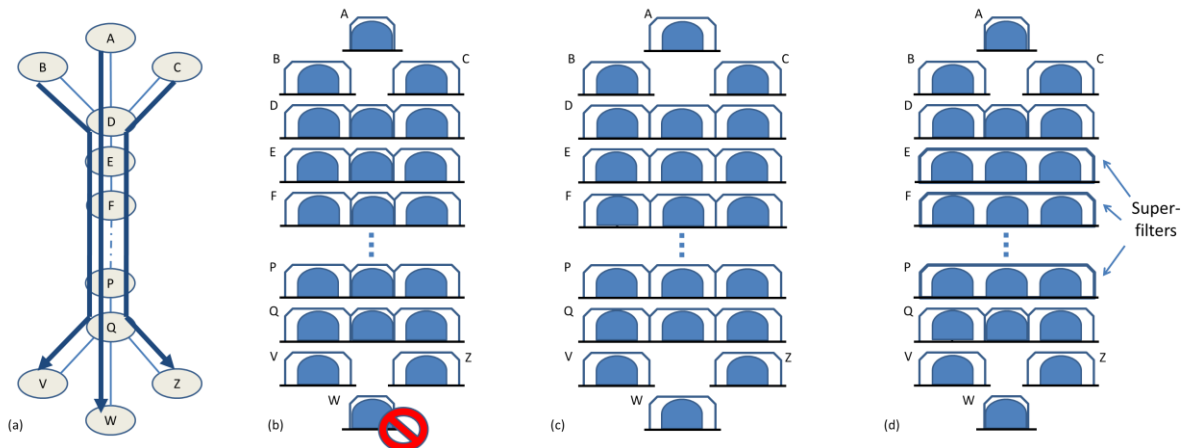


Figure 4: (a) reference network; (b) 25GHz reserved to A-W but unacceptable QoT is experienced; (c) 37.5GHz reserved to A-W at the expenses of additional reserved resources; (d) Super-filter technique.

An example is reported in Figure 3. Then Figure 3a shows the reference network scenario including three lightpaths.

In Figure 3b, lightpath A-W traverses filters grid configured with $m=2$ (e.g., 25 GHz). In this case, lightpath A-W experiences excessive detrimental filtering effects.

In Figure 3c, lightpath A-W traverses filters grid configured with $m=3$ (e.g., 37.5 GHz). QoT is guaranteed at the expenses of additional reserved spectrum resources.

In Figure 3d, the super-filter technique is applied. A unique filter for all three lightpaths is configured on E-P. These nodes do not induce additional detrimental filtering effects and QoT on lightpath A-W is guaranteed even if it occupies only 25 GHz.

The proposed super-filter technique accounts for the actual configuration of flexi-grid filters in the network nodes. That is, filter configuration is computed such that different lightpaths (e.g., with different source or destination nodes) can co-exist within the same flat region of a single filter configuration.

Moreover, differently with respect to traditional networking solutions, also in case of tear down, the behaviour has to be modified. With reference to the figure above, if the B-V lightpath is torn-down, control plane has to account for the presence of the adjacent connections. Thus, all the A-W spectrum resources will not be completely released. In particular, a slice of 12.5GHz contiguous to E-P resources will be maintained reserved to A-W.

In GMPLS/PCE-controlled optical networks, the implementation of super-filters is quite complex. Indeed, node resources are reserved according to head-end lightpath parameters. That is, hierarchical LSP instances would be required to aggregate multiple lightpaths and apply a unique filter configuration. However, such hierarchical LSP may even be only one-hop long, driving to inefficient control plane solutions.

On the other hand, the super-filter technique appears to be particularly suitable for the SDN architecture. In [38] the super-filter technique is introduced, evaluated and successfully implemented in a testbed controlled by a specifically enhanced SDN-based architecture.

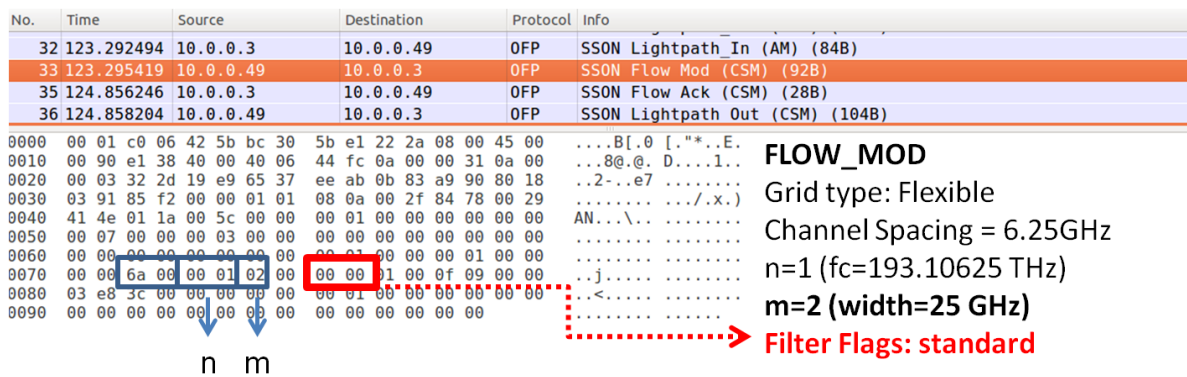


Figure 5. Message capture at node-controller of the ingress node A (IP: 10.0.0.3).

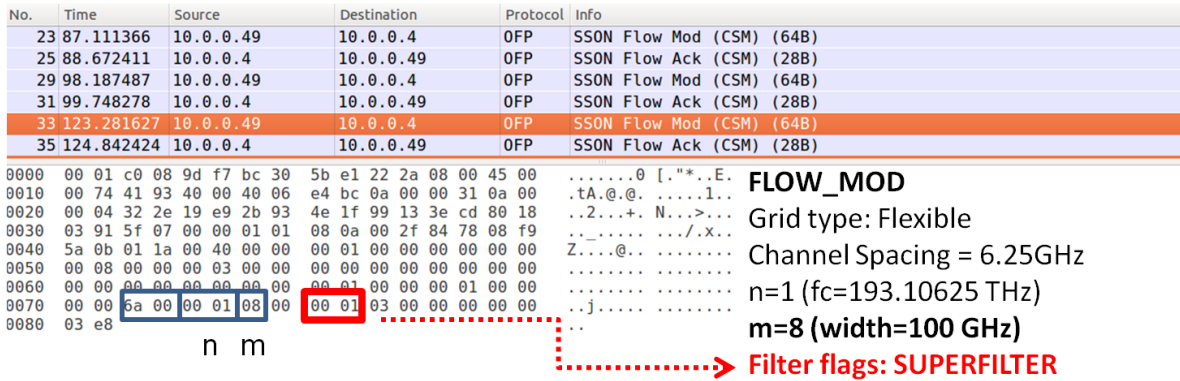


Figure 6: Message capture at node-controller of the intermediate node E applying super-filters (IP: 10.0.0.4).

2.2.2 Transceiver Control

The use of Multi-Flow Optical Transponders (also referred to as Sliceable Bandwidth Variable Transceivers, SBVTs) is considered as a key technology to increase the flexibility and re-configurability in flexgrid networks [12][13]. Specifically, MF OTP allows generating multiple optical flows (sub-carriers) that can be flexibly associated to client layer demands (IP). Such flows can be either aggregated (forming super-channels) or directed towards different destinations based on the traffic demands. In the following, we present two control plane architectural solutions for the control and configuration of MF OTPs. The former relies on an active stateful PCE which aims at performing adjustments on the reserved resources (e.g., number of sub-carriers) to deal with the degradation of the transmission quality. The second solution addresses the required extensions to GMPLS protocols (routing and signaling) for the control and configuration of MF OTPs at the LSP endpoints. Such extensions pave the way to devise a distance-adaptive Routing, Spectrum and Modulation Assignment (RSMA) algorithm which dynamically computes flexgrid paths between two remote MF OTPs.

2.2.2.1 Elastic Operation on Code Adaptation

The Sliceable Bandwidth Variable Transponders (SBVT) designed in WP2 enables the dynamic adaptation of several transmission parameters, including modulation format, number of active sub-carriers, and applied Low-Density Parity-Check (LDPC) coding. The adaptation of such parameters requires adequate control plane mechanisms.

In this section, we designed and validate a control plane architecture and solution enabling the dynamic adaptation of both the number of active sub-carriers and the LDPC coding.

An active stateful PCE architecture has been designed (Figure 6a) and applied to a flexgrid optical testbed (Figure 6b) also including Spectrum Selective Switches (SSS) and a Bandwidth Variable Transponder (BVT) implementing up to 1Tb/s transmission through 7 carriers at up to 160Gb/s PM-QPSK transmission with coherent detection. The transmission relies on time-frequency packing technique through configurable LDPC coding, e.g. of type 8/9 (8 information bits out of 9 transmitted). More redundant coding could also be applied, e. g. of type 4/5, thus enabling more robust transmission.

Adjustments on reserved spectrum resources of existing lightpaths can be operated either to cope with requests for additional bit rate or to deal with the degradation of transmission performance. The implemented active stateful PCE has been here applied to address the

latter case. To reveal the degradation, monitoring on post-FEC Bit Error Rate (BER) and the variance of the acquired data samples at the 1Tb/s coherent receiver has been implemented. The variance provides useful information on the quality of the received signal and indicates whether a working-limit condition is approaching for the code in use before post-FEC errors occur. Indeed, when no post-FEC BER degradation is experienced, but a pre-defined threshold on error variance is exceeded (e.g., 0.06 for 8/9 coding), a warning alarm is locally triggered.

According to pre-defined policies, a notification message (i.e., PCEP Ntf extended with a dedicated QoT object, see Figure 6c) is then sent to the PCE to notify the degradation and trigger the improvement of the transmission robustness. The PCE elaborates the parameters to adapt the LSP, considering network resources, modulation format, coding, or a combination of the above. In this experiment, the PCE computes LDPC coding adaptation from 8/9 to 4/5. To preserve the information rate, an additional carrier is also dynamically activated. This imposes an increase of the occupied spectrum resources, from 200GHz to 237.5GHz. To make room for such increase, the PCE has the capability to first trigger network de-fragmentation. Path computation can be performed by elaborating a strategy where a combination of rerouting operations of existing paths is considered. To this extent, the PCE possibly considers non-disruptive de-fragmentation operations, including the push-pull technique implementing the hitless shift of lightpaths. The computed strategy can be applied thanks to the active functionality. The Action Handler module is utilized to synchronize re-optimization operations, thus handling deadlock conditions and enabling the de-fragmentation of the network resources contiguous to the considered LSP.

To improve transmission robustness, the PCE sends a PCUpd message to increase the reserved spectrum resources of the degraded LSP and apply the code adaptation (Figure 6c). A novel Explicit Route Object (ERO) sub-TLV is enclosed to specify the updated LDPC code rate. The source node runs the RSVP-TE signaling (using make-before-break mechanism) to adapt the node configurations along the path. Then, the new coding is activated in the transmitted data: within the preamble of each data block, a specifically introduced 3-bit field is configured as part of the overhead to communicate to the receiver the applied 4/5 coding.

This way, the receiver becomes aware of the code to apply. It processes incoming data through the new coding, successfully supporting the more robust transmission which aims at preventing post-FEC BER degradation. Code adaptation is performed with no traffic disruption.

Figure 6a reports the collected variance statistics before and after degradation is induced. The figure also shows the related spectral efficiency of the transmitted data. Figure 6b shows the variance performance vs. OSNR. For each code, the minimum reported OSNR value is the one which guarantees error free post-FEC BER. A margin of 1dB is here considered to fix the variance threshold value and guarantee error-free operations. The bars at the bottom of Figure 6b summarize the range of use for each considered LDPC code (i.e., the OSNR value at the left side indicates the variance threshold). Additional details can be found in [37].

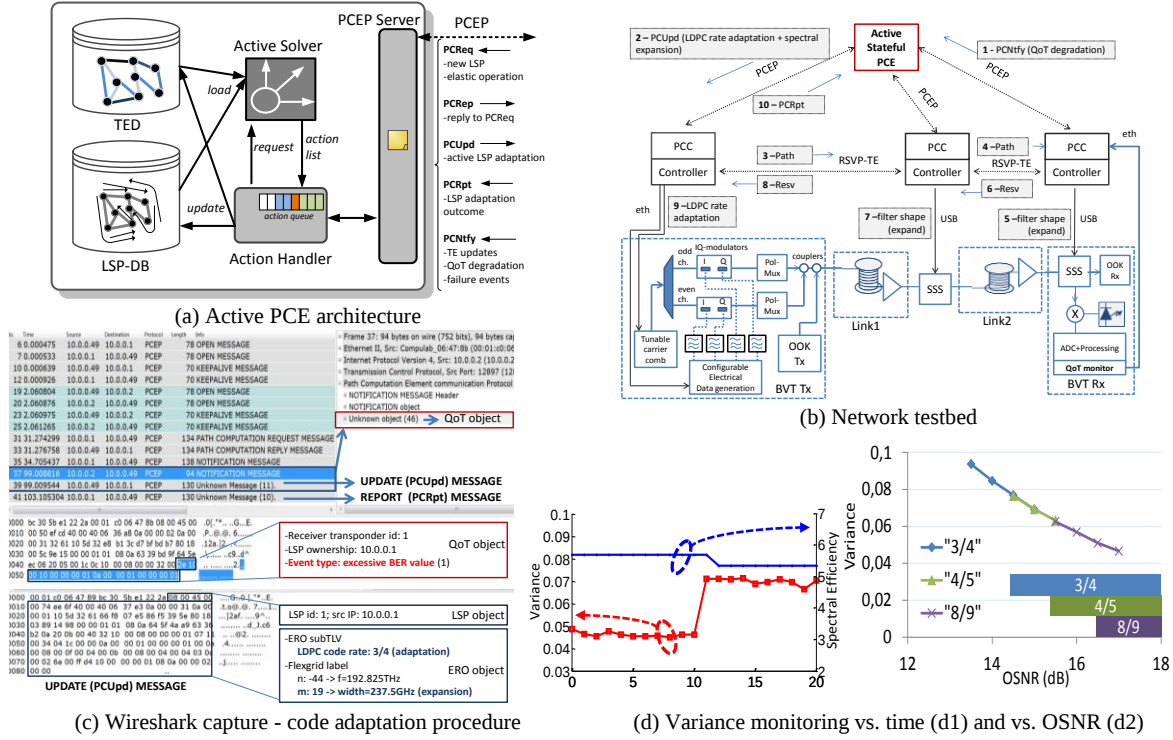


Figure 7: elastic operation on code adaptation

2.2.2.2 GMPLS/PCE Control and Configuration of MF OTPs

The considered MF OTP follows the architecture described in [12]. That is, several laser sources (sub-transponders) are equipped in the MF OTP. Each sub-transponder is then able to generate a single optical flow / sub-carrier. The generated sub-carriers are then multiplexed over a single line interface connected to the optical switch node (BV OXC).

Relying on the above MF OTP architecture, let's assume the dynamic establishment of LSP requests (r) between two remote MF OTPs. For the sake of completeness, herein we consider that all LSPs are triggered / initiated by an active PCE. Each demand r must specify the endpoints (source s and destination d nodes) as well as the bit rate (b in Gb/s). A RSMA algorithm is then launched at the active PCE to compute a feasible path for r . If the RSMA succeeds, a PCInitiate messages is sent from PCE to the ingress node of r with the spatial path (i.e., nodes and links) and spectral resources (Frequency Slot, FS) for accommodating r . MF OTPs enables that one or several co-routed floes contiguous in the spectrum (super-channel) are used. Therefore, besides the physical path and selected FS, the PCE response must also specify MF OTP-specific resources to be configured at LSP endpoints, such as selected sub-carriers (IDs), modulation format, optical spectrum (n and m) per sub-carrier, etc. Consequently, the RSMA needs details of the MF OTP attributes and capabilities to perform such path computations.

The TE and resource information of MF OTPs are flooded by the routing protocol and gathered as a Traffic Engineering Database and provided to the PCE. Three different MF OTP flooding information models are considered:

- No information: RSMA only operates with the topology and network link attributes between BV OXCs. The selection/configuration of MF OTP resources is locally realized at the endpoint.

- Partial model: the PCE's TED collects limited details of the MF OTP interfaces, namely the number of available sub-transponders.
- Full model, the partial view is extended with aggregated optical spectrum utilization (for both Tx and Rx directions) over the MF OTP - BV OXC interface. This would facilitate the RSMA selecting a feasible FS avoiding the overlapping with the optical spectrum occupied by other optical flows/sub-carriers.

Herein, we exclusively focus on partial and full models. For a request r the RSMA must compute a (shortest) path ensuring sufficient continuous and contiguous unused optical spectrum through the entire path. The FS is selected from a set of feasible solutions using the first-fit heuristic. Bearing this in mind let's use the example shown in Figure 7a to illustrate the performance differences achieved by either partial or full models.

In the example, 3 LSPs (P1, P2 and P3) are sequentially established. P1 is a super-channel with 2 sub-carriers (at both MF OTP C and D) and is set up through the path 2C-4-3B and occupies FS 1-4. Next, P2 is set up between 1A and 4D. Regardless of the MF OTP information model, P2 is provisioned as a super-channel flow (2 sub-carriers) via the path 1A-2-4D and allocates the FS 5-8. Observe that P2 is set up over an available FS without optical spectrum collision with P1. Finally, P3 must be set up from 1A to 3B. Using MF OTP partial model (i.e., no NCF utilization of MF OTP interfaces), P3 is computed through the path 1A-3B allocating the FS 1-2. Nevertheless, although such FS is available on the link 4 (nodes 1-3), P3 would fail (see Figure 7b) since that FS is already used by P1. Thus, partial model complicates the FS selection in MF OTP interfaces. The full model mitigates such problems, allowing the RSMA using both the available NCFs on every network link and the aggregated spectrum (NCFs) status over the MF OTP interfaces, as shown in Figure 7c: the RSMA computes P3 through the route 1A-3B but selecting the FS 9-10 which does not incur on spectrum collision with existing P1 and P2.

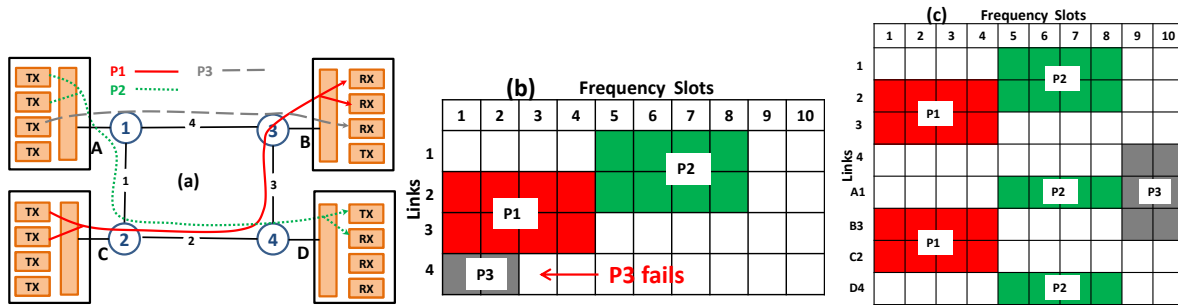
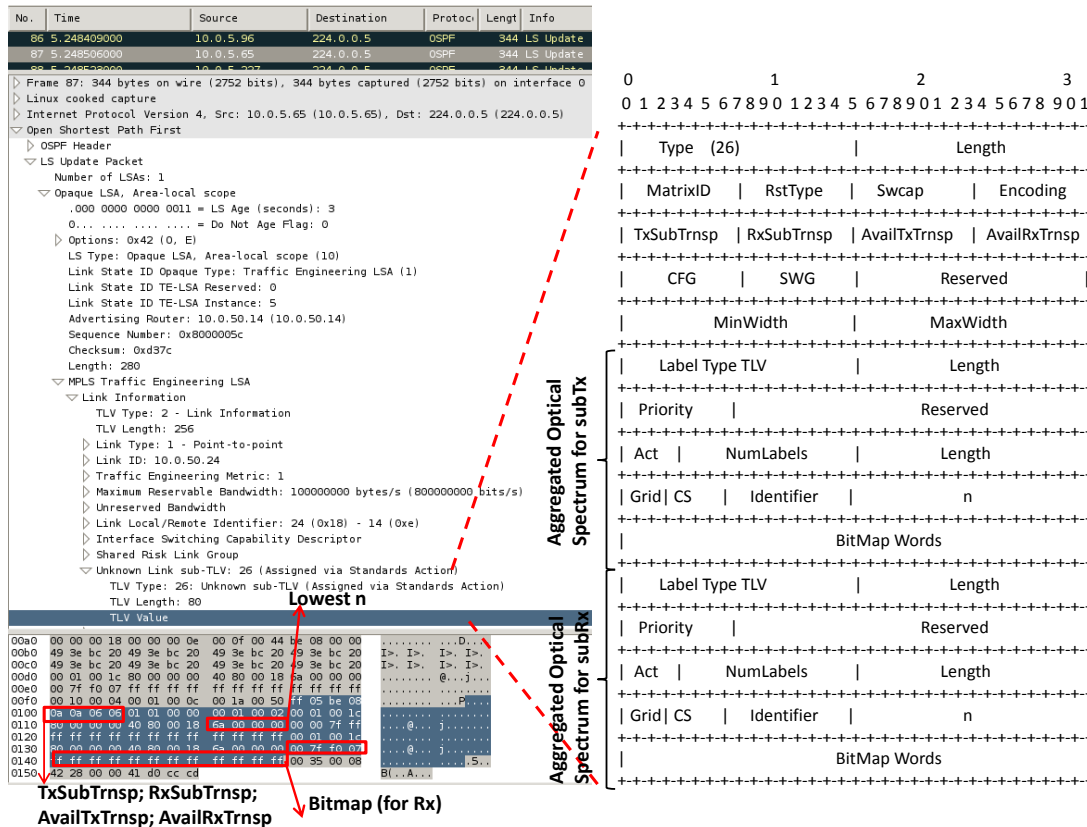


Figure 8 (a) Example of Spectrum Allocation for P1, P2 and P3 LSPs using: (b) Partial vs (c) Full MF OTP Information

The strict control of MF OTPs within GMPLS requires specific extensions to both routing (OSPF-TE) and signaling (RSVP-TE) protocols. In the following the proposed and implemented enhancements to such protocols are thoroughly detailed in Figure 8.

○ MF OTP-specific OSPF-TE Protocol Extension

For both full and partial MF OTP information models, the OSPF-TE floods MF OTP TE attributes using the Port Label Restriction (PLR) sub-TLV. To this end, we extend the PLR sub-TLV solution proposed in [14]. In particular, focusing on the full model (which has the most completed information of each MF OTP), it is carried both the total equipped and unused number of sub-transponders for each direction (i.e., $TxSubTrnsp$, $RxSubTrnsp$, $AvailTxTrnsp$ and $AvailRxTrnsp$) as well as the aggregated NCF status per direction on the interface with the BV OXC. For the latter, a bitmap coding is used where a NCF is represented by a bit: '1' available and '0' occupied. The detailed MF OTP attributes carried by the implemented PLR sub-TLV are depicted in Figure 8.



- MF OTP-specific RSVP-TE Protocol Extension

Once the route is computed, the path is passed as ERO to the RSVP-TE to set up the LSP. The ERO contents or sub-objects (Figure 9) are formed by unnumbered interface IDs, followed by the Label subobject specifying the FS (i.e., n and m), and at the LSP endpoints, the new *Transponder* subobjects. The latter allows s and d configuring their respective MF OTPs. The proposed Transponder subobject (Type 10) is formed by a variable list of Transponder TLVs. This allows the configuration of a set of sub-carriers forming a super-channel LSP.

