31-03-2015

Open Call Deliverable OCS-DN1.1
Final Report (REACTION)

Open Call Deliverable OCS-DN1.1

Grant Agreement No.:	605243
Activity:	NA1
Task Item:	10
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	CNIT
Partners:	CNIT, UPC, Telefonica
Document Code:	GN3PLUS14-1300-42
Authors:	Filippo Cugini, Matteo Dallaglio, Luis Velasco, Victor Lopez, Juan Pedro Fernandez-Palacios

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7 2007–2013) under Grant Agreement No. 605243 (GN3plus).

Abstract
The REACTION project has designed a flexible optical network scenario enabling software-controlled super-channel transmission. Innovations have been introduced in the context of data plane architectures (e.g., sliceable bandwidth variable transponder), control plane (e.g., back-end/front-end PCE architecture) and advanced routing and spectrum assignment strategies.
The REACTION solutions have been applied to selected use cases, showing the benefits of the proposed technologies.
The OPNET Modeler developed within the REACTION project has been also released as project outcome.
The REACTION results have been presented in eight top-ranked peer-reviewed international journals and conferences.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

The recent evolution of optical networks, driven by the introduction of the flexible grid technology and the availability of high rate coherent transmission systems, has determined the need to design and investigate novel networking solutions.

The REACTION project has designed a flexible optical network scenario enabling software-controlled super-channel transmission. Innovations have been introduced in the context of data plane, control plane and routing and spectrum assignment strategies.

To evaluate the proposed REACTION solutions, a specifically designed OPNET modeler has been first implemented. The OPNET modeler accurately reproduces the flexible optical network behaviour, including node and transponder architectures from a data plane perspective and the support of the GMPLS routing and signalling protocol, extended for flexile grid. Moreover, it supports the Path Computation Element (PCE) also including the PCE communication Protocol (PCEP).

The OPNET Modeler developed within the REACTION project has been released as project outcome, to be used within GEANT community.

In the REACTION project, from a data plane architectural perspective, a bandwidth variable transponder (BVT) supporting 1 Tb/s multi-carrier transmission has been enhanced to support, besides dynamic adaptation of transmission parameters, the sliceable functionality. Specifically, such Sliceable BVT (SBVT) transponders are capable of creating multiple optical flow units (i.e., sub-carriers) that can be aggregated or independently routed according to the traffic requirements. In this report, the use of slice-ability during provisioning and restoration in flexible grid optical networks is addressed. Specifically, a scheme is proposed to exploit the possibility of establishing/recovering an optical connection as a single super-channel or as a number of independent sub-carriers. Both centralized and distributed implementations of the proposed schemes are evaluated through simulations in a GMPLS-based scenario. Results show that, despite the introduced spectrum overbuild, the utilization of slice-ability permits to increase the amount of established/recovered traffic.

From a control plane perspective, the REACTION project has developed a solution relying on a GMPLS-based distributed control plane with a Path Computation Element (PCE) architecture. Specifically, a novel PCE architecture has been investigated. The architecture relies on an active stateful front-end PCE, in charge of routing and spectrum assignment (RSA) computations and a back-end PCE in charge of performing complex network re-optimization solutions. The PCE architecture also relies on the North-Bound Distribution of Link-State and TE Information through BGP (i.e., BGP-LS), utilized to provide the PCE architecture (also in the context of a hierarchical implementation for multi-domain scenarios) with adequate networking information.

In the REACTION project, novel routing and spectrum allocation (RSA) algorithms have been designed and evaluated in the context of flexible optical networks and, specifically, to be encompassed within the proposed PCE architecture.

In this report, the application of the proposed REACTION solutions in the context of three selected use cases is reported.

In the first use, SBVTs capabilities are evaluated in the context of restoration. In particular, multipath recovery and bitrate squeezing are applied to maximize the amount of restored bitrate, also exploiting limited portions of spectrum resources along multiple routes. An ILP model and heuristic strategy are proposed. A software defined network (SDN) architecture is then introduced to adequately support the SBVT configuration. The SDN architecture is applied to experimentally assess that the overall re-configuration time upon failure detection is included within two seconds, largely dominated by the proprietary control of optical nodes. Finally, extensive simulation results show the relevant restoration capabilities achieved by the proposed multipath recovery and bitrate squeezing scheme.

In the second use case, in-operation network planning operations are considered. To increase traffic restorability in flexgrid networks, multipath after failure repair optimization (MP-AFRO) problem is applied to reduce sub-connections count by aggregating those belonging to the same original connection and re-routing the resulting connection to release spectral resources. The MP-AFRO problem is modelled using either a Mixed Integer Linear Program (MILP) formulation and a heuristic algorithm. The performance is firstly validated by simulation. Next, the heuristic algorithm is deployed inside an in-operation planning tool in the form of back-end PCE (bPCE) inside the Application-based Network Operations (ABNO) architecture controlling a network. The bPCE is connected to the centralized active stateful PCE. MP-AFRO is experimentally demonstrated using a distributed field trial test-bed connecting the premises of Telefonica (Madrid), CNIT (Pisa), and UPC (Barcelona).

The third use case evaluates the benefits of flexi-grid in the context of the UNINETT NREN network. The expected evolution of the NREN traffic matrix is assumed to evaluate the benefits provided by the adoption of high rate transmission systems with and without the flexi-grid technology. Results show that fiber exhaustion will occur after around 7 years from now, further postponed in the case of flexi-grid networks.