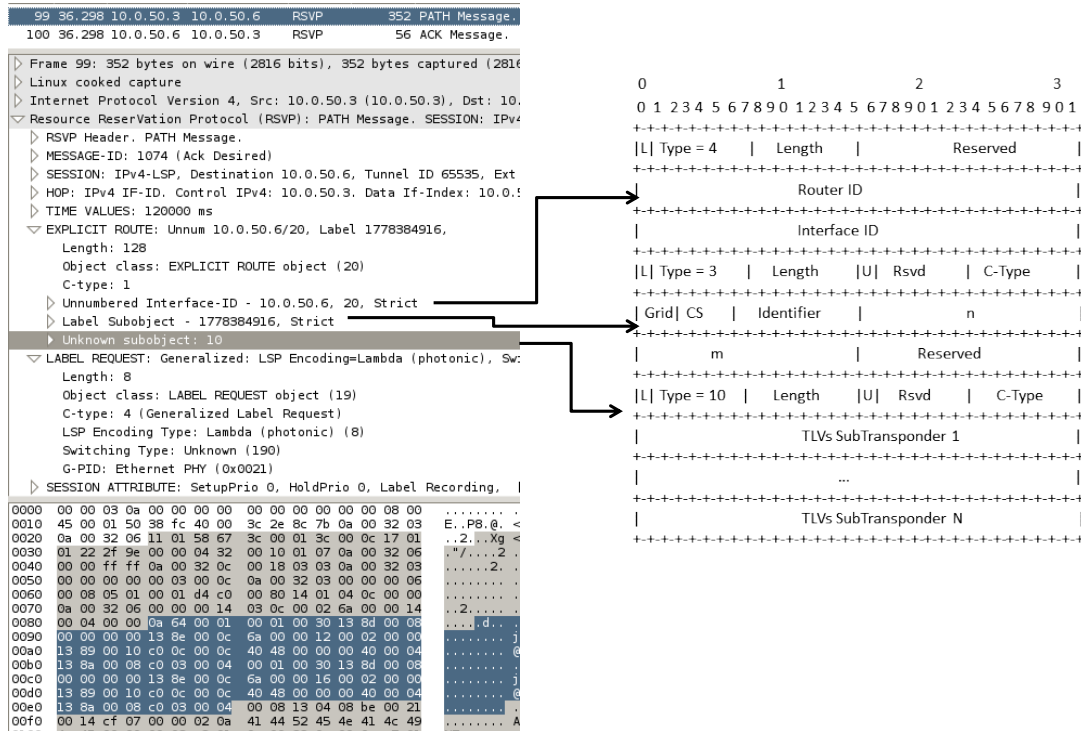
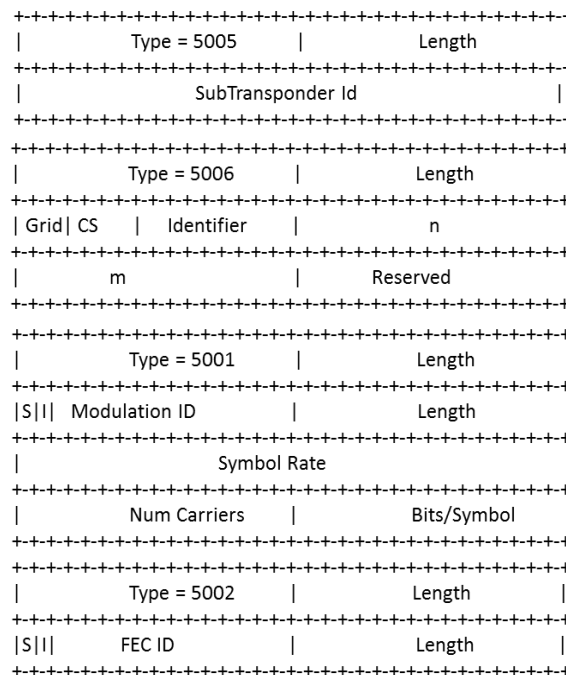


Figure 10 Transponder Sub- Object in RSVP-TE ERO Sub-Object

Each Transponder TLV (Type 1) contains 4 sub-TLVs (Type 5005, 5006, 5001 and 5002) used to determine: the sub-transponder ID, the sub-carrier FS, the modulation format and the FEC, respectively (Figure 10).



**Figure 11 Transponder TLV contents: Transponder Id, sub-carrier FS, Modulation
Format and FEC**

Proposed online distance-adaptive RSMA algorithm

The GMPLS protocol extensions for MF OTP control and configuration allow devising RSMA algorithms to compute routes between pair of remote MF OTPs. In this regard, an iterative Constrained Shortest Path First (CSPF) mechanism is proposed where for each demand r the shortest path (with respect to the TE metric) is computed to achieve the most efficient use of both the optical spectrum and the MF OTP resources. The algorithm output is composed of the spatial path, the FS and the selected sub-transponders attributes (i.e., FS and modulation format). The RSMA algorithm operates iteratively starting from the most efficient modulation format until a feasible solution is found [15]. If the algorithm succeeds, the ERO is constructed and sent to the s node. If all modulation formats are checked and no solution is found, r is blocked.

For a given modulation format (mf_i), first it is checked whether the requested bit rate (b) is supported by mf_i . We consider that each MF OTP has N sub-transponders/sub-carriers operating at a fixed baud rate (25 Gbauds/s). Hence, if b is not multiple of the transport capacity of a sub-carrier at mf_i (i.e., using one or more sub-carriers), then mf_i is discarded. Otherwise, the number of required sub-carriers (n_{sc}) is computed. Next, at s and d nodes, it is verified whether n_{sc} sub-transponders are available; if not, mf_i is discarded. After that, the FS width (SW in GHz) to accommodate r using n_{sc} at mf_i is computed; SW is obtained multiplying the FS width of each sub-carrier (e.g., $m = 2$ corresponding to 25 GHz using a channel spacing of 6.25 GHz) by the required n_{sc} sub-carriers. Next, the Dijkstra algorithm is triggered to compute a path from s to d ensuring the spectrum continuity constraint (SSC) for SW GHz. If a path is found, it is checked whether the total path length (in Km) is supported by mf_i . That is, the RSMA applies a distance-adaptive mechanism where the computed path length must not exceed the maximum supported by mf_i . If this occurs, mf_i is discarded. Otherwise, the route is returned to the ingress node. For the sake of completeness, the path length is computed since the TE metric of each network links represents the distance (in Km) of such a fiber link.

The above RSMA algorithm is applied for both partial and full models. In the full model, however, the aggregated optical spectrum availability on the MF OTP interfaces at s and d is added to ensure the SSC for SW GHz. This would avoid optical spectrum collision with other occupied sub-carriers in the same MF OTP.

Given the following notation:

- s and d being the source and destination nodes
- B is the requested bit rate in Gb/s
- Mf : Set of spectral efficiency ordered modulation formats; $|Mf|$: number of modulation formats
- F_n : set of MF OTPs at node n ; $|F_n|$: number of MF OTPs at node n
- Tf,n : set of available sub-carriers for Tx at MF OTP $f \in F_n$; $|Tf,n|$: number of available sub-carriers for Tx at MF OTP $f \in F_n$
- Rf,n : set of available sub-carriers for Rx at MF OTP $f \in F_n$; $|Rf,n|$: number of available sub-carriers for Rx at MF OTP $f \in F_n$
- UTf,n : set of occupied sub-carrier for Tx at MF OTP $f \in F_n$;
- URf,n : set of occupied sub-carrier for Rx at MF OTP $f \in F_n$;
- n_{sc} : number of required sub-carriers at s and d according to $m \in M$;
- W : in GHz, optical spectrum occupied by any sub-carrier $t \in Tf,n$ or $r \in Rf,n$;
- SW : in GHz, required optical spectrum by d according to $m \in M$; $SW = W * n_{sc}$
- Pd,m : Shortest Path dealing with optical spectrum continuity for SW
- Dp,m : distance in Km for Pd,m
- $MaxD,mf$: maximum distance supported in Km for modulation format mf

The pseudo-code of the RSMA algorithm is:

```

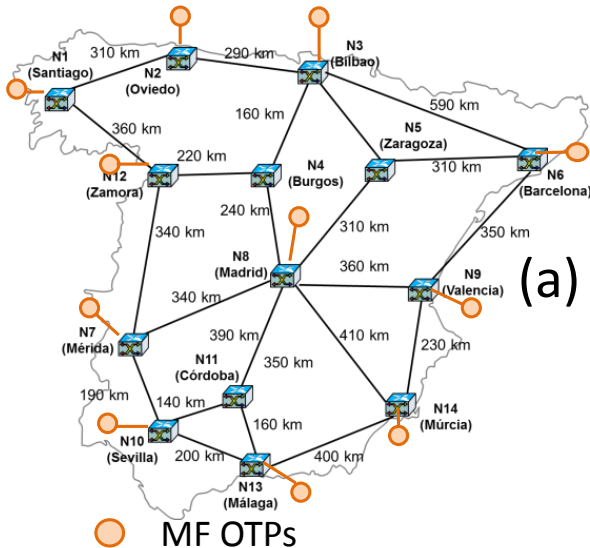
1. For all  $mf \in Mf$ 
2.   If  $B$  is not supported by  $mf$ , then continue //check table of modulation formats
3.   Compute  $n_{sc}$  //Using  $mf$  compute required number of sub carrier to support  $B$ 
4.   For each  $f \in Fs$  //for each MF OTP check whether sufficient available Tx sub carriers at  $s$ 
5.     If  $|Tf,s| < n_{sc}$  then  $UTf,s = NULL$ 
6.     Else compute  $UTf,s$  //select using FF the  $n_{sc}$  available Tx sub carriers
7.   end for
8.   For each  $f \in Ft$  //for each MF OTP check whether sufficient available Rx sub carriers at  $d$ 
9.     If  $|Rf,t| < n_{sc}$  then  $URf,d = NULL$ 
10.    Else compute  $URf,d$  //select using FF the  $n_{sc}$  available Rx sub carriers
11.  end for
12.  If  $!UTf,s$  OR  $!URf,d$  then continue
13.  else
14.    compute SW // compute the required optical spectrum considering the number of sub carriers
15.    compute  $Pd,m$ , // compute SP ensuring SSC for SW and  $MaxD,mf$  and is ensured
16.    if  $Pd,m == NULL$  then continue
17.  else
18.    Generate ERO using  $p$  // ERO formed by  $p$  (route, FS) +  $URf,s + URf,d$ 
19.  end for

```

Figure 12 Pseudo-code for the Proposed RSMA Algorithm Considering MF OTP Attributes at the Endpoints

Experimental performance Evaluation

Figure 12a shows network topology (Spanish optical core network) used for the experimental results. The transport data is emulated and the results are obtained at the control plane level. Selected nodes are equipped with a MF OTP. The MF OTP has 10 sub-transponders supporting 3 modulation formats: DP-QPSK, DP-8QAM and DP-16QAM. The LSP requests arrive according to a Poisson process with mean inter-arrival time of 10s. The Holding Time (HT) is exponentially modeled whose mean is varied to obtain different traffic values. The s and d nodes are randomly selected. The requested bandwidth is uniformly distributed as multiples of 100 Gb/s up to 500 Gb/s. Table in Figure 12b depicts the maximum path length for the each mf_i along with the number of required sub-carriers for a b bit rate. Each sub-carrier has a fixed FS width of $m = 2$ (i.e., 25 GHz) and the network links support 128 NCFs. A 400 Gb/s LSP using 2 sub-carriers at DP-16QAM uses a FS width of 50 GHz whilst the same LSP using 4 sub-carriers with DP-QPSK requires a FS width of 100 GHz.



(b)

	DP-QPSK	DP-8QAM	DP-16QAM
Bits/symbol	4;	6;	8;
Bit rate (Gb/s)	100	150	200
Max. Path(Km)	3000	1000	650
bit rate (Gb/s)	# of sub-carriers (n_{sc})		
100	1	-	-
200	2	-	1
300	3	1	-
400	4	-	2
500	5	-	-

Figure 13 Network Topology w/ MF OTPs; (b) Number of Sub-carriers per mf_i and bit-rate

For all HT values, the full MF OTP model does lower the connection Blocking Probability (BP) compared to the partial model. LSP failures are split into: path computation failure due to Routing, Modulation and Spectrum Assignments (RSMA) and signaling errors Figure 13. In both models RSMA errors are mainly caused by the lack of available sub-transponders. On the other hand, signaling errors occur when the LSPs are being actually set up. The main difference between both models lies on the signaling errors. In the partial model these errors principally happen due to the difficulties to deal with the SSC. Thus contention on the MF OTP interfaces with existing LSPs occurs. Applying the full model, these problems are mitigated. Indeed, the RSMA algorithm is aware of the optical spectrum status of MF OTP interfaces which in turn facilitates dealing with the end-to-end SSC. Signaling errors appearing in the full model could be due to either concurrent LSPs being signaled or the RSMA algorithm operates with out-of-dated PCE's TED. Observe that the differences with respect to the setup and PCE delays are negligible, since both models use (practically) the same RSMA algorithm.

(c)		Delay (ms)		Error		
HT (s)	MF OTP model	RSMA	Setup	RSMA	Signaling	BP (%)
25	Partial	11,2	51,4	1	225	22,6
	Full	12,3	50,9	6	12	1,8
50	Partial	11,5	51,8	9	278	28,7
	Full	11,3	50,9	24	3	2,7
75	Partial	10,7	51,7	23	322	34,5
	Full	10,9	50,4	34	18	5,2
100	Partial	10,5	50,9	26	331	35,7
	Full	11,4	50,8	60	8	6,8

Figure 14 Experimental delays and BP results for Partial and Full MF OTP Models

2.3 Control Plane in a Multi Domain Network

Even when under the control of a single administrative entity, transport networks may be segmented into domains for technical or scalability reasons (e.g., in the form of vendor islands). Such multi-domain networks are characterized by the fact that no single entity has full TE topology visibility, affecting optimality and efficient resource usage.

In IDEALIST, we rely on a Hierarchical PCE (H-PCE) [16] approach, scoped in the framework of what we refer to as interconnected traffic engineered networks. The fundamental components are the following, detailed later in Sections 2.3.1.1 to 2.3.1.5:

- Hierarchical PCE for multi-domain path computation for coordinated path computation, where the parent PCE (pPCE) performs domain selection and the children perform the actual path computation within their domain.
- The BGP-LS protocol [6] for Northbound topology distribution children to the parent, which allows the pPCE to construct an abstracted multi domain topology.

- Policy based network abstraction, with algorithms and approaches to represent a domain, which can be as simple as a virtual domain representing the domain internal connectivity, or open to advanced, policy controlled network abstraction methods
- Two considered modes for provisioning a end to end connection:
- Based on a NMS or Adaptive Network Manager (ANM) which triggers the provisioning using a dedicated provisioning interface. This implies, for the considered scenario, and end-to-end RSVP-TE session for signalling purposes (i.e., connection establishment).
- Based on extending the stateful H-PCE capabilities and instantiation, with coordination of segments and "stitching on the wire". This means that the H-PCE architecture is extended to support stateful capabilities and distributed instantiation.

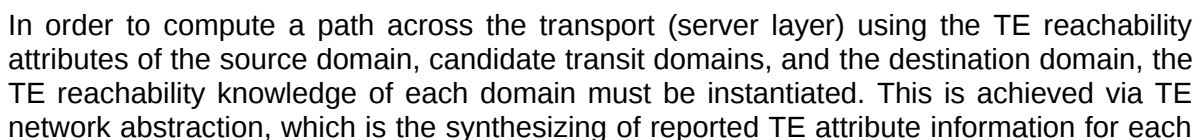
2.3.1 Architecture

In this section we detail the main components of the control plane architecture, which are based on the initial definitions and requirements as stated in [17]. It is worth noting that the architecture is being developed and evaluated in WP4, within the IDEALIST multi-partner testbed, which interconnects four European research institutions, located in Madrid (Telefónica I+D), Barcelona (CTTC), Torino (TI) and Pisa (CNIT). Partners' premises are connected (at the control plane level) by means of dedicated IPsec tunnels. The resulting low level connectivity layout is a hub, centered at CTTC. Static routing entries provide full connectivity between partners' private addresses, secured and isolated from the rest of Internet traffic. On top of this distributed control plane connectivity network, logical relationships between PCEs are established, in particular between Telefónica I+D PCE, acting as pPCE, and the other PCEs, acting as cPCE. The PCEs of the testbed have been independently developed by each partner. BGP-LS speakers are implanted by each partner.

As mentioned, IDEALIST defined architecture is built around the hierarchical PCE (HPCE) framework, in which a parent PCE (pPCE) coordinates several children PCEs (cPCE), one per network domain (Figure 14). The pPCE is in charge of domain selection and inter-domain path computation. Children PCEs are responsible for segment expansion, i.e. for path computation in their respective domains. Both stateless and stateful PCEs (with instantiation capabilities) are considered.

BGP-LS is deployed for topology abstraction and to export inter-domain TE information to the pPCE and LSP provisioning is triggered by an SDN controller that monitors network resources utilizations and is able to decide the optimal network configuration based on the status, bandwidth availability and user service. It leverages recently proposed extensions to the PCEP protocol for the so-called stateful and active PCE and includes instantiation capabilities.

A single end-to-end RSVP-TE signaling session is used for setting up the connection between the domains; this allows simplified setup and teardown procedures, especially in case of exceptions handling, and in the case of separated signaling sessions (one per domain). Alternatively, the pPCE may coordinate the establishment of segments in a vertical top-down manner, ensuring continuity of the assigned frequency slot as required.



domain and inter-domain link. This provides the aggregated TE reachability information and subsequent abstracted topology representation, known as virtual links and nodes (virtual topology). This transport network abstraction and creation of a virtual topology does not represent all possible connectivity options, but instead provides a connectivity matrix based on current TE attributes that are being reported from within each domain. While abstraction uses available TE information, it may also be subjected to network policy and management choices. Thus, not all potential connectivity would be advertised.

If the current transport connectivity does not meet required or expected client demands new peer connections can be established. These TE LSP tunnels will span the transport domains used to achieve the required or expected connectivity. These transit LSPs are the key building blocks of the end-to-end connectivity for the client. It is expected that planning will be required to ensure the required connectivity is available, but dynamic or on-demand requests could be supported, but should be subject to policy considerations.

Once a suitable topology exists it can be abstracted into a virtual topology provided as a TED to the client node, ANM or PCE for computing end-to-end TE-based services across the multi domain network without exposing the internal domain topologies and exhaustive TE information of the transport network.

2.3.1.2 Multi-domain Topology Management and Inter-domain Routing

In IDEALIST proposed hierarchical architecture, it is required to maintain a domain topology map at the pPCE, representing a view of the child domains and their interconnectivity, e.g. its abstract representation. The procedure and protocol mechanisms for disseminating and constructing of the pPCE TED may be provided using a number of mechanisms, subject to policy and currently being evaluated: the pPCE could join the IGP instance of each child PCE domain, while the attributes of the inter-domain links may be distributed within a domain by TE extensions to the IGP, as in [19]. However, it would break the domain confidentiality principles and it is subject to scalability issues. Alternatively, [18] points out that in ASON models it is possible to consider a separate instance of an IGP running within the parent domain with the participation of the child PCEs. Alternative options is the embedding in PCEP Notifications both intra domain and inter-domain LSAs. However, it is argued that the utilization of PCEP is beyond the scope of such protocol. Finally, at the retained option in this work, is the use north-bound distribution of TE information, by means of the BGP-LS protocol. With this approach, there is a BGP speaker in each domain that sends the necessary information to a BGP speaker in the parent domain. A separated policy can be configured to decide which information can be exported.

Note that the number of "border links" that create the inter-domain network is usually quite low; new links are seldom added (involving also commercial agreements between carriers) and connections crossing domain boundaries are less frequent than the intra-domain ones. Due to the implementation of different policies in the domains, routing information updates could be uncoordinated impacting on routing protocol convergence and leading to connection setup failures. Moreover, in the multi-carrier scenario, certain coordination about the kind of TE information to distribute may be required to avoid issues to the routing algorithms implemented at the inter-domain level. Consequently, another viable approach could be to configure the inter-domain links statically into the pPCE.

Additionally, in a multi-carrier scenario, it may also be desirable that the inter-domain routing is managed by a third party and this approach reduces the need for coordination or agreement among the actors.

The available resources updates can be achieved employing a stateful PCE. Depending on the routing model adopted (R\&SA, R+SA or R+DSA) or if some nodes, due to physical restrictions, must allocate a wider frequency slot than the one requested by the LSP, the PCE could not have all the information about the resources that the LSP utilizes; in this case, the PCEP PCRpt message must contain all the information required (i.e. the labels, contained, for example in the Record Route Object or RRO).

2.3.1.3 Hierarchical Stateful PCE

The H-PCE has been retained as the most suitable technology to compute optimum routes for LSPs crossing multiple domains: The pPCE is responsible for domain sequence computation. Then, in each identified domain, a child PCE (cPCE) performs segment expansion. The pPCE exploits an abstracted domain topology map that contains the child domains and their interconnections. In IDEALIST, several innovative enhancements to this approach are under investigation.

- First, besides reachability information, a mesh of abstracted links between border nodes is introduced in the parent TED to improve the effectiveness of domain sequence computation.
- Second, the north-bound distribution of Link-State and TE information using BGP (i.e., BGP-LS) is the protocol solution proposed to provide link information to the pPCE. Third, specific extensions to BGP-LS for elastic optical networks are also introduced.
- Finally, a stateful condition is introduced at the pPCE to enable advanced TE solutions, e.g. multi-domain reoptimization. The approach completes the hierarchical path computation composed of domain sequence selection and segment expansion with a subsequent route segmentation and segment provisioning, as explained next.

2.3.1.4 Signalling Aspects

According to [20], inter-domain TE LSPs can be supported by one of three options: contiguous LSPs, stitched LSPs and nested LSPs. In the flex-grid context, the latter solution is not applicable. Since these solutions require a high degree of control plane interoperability both for routing and for signalling, we are considering:

- A single RSVP-TE end-to-end session, This means that the H-PCE framework is used only for path computation (even if in a stateful mode, in which local end multi-domain LSPDBs are maintained), but not for the actual provisioning, which is requested using a provisioning interface. Note that this provisioning interface may also be based on the PCEP protocol, for convenience.
- Second, taking advantage of the H-PCE structure, where the pPCE can orchestrate the cPCEs, acting as the responsible within its own domain, for the establishment (and release) of connections to an underlying GMPLS control plane. By this approach, all PCEs are stateful and have instantiation capabilities. That is, every domain has its own "local" RSVP-TE session and the connectivity at the data plane level is insured by the concatenation of media channels at each domain, while the coordination among the domains (i.e. ingress/egress ports, labels, etc.) is the responsibility of the pPCE. In this case, interoperability requirements are scoped to PCEP extensions for stateful PCE with instantiation capabilities and no protocols are required at the inter-domain boundaries.

2.3.1.5 Control Plane Procedures

Let us detail the main procedure for the establishment of a LSP with the help of Figure 15, and as implemented in WP4. Upon request (1), the SDN controller triggers the provisioning. First, it requests a multi-domain path computation (2), which is a two-step process in which the pPCE obtains the domain sequence and then requests the children to expand the domain path within their respective domains. The pPCE composes and end-to-end ERO, which, by default, uses unnumbered interfaces represent outgoing TE links but, to convey information about the ingress port, it can be prepended with an additional ingress interface (facing the client) and may either end with a IPv4 prefix address or an unnumbered interface meaning that the LSP ends at the output interface with an additional cross-connect. Explicit label control (ELC) conveys information about the outgoing label (frequency slot) that will be used by the downstream node in switching. The actual LSP provisioning takes place after the end to end path has been computed (3), and the provisioning manager uses the PCEP interface with the ingress node to request a Path establishment (4). It is based on the use of PCInitiate and the PCRpt messages: the PCInitiate includes the SRP, LSP, ENDPOINTS, ERO objects, and instructs the ingress node to initiate the signaling procedure, based on the Path/Resv RSVP-TE message exchange with an end-to-end session (5). Upon completion of the signaling process, the PCRpt message is sent back to the provisioning manager, additionally including the route object RRO and the allocated frequency slot.