The REACTION results have been published on the most relevant international peer-reviewed conferences and journals. In particular, five conference papers have been presented, including a prestigious post-deadline paper and a top-scored paper at the OFC conference, the most relevant international conference on optical communication. Moreover, three journal papers have been already published, including the top-ranked OSA/IEEE Journal of Lightweight Technology (JLT) and OSA/IEEE Journal of Optical Communication and Networking (JOCN)

.

1 Introduction

Recent advances in optical modulation formats, filtering, and digital signal processing, among others, are enabling Flexgrid components to be produced; finer granularity and flexibility in the use of the optical spectrum compared to that of the fixed grid can be achieved [Jin-CM09].

For flexgrid-based optical core networks to be deployed, bandwidth-variable optical cross-connects (BV-OXC), equipped with bandwidth-variable wavelength selective switches (WSSes) and bandwidth variable transponders (BVTs), have to be designed and validated.

Moreover, a novel functionality, called *slice-ability*, may be introduced within BVTs to further improve the overall network flexibility [Ger-CM12], [Jin-CM12]. In particular, SBVTs allow several optical connections (lightpaths or sub-carriers) to be terminated in different destination nodes.

A sliceable BVT has the potential to combine the cost-reduction provided by the integration and sharing of multiple components in a single transponder with the flexibility guaranteed by the adaptation of each sub-carrier transmission parameter. This way, SBVTs can avoid the need to purchase many different transponders with fixed transmission parameters and rigid number of sub-carriers.

Moreover, and differently with respect to previous solutions, sliceable functionality will allow a single SBVT to be connected to different destinations, each served by a sub-set of sub-carriers. This will enable the effective pay-as-you-growth strategy. Furthermore, it has the potential to significantly improve the network re-configurability and survivability.

Within the REACTION project, the sliceable functionality is specifically investigated, providing both architectural and networking solutions. In particular, SBVT architectures are designed and evaluated in the context of high rate communication (i.e., exploiting super-channel transmissions).

Two different control plane architectures have been considered within REACTION to operate an EON.

First, a Software Defined Networking (SDN) architecture, where the SDN controller uses the OpenFlow protocol (with extensions) to configure flexible transmitters, receivers and intermediate BV-OXCs of optical connections (see Figure 1).

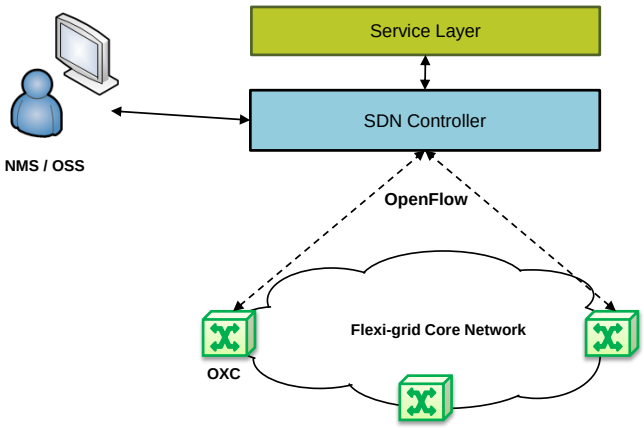

Figure 1: Software Defined Networking architecture.

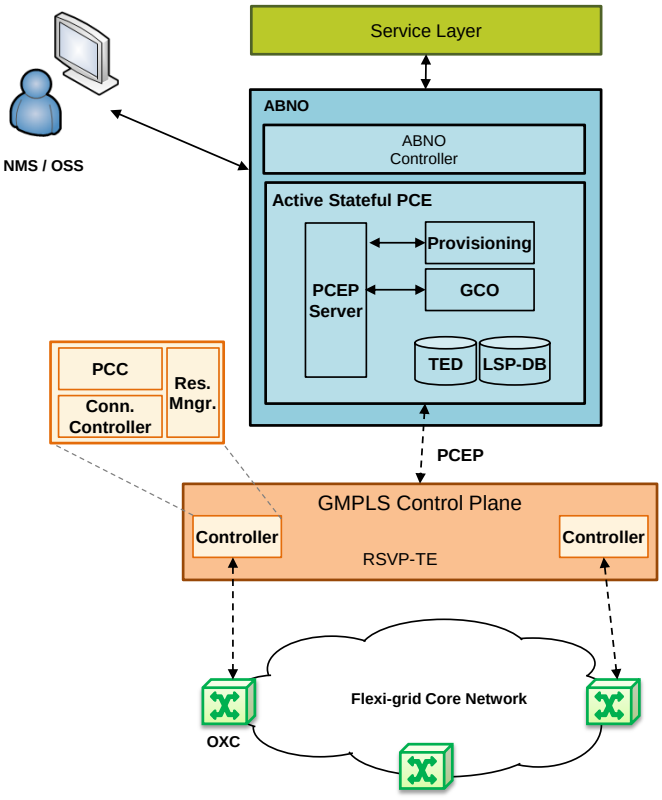

Figure 2: ABNO-based architecture.

Second, a Path Computation Element (PCE) [RFC4655] -based Architecture for Application-based Network Operations (ABNO) is considered, where the PCE is active and stateful with initiation capabilities (see Figure 2). The ABNO architecture consists of a number of standard