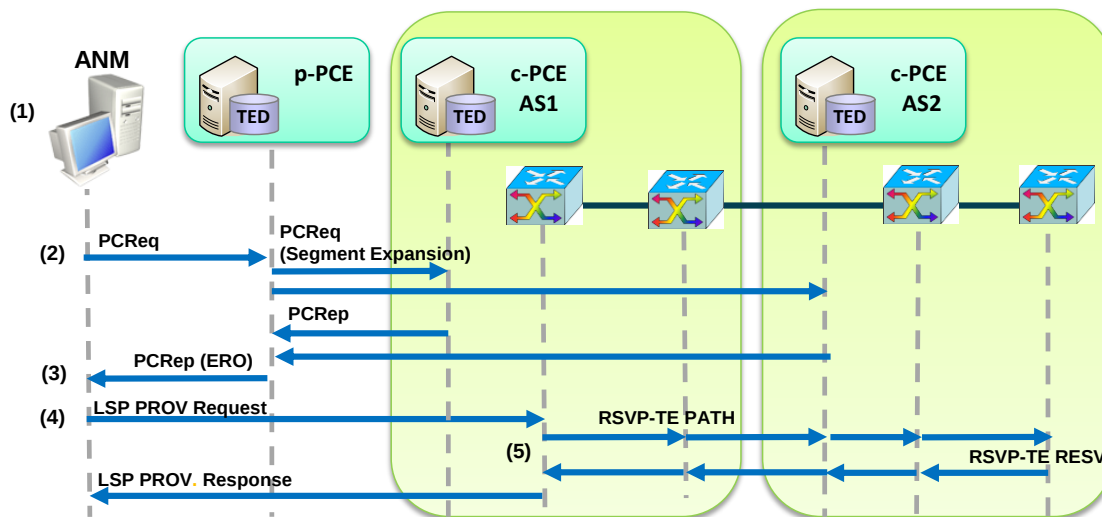


Figure 16 Message flow as defined in WP3 for the end to end RSVP-TE session and implemented in WP4, multi-partner control plane testbed

2.3.2 Selected Implementation Scenarios

2.3.2.1 Scenario 1: Architecture, Interfaces and Protocol Functions

The first scenario (Scenario 1) is characterized by the following:

- The path computation function is based on the hierarchical PCE (H-PCE)

- Topology abstraction is provided from the child to the parent using BGP-LS, this process is detailed later in this document.
- Provisioning is triggered by the Adaptive Network Manager (ANM) using a provisioning interface based, on a first step, on using a modified PCEP
- The existence of an end-to-end RSVP-TE session to signal the LSP

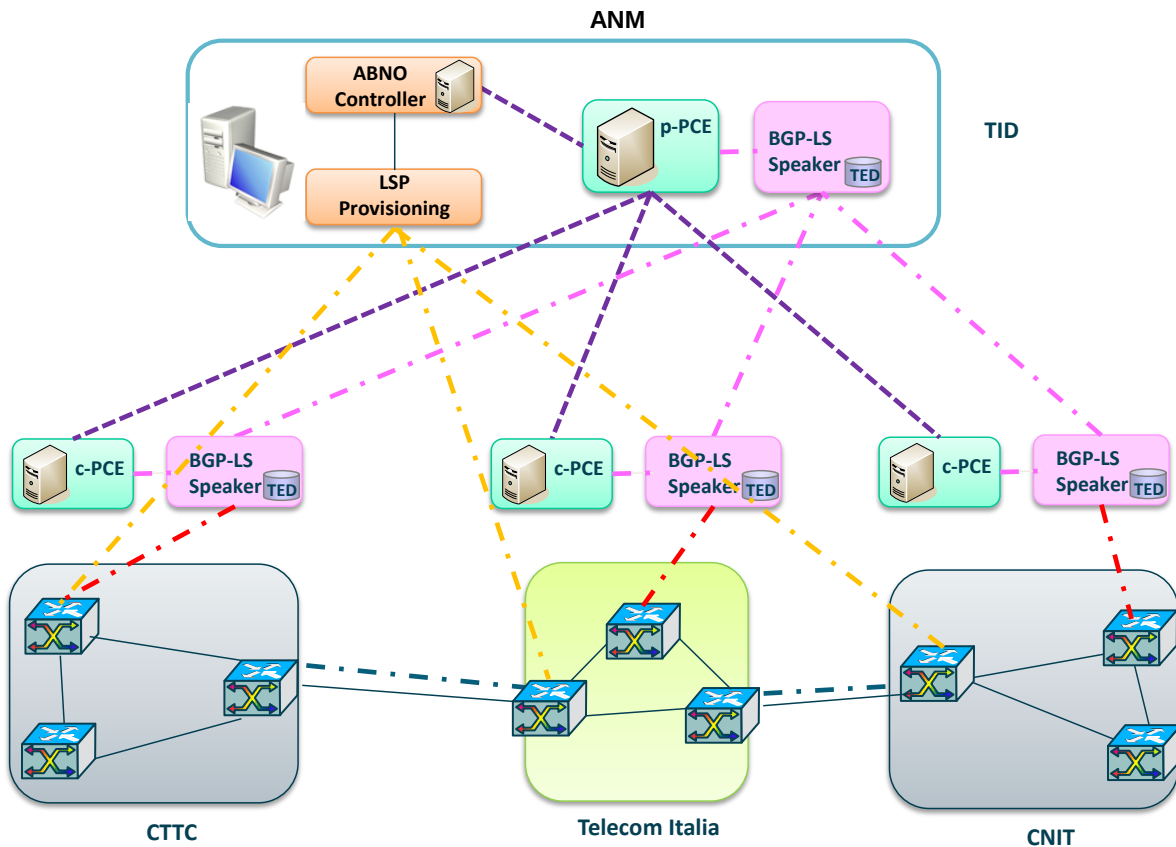


Figure 17 Architecture of the Scenario 1, based on the ANM-driven provisioning and a single RSVP-TE session

2.3.2.2 Scenario 2: Architecture, Interfaces and Protocol Functions

The second scenario (Scenario 2) is characterized by the following:

- The path computation function is based on the hierarchical PCE (H-PCE) as in the Scenario 1
- Topology abstraction goes in the child to parent direction using BGP-LS as in the Scenario 1
- Provisioning is based on instantiation extensions to PCEP. A single connection happens by stitching “on the wire” as many segments as required. The systems thus proceed in a two-step process, a full path computation (the strict ERO is retrieved by p-PCE) and the subsequent provisioning.

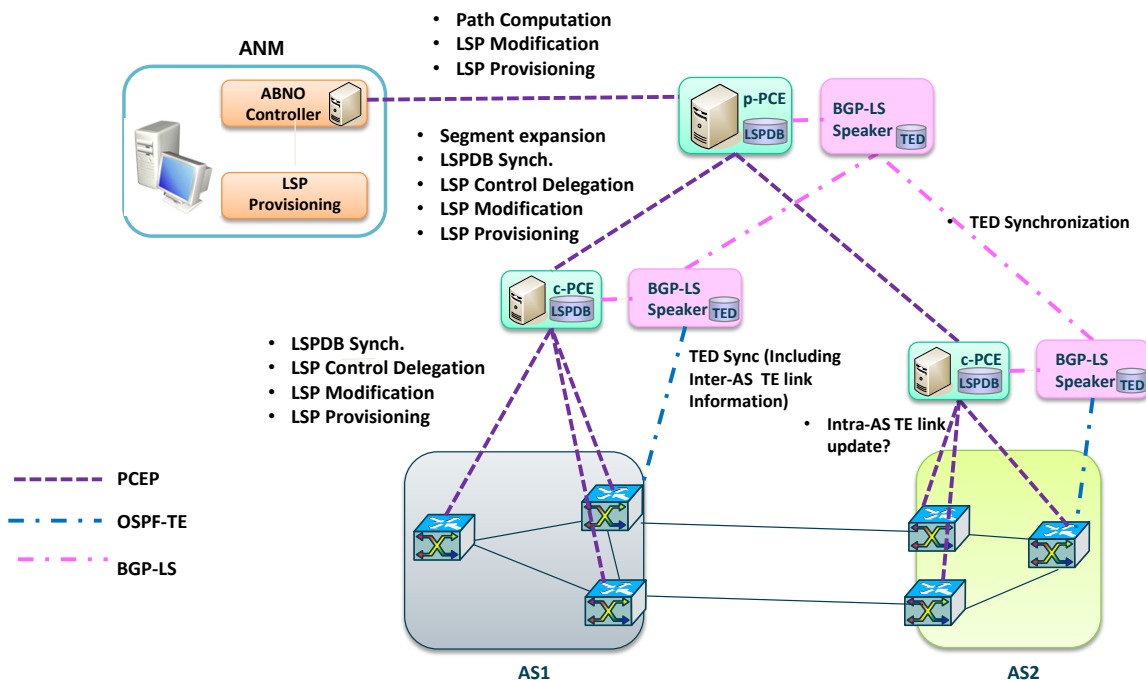


Figure 18 Architecture of the Scenario 2, based on the H-PCE path computation and provisioning as well as stitching different connection segments.

In short, the pPCE receives a PCInitiate and allocates an inter-domain LSP entry in the inter-domain LSPDB.

When the ERO has been computed, it segments the ERO into domain segments and sends the corresponding PCInitiate with symbolic names, maintaining a 1:N relationship between inter-domain LSP and intra-domain segments.

The parent PCE needs to operate e.g. with transactional semantics and ensure an "all or nothing" process is executed.

3 GMPLS/PCE Protocol Extensions

3.1 Label format

3.1.1 Single Label Component

This section defines the generalized label encoding for use in flexi-grid systems. As with the other GMPLS lambda label formats defined in [21], the use of this label format is known a priori, that is, since the interpretation of all lambda labels is determined hop-by-hop. The use of this label format requires that all nodes on the path expect to use this label format. The encodings are aligned with IDEALIST efforts in standardization at the IETF [11]. It is worth noting that previous versions of the label format used an m parameter encoded with 8 bits. Feedback obtained during the standardization process justified extending it to 16 bit, in order to be able to cover frequency slots with widths up with $m > 256$, which was considered to be limiting.

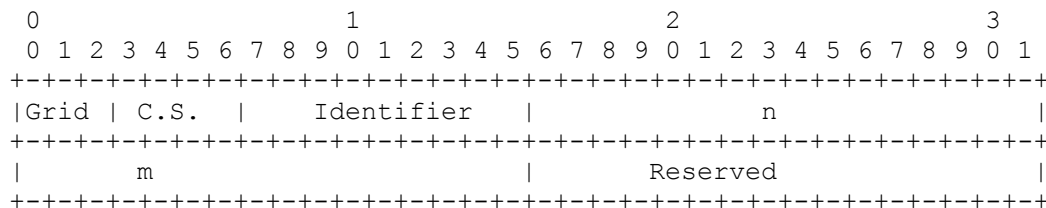


Figure 19 Selected label format with 64 bit encoding and m with 16 bitss.

As shown in Figure 18, the fields of the label encoding are the following

- The Grid value with a fixed value of 3, which is the selected encode to state that it is based on ITU-T flexi grid.
- Within the fixed grid network, the C.S. value is used to represent the channel spacing, as the spacing between adjacent channels is constant. For the flexible grid situation, this field is used to represent the nominal central frequency granularity. A new C.S. value to supplement those in [21] is 6.25 GHz encoded as 5.
- The meaning of the Identifier field is maintained from [21].
- The meaning of n is maintained from [21].
- The m field is used to identify the slot width according to the formula given in [G.694.1] as in Slot Width (GHz) = 12.5 GHz * m
- The Reserved field MUST be set to zero on transmission and SHOULD be ignored on receipt.

```

> Internet Protocol Version 4, Src: 10.0.50.3 (10.0.50.3), Dst: 10.0.50.2 (10.0.50.2)
> Resource ReserVation Protocol (RSVP): RESV Message. SESSION: IPv4-LSP, Destination: 10.0.50.2
  ▸ RSVP Header. RESV Message.
  ▸ MESSAGE-ID: 10 (Ack Desired)
  ▸ SESSION: IPv4-LSP, Destination 10.0.50.3, Tunnel ID 65535, Ext ID a003201.
  ▸ HOP: IPv4 IF-ID. Control IPv4: 10.0.50.3. Data If-Index: 10.0.50.2, 3.
  ▸ TIME VALUES: 120000 ms
  ▸ STYLE: Fixed Filter (10)
  ▸ FLOWSPEC
  ▸ FILTERSPEC: IPv4-LSP, Tunnel Source: 10.0.50.1, LSP ID: 1.
  ▸ LABEL: Generalized: 0x6a000001, 0x10000
  ▸ RECORD ROUTE: IPv4 10.0.50.3

```

Figure 20 Wireshark capture of a GMPLS Resv message in which the Generalized Label is using the 64 bit format, with $n = 1$ and $m = 1$

This label format is used uniformly, notably in the PCEP and RSVP-TE protocols, as shown in Figure 19, which shows a Wireshark capture of the implemented control plane, showing the RSVP-TE Resv message and detailing the use of the label format in the GENERALIZED_LABEL of the Flow descriptor.

3.1.2 Multiple Label Components

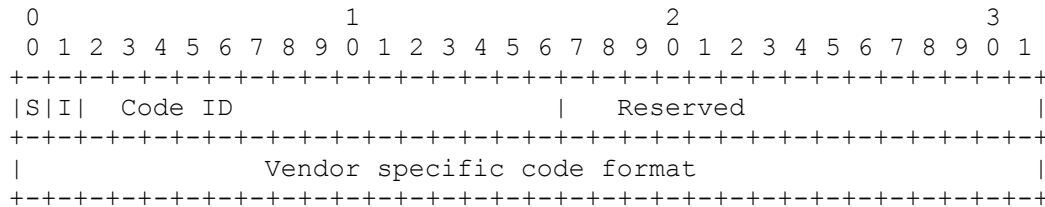
The creation of a composite of multiple channels to support inverse multiplexing is already supported in GMPLS for TDM and OTN [22][23][24]. The mechanism used for flexi-grid is similar. To signal an LSP that uses multiple flexi-grid slots a "compound label" is constructed. That is, the LABEL object is constructed from a concatenation of the 64-bit Flexi-Grid Labels. The number of elements in the label can be determined from the length of the LABEL object. The resulting LABEL object is shown below. Note that r is the count of component labels, and this is backward compatible with the label shown above where the value of r is 1. The order of component labels MUST be presented in increasing order of the value n . Implementations MUST NOT infer anything about the encoding of a signal into the set of slots represented by a compound label from the label itself. Information about the encoding MAY be handled in other fields in signaling messages or through an out of band system.

0																1																2																3																															
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																																																																															
Object Length (4 + 8r)																Class-Num (16)																C-Type (2)																																															
Grid C.S.																Identifier																n																																															
																m																																Reserved																															
~																																																~																															
Grid C.S.																Identifier																n																																															
																m																																Reserved																															

- Grid MUST show "ITU-T Flex" value 3 in each component label.

the estimated optical impairments in order to convey the signal in the m slots indicated in the label:

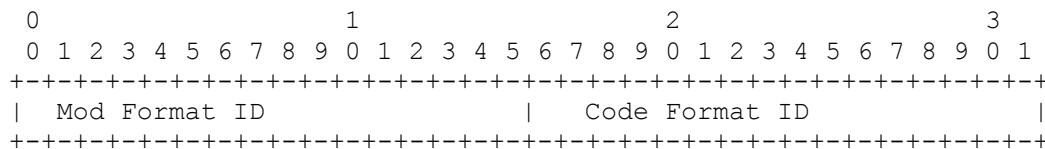
- type (2 octets): 51
- length (2 octets): 8
- value: see figure below with S bit: 1(standard), 0(vendor-specific), with I bit : 1 (input), 0 (output) typically set to 0.



- The encoding is based on draft-ietf-ccamp-rwa-wson-encode.

ALTERNATIVE PROPOSAL:

- Novel ERO subobject called TRANSPONDER_SPEC (type 100, experimental) indicating the modulation format and the code format suggested by the pPCE in order to convey the signal in the m slots indicated in the label:



The pPCE is supposed to be aware of the available bitrate $\rightarrow$ (m , modulation, code) match.

The BANDWIDTH object of the ANM $\rightarrow$ pPCE PCReq message (scenario 1) and of the ANM $\rightarrow$ pPCE PCInitiate (scenario 2) should include the required information bitrate of the LSP (option A) or the requested amount of spectrum m (option B).

In the case of option A, the BANDWIDTH object specifies the information rate (in Gb/s) of the client signal. In this case, m is an output parameter of the path computation. As a result, the effective line rate of the transponder will be computed by considering the requested information bitrate, the identified modulation format and code rate.

In the case of option B, m is the input path computation parameter enclosed in the BANDWIDTH object. In this case, the PCRep (scenario 1) or the PCInitiate towards the cPCEs (scenario 2) should include the information bitrate in the BANDWIDTH object in order to properly configure the transponders.

SCENARIO 1

If a cPCE is queried by a pPCE with a PCReq including a BANDWIDTH object enclosing the required m value to request a segment expansion, the cPCE should not provide a computed segment with a suggested label. First, if continuity constraint among domains is considered, the pPCE is in charge of suggesting the e2e label. Second, the match between bitrate and m may not be known at cPCE level.

The PCReq of a segment expansion can include the GENERALIZED_ENDPOINTS (unnumbered interfaces) if needed. For example, in the case of src/dst nodes and in the case the outgoing inter-AS link is required to be crossed. Otherwise, standard

END_POINTS (e.g., IPV4) may be used. The PCReq should specify the requirements in terms of spectrum by including SUGGESTED_LABEL in the case m or the (n, m) pair is suggested for same reasons by the pPCE. Otherwise, the cPCE should provide end-to-end (i.e., border-to-border) available LABEL_SET of the computed path, in order to enable the e2e spectrum selection by the pPCE.

For these reasons, the cPCE should provide a ERO without label objects, with the indication of the available spectrum of the computed end-to-end segment, if required. Thus, it should provide the LABEL_SET or the SUGGESTED_LABEL object proposed in MS6.

SCENARIO 2

Segment expansion is realized through a PCInit message sent to cPCE in order to setup the domain segment. The PCInit should include SRP, LSP, ENDPOINTS (standard or generalized, if needed) and ERO. In this case the ERO should enclose the label objects since path computation and spectrum suggestion has been already performed by pPCE.

3.2.2 PCEP-based Inter-AS TE Link Update

The inter-AS TE link update is performed by means of the BGP-LS update message generated by the BGP-LS speaker co-located at the involved cPCE. The update mechanism to cPCE is realized through multiple options, depending on the considered domain (e.g., OSPF-TE, PCEP, SNMP, dedicated protocol).

In the case PCEP is utilized within the considered domain, we may consider a NOTIFICATION message upon the LSP segment is established.

The notification includes novel notification type (i.e., reserved spectrum) and the RRO object of the provisioned LSP including the outgoing inter-AS link. RRO labels identify the reserved resources.

Similarly, upon LSP tear down or modification, the same message is sent to cPCE with the "released spectrum" notification type. This way, the PCE TED can be updated accordingly.

3.2.3 Label and Label Set Extensions

For distributed RSA models, and to address spectrum continuity constraints, within IDEALIST we consider attaching a new LABEL_SET object (class 130, type 1) as an attribute of the computed path, which encodes the free NCF as a bitmap

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Object-Class |   OT   |Res|P|I|   Object Length (bytes)   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   4   |   Num Labels   |   Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Grid   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Bit Map Word #1 (Lowest numerical labels)   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
:
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   Bit Map Word #N (Highest numerical labels)   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

For example, the following hex byte stream encodes a LABEL_SET object (0x82) type 1, with 28 (0x1c) total bytes length, encoding a bitmap of 128 labels starting with flexi-grid $n = 0$ and all being available (bit set to 1, 0xff...ff)

```
82 10 00 1c
40 80 00 18
6a 00 00 00
ff ff ff ff
ff ff ff ff
ff ff ff ff
ff ff ff ff
```

Likewise, a PCC or PCE may attaching a LABEL object (class 129, type 1) as an attribute of the computed path or as a path computation request constraint.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Object-Class										OT										Res P I										Object Length (bytes)									
label																																							

If there is only one label object it is the SUGGESTED_LABEL. For bidirectional connections (to be evaluated) the PCE MAY attach a second LABEL object for the UPSTREAM LABEL. The following byte stream includes a label with $n = 1$ and $m = 1$

```
81 10 00 0c
6a 00 00 01
00 01 00 00
```

3.2.4 LSPDB Synchronization

LSP synchronization between two peers relies on the following:

- The use of PCRpt messages sent by the PCC or cPCE to the PCEP peer, using SRP object with srpid 0.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	backend	frontend	PCEP	OPEN MESSAGE
2	0.009641	frontend	backend	PCEP	OPEN MESSAGE
3	0.009972	frontend	backend	PCEP	KEEPALIVE MESSAGE
4	0.010783	backend	frontend	PCEP	KEEPALIVE MESSAGE
5	0.011045	frontend	backend	PCEP	PATH COMPUTATION REPORT
6	0.011111	frontend	backend	PCEP	PATH COMPUTATION REPORT
7	0.011120	frontend	backend	PCEP	PATH COMPUTATION REPORT
8	0.011123	frontend	backend	PCEP	PATH COMPUTATION REPORT

Path Computation Element communication Protocol

- ▷ PATH COMPUTATION REPORT MESSAGE Header
- ▷ SRP object
- ▷ LSP object
- ▷ EXPLICIT ROUTE object (ERO)
- ▷ BANDWIDTH object

Figure 21: Example of LSPDB synchronization process

- Ending the synchronization with an empty report, that is the LSP object with the end-of-sync flag and no ERO or BANDWIDTH objects

3.2.5 LSP Provisioning

LSP (segment) provisioning between a PCE and an attached PCC/LSR node is based on the use of PCInitiate and the PCRpt messages, as shown below:

- The PCInitiate includes the SRP, LSP, ENDPOINTS, ERO objects (ENDPOINT objects can be either RFC5440 if just the nodes are specified, assuming an LSP that starts at an output interface and ends at a input interface or using the generalized form that allows convening an input interface and output interface for the ingress and egress nodes, respectively).

```
Internet Protocol Version 4, Src: pPCE (84.88.62.83), Dst: cPCE-1
Transmission Control Protocol, Src Port: pcep (4189), Dst Port: 54
Path Computation Element communication Protocol
  ▸ PATH COMPUTATION INITIATE MESSAGE Header
  ▸ SRP object
  ▸ LSP object
  ▸ END-POINT object
    ▾ EXPLICIT ROUTE object (ERO)
      Object Class: EXPLICIT ROUTE OBJECT (ERO) (7)
      0001 .... = Object Type: 1
      ▸ Flags
      Object Length: 88
      ▸ SUBOBJECT: Unnumbered Interface ID: 10.0.50.1:1
      ▸ SUBOBJECT: Unnumbered Interface ID: 10.0.50.1:2
      ▸ SUBOBJECT: Unnumbered Interface ID: 10.0.50.3:2
      ▸ SUBOBJECT: Unnumbered Interface ID: 10.0.50.5:5
      ▸ SUBOBJECT: Unnumbered Interface ID: 10.0.50.7:3
      ▸ SUBOBJECT: Unnumbered Interface ID: 10.0.50.12:4
      ▸ SUBOBJECT: Unnumbered Interface ID: 10.0.50.13:10
      ▸ LABELSET object
    ▾ SL object
      Object Class: SL OBJECT (129)
      0001 .... = Object Type: 1
      ▸ Flags
      Object Length: 12
```

Figure 22: Example of LSP provisioning procedure

- The cPCE PCRpt messages includes the SRP object, the LSP object and the ERO of the LSP. Attributes such as the LABELSET and LABEL (SL) objects are also provided, including the allocated frequency slot,

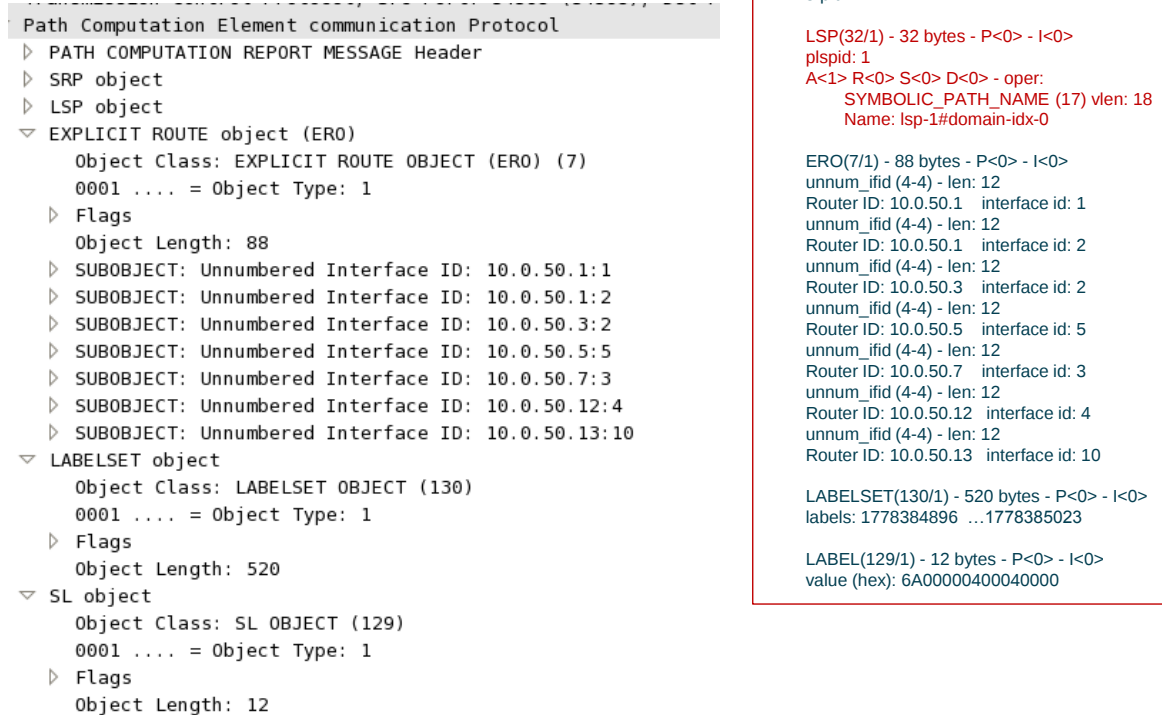


Figure 23: PCRpt message sent by an LSR/PCC after successful setup

3.2.6 LSP Concatenation (ERO format)

The format of the ERO is as follows:

- By default, ERO unnumbered interfaces represent outgoing TE links
- To convey information about the ingress port, if the first node router ID (router IDs and node IDs are assumed to be equal) appears twice consecutively in the ERO, it is assumed to be an ingress interface (facing the client)
- The ERO may end with a IPv4 prefix address (/32) meaning that the LSP ends at the egress node incoming interface or an unnumbered interface, meaning that the LSP ends at the output interface with an additional cross-connect.
- In all the cases, explicit label control (ELC) conveys information about the outgoing label (frequency slot) that will be used by the downstream node in switching.

In the case the endpoint transponders are not indicated in the ANM->pPCE PCReq, the pPCE has to select the transponders and enclose their indication within the ERO of the final PCRep (scenario 1) or of the PCInit sent to the cPCEs of the ingress and egress domains, as well as in the PCRpt sent to the ANM (scenario 2). In order to enable the selection of the end-points transponders, the ERO provided by the pPCE to the source

cPCE has to be expanded accordingly. To identify the selected transponders within the ERO without introducing a new ERO subobject, the following agreement is proposed:

- The first and the second ERO subobjects of the e2e path are referred to the selected transponder located at the source node. The penultimate and the ultimate ERO subobjects are referred to the selected transponder at the destination.
- The first (and the penultimate) ERO subobject is of type "Unnumbered" and identifies the transponder.

The second (and the last) ERO subobject is of type "Label" and identifies the m and n values "generated" by the transponder (source node case) or used to switch the traffic to the receiver and to tune its central frequency (destination node case). The value of m and n may in general be different from those associated to the outgoing optical link. In particular m can be different based on the specific architecture of the optical node.

Referring to Figure 23, the ERO is: A[a1],label(tx),A[a2],label,B[b2],label,C[c1], label (rx).

In particular label(tx) is the information needed by the transponder to setup the tunable laser and the electrical (or optical) filter. In general, such values could be different from the label values of the outgoing link interface a2. This because the granularity required from such devices could be different from the standard flexi-grid (6.25GHz cs).

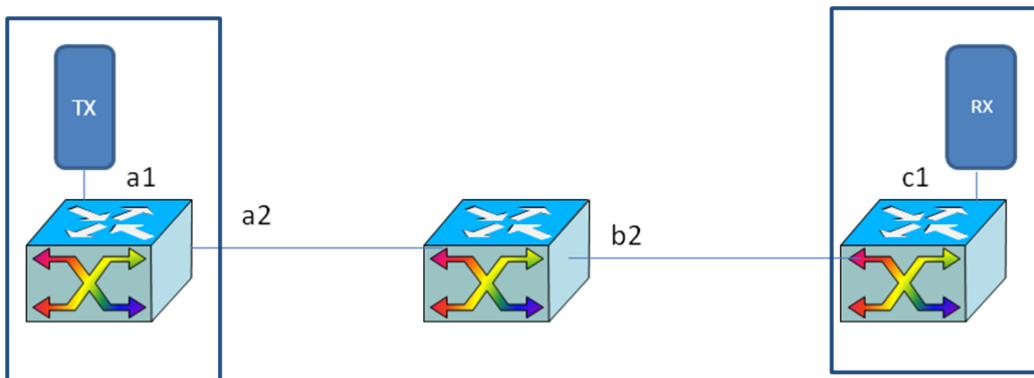


Figure 24: reference scenario

3.2.7 LSP Modification

A PCE may initiate a modification of a LSP or LSP segment by sending a PCUpdate message with the new LSP attributes. In this milestone and joint WP3/WP4 activity only modification of LSPs using a Break before Make (BBM) is targeted / supported.

3.3 RSVP-TE Signalling Protocol

3.3.1 Signalling Scenarios

Two options are examined for multi-domain connection instantiation:

1. Single end-to-end signalling session.
2. Segment concatenation.

In the first case, a single signalling session is created by the head-end node of the ingress domain with destination the tail-end node of the egress domain (Figure 24). Upon receiving a request from a Provisioning Element, the head-end node instantiates a signalling session by creating a PATH message in which the LSP_IPv4_SESSION object contains the tail-end node-ID. The ERO includes all the hops (in the form of UNNUMBERED sub-objects) and the labels to be used at every hop.

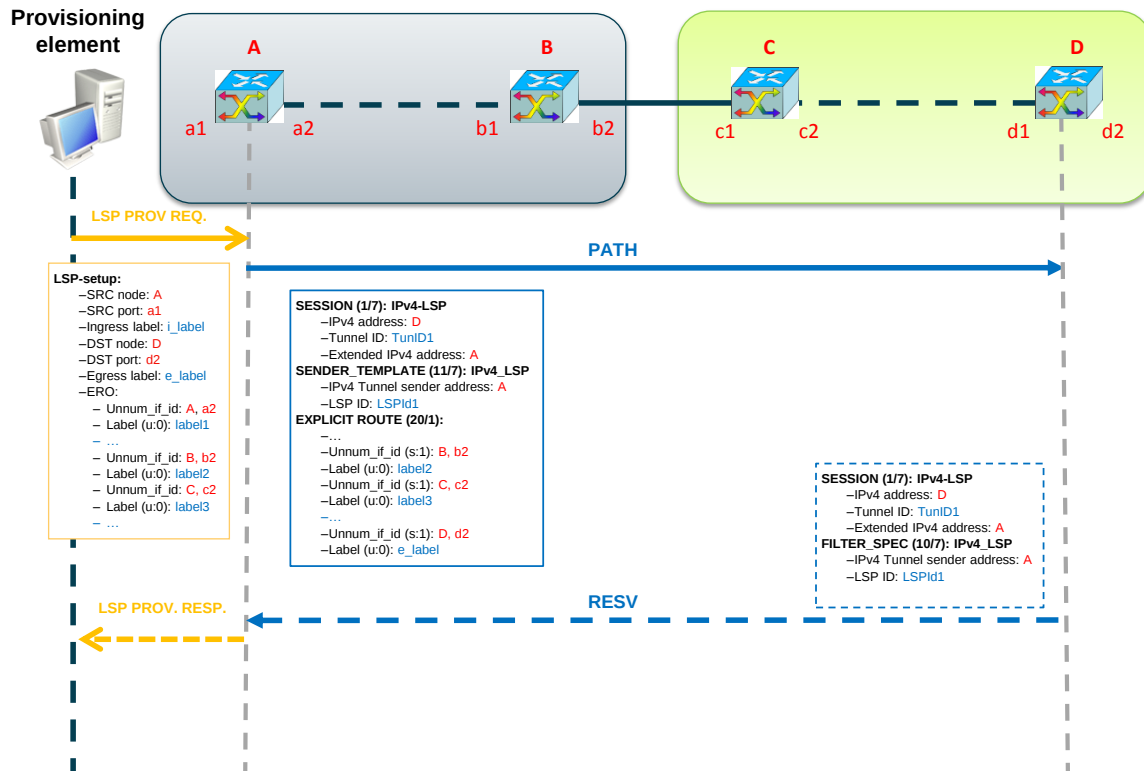


Figure 25 - Single end-to-end signalling session

A variant to this approach is the use of path-keys inside the ERO to maintain confidentiality within the domains (Figure 25). In this case, during the segment expansion phase of the path computation procedures, (some of) the children PCEs returned the ERO in the form of path-key. The PATH_KEY sub-object contained in the ERO should be preceded by an IPv4_PREFIX (or IPv6_PREFIX) sub-object pointing to the domain ingress node that will be in charge of the Path Key expansion procedure. In this procedure, that is executed during the signalling phase, the domain ingress node creates a PCEP session with the PCE whose identifier is contained in the PATH_KEY sub-object, typically its domain PCE (or one PCE in its domain).

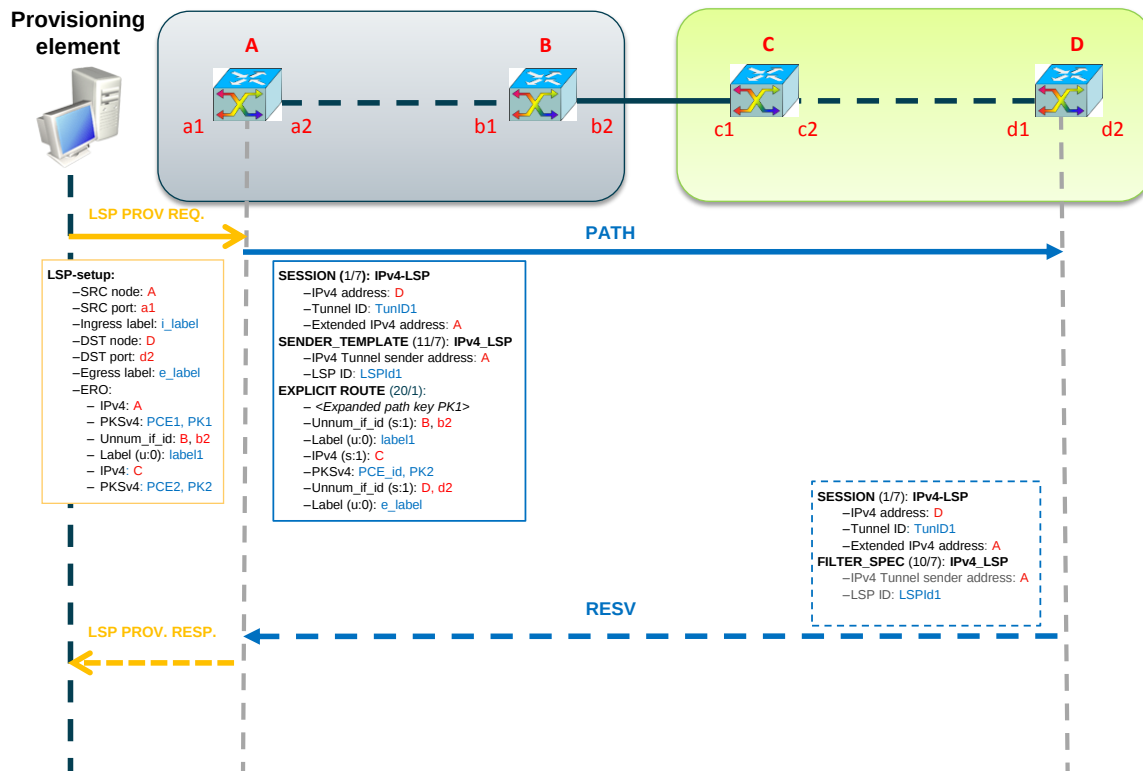


Figure 26 - Single end-to-end signalling session with the use of Path-Keys

In the second option, there are separated signalling session in each domain (Figure 26). The provisioning element (or one provisioning element per domain) requires the creation of a connection between the domain ingress and egress nodes. Every ingress node instantiates an LSP by creating a PATH message in which the LSP_IPv4_SESSION object contains the node-ID of the corresponding domain egress node. This approach has the advantage that no interworking is required between the domains (even the signalling protocol could be different in the domains). However, in this case, some kind of coordination is required to align the egress port and label of a domain to the ingress port and label of the next one. In other words, the provisioning element(s) must convey, in the setup request, the same information that are carried in the RSVP_HOP and LABEL (and UPSTREAM_LABEL for bidirectional connections) objects. The EXPLICIT_ROUTE object can be used for this purpose: while extensions for label disseminations have already been proposed for (flexi-grid) DWDM networks, some limitations exist for the routing protocol that should disseminate also the data-link identifiers, besides the TE-link ones. This limitation leads to the impossibility of supporting link-bundling in such a scenario.

Moreover, the “segment concatenation” approach requires strict coordination between the provisioning elements to handle exceptions correctly (i.e. in case of LSP setup errors and/or fault management) because the error condition must be propagated backward in order to take all the necessary actions (e.g. teardown of already established segments). On the other hand, this approach allows for local fault handling, i.e. a fault could be locally repaired adopting routing (new route calculation) and signalling (new route provisioning) procedures involving only nodes within the domain. Some interworking, however, is still necessary to avoid incorrect handling.

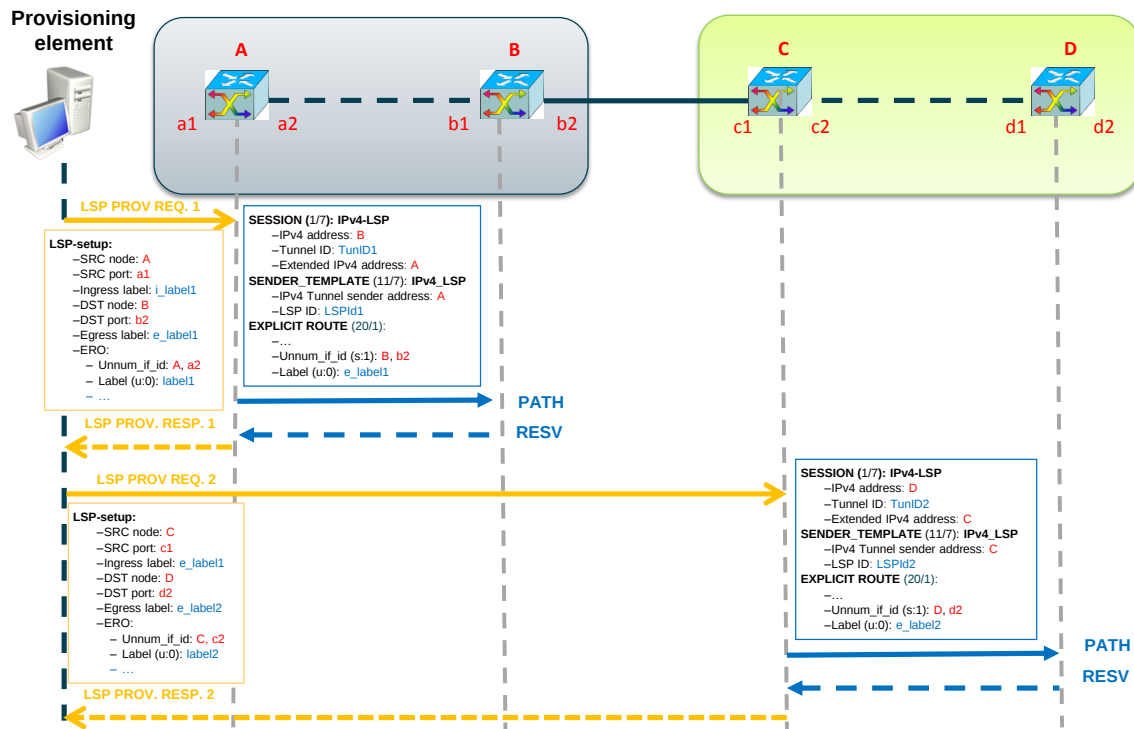


Figure 27 - Segment concatenation

3.3.2 Signalling Procedures

3.3.2.1 Connection Establishment

Connection established MUST be supported by the exchange of PATH and RESV messages. A PATH message MUST contain, at least, the following objects:

MESSAGE_ID
LSP_IPv4_SESSION
IPv4_IF_ID_RSVP_HOP
TIME_VALUES
EXPLICIT_ROUTE
GENERALIZED_LABEL_REQUEST
SENDER_TEMPLATE
SENDER_TSPEC

At least one between the LABEL_SET or the SUGGESTED_LABEL objects MUST also be present for Explicit Label Control (sect. 3.3.5).

A RESV message MUST, at least, contain the following objects:

MESSAGE_ID
LSP_IPv4_SESSION

IPv4_IF_ID_RSVP_HOP
TIME_VALUES
STYLE
FLOWSPEC
FILTER_SPEC
GENERALIZED_LABEL

All other objects **MUST** be processed according to RFC2205-sect. 3.10.

The usage of the CONFIRM message is optional.

3.3.2.2 Connection release

Connection release **MUST** be implemented by using the PATH_TEAR message ("Forced deletion"). Only forward connection release must be supported, i.e. the connection release **MUST** be initiated by the node that created the signalling session.

A PATH_TEAR message **MUST** contain the following objects:

MESSAGE_ID
LSP_IPv4_SESSION
IPv4_IF_ID_RSVP_HOP
SENDER_TEMPLATE
SENDER_TSPEC

The "graceful deletion" procedure (RFC3473 - sect.7.2) is optional.

3.3.2.3 Refresh Overhead Reduction

The BUNDLE message **MUST** not be used.

Reliable message delivery, by means of the MESSAGE_ID and MESSAGE_ID_ACK objects and the ACK message **MUST** be supported for trigger messages and **MAY** be used also for refresh messages. The usage of the SREFRESH is optional.

3.3.3 RSVP Objects Structure

RSVP Object	C-Num C-Type	Reference
LSP_TUNNEL_IPv4_SESSION	1/7	RFC3209
IPv4_IF_ID_RSVP_HOP	3/3	RFC3473, RFC3477
TIME_VALUES	5/1	RFC2205
IPv4_ERROR_SPEC	6/1	RFC3473
STYLE	8/1	RFC2205, RFC3209
FLOWSPEC	9/10	Sect.3.3.6
LSP_TUNNEL_IPv4_FILTER_SPEC	10/7	RFC3209
LSP_TUNNEL_IPv4_SENDER_TEMPLATE	11/7	RFC3209

RSVP Object	C-Num C-Type	Reference
SENDER_TSPEC	12/10	Sect.3.3.6
IPv4_RESV_CONFIRM	15/1	RFC2205
GENERALIZED_LABEL	16/2	RFC3473
GENERALIZED_LABEL_REQUEST	19/4	RFC3473, Sect. 3.3.4
EXPLICIT_ROUTE	20/1	RFC3209, RFC3473, RFC3477 sect. 3.3.5
MESSAGE ID	23/1	RFC2961
MESSAGE ID_ACK	24/1	RFC2961
MESSAGE ID_NACK	24/2	RFC2961
MESSAGE ID_LIST	25/1	RFC2961
LABEL_SET	36/1	RFC3473, Sect. 3.3.5
SUGGESTED_LABEL	129/1	RFC3473, Sect. 3.3.5
ADMIN_STATUS	196/1	RFC3473
SESSION_ATTRIBUTE	207/7	RFC3209

3.3.4 Switching Capability

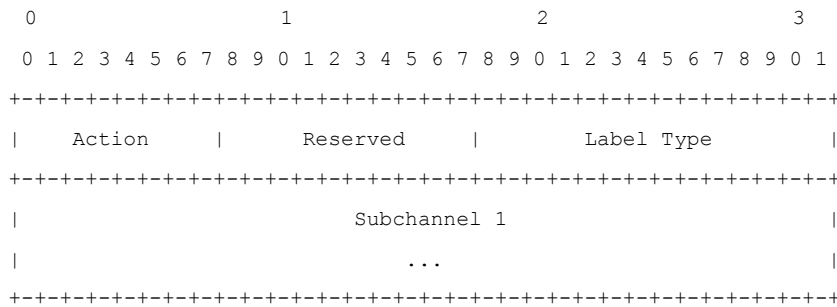
The GENERALIZED_LABEL_REQUEST object MUST contain the following values:

Switching Capability	190 (SSON media layer)
Encoding	8 (Lambda)
GPID	<don't care>

3.3.5 Explicit Label Control

The EXPLICIT_ROUTE object contains a list of sub-objects as computed by the parent and the children PCEs combination. The RSA model that MUST be implemented is the R&SA (Combined Routing and Spectrum Assignment) where the (Hierarchical-)PCE is in charge of performing both spatial (node) and spectrum assignment. Explicit Label Control (ELC) MUST be performed according to the procedures described in RFC3473-sect. 5.1.1.

The LABEL_SET object included on the outgoing PATH message MUST have the "Inclusive list" format and be composed of a single sub-channel element.



Action	0 (Inclusive List)
Label Type	2 (generalized label)
Subchannel 1	Label (sect. ¡Error! No se encuentra el origen de la referencia.)

The Reserved bits MUST be set to zero and ignored upon receipt.

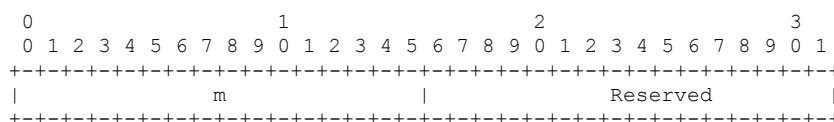
Alternatively, the upstream node can insert the label into the SUGGESTED_LABEL object.

3.3.6 Traffic Parameters

The SENDER_TSPEC object in the PATH message indicates the requested resource reservation. The FLOWSPEC object in the RESV message indicates the actual resource reservation. The traffic parameters for a flexi-grid LSP encode the slot width.

A new C-Types for the SENDER_TSPEC and FLOWSPEC objects to carry Spectrum Switched Optical Network (SSON) traffic parameters has been defined (not yet assigned by IANA):

1. SSON SENDER_TSPEC: Class = 12, C-Type = 10.
2. SSON FLOWSPEC: Class = 9, C-Type = 10.



m	slot width according to the formula: <i>Slot Width (GHz) = 12.5 GHz * m</i>
---	--

The Reserved bits MUST be set to zero and ignored upon receipt.

3.3.7 OSPF-TE

Within IDEALIST WP3, the OSPF-TE protocol has been extended to support flexi-grid networks. The extensions inherit the previous work done in the scope of Wavelength Switched Optical Networks (WSO), for which the framework was defined in [25]. With the

exception of except wavelength-specific availability information, the connectivity topology and node capabilities are the same, which can be advertised by the GMPLS routing protocol (refer to section 6.2 of [25]).

For Elastic optical networks based on flexi-grid, a set of non-overlapping available frequency ranges should be disseminated in order to allow efficient resource management of flexi-grid DWDM links and RSA procedures, i.e., in the flexi-grid case, the available frequency ranges are advertised for the link instead of the specific “wavelengths”.

The proposed extensions, being pushed for standardization in [14], mainly disseminate the status of the Nominal Central Frequencies. Such extensions are carried into the Interface Switching Capability Descriptor (ISCD), and more specifically in the Switching Capability Specific Information (SCSI), as shown in Figure 27. In the next sections, we detail the extensions and their use in the testbeds within WP4.

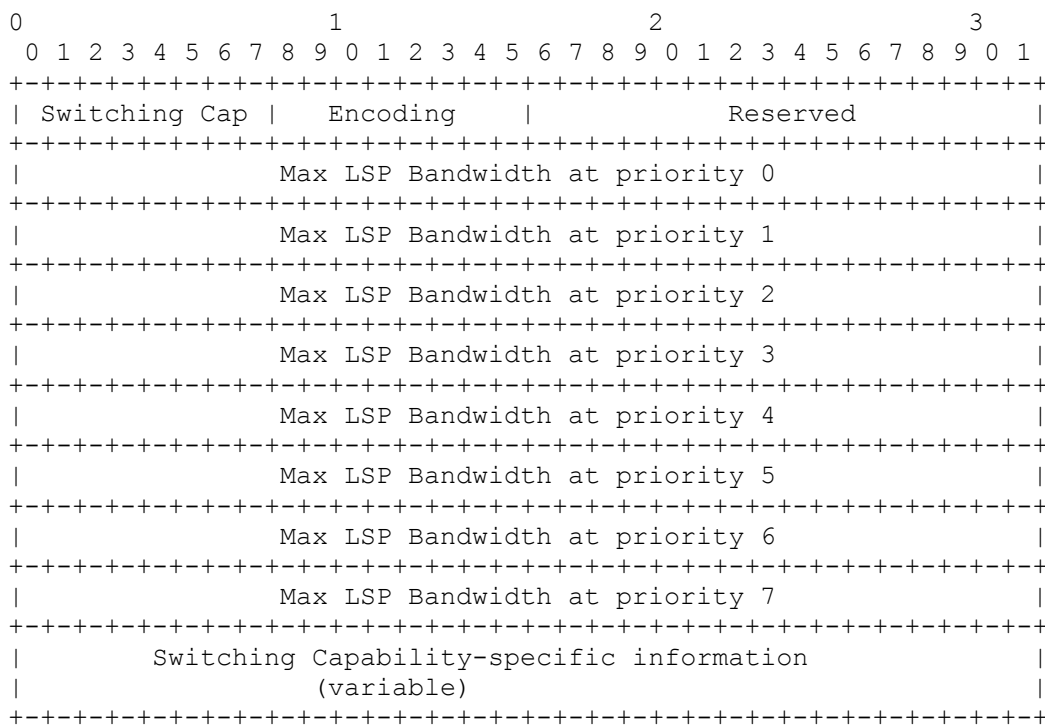


Figure 28: OSPF-TE interface switching capability descriptor (ISCD)

3.3.8 Available Label Set Sub-TLV

3.3.8.1 Inclusive Label Lists

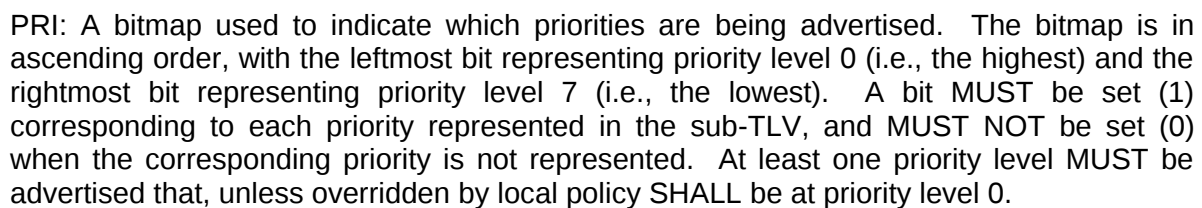
The inclusive/exclusive label lists format of Available Labels Set sub-TLV defined in [10] can be used for specifying the available central frequencies of flexi-grid DWDM links.

3.3.8.2 Bitmap Label Set

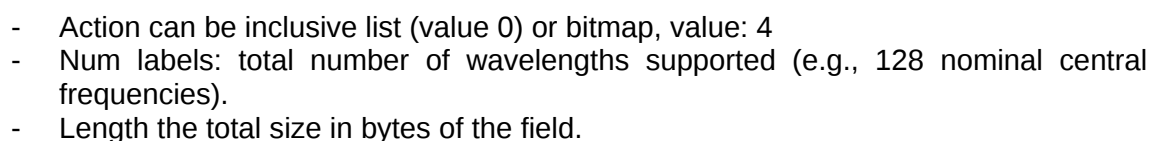
The bitmap format of Available Labels Set sub-TLV defined in [10] can also be used. In this case, the Base Label field specifies the lowest supported central frequency. Each bit in the bitmap represents a particular central frequency with a value of 1/0 indicating whether the central frequency is in the set or not. Bit position zero represents the lowest central



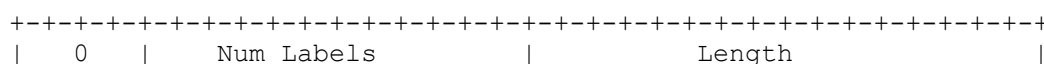
The SCSI is used to carry the technology specific part of the flexi-grid DWDM. This may include a variable number of sub-TLVs called Bandwidth sub-TLVs: bandwidth sub-TLVs is available labels; Type: 1



The Label Set Field is as follows:



For the specific Action case set to inclusive list, each “entry” is a NCF with format: grid-cs-identifier-n:



```

+-----+
|                                     |
|               ncf-number 1         |
+-----+
|               ncf-number 2         |
+-----+
|               ncf-number ..        |
+-----+
|               ncf-number M         |
+-----+

```

For the specific Action case set to the Bitmap:

```

      0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+
|      4      | Num Labels |      Length      |
+-----+
|               start NCF               |
+-----+
| Bit Map Word #1 (Lowest numerical labels) |
+-----+
:                                                         :
+-----+
| Bit Map Word #N (Highest numerical labels) |
+-----+

```

With start label / NCF:

```

      0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+
|Grid | C.S. | Identifier |           n           |
+-----+

```

- n is set to the lowest supported central frequency.

The following Figure shows the implemented OSPF-TE extensions, focusing on the selected bitmap encoding, as used in WP4.

```

Area-Local Opaque-LSA (Area 0.0.0.0)

LSA Age: 73
Link State ID: 1.0.0.2
Advertising Router: 10.0.50.1
LS Sequence Number: 0x80000001
Checksum: 0x39cd
LSA Len: 196
Opaque-Type 1 (Traffic Engineering LSA)
Opaque-ID 0x2
Opaque-Info: 176 octets of data
Link: 172 octets of data
Link-Type: Point-to-point (1)
Link-ID: 10.0.50.2
TE Metric: 300
Maximum Reservable Bandwidth: 1e+08 (Bytes/sec)
Unreserved Bandwidth (pri 0): 1e+08 (Bytes/sec)
Unreserved Bandwidth (pri 1): 1e+08 (Bytes/sec)
Unreserved Bandwidth (pri 2): 1e+08 (Bytes/sec)
Unreserved Bandwidth (pri 3): 1e+08 (Bytes/sec)
Unreserved Bandwidth (pri 4): 1e+08 (Bytes/sec)
Unreserved Bandwidth (pri 5): 1e+08 (Bytes/sec)
Unreserved Bandwidth (pri 6): 1e+08 (Bytes/sec)
Unreserved Bandwidth (pri 7): 1e+08 (Bytes/sec)
Link Local ID: 2
Link Remote ID: 1
Interface Switching Capability Descriptor:
Switching Capability: SSC
Encoding: 8
Max LSP BW (pri 0): 781250 (Bytes/sec)
Max LSP BW (pri 1): 781250 (Bytes/sec)
Max LSP BW (pri 2): 781250 (Bytes/sec)
Max LSP BW (pri 3): 781250 (Bytes/sec)
Max LSP BW (pri 4): 781250 (Bytes/sec)
Max LSP BW (pri 5): 781250 (Bytes/sec)
Max LSP BW (pri 6): 781250 (Bytes/sec)
Max LSP BW (pri 7): 781250 (Bytes/sec)
Label TLV Type: 1 Length: 28
Priority (PRI): 128
Action: 4 NumLabels: 128 Length: 24
LowestNCF: 0x6a000000 (grid: 3, cs: 5, lowest N: 0)
BitmapWord: 11111111111111111111111111111111
BitmapWord: 11111111111111111111111111111111
BitmapWord: 11111111111111111111111111111111
BitmapWord: 11111111111111111111111111111111
Shared Risk Link Group 1: 65538

```

Figure 29: Wireshark capture showing the ISCD for Spectrum Switching, with the Bitmap encoding of a 128 nominal central frequencies.

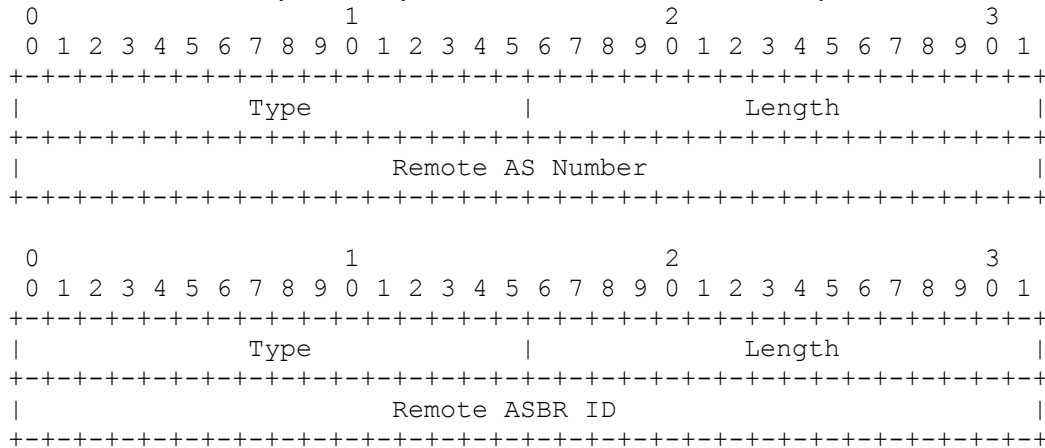
3.3.9 Inter-AS TE Link Update

Inter-AS TE link updates rely on [19], except wavelength-specific availability information. For the advertisement of OSPFv2 inter-AS TE links, we extend the new Opaque LSA, the Inter-AS-TE-v2 LSA, defined in the aforementioned reference, which has the same format as "Traffic Engineering LSA". The advertisement may be carried in a Type 10 Opaque LSA if the flooding scope is to be limited to within the single IGP area to which the ASBR belongs, or MAY be carried in a Type 11 Opaque LSA if the information is intended to reach all routers (including area border routers, ASBRs, and PCEs) in the AS.

The Link State ID of an Opaque LSA is divided into two parts. One of them is the Opaque type (8-bit), the other is the Opaque ID (24-bit). The value for the Opaque type of Inter-AS-TE-v2 LSA is 6. The TLVs within the body of an Inter-AS-TE-v2 LSA have the same format as used in OSPF-TE. The payload of the TLVs consists of one or more nested

Type/Length/Value triplets. Notably, the following remarks should be observed in WP4 implementations:

- The Link ID sub-TLV must not be used in the Link TLV.
- The remote ASBR is identified by the inclusion of the sub-TLVs 21 (Remote AS Number sub-TLV) and 22 (IPv4 Remote ASBR ID sub-TLV)



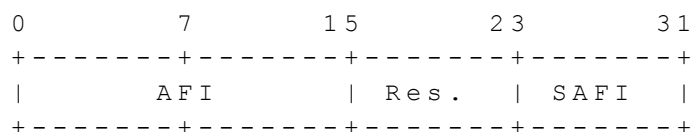
3.4 BGP-LS

The BGP-4 protocol has been extended by IETF to support the exchange of link-state information between two entities [<http://tools.ietf.org/html/draft-ietf-idr-ls-distribution-06>]. In the context of elastic optical networks, and in particular, multi-domain optical networks, BGP-LS can be used as a mean to send the Traffic Engineering Database to a remote Path Computation Element. This PCE can be a CPU intensive Planning Tool, specialized in complex optimizations, running in a dedicated server. Also, the PCE that receives via BGP-LS information can be the parent PCE in a Hierarchical Path Computation context. Summing up, BGP-LS fulfills the function of exporting a Traffic Engineering Database.

In this section, first the procedure to start a BGP-LS session is explained. Then, the encoding of an elastic optical traffic-engineering database is described.

3.4.1 BGP-LS session establishment

A BGP-LS session between two peers starts as a regular BGP-4 session with an exchange of OPEN Messages. The OPEN messages, encoded as per RFC 4271, in BGP-LS are sent with a Capabilities Optional Parameter (Parameter Type 2) (see <http://tools.ietf.org/html/rfc3392#section-4>) with a Multiprotocol Extensions capability (Capability Code 1) (see <http://tools.ietf.org/html/rfc4760#section-8>). This capability is encoded as follows:



In BGP-LS, the Family identifiers, as per early allocation procedure of IANA for <http://tools.ietf.org/html/draft-ietf-idr-ls-distribution-06> are:

- Address Family Number (AFI) -> 16388

- Subsequent Address Family Numbers (SAFI)->71

```

Border Gateway Protocol - OPEN Message
Marker: ffffffffffffffffffffffffffffffffff
Length: 37
Type: OPEN Message (1)
Version: 4
My AS: 1
Hold Time: 180
BGP Identifier: TID_PARENT_PCE (172.16.104.201)
Optional Parameters Length: 8
▼ Optional Parameters
  ▼ Optional Parameter: Capability
    Parameter Type: Capability (2)
    Parameter Length: 6
    ▼ Capability: Multiprotocol extensions capability
      Type: Multiprotocol extensions capability (1)
      Length: 4
      AFI: Link State and TE information (16388)
      Reserved: 00
      SAFI: Link State (71)
  
```

Figure 30: Wireshark capture showing OPEN Message in a BGP-LS session.

Each BGP speaker (peer) is both a client and a server. It will initiate connections and will attend others trying to connect to it. After the exchange of OPEN messages, KA messages are sent from both sides if peers accept the capabilities.

3.4.2 Topology exchange

In order to distribute the Topology, a peer can send Update messages without waiting for any reply from the remote peer. After the BGP session has been established as per the procedure described in the previous subsection, the peer that wants to export the topology should start sending the messages (right after sending the KA messages). How often are the updates sent later will be implementation dependent.

Each update message is encoded as per RFC 4271. An update message is sent per node or link to be exported. Which node or link to send is a policy decision. In the use case of the Planning Tool, policy should be set to send all nodes and links. In the use case of hierarchical PCE, the minimum topology to be sent is the border nodes and the inter-domain links. The base for encoding the information of nodes and links is described in draft-ietf-idr-ls-distribution-06. In the next section, the encoding for elastic optical nodes and links is explained.

3.4.3 Describing Elastic Optical Nodes and Links in BGP-LS

The information of an Elastic Optical Node or Link is sent within an update message. The UPDATE message contains a set of Path Attributes. The node and link information is carried in these Path Attributes. Specifically, when informing of a new node/link (or updating information a new node, the MP_REACH attribute has to be used for part of the information and the rest of the information, related to traffic engineering, is carried in the BGP-LS Attribute.

In particular, the information in the MP_REACH is mainly about addressing of the node/link and the domain it belongs. The MP_REACH and MP_UNREACH attributes are BGP's

containers for carrying opaque information. The MP_REACH is characterized by an AFI, SAFI, and a variable NLRI.

```

+-----+
| Address Family Identifier (2 octets) |
+-----+
| Subsequent Address Family Identifier (1 octet) |
+-----+
| Length of Next Hop Network Address (1 octet) |
+-----+
| Network Address of Next Hop (variable) |
+-----+
| Reserved (1 octet) |
+-----+
| Network Layer Reachability Information (variable) |
+-----+

```

The information of both Elastic Optical Nodes and links, according to BGP-LS specification, are encoded using **AFI 16388 / SAFI 71**. The Network Layer Reachability Information contains the information of nodes and links.

```

      0           1           2           3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+
|           NLRI Type           | Total NLRI Length |
+-----+-----+-----+-----+
|                               |                     |
//                               Link-State NLRI (variable)                               //
|                               |                     |
+-----+-----+-----+-----+

```

The Node NLRI has NLRI Type 1, and will indicate a Node is described.

The Link NLRI has NLRI Type 2, and will indicate a Link is described.

The Optical Node NLRI should be filled as follows:

- Protocol ID: 3 if OSPF is the original source of the configuration. 5 in case of static configuration
- Identifier: 1, Optical Topology
- Mandatory information to be filled: IGP Router ID (515) and Autonomous System (512)
- The IGP Router ID must be an IPv4 (OSPFv2 and OSPFv3 non-pseudo)

Node attributes

1. IPv4 Router-ID of local node is the only required info. Redundant with IGP Router ID
2. Other attributes are optional and considered complementary information.
3. Proposal to add characterization of optical node (port restriction, connectivity matrix, etc)? (2nd stage, yes, but not now)

3.4.1 Describing a Flexi-Grid Link

As in the node case, a flexi-grid link needs to be described by its source and destination (node, interface and domain it belongs to) and the TE information.

Also, the same considerations in terms of source of the information and type of technology apply. Thus, the Optical Link NLRI needs to indicate the source of the information in the protocol ID (3 for OSPF sources, 5 for static configuration). As in the node case, the identifier is 1 (L1 Optical Topology).

The source and destination endpoints of the link are indicated by IPv4 addresses of the nodes, which are encoded in the Link NLRI, in the Local Node Descriptors TLV (information of the source) and Remote Node Descriptors TLV (information of the destination), in the IGP Router-ID fields. Also, the endpoints of the link are characterized by the Autonomous System ID (in the Autonomous System TLV) and Area ID. Thus, in the case of a link between two Autonomous System will have different AS IDs in the Local Node Descriptor and Remote Node Descriptor. In the case of intra-domain link both IDs will be the same. The following figure shows a capture of a Link-State NLRI.

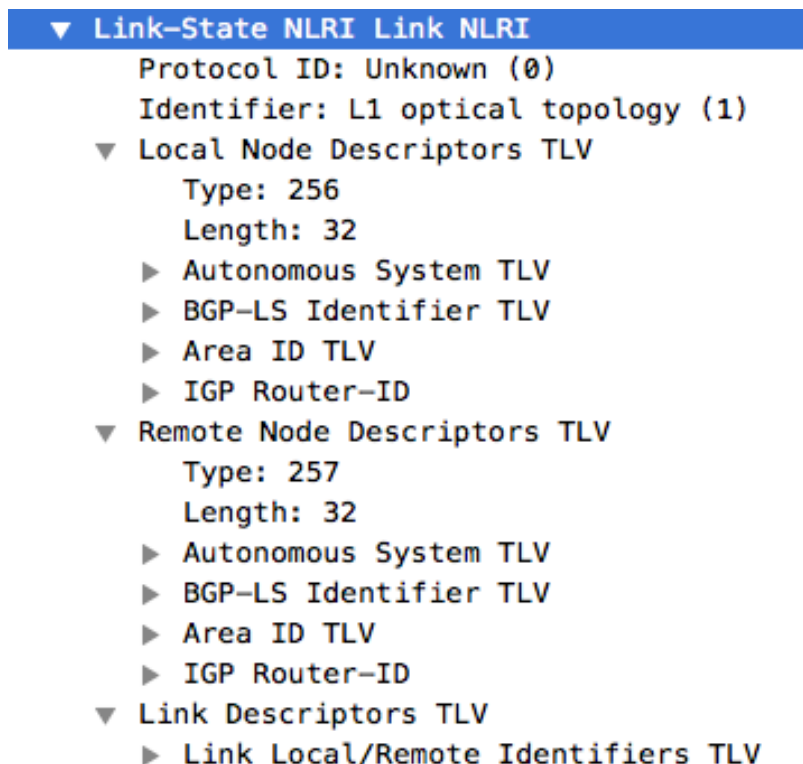


Figure 31: Wireshark capture showing Link-State NLRI.

In a flexi-grid link it is necessary to indicate not only the source and destination nodes, but also the interface where the fiber is connected. The interfaces are identified by the use of unnumbered interfaces, which are encoded in the link local/remote identifiers TLV of the Link NLR. An example is shown bellow:

```

▼ Link Descriptors TLV
  ▼ Link Local/Remote Identifiers TLV
    Type: 258
    Length: 8
    Link Local Identifier: 0x0000000b (11)
    Link Remote Identifier: 0x0000000b (11)
  
```

Figure 32: Wireshark capture showing Unnumbered interfaces

The traffic engineering information of the flexi-grid link attributes is carried in the BGP-LS Attribute. The information to send is:

- IPv4 router IDs of both local and remote nodes (it is redundant with the information sent inside Link-State NLRI, so it can be not sent)
- Optionally: possibility of sending TE information (maximum link BW, unreserved BW, reservable BW, metric, default metric)
- TE Default Metric (1092),
- Optional SRLG (1096)
- For optical network scenarios: New available labels subtlv (NCF set with bitmap without m) is defined (type code: 1200)

Traffic Engineering

```

▼ Path Attribute - LINK_STATE
  ► Flags: 0x10, Length: Well-known, Non-transitive, Complete, Extended Length
  Type Code: LINK_STATE (99)
  Length: 43
  ▼ Link State
    ▼ TE Default Metric TLV
      Type: 1092
      Length: 3
      TE Default Metric: 0x000001 (1)
    ▼ Shared Risk Link Group TLV
      Type: 1096
      Length: 4
      Shared Risk Link Group Value: 0x0065000b (6619147)
    ► [Expert Info (Error/Protocol): Unknown Prefix Descriptor TLV Code (1200)!]
  
```

4 Carrier SDN Orchestrator Architecture: Adaptive Network Manager

4.1 Motivation

Transport networks are in charge of transporting aggregated traffic pipes from multiple users and services among different cities, regions or continents. Traditional carriers' networks operation is very complex and is neither readily adaptable nor programmable to traffic changes. Multiple manual configuration actions are needed in metro and core network nodes. Furthermore, network solutions from different vendors typically use vendor-specific Network Management System (NMS) implementations.

Software Defined Networking (SDN) and network programmability offer the ability to direct application service requests towards the IP/MPLS and optical network. SDN approach can help operators to reduce the CAPEX and OPEX in the networks, thanks to the optimization of the resources and the reduction of the complexity in the operation of the network. However, the proposed SDN controllers in the market are based on monolithic software, which are not adapted to current heterogeneous network environments. Such SDN solutions are like a "black-box" and its deployment leads to different problems for the operator:

1. Vendor lock-in (solutions in the market are mono-vendor)
2. Lack of support for a wide range of forwarding technologies
3. Problems to support E2E multi-domain path establishment

The IETF ABNO architecture is based on existing standard blocks defined within the IETF (PCE, ALTO, VNTM...), which could be implemented either in a centralized or distributed way according to network operator requirements. Thanks to the modular nature of ABNO architecture, building blocks can be deployed by different vendors or third parties and even by a single provider. This modularity and the standard interfaces between the modules solve the problem of vendor lock-in for the operators. On the other hand, ABNO is specially adapted to multi-domain and multi-vendor networks, enabling interoperability between control plane based and OpenFlow based domains.

4.2 Requirements

As data centers and related technology has evolved the concept of elastic computing has been developed, allowing scaling of Virtual Machines (VMs) for existing applications and deployment of new VMs for new applications. What failed to evolve in parallel to data center developments, was the elastic capability in the optical transport network. That is, the ability to scale up or down network connections (including inter-data center connectivity).

As variable bit-rate optical technologies were developed, the question of how to control and manage this new network flexibility, and mirror the flexibility of data centers. The concept of SDN, and discussed in the previous section, offered an architectural approach to facilitate control of transport technologies, and optimize and simplify network operations by more closely binding the interaction (i.e., provisioning, resource messaging, and operations and maintenance) among applications and network services and devices:

1. SDN breaks the vertical integration by separating the network's control logic (the control plane) from the underlying routers and switches that forward the traffic (the data plane).
2. With the separation of the control and data planes, network switches may be controlled via control logic implemented in a centralized controller.
3. The underlay network resources are used in response to requests from network applications. The controller is aware of all switching domains and can set up the end-to-end path.

This simplified architecture is showing the in the following figure.

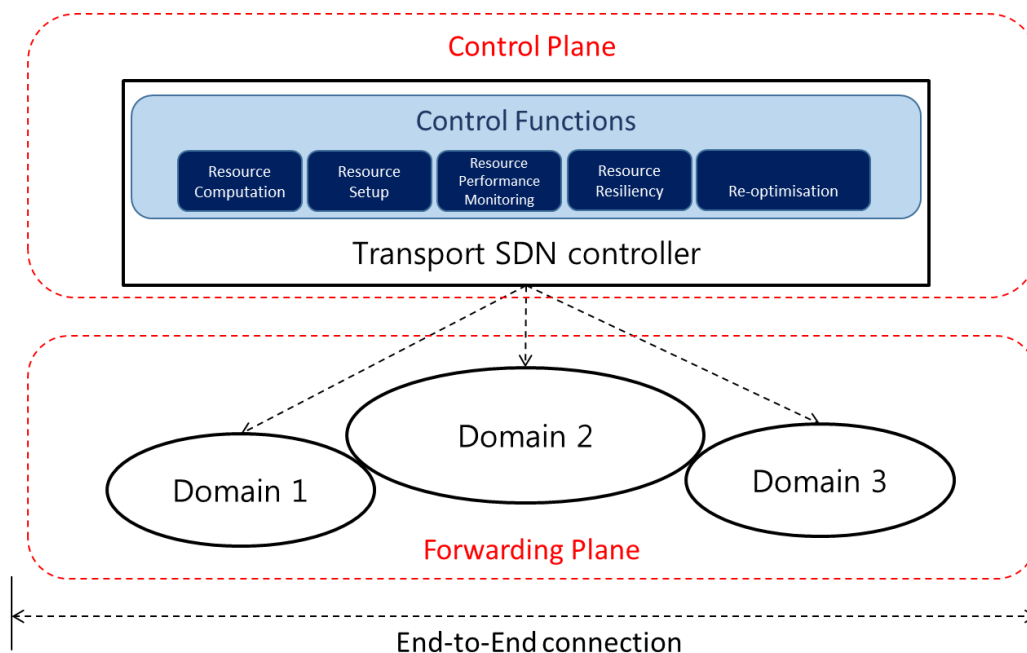


Figure 33: Simplified Network Infrastructure

A fundamental principle of SDN is centralized network control of network resources, this provides a key capability for control and operation of elastic optical networks where a variety of factors need to be considered for optimal operation of the optical transport network resources. These include: wavelength availability and variable bit rate required, non-linearity effects, end-to-end path constraints, metrics and objectives. Therefore, centralization of the network path computation and overall network optimization, sometimes comprise of multi-layers required the use of a central controller architecture.

It is also important to underline that a logically centralized architecture does not assume a physically centralized system. A centralized controller is a single entity that manages all forwarding devices of the network. Naturally, it represents a single point of failure and may have scaling limitations. A hybrid control architecture could receive all requests for services

a central point, but delegate service setup to agents running distributed control plane elements for actual service setup, and local protection switching decisions in the event for failure.

Typically underlay resources would be provisioned via a southbound APIs, device drivers or signaling protocol. By providing a common interface for the upper layers, while allowing a control platform to use different southbound APIs as transport technologies evolves or multiple layers are introduced.

It is important to design a platform that is modular and extensible. As new use cases are developed function and interfaces can be added to existing platform, rather than having to replace the entire platform.

4.2.1 IDEALIST Platform Principles

Generally, the operational requirements include:

- Enable significantly greater network efficiency, leveraging the variable bit-rate capabilities of the technologies available
- Provide a wide variety of agile (elastic) services in rapid time
- Significantly reduce operational costs while simplifying network operations
- Enable more efficient setup, monitoring and diagnostics of end-to-end services
- Provide the means to control and operate a converged (optical, ethernet, packet, IP) infrastructure

These requirements become the core principles which the SDN-based central controller would have to solve.

4.2.2 Capability Requirements

Future operation of the elastic optical transport network would require a number of core capabilities, these include:

- Capable of representing the flexible (variable bit rate) technologies
- Centralized control functions, including resource discovery, computation and explicit path based on optimization of multi-constraints
- Southbound control interfaces for a variety of flexible optical technologies
- Extensible future northbound client or application requests

4.3 Framework and Evolution (ODC)

Once we established the technical requirements and core principles we needed to design a framework and identify the most suitable SDO to develop our proposal. The purpose of a functional architecture is to decompose a problem space.

1. Define the framework

A framework is not a blue-print for implementation, as components can be used or disregarded as the application dictates.

2. Separate distinct and discrete functions into required components

Which functions are separate, either logically or physically? Typically this can be done by identifying if a protocol mechanism is required to communicate data.

3. Identify the functional interactions between components

A protocol provides a realization of the interaction between two functional components. If functions can co-existing interfaces (protocol mechanisms) are unlikely to be needed.

4. Address a range of network operation and management scenarios, including the implementation and deployment options.

These became the operational use cases from the IDEALIST operators and content providers.

5. Highlight existing protocols and components that may be suitable within the framework and derived architectures.

A clear design goal of the IDEALIST control architecture was to develop innovative technology where required, but extend or adapt existing technology where possible, in order to meet the overall operational objectives of the platform. This would also allow well-defined (standardized) interfaces to be used, and extended for new capabilities.

Based on the use cases and clear design goals ABNO was developed to integrate together many existing technologies, to meet the control platform objectives discussed previously. ABNO Functions include gathering information about the resources available in a network, consideration of topologies and how those topologies map to underlying network resources, methods for requesting a path, and for provisioning or reserving network resources. Thus, ABNO may be seen as the use of a toolbox of existing components enhanced with a few new elements. The key component within an ABNO is the PCE and its multiple architectures (hierarchical, stateful and active), which can be used for computing paths across single or multi-domain networks.

It was also important that the controller framework and development achieved within the IDEALIST project, might be standardized and benefit from additional input from the wider industry and academia. Therefore, it became compelling to take our vision for controller and propose the framework within the IETF.

How the ABNO architecture was constructed for the ANM platform and which key modules are used, is described in the following sections.

4.4 Modules within the architecture

The main components of the ABNO architecture are listed below:

- **ABNO Controller.** The ABNO Controller is the main component of the architecture and is responsible of orchestrating, and invokes the necessary components in the right order. It listens for request from the NMS/OSS and selects the appropriate workflow to follow in order to satisfy each request.
- **Policy Agent.** The Policy Agent is the entity that stores the restrictions and policies, and is responsible for propagating those policies into the other components of ABNO architecture.
- **Path Computation Element.** The PCE is defined in [RFC4655](#), and it's the unit that handles the path computation across the network graph. It can calculate traffic engineered end-to-end paths in order to optimize the optical spectrum consumption within the network.

The PCE is capable of computing a TE LSP by operating on the TED regarding to the available bandwidth and network constraints. Coordination between multiple PCEs operating on different TEDs is also required for performing path computation in multi-

domain (for example, inter-AS) or multi-layer networks. The PCE has different modes of operation, stateless, stateful and active.

- **Virtual Network Topology Manager (VTNM).** VNTM is defined in RFC5212 and it's in charge of maintaining the topology of the upper layer by connections in the lower layer. The LSPs established at the layer 0 network are advertised to the layer 3 resources as virtual links to provide connectivity.

This entity simplifies the upper-layer routing and traffic engineering decisions as it hides the optical connections set up by the LSP. It can also respond to traffic demands, topology changes or network failures by releasing unused resources.

- **ALTO Server.** The ALTO protocol is defined as a service that provides network information to the application layer based on abstract maps of the underlying network. This information provides a simplified view, but it is useful to route the traffic of the application layers. ALTO Services enable to share information about network locations and costs between them. The selection criteria to choose between two locations may depend of information such as maximum bandwidth, minimum cross-domain traffic, lower cost to the user, etc.
- **I2RS Client.** The Interface to the Routing System (I2RS) is designed to provide a programmatic way to access control plane capability that exists in routers. It will also report routing state and policy information, and inject routing rules and access lists into the routing information base.
- **OAM Handler.** The Operations, Administration, and Maintenance Handler is the entity responsible for monitoring the behavior of the network. This module does not have to carry out the monitoring itself, but to retrieve and to forward the required information to its surrounding modules. Thus is, OSS systems, ABNO controller and specific measurements to the monitoring probes in the network.

The OAM Handler can also be used to verify new connections, links and performance properties within the network.

Due to the novelty of this approach is out of the scope of this architecture for the moment. Future documents take into account this module if the technology is accepted.

- **Topology Module.** The Topology Module is responsible for storing and providing network topology information, both per-layer topologies as well as inter-layer topology. One part of the module is devoted to getting and maintaining up to date the topology from the available sources (e.g. routing protocols, mainly for the more dynamic information). The other part of the module is devoted to providing the information to the requesting parties such as the Provisioning Manager, PCE or VTNM.

The Topology Module has multiple databases: a view of each layer, the inter-layer information and an inventory DB with the configuration parameters of each of the resources (location, address, vendor...) and the details of the available interfaces of each the resources.

- **Provisioning Manager.** The Provisioning Manager is the unit in charge of configuring the network elements so the LSP can be established. It can do so both by configuring the resources through the data plane or by triggering a set of actions to the control plane.

There are several protocols that allow the configuration of specific network resources such as Openflow, Netconf, CLI and PCEP.

4.5 Interfaces Definition

ABNO architecture is under definition now. Most of the interfaces are defined in the architecture and they will be used as a proof of concept in this project. **Figure 30** shows the preliminary definition of interfaces between the main components of the ABNO architecture.

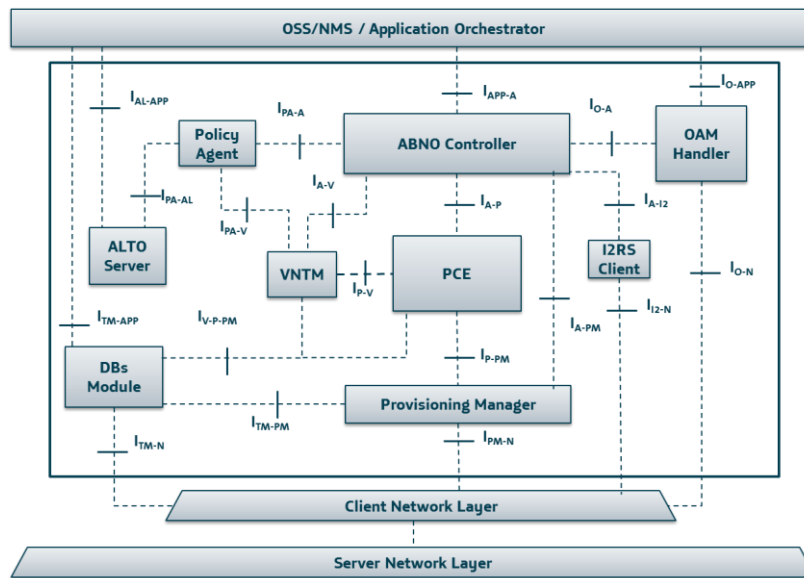


Figure 34: Interfaces for Application-Based Network Operation Architecture

- **I_{APP-A}**. This is the interface between the Application layer/NMS/OSS and the ABNO controller. Application layer makes requests to set up connections or to trigger any other workflow using HTTP/JSON. This interface is not yet defined in IETF.
- **I_{AL-APP}**. This is the interface between the ALTO Server and Application layer/NMS/OSS, where the Application layer acts as an ALTO Client. They communicate using the ALTO Protocol.
- **I_{TM-APP}**. This interface connects the Databases Module (DBs Module) and the Application layer/NMS/OSS. They communicate over HTTP/JSON. An information model has to be defined for this interface to support TED, LSPs and inventory requests.
- **I_{A-12}, I_{12-N}**. The Interface to the Routing System (I2RS) is, for the moment, a very immature technology, so it is not considered in the workflows. It is included in the ABNO architecture, because it is expected that it will become more mature during the project duration.
- **I_{PA-A}, I_{PA-V}, I_{PA-AL}**. All the interactions between the different modules and the Policy Agent should be done through HTTP/JSON when the latter is implemented.
- **I_{A-P}**. This is the interface between the ABNO controller and the PCE. The ABNO controller queries the PCE using PCEP to determine which services can be provisioned. A PCE can be either stateful or stateless. Thus, this interface will support queries for both types of PCEs.
- **I_{A-V}**. This interface connects the ABNO controller and the VNTM. They communicate through PCEP.

- **I_{A-PM}**. This is the interface between the ABNO controller and the Provisioning Manager. The requests are channeled through PCEP.
- **I_{V-TM}, I_{P-TM}**. This interface uses NETCONF/YANG to interconnect the VNTM and PCE with the Provisioning Manager.
- **I_{P-V}**. This is the interface between the PCE and the VNTM. PCE makes path computation requests using PCEP.
- **I_{P-PM}**. This is the interface between the PCE and the Provisioning Manager. If the PCE is stateless, the interface **I_{P-PM}** does not apply. If the PCE is stateful and active, the interface **I_{P-PM}** is used to forward path instantiations to the PM. Note that another possibility is to use PCEP towards the head end LSR directly from the PCE.
- **I_{TM-PM}**. This is the interface between the Provisioning Manager and the Topology Module. They communicate over HTTP/JSON,
- **I_{TM-N}**. This is the interface between the Topology Module and the Network Layer. The Topology Module can update the link-state information using multiple protocols like OSPF or BGP-LS.
- **I_{PM-N}**. This is the interface between the Provisioning Manager and the network elements. Depending on the router description, the configuration can be done using Openflow, Netconf or PCEP.
- **I_{O-APP}, I_{O-A}**. This is the interface between the OAM Handler and the application layer, ABNO controller, respectively.
- **I_{O-N}**. This is the interface between the OAM Handler and the network elements. There are several options for this interface like SNMP, Netflow, depending on the measurements to be done in the network elements.

4.6 Summary of the Implementation Advances in the ANM

The ABNO architecture groups a number of standard components achieving different functions. For the set of use cases developed in this project, a *lite* version of the architecture is implemented in the project as a proof-of-concept. More complex services require additional modules, like I2RS, ALTO or policy agent, that are not required for the use cases defined in this project. The flexibility provided by the ABNO framework proposed in the innovation project can be exploited to pursuit other benefits, such as a better utilization of the network resources, in terms of capacity links, physical server consolidation, etc.

The modules of the ABNO architecture implemented in the innovation project to support the use cases demonstrations are highlighted in blue color in Figure 31.

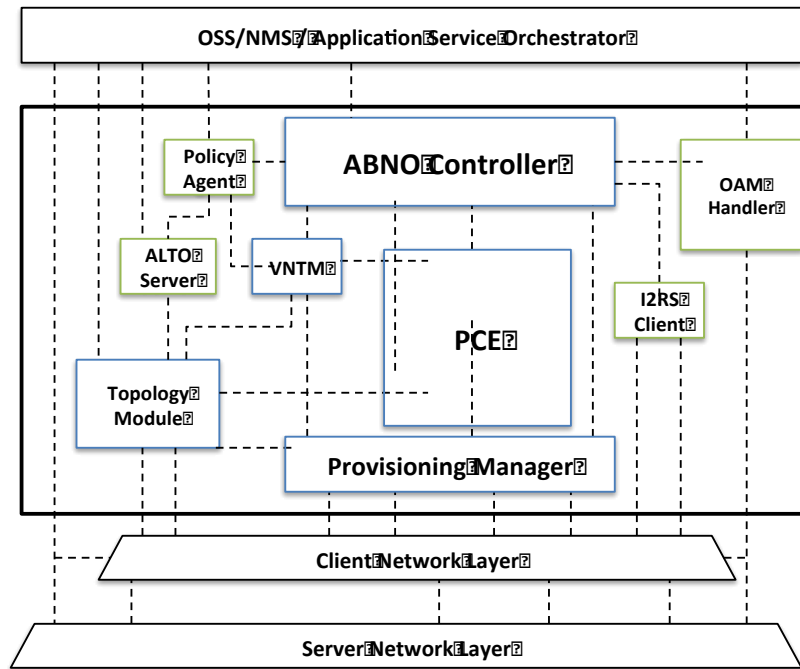


Figure 35: ANM architecture implemented

4.7 Implementation feedback

The following table contains the protocols that are used and the missing interfaces for the use cases tested. There are some interfaces that are defined but not shown here because they are not used in the use cases.

Table III: Implementation feedback to WP3

	L0 Service	IP Link Provisioning	MPLS Service	Re-optimization process
I _{APP-A}	NBI	NBI	NBI	NBI
I _{AL-APP}				
I _{TM-APP}				
I _{A-I2} , I _{I2-N}				
I _{PA-A} , I _{PA-V} , I _{PA-AL}	Not def	Not def	Not def	Not def
I _{A-P}	PCEP	PCEP	PCEP	PCEP
I _{A-V}	-	PCEP	PCEP	-
I _{A-PM}	PCEP (L0 PCE stateless)	PCEP (L0 PCE stateless)	PCEP (L0 or L2 PCE stateless)	PCEP (L0 PCE stateless)
I _{V-TM} , I _{P-TM}	YANG model	YANG model	YANG model	YANG model
I _{P-V}	-	PCEP	PCEP	-
I _{P-PM}	PCEP (L0 PCE stateful)	PCEP (L0 PCE stateful)	PCEP (L0 or L2 PCE stateful)	PCEP (L0 PCE stateful)
I _{TM-PM}	YANG model	YANG model	YANG model	YANG model

I_{TM-N}	OSPF	OSPF REST/API vendor	OSPF REST/API vendor	OSPF
I_{PM-N}	PCEP	CLI Netconf PCEP REST/API vendor	CLI Netconf PCEP REST/API vendor	PCEP

In the following section we provide feedback on several aspects of the ABNO architecture and the protocols being used.

As it has already been mentioned throughout the document, the Policy Agent is not defined yet in any standardization group. Consequently, we cannot specify the protocol to be used in such module yet. However, it is not mandatory to justify a behavioural test. Further research will be required to identify the key requirements and protocol mechanisms that could be suitable.

Regarding PCEP, the main drawback detected is related to its inherent request-response nature; PCRep messages can be used to specify the route and spectrum allocation for new or already established LSPs in response to PCReq messages where those LSP are explicitly identified. To deal with this PCEP behaviour in the context of re-optimization, the fPCE must select a set of LSPs candidate to be re-optimized. Notwithstanding, in multilayer networks, new LSPs could need to be created, e.g. for VNT reconfiguration purposes, which is against current standards. Therefore, new IETF PCEP extensions to cope with this problem are needed.

Another issue is related to the lack of information about the physical equipment, e.g. nodes, cards, ports and fibres and its relation to the data stored in the VNTM. This fact, limits the capability of using a planning tool to perform network capacity planning related to network operation. Using standard modelling languages, such as YANG, to model not only the data in the TM, but also correlate those to inventory data would bring new opportunities to reconfigure not only the LSPs but also the network itself to adapt quicker its capacity in response to mayor changes, such as disaster recovery.

An important characteristic must be taken into consideration when working with Rest/API interfaces. They are vendor specific, thus the PM and the TM must be adapted accordingly every time a different vendor is used.

Regarding the NBI, within the IDEALIST project we developed simple NBI capability, this is documented in the following section. However, no formal specifications exists across the various standards organisations for detailed communication from the client or application layer for a request, status, and tear down, of a single or set of services. There are some attempts by the ONF to define it, but for the moment they are not working in any implementation. Again, further research would allow for requirements development, but it is unlikely existing solutions would be suitable, therefore a new political mechanism would need to be developed.

5 Interface between network and applications

5.1 North Bound Interface for the Adaptive Network Manager

This section shows how to ask the ABNO controller for different kinds of provisioning using the North Bound Interface. For every provisioning use case are shown all the demo steps and the parameters for the GET or POST requests. This document is focused in three use cases: IP Link provisioning, MPLS Service Provisioning and Layer Zero provisioning.

5.1.1 North Bound Interface for Layer Zero Provisioning

The parameters in the NBI for the Layer Zero provisioning request are the following:

- **ID_Operation (mandatory).** This parameter is a unique number to identify the operation from each user. This value is used to correlate different operations with the ABNO controller like deletion, re-optimization.
- **Operation Type (mandatory).** This string is mapped to the workflow that the ABNO controller has to execute. For this workflow the value is: "L0ProvisioningWF".
- **Source and Destination Node (mandatory).** These variables contain the management IPs used to identify (and configure) the routers.
- **Operation (mandatory).** This parameter can take two values: add, to create a new L0 service, and delete, to turn off an existing service.

Example:

```
{  
  "ID_Operation": "3456"  
  "Operation Type": "L0ProvisioningWF",  
  "Source Node": "172.16.2.1",  
  "Destination Node": "172.16.2.3",  
  "Operation": "add"  
}
```

5.1.2 North Bound Interface for IP Link Provisioning

The parameters in the NBI for the IP Link provisioning request are the following:

- **ID_Operation (mandatory).** This parameter is a unique number to identify the operation from each user. This value is used to correlate different operations with the ABNO controller like deletion, re-optimization.
- **Operation Type (mandatory).** This string is mapped to the workflow that the ABNO controller has to execute. For this workflow the value is: "IPProvisioningWF".
- **Source and Destination Node (mandatory).** These variables contain the management IPs used to identify (and configure) the routers.
- **Interfaces ID (optional).** These parameters contain the IPs to configure the interfaces on the routers. From the control plane point of view they are Numbered

Interfaces. In case these values are not included, the provisioning manager assigns a value to them.

- **IGP ID (optional).** This parameter is the AS identifier used to export the node topology via OSPF. When this value is not defined, the provisioning module carries out a default operation (assigning an OSPF value or adding a static route).
- **Operation (mandatory).** This parameter can take two values: add, to create a new MPLS service, and delete, to turn off an existing service.

Example:

```
{  
  "ID_Operation": "1234"  
  "Operation Type": "IPProvisioningWF",  
  "Source Node": "10.95.73.72",  
  "Destination Node": "10.95.73.74",  
  "Interfaces ID": ["70.70.70.1", "70.70.70.2"],  
  "Operation": "add",  
  "IGP ID": "100"  
}
```

5.1.3 North Bound Interface for MPLS Service Provisioning

The parameters in the NBI for the MPLS Service provisioning request are the following:

- **ID_Operation (mandatory).** This parameter is a unique number to identify the operation from each user. This value is used to correlate different operations with the ABNO controller like deletion, re-optimization.
- **Operation Type (mandatory).** This string is mapped to the workflow that the ABNO controller has to execute. For this workflow the value is: "MPLSProvisioningWF".
- **Source and Destination Node (mandatory).** These variables contain the management IPs used to identify (and configure) the routers.
- **Operation (mandatory).** This parameter can take two values: add, to create a new MPLS service, and delete, to turn off an existing service.
- **Bandwidth (mandatory MPLS).** It contains the amount of the resources required for the connection.

Example:

```
{  
  "ID_Operation": "2345"  
  "Operation Type": "MPLSProvisioningWF",  
  "Source Node": "10.95.73.72",  
  "Destination Node": "10.95.73.74",  
  "Operation": "add",  
}
```

"Bandwidth": 200

}

5.1.4 North Bound Interface for Re-optimization Process

The parameters in the NBI for the Layer Zero provisioning request are the following:

- **ID_Operation (mandatory).** This parameter is a unique number to identify the operation from each user. This value is used to correlate different operations with the ABNO controller like deletion, re-optimization.
- **Operation Type (mandatory).** This string is mapped to the workflow that the ABNO controller has to execute. For this workflow the value is: "LOProvisioningWF".
- **Source and Destination Node (mandatory).** These variables contain the management IPs used to identify (and configure) the routers.
- **Operation (optional).** This parameter can take two values: add, to create a new L0 service, and delete, to turn off an existing service.

Example:

```
{  
  "ID_Operation": "3456"  
  "Operation Type": "LinkProvisioning_ReOpt_WF",  
  "Source Node": "10.10.0.6",  
  "Destination Node": "10.10.0.7",  
  "Operation": "add"  
}
```

5.1.5 North Bound Interface for dynamic bandwidth allocation

The parameters will be similar as a MPLS Service provisioning request, but in this case minimum and nominal bandwidths will be specified:

- **ID_Operation (mandatory).** This parameter is a unique number to identify the operation from each user. This value is used to correlate different operations with the ABNO controller like deletion, re-optimization.
- **Operation Type (mandatory).** This string is mapped to the workflow that the ABNO controller has to execute. For this workflow the value is: "MPLSProvisioningWF".
- **Source and Destination Node (mandatory).** These variables contain the management IPs used to identify (and configure) the routers.
- **Operation (mandatory).** This parameter can take two values: add, to create a new MPLS service, and delete, to turn off an existing service.
- **Minimum Bandwidth (mandatory).** It contains the minimum amount of resources for the connection
- **Bandwidth (mandatory).** It contains the nominal bandwidth



Example:

```
{  
  "ID_Operation": "2345"  
  "Operation Type": "MPLSProvisioning_DBA_WF",  
  "Source Node": "10.95.73.72",  
  "Destination Node": "10.95.73.74",  
  "Operation": "add",  
  "Minimum Bandwidth": 20  
  "Bandwidth": 200  
}
```

5.1.6 ABNO REST API Summary

REST API is the chosen interface to use the ABNO features. The URI is always the same: http://<IP_Server>:<Port> (Replacing the inet address with the proper one), setting in the parameter "Operation_Type" the requested functionality. The method is GET for all the workflows.

Example:

Curl

```
'172.16.1.3:4445?Source_Node=10.95.73.72&Destination_Node=10.95.73.74&Operation=add&Bandwidth=3000&Operation_Type=MPLSProvisioningWF'
```

The next table shows the "Operation_Type" for these use cases a short description and the parameters:

Operation_Type	Description	Parameters
IPLinkProvisioningWF	Creates a new IP Link between two routers, configuring (if necessary) optical equipment	ID_Operation, Operation_Type, Source_Node, Destination_Node, Interfaces_ID, Operation, IGP ID
MPLSProvisioningWF	Creates a new path between two routers using existing IP links	ID_Operation, Operation_Type, Source_Node, Destination_Node, Operation, Bandwidth
L0ProvisioningWF	Creates a new optical path	ID_Operation, Operation_Type, Source_Node, Destination_Node, Operation
LinkProvisioning_ReOpt_WF	Request for a connection and a re-optimization process if there is no space	ID_Operation, Operation_Type, Source_Node, Destination_Node, Operation
MPLSProvisioning_DBA_WF	Creates a new path between two routers using existing IP links with dynamic bandwidth	ID_Operation, Operation_Type, Source_Node, Destination_Node, Operation, Minimum Bandwidth, Bandwidth

5.2 YANG Data Model for a Traffic Engineering Database of Optical Networks with Flex-grid

One of the objectives of the project is to dynamically allocate network resources at both IP and elastic optical layers, applying new optimization tools that enable both off-line planning and on-line network re-optimization in elastic optical networks. In order to meet this objective, it is necessary to develop new ways to easily configure the optical switches, dealing with flexi-grid specificities. Such configuration has to be able to describe an optical path among several switches, describing the optical channels that are used on each link.

Several interfaces can be used right now by a network operator to configure switches, such as Command Line Interface or a Web interface, being NETCONF [26] the most promising solution to automate the configuration of the network devices. However, the adoption of NETCONF by vendors has been very slow, due to a lack of data models that describe the forwarding technology capability and configuration of the forwarding technology in a standard way. To solve this situation, YANG [27] has been proposed by the IETF as the data modeling language for NETCONF, to define different configuration models.

As stated in [28], Capacity Planning and Traffic Engineering applications need a normalized view of the network topology. Following this idea, this chapter presents a YANG model that provides a holistic view of an optical network, including the nodes, transponders and links between them, as well as how such links interconnect nodes and transponders. The model has taken into account that two different optical technologies will be present in future devices: DWDM and flexi-grid. It also allows the representation of the optical layer of a network, combined with the underlying physical layer. The model is defined in two YANG modules:

- Optical-TED (Traffic Engineering Database): This module defines all the information needed to represent an optical node, an optical transponder and an optical link.
- Media-channel: This module defines the whole path from a source transponder to the destination through a number of intermediate nodes.

This model will be a valuable tool in a scenario based on Application-Based Network Operations (ABNO) [29]. The specification of the model proposed in this chapter provides an ABNO controller means for the on-demand provisioning or optimization of optical paths. This application can be seen as a feasible Software Defined Network (SDN) approach for next-generation optical networks [30].

The development of the holistic model for dynamic optical networks in the YANG format has also been submitted as a proposal in the IETF.

The rest of this chapter is structured as follows. Next section an overview of the optical network topology model. For this, we explain what a YANG data model is, what tools are currently available for YANG specifications, and also already defined models that are related to this proposal. After this, the proposed optical TED and media channels YANG models are detailed in section 6.2. An application example is later provided in section 6.3 to better understand their utility.

5.2.1 Optical Network Topology Model Overview

YANG is a data modeling language used to model configuration data manipulated by the NETCONF protocol. In order to explain the proposed information model to describe an optical network topology, it is necessary to pinpoint first the main YANG characteristics, which are specified in [27]:

- YANG models the hierarchical organization of data as a tree in which each node has a name, and either a value or a set of child nodes. YANG provides clear and concise descriptions of the nodes, as well as the interaction between those nodes.
- YANG structures data models into modules and submodules. A module can import data from other external modules, and include data from submodules. The hierarchy can be augmented, allowing one module to add data nodes to the hierarchy defined in another module. This augmentation can be conditional, with new nodes appearing only if certain conditions are met.
- YANG models can describe constraints to be enforced on the data, restricting the appearance or value of nodes based on the presence or value of other nodes in the hierarchy. These constraints are enforceable by either the client or the server, and valid content **MUST** abide by them.
- YANG defines a set of built-in types, and has a type mechanism through which additional types may be defined. Derived types can restrict their base type's set of valid values using mechanisms like range or pattern restrictions that can be enforced by clients or servers. They can also define usage conventions for use of the derived type, such as a string-based type that contains a host name.
- YANG permits the definition of reusable groupings of nodes. The instantiation of these groupings can refine or augment the nodes, allowing it to tailor the nodes to its particular needs. Derived types and groupings can be defined in one module or submodule and used in either that location or in another module or submodule that imports or includes it.
- YANG data hierarchy constructs include defining lists where list entries are identified by keys that distinguish them from each other. Such lists may be defined as either sorted by user or automatically sorted by the system. For user-sorted lists, operations are defined for manipulating the order of the list entries.

For more information on YANG, the document [31] provides a tutorial with some examples on how to model the information and use the YANG structures.

Several YANG models have already been specified for network configurations. For instance, the work in [32] has proposed a YANG model of a TED, but only covering the IP layer. A YANG model has also been proposed in [33] to configure optical DWDM parameters. On the other hand, a TED has been proposed for optical networks in [34], but this approach did not specify a YANG model to enable its configuration.

As stated before, we propose a model to describe an optical topology that is split in two YANG modules:

- **Optical-TED:** In order to be compatible with existing proposals, we augment the definitions contained in [32], by defining the different elements we find in an optical network: a node, a transponder and a link. For that, each of those elements are defined as containers that include a group of attributes. References to the elements are provided to be later used in the definition of a media channel. It also includes the data types for the type of modulation, the optical technology, the FEC, etc.
- **Media-channel:** This module defines the whole path from a source transponder to the destination through a number of intermediate nodes and links. For this, it takes the information defined before in the optical TED.

Next section provides a detailed view of each module.

<output-port>: Boolean value that defines whether the interface is output or not.

<description>: Description of the usage of that interface.

<interface-type>: Determines if the interface is numbered or unnumbered.

<numbered-interface> ::= <n-i-ip-address>

<numbered-interface>: A interface with its own IP address

<n-i-ip-address>: Only available if <interface-type> is "numbered-interface". Determines the IP address of the interface.

<unnumbered-interface> ::= <u-i-ip-address> <label>

<unnumbered-interface>: A interface that needs a label to be unique

<u-i-ip-address>: Only available if <interface-type> is "numbered-interface". Determines the IP address of the node, which with the label, defines the interface.

<label>: Label that determines the interface, joint with the node IP address.

<connectivity-matrix> ::= <connections>

<connectivity-matrix>: Determines whether a connection port in/port out exists.

<connections> ::= <input-port-id> <output-port-id>

<connections>: The actual connection between an input port and an output port

<input-port-id>: The input port associated with the output port.

<output-port-id>: The output port associated with the input port.

<optical-transponder> ::= <optical-transponder-attributes> <optical-node-attributes>

<optical-transponder>: Determines an optical transponder in the network

<optical-transponder-attributes> ::= <available-modulation> <modulation-type> <available-FEC> <FEC-enabled> [<FEC-type>]

<optical-transponder-attributes>: Contains all the attributes related to the transponder, such as whether it has FEC enabled or not, or its modulation type..

<available-modulation>: It provides a list of the modulations available at this transponder.

<modulation-type>: Determines the type of modulation in use: QPSK, QAM16, QAM64...

<available-FEC>: It provides a list of the FEC algorithms available at this transponder.

<FEC-enabled>: Boolean value that determines whether is the FEC enabled or not.

<FEC-type>: Determines the type of FEC in use: reed-solomon, hamming-code, enum golay, BCH...

<optical-node-attributes>: See above, node attributes are reused also for transponders.

<optical-sliceable-transponder> ::= <carrier-id> <optical-transponder-attributes>

<optical-sliceable-transponder>: It provides a list of transponders.

<carrier-id>: An identifier for each one of the transponders in the list.

<optical-transponder-attributes>: See above, transponder attributes are reused also for sliceable transponders.

<link> ::= <optical-link-attributes>

<link>: This element describes all the information of a link.

<optical-link-attributes> ::= <link-id> <technology-type> <available-label-flexigrid> <available-label-WSO> <N-max> <base-frequency> <nominal-central-frequency-granularity> <slot-width-granularity>

<optical-link-attributes>: Contains all the attributes related to the link, such as its unique id, its N value, its latency, etc.

<link-id>: Unique id of the link

<technology-type>: Optical technology used in this link: Flexigrid, WDM50, WDM100...

<available-label-flexigrid>: Array of bits that determines, with each bit, the availability of each interface for flexigrid technology.

<available-label-WSO>: Array of bits that determines, with each bit, the availability of each interface for WSO technology.

<N-max>: The max value of N in this link, being N the number of slots.

<base-frequency>: The default central frequency used in the link.

<nominal-central-frequency-granularity>: It is the spacing between allowed nominal central frequencies and it is set to 6.25 GHz (note: sometimes referred to as 0.00625 THz).

<slot-width-granularity>: 12.5 GHz, as defined in G.694.1.

5.2.2.2 Media-channel.yang

The figure below shows a class diagram of the media channel data model.

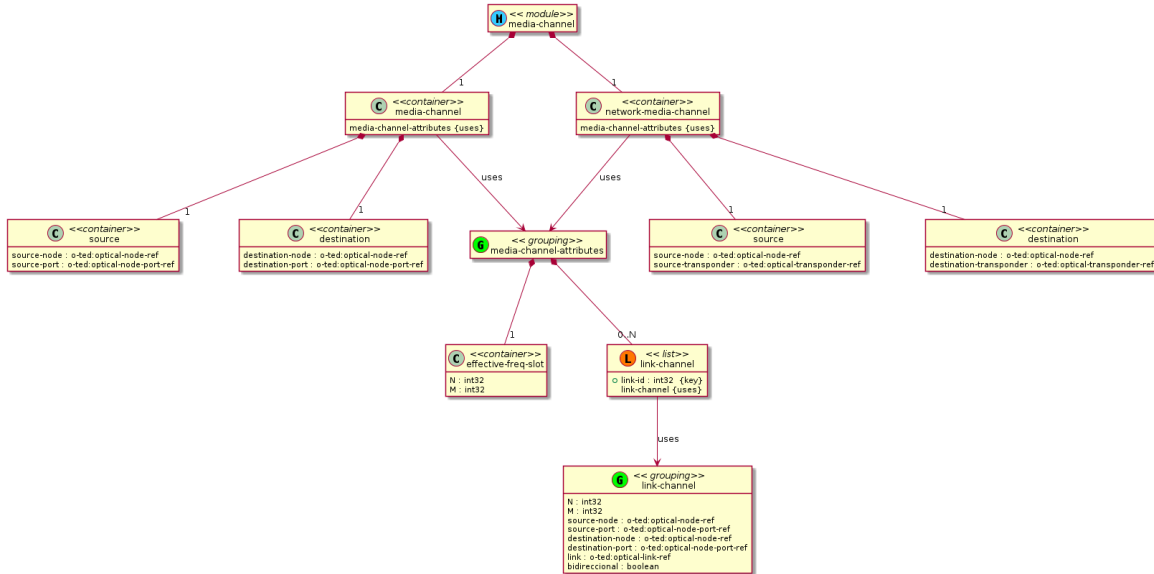


Figure 37. Media channel class diagram

We have defined two types of media channel, depending on where they extend: media-channel, if the channel is between nodes, and network-media channel, if the channel is between transponders. The description of these components is provided below:

<media-channel> ::= <source> <destination> <link-channel> <effective-freq-slot>

<media-channel>: Determines a media-channel and its components.

<source> ::= <source-node> <source-port>

<source>: In a media-channel, the source is a node and a port.

<source-node>: Reference to the source node of the media channel.

<source-port>: Reference to the source port in the source node.

<destination> ::= <destination-node> <destination-port>

<destination>: In a media-channel, the destination is a node and a port.

<destination-node>: Reference to the destination node of the media channel.

<destination-port>: Reference to the destination port in the destination node.

**<link-channel> ::= <link-id> <N> <M> <source-node> <source-port>
<destination-node> <destination-port> <link> <bidirectional>**

<link-channel>: Defines a list with each of the links between elements in the media channel.

<link-id>: Unique identifier for the link channel

<N>: N used for this link channel.

<M>: M used for this link channel.

<source-node>: Reference to the source node of this link channel.

<source-port>: Reference to the source port of this link channel.

<destination-node>: Reference to the destination node of this link channel.

<destination-port>: Reference to the destination port of this link channel.

<link>: Reference to the link of this link channel.

<bidirectional>: Indicates if this link is bidirectional or not.

<effective-freq-slot> ::= <N> <M>

<effective-freq-slot>: Defines the effective frequency slot of the media channel, which could be different from the one defined in the link channels.

<N>: Defines the effective N for this media channel.

<M>: Defines the effective M for this media channel.

<network-media-channel> ::= <source> <destination> <link-channel> <effective-freq-slot>

<network-media-channel>: Determines a network media-channel and its components.

<source> ::= <source-node> <source-transponder>

<source>: In a network media channel, the source is defined by a node and a transponder.

<source-node>: Reference to the source node of the media channel.

<source-transponder>: Reference to the source transponder in the source node.

<destination> ::= <destination-node> <destination-transponder>

<destination>: In a network media channel, the destination is defined by a node and a transponder

<destination-node>: Reference to the destination node of the media channel.

<destination-port>: Reference to the destination port in the destination node.

<link-channel>: See above, the information is reused for both types of media channels.

<effective-freq-slot>: See above, this information is reused for both types of media channels.

5.2.3 Example of Use

In order to explain how this model is used, we provide the following example. An optical network usually has multiple transponders, switches (nodes) and links between them. Figure 34 shows a simple topology, where two physical paths interconnect two optical transponders.

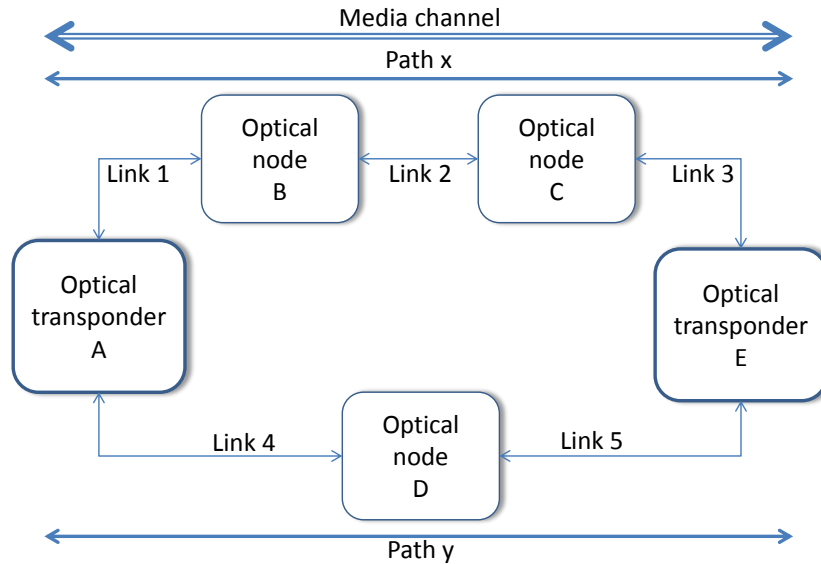


Figure 38. Topology example

In order to configure a media channel to interconnect transponders A and E, first of all we have to populate the optical TED YANG model with all elements in the network:

1. We define the transponders A and E, including their FEC type, if enabled, and modulation type. We also provide node identifiers and addresses for the transponders, as well as interfaces included in the transponders. It is also possible sliceable transponders if needed.
2. We do the same for the nodes B, C and D, providing their identifiers, addresses and interfaces, as well as the internal connectivity matrix between interfaces.
3. Then, we also define the links 1 to 5 that interconnect nodes and transponders, indicating which labels are available, both in flexi-grid or WSON. Other information, such as the slot frequency and granularity are also provided.

Next, we can configure the media channel from the information we have stored in the optical TED, by querying which elements are available, and planning the resources that have to be provided on each situation. Note that every element in the optical TED has a reference, and this is the way in which they are called in the media channel.

4. Depending on the case, it is possible to define either the source and destination node ports, or the source and destination node and transponder. In our case, we would define a network media channel, with source transponder A and source node B, and destination transponder E and destination node C. Thus, we are going to follow path x.
5. Then, for each link in the path x, we indicate which channel we are going to use, providing information about the slots, and what nodes are connected.

Finally, the optical TED has to be updated with each element usage status each time a media channel is created or torn down.

6 Traffic Monitoring and OAM

6.1 Monitoring Architecture

As it was described in previous deliverables D3.1 and D4.1, monitoring probes will be added at the edge routers in order to guarantee the QoS and dynamically adapt optical bandwidth to the traffic requirements (dynamic bandwidth allocation use case). Typically, those input interfaces will be implemented as 10, 40 or 100 Gb/s Ethernet ports. It was previously shown (D3.1) that higher-level statistics such as SNMP counters (MRTG) are not enough to predict bandwidth demands or infer QoS in the links. Therefore we opted to work at the network flow level, which provides enough detail but yet reduces one order of magnitude the amount of information with respect to raw traffic.

The monitoring architecture is shown in Figure 35. Monitoring probes will be connected to the span ports of the edge routers in order to examine incoming traffic. The monitoring probes will be in charge of generating network flows, that will be exported to the OAM. Inside the OAM, the DBA (dynamic bandwidth allocation) Monitor will execute bandwidth estimation algorithms and will send updates to the ABNO Controller.

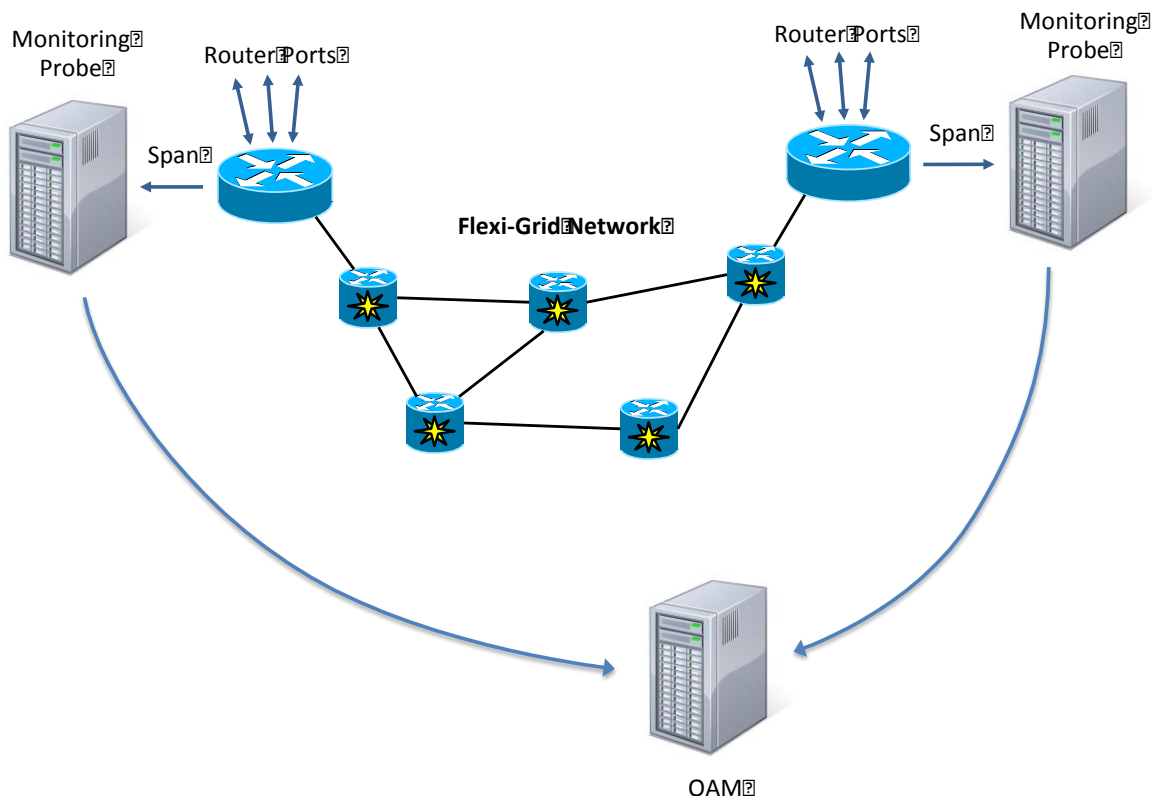


Figure 39. Monitoring architecture

Even considering that network flows approximately reduce by an order the magnitude the amount of data with respect to the original traffic, the information that has to be exchanged between the monitoring probes and the OAM can be too high. Therefore, we envision an

alternative architecture where the DMA Monitor block of the OAM is distributed in several instances close to the monitoring probes, and the information that is sent to the OAM is just the output of the bandwidth estimation algorithm.

6.2 Bandwidth Estimation Algorithms

Current Internet traffic demand is highly burst and a-priori capacity planning is extremely difficult. To complicate matters, new services are appearing on a week-by-week basis, which have heterogeneous demands in terms of bandwidth and latency requirements.

To circumvent this issue, dynamic bandwidth allocation algorithms provide capacity on-demand, based on the predicted changes in the traffic demand curve. In order not to continuously change the allocated capacity, only structural changes should be taken care of, and not sudden traffic peaks. The following figure shows the offered traffic at the edge of a large datacenter, which shows instantaneous traffic peaks in small timescales. Such peaks should be dealt with by means of buffering, and not changes in bandwidth allocation.

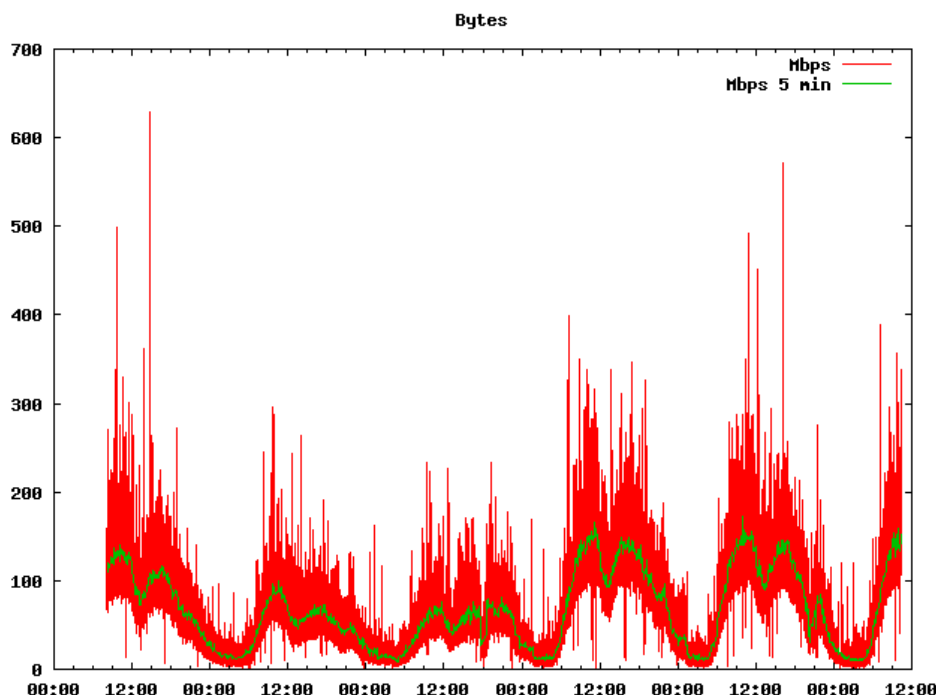


Figure 40. Offered traffic at the datacenter edge

Precisely, the detection of structural changes in the traffic requires new variables in the equation beyond the mere bits per second time series. The following figure shows that while the bits per second time series is actually the flows per second time series remains more constant. Consequently, it is more convenient for the detection of a structural change in the offered load. Furthermore, the number of flows per second portray the end-user activity better than the bits per second, whose peaks may result from a sudden file transfer, for example.

6.2.1 Correlation between flows and bits per second

A natural question that arises at this point, given the apparent invariance of the flows per second time series, is whether the flow per second time series and bits per second time

series are related or not. Fortunately, we have found a relationship between both through an indirect statistics: the flow size variance.

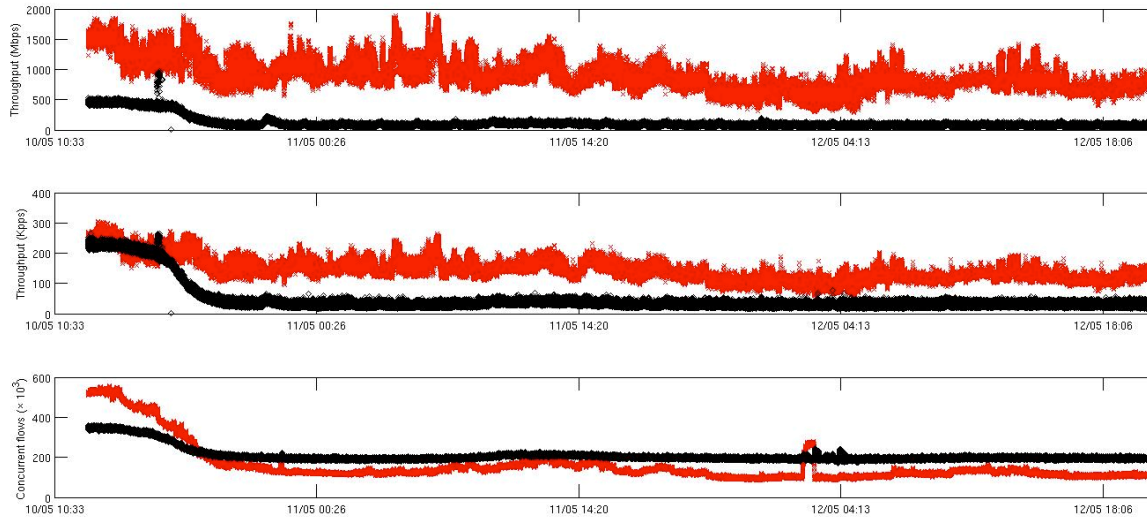


Figure 41. Time series of throughput and flows per second

In order to prove such relationship extensive trace-drive experiments have been performed using the following traces:

Trace Set	Number of traces	Avg. Rate (Mb/s)	Max Rate (Mb/s)	Additional Info
Bank1	30	6.4	82.3	LDAP,FTP,XCOM and bank transactions
Bank2	30	3.7	78.6	LDAP,FTP,XCOM and bank transactions
Bank3	30	3.5	68.3	LDAP,FTP,XCOM and bank transactions
Bank4	30	0.1	5.9	LDAP,FTP,XCOM and bank transactions
VoIP1	5	1.5	75	SCCP,RTP,FTP and HTTP traffic
VoIP2	5	0.4	16	SCCP,RTP,FTP and HTTP traffic
VoD	2	33.11	145	RTP and HTTP
Operator	4	3	17.3	-

The following curve shows the cumulative distribution function of cross-correlation (lag 0) between bits per second and flow variance in time intervals of five minutes.

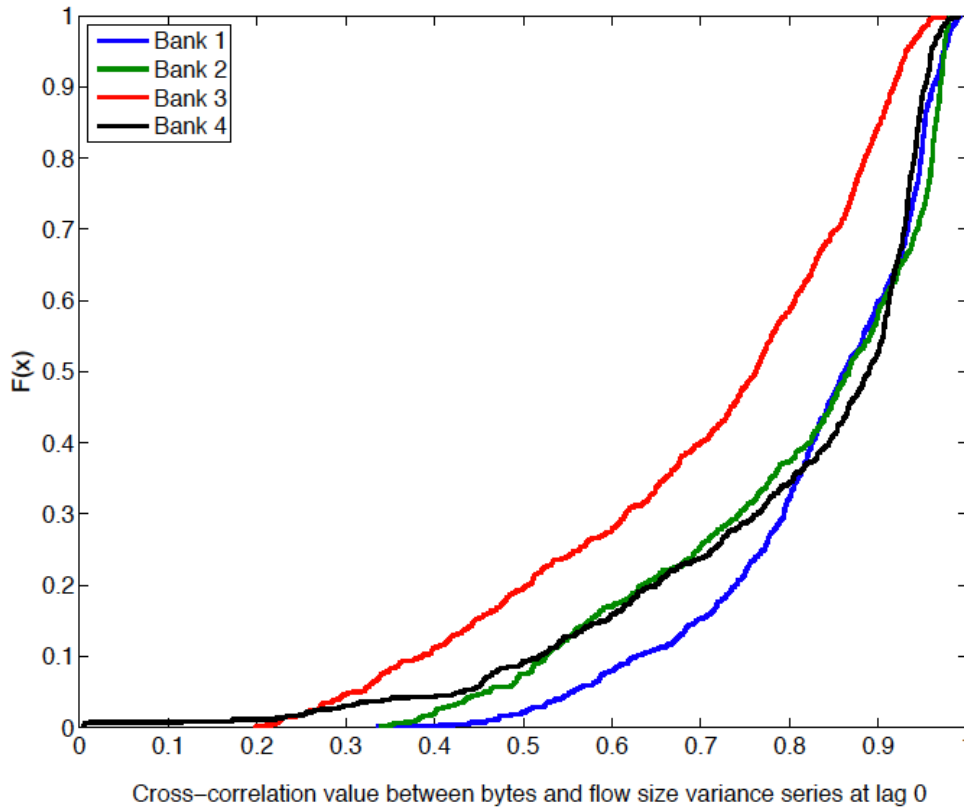


Figure 42. ECDF for cross correlation between bps and flow size variance (I)

The results are rather striking, with cross-correlation larger than 0.7 for more than 30% of the time intervals. The results are even better for multimedia traces, as shown in the following figure.

6.3 Example of Use and Implementation Feedback

Regarding the use cases identified in deliverable D1.5, dynamic bandwidth allocation could be useful at least for the elastic connectivity for cost optimization in federated dataservers and BRAS centralization in metro networks. In general, those cases where link usage is not constant could benefit from dynamic bandwidth allocation techniques. The goal is not only that underutilized links could release optical spectrum to be used by other links, but also reduce power consumption. Actually, this second point could be more relevant, since power is strongly dependent on the binary rate of the link. For example, in table 4 of D1.5 it is shown that the power savings due to reducing the binary rate from 100 Gb/s to 25 Gb/s is 46%. This power saving is obtained without changing the modulation format or reducing the reach. Therefore, further power savings could be achieved if modulation is changed to energy-efficient formats when links are underutilized.

For example, in a connection between federated dataservers, monitoring probes will be connected to the span ports of ingress routers. Therefore, a passive monitoring scheme will be deployed, that will capture all traffic in order to create network flows. These network flows will feed the bandwidth estimator, which will forecast link usage and make recommendations for dynamic bandwidth variations.

The feedback from the implementation WP4 is that flow creation at 40/100 Gb/s is a challenging task. At these speeds not only the number of packets per second constitutes a problem, but also the number of concurrent flows that could happen in the link. Actually, the number of concurrent flows in a link is very variable, and will depend on the type of network applications that are being used. In extreme cases, for example during security attacks, the number of concurrent flows can easily reach hundreds of millions. That is, it is necessary to have a memory of tens of gigabytes to store the flows. Such memory size is incompatible with a high packet rate. Actually, we have measured that for a conventional DDR3 memory, the worst-case read/update rate is 10 Mtransaction/s.

Therefore, we are working towards schemes where flows are generated in FPGA (see D4.2) in order to maximize performance, but also alternatives schemes where the FPGA can divide the traffic into several network flow engines, or perform a selective sampling that maintains flow coherence, as opposed to packet-level sampling implemented by routing.



7 Conclusions

This document described the different building blocks of IDEALIST control architecture including both distributed control plane enabling optimized survivability in elastic optical networks and centralized SDN orchestrator enabling multilayer operation and dynamic network-application. This architecture will be implemented in WP4.

I. Appendix YANG.1: optical-ted.yang file

```
module optical-TED {  
  
    namespace "http://www.tid.es/IDEALIST";  
  
    prefix o-ted;  
  
    import ietf-inet-types {  
        prefix inet;  
    }  
  
    import network-topology {  
        prefix nt;  
    }  
  
    //import optical-LSP {  
    //    prefix o-lsp;  
    //}  
  
    revision 2015-05-04;  
  
    typedef optical-node-type {  
        description "Determines the node type: optical-node, optical-transponder or  
optical-sliceable-transponder";  
        type enumeration {  
            enum optical-node;  
            enum optical-transponder;  
            enum optical-sliceable-transponder;  
        }  
    }  
  
    typedef modulation {  
  
        description "Enumeration that defines the type of wave modulation";  
  
        type enumeration {  
            enum BPSK;  
            enum DC_DP_BPSK;  
            enum QPSK;  
            enum DP_QPSK;  
            enum QAM16;  
            enum DP_QAM16;  
            enum DC_DP_QAM16;  
        }  
    }  
  
    typedef optical-technology {  
  
        description "Enumeration that defines the type of optical technology";  

```

```
        type enumeration {
            enum Flexigrid;
            enum WDM50;
            enum WDM100;
        }
    }

typedef FEC {

    description "Enumeration that defines the type of FEC";

    type enumeration {
        enum reed-solomon;
        enum hamming-code;
        enum golay;
        enum BCH;
    }

// }

// typedef fiber-type {
//     description "Fiber type";
//     type enumeration {
//         enum G652;
//         enum G653;
//         enum G655;
//     }
// }

// typedef node-type {
//     type enumeration {
//         enum ROADM;
//         enum optical-cross-connect;
//     }
// }

typedef interface-type {

    description "Enumeration that defines if an interface is numbered or
unnumbered";

    type enumeration {
        enum numbered-interfaces;
        enum unnumbered-interfaces;
    }

}

typedef optical-transponder-ref {
```

```

    type leafref {
      path "/nt:network-topology/nt:topology/nt:node/nt:node-id";
    }
    description
      "This type is used by data models that need to reference
      an optical transponder.";
  }

  typedef optical-node-ref {
    type leafref {
      path "/nt:network-topology/nt:topology/nt:node/nt:node-id";
    }
    description
      "This type is used by data models that need to reference
      an optical node.";
  }

  typedef optical-link-ref {
    type leafref {
      path "/nt:network-topology/nt:topology/nt:link/nt:link-id";
    }
    description
      "This type is used by data models that need to reference
      an optical link.";
  }

  typedef optical-node-port-ref {
    type leafref {
      path "/nt:network-topology/nt:topology/nt:node/o-ted:interface/o-
ted:port-number";
    }
    description
      "This type is used by data models that need to reference
      an optical link.";
  }

  grouping optical-ted-topology-type {
    container optical-ted-topology {
      presence "indicates optical TED Topology";
    }
  }

  grouping optical-ted-topology-attributes {
    container optical-ted-topology-attributes {
      leaf name {
        description "Name of the topology";
        type string;
      }
      // leaf-list flag {
      //   description "Topology flags";

```

```
//          type flag-type;
//      }
    }

    grouping optical-node-type {
        description "Used to determine the type of the optical node.";

        leaf type {
            type optical-node-type;
        }
    }

    grouping optical-node-attributes {
        description "Set of attributes of an optical node.";

        // leaf node-id {
        //     type uint32;
        //     description "Unique identifier for the node";
        // }

        // leaf ipv4-management-address {
        //     type inet:ipv4-address;
        //     description "IPv4 Address for the management of the optical node.";
        // }
        // leaf ipv6-management-address {
        //     type inet:ipv6-address;
        //     description "IPv6 address for the management of the optical node.";
        // }

        list interface {
            key "name";

            unique "port-number";

            description "List of interfaces contained by the node";

            leaf name {
                type string;
            }

            leaf port-number {
                type uint32;
                description "Number of the port used by the interface";
            }

            leaf input-port {
                type boolean;
            }
        }
    }
}
```

```

        description "Determines if the port is an input port";
    }

    leaf output-port {
        type boolean;
        description "Determines if the port is an output port";
    }

    leaf description {
        type string;
        description "Description of the interface";
    }

    leaf interfaces-type {
        type interface-type;
        description "Determines the type of the interface";
    }

    container numbered-interface {

        when "interfaces-type == numbered-interfaces";

        description "Grouping that defines an numbered interface with an
ip-address";

        leaf n-i-ip-address{
            type inet:ip-address;
        }
    }

    container unnumbered-interface {

        when "interfaces-type == unnumbered-interfaces";

        description "Grouping that defines an unnumbered interface with
an ip-address and a label";

        leaf u-i-ip-address{
            type inet:ip-address;
        }
        leaf label {
            type uint32;
        }
    }
}

    container connectivity-matrix {

        list connections {

            key "input-port-id";

```

```

        leaf input-port-id {
            type optical-node-port-ref;
        }

        leaf output-port-id {
            type optical-node-port-ref;
        }
    }
}

```

```

grouping optical-transponder-attributes {
    description "Set of attributes of an optical transponder.";

    leaf-list available-modulation {
        type modulation;
        description "List determining all the available modulations";
    }

    leaf modulation-type {
        type modulation;
        description "Modulation type of the wave";
    }

    leaf-list available-FEC {
        type FEC;
        description "List determining all the available FEC";
    }

    leaf FEC-enabled {
        type boolean;
        description "Determines whether the FEC is enabled or not";
    }

    leaf FEC-type {
        type FEC;
        description "FEC type of the transponder";
    }

    uses optical-node-attributes;
}

```

```

grouping optical-sliceable-transponder-attributes {
    description
        "Grouping that defines a sliceable transponder which is composed by
several transponders.";
    list transponder-list {
        key "carrier-id";

        leaf carrier-id {

```

```

        type uint32;
    }

    uses optical-transponder-attributes;
}

}

grouping optical-link-attributes {

    description "Set of attributes of an optical link";

    leaf-list available-label-flexigrid {
        type bits {
            bit is-available;
        }
        description "Array of bits that determines whether a spectral slot is
available or not.";

        when "technology-type == Flexigrid";
    }

    leaf-list available-label-WSO {
        type bits {
            bit is-available;
        }
        description "Array of bits that determines whether a wavelength is
available or not.";

        when "technology-type != Flexigrid";
    }

    leaf N-max {
        type int32;
        description "Maximum number of channels available.";
    }

    leaf base-frequency {
        type decimal64 {
            fraction-digits 5;
        }
        units THz;
        default 193.1;
        description "Default central frequency";
        reference "draft-ietf-ccamp-flexi-grid-fwk-01";
    }

    leaf nominal-central-frequency-granularity {
        type decimal64 {
            fraction-digits 5;
        }
        units GHz;
    }
}

```

```

        default 6.25;
        description "It is the spacing between allowed nominal central
frequencies and it is set to 6.25 GHz";
        reference "draft-ietf-ccamp-flexi-grid-fwk-01";
    }

    leaf slot-width-granularity {
        type decimal64 {
            fraction-digits 5;
        }
        units GHz;
        description "Minimum space between slot widths";
        reference "draft-ietf-ccamp-flexi-grid-fwk-01";
    }

    leaf technology-type {
        type optical-technology;
        description "Determines which technology is used at optical-level";
    }

    // leaf max-link-latency {
    //     type int32;
    //     description "Maximum latency allowed by the link.";
    // }

    // leaf refraction-index {
    //     type decimal64 {
    //         fraction-digits 5;
    //     }
    //     description "Index of refraction.";
    // }

    // leaf link-length {
    //     type int32;
    //     description "Length of the link in km.";
    // }

    // leaf spans-count {
    //     type int32;
    //     description "Number of spans in the link.";
    // }

    // leaf attenuation {
    //     type decimal64 {
    //         fraction-digits 5;
    //     }
    //     description "Link attenuation in dB.";
    // }

    // leaf link-extra-losses {
    //     type decimal64 {
    //         fraction-digits 5;
    //     }

```

```

//    description "Link extra losses due to splices, ODF connections,
amplifier connectors.";
// }

// leaf-list amplifiers-locations {
//   type decimal64 {
//     fraction-digits 5;
//   }
//   description "Array of the distances where the amplifiers are located in
km.";
// }

// leaf pmd-coefficient {
//   type decimal64 {
//     fraction-digits 5;
//   }
//   description "PMD coefficient for the link.";
// }

// leaf chromatic-dispersion {
//   type decimal64 {
//     fraction-digits 5;
//   }
//   description "Chromatic dispersion for the link (ps/nm*Km).";
// }

// leaf chromatic-dispersion-slope {
//   type decimal64 {
//     fraction-digits 5;
//   }
//   description "Chromatic dispersion slope for the link (ps/nm2*Km).";
// }
}

augment "/nt:network-topology/nt:topology/nt:topology-types" {
  uses optical-ted-topology-type;
}

augment "/nt:network-topology/nt:topology" {
  when "nt:topology-types/optical-ted-topology";
  uses optical-ted-topology-attributes;
}

augment "/nt:network-topology/nt:topology/nt:node" {
  when "../nt:topology-types/o-ted:optical-ted-topology";
  uses optical-node-type;
}

augment "/nt:network-topology/nt:topology/nt:node" {
  when "../nt:topology-types/o-ted:optical-ted-topology";

```

```

    uses optical-node-attributes;
}

augment "/nt:network-topology/nt:topology/nt:node" {
    when "o-ted:optical-node-type/o-ted:optical-transponder";
    uses optical-transponder-attributes;
}

augment "/nt:network-topology/nt:topology/nt:node" {
    when "o-ted:optical-node-type/o-ted:optical-sliceable-transponder";
    uses optical-sliceable-transponder-attributes;
}

augment "/nt:network-topology/nt:topology/nt:link" {
    when "../nt:topology-types/o-ted:optical-ted-topology";
    uses optical-link-attributes;
}
}

```

II. Appendix YANG.2: media-channel.yang file

```

module media-channel {

    namespace "http://www.tid.es/IDEALIST-lsp";

    prefix m-c;

    import optical-TED {
        prefix o-ted;
    }

    revision 2014-06-05;

    container media-channel {

        description "Media association that represents both the topology
            (i.e., path through the media) and the resource (frequency slot)
            that it occupies. As a topological construct, it represents a (effective)
            frequency slot supported by a concatenation of media elements
            (fibers, amplifiers, filters, switching matrices...). This term is
            used to identify the end-to-end physical layer entity with its
            corresponding (one or more) frequency slots local at each link
            filters.";

        reference "draft-ietf-ccamp-flexi-grid-fwk-01";

        container source {

            leaf source-node {

```

```

        type o-ted:optical-node-ref;
    }

    leaf source-port {
        type o-ted:optical-node-port-ref;
    }
}

container destination {
    leaf destination-node {
        type o-ted:optical-node-ref;
    }

    leaf destination-port {
        type o-ted:optical-node-port-ref;
    }
}

uses media-channel-attributes;
}

container network-media-channel {

    description "It is a media channel that transports an Optical Tributary Signal";

    reference "draft-ietf-ccamp-flexi-grid-fwk-01";

    container source {
        leaf source-node {
            type o-ted:optical-node-ref;
        }

        leaf source-transponder {
            type o-ted:optical-transponder-ref;
        }
    }

    container destination {
        leaf destination-node {
            type o-ted:optical-node-ref;
        }

        leaf destination-transponder {
            type o-ted:optical-transponder-ref;
        }
    }

    uses media-channel-attributes;
}

grouping media-channel-attributes {

```

```

container effective-freq-slot {

    description "The effective frequency
    slot is an attribute of a media channel and, being a frequency slot,
    it is described by its nominal central frequency and slot width";

    reference "draft-ietf-ccamp-flexi-grid-fwk-01";

    leaf N {
        type int32;
        description
            "Is used to determine the Nominal Central Frequency. The
set of nominal
expression f =
"anchor
or
            central frequencies can be built using the following
            193.1 THz + n x 0.00625 THz, where 193.1 THz is ITU-T
            frequency" for transmission over the C band, n is a positive
            negative integer including 0.";
        reference "draft-ietf-ccamp-flexi-grid-fwk-01";
    }

    leaf M {
        type int32;
        description
            "Is used to determine the slot width. A slot width is
constrained
integer greater
            to be M x SWG (that is, M x 12.5 GHz), where M is an
            than or equal to 1.";
        reference "draft-ietf-ccamp-flexi-grid-fwk-01";
    }
}

list link-channel {

    key "link-id";

    leaf link-id {
        type int32;
    }

    uses link-channel;
}

grouping link-channel {

    description "A link channel is one of the concatenated elements of the media
channel.";

    leaf N {

```

```

    type int32;
    description
        "Is used to determine the Nominal Central Frequency. The set of
nominal          central frequencies can be built using the following expression f
=
        193.1 THz + n x 0.00625 THz, where 193.1 THz is ITU-T "anchor
        frequency" for transmission over the C band, n is a positive or
        negative integer including 0.";
        reference "draft-ietf-ccamp-flexi-grid-fwk-01";
    }

    leaf M {
        type int32;
        description
            "Is used to determine the slot width. A slot width is constrained
greater          to be M x SWG (that is, M x 12.5 GHz), where M is an integer
            than or equal to 1.";
            reference "draft-ietf-ccamp-flexi-grid-fwk-01";
        }

    leaf source-node {
        type o-ted:optical-node-ref;
    }

    leaf source-port {
        type o-ted:optical-node-port-ref;
    }

    leaf destination-node {
        type o-ted:optical-node-ref;
    }

    leaf destination-port {
        type o-ted:optical-node-port-ref;
    }

    leaf link {
        type o-ted:optical-link-ref;
    }

    leaf bidireccional {
        type boolean;
        description "Determines whether the link is bidireccional or not";
    }
    }
}

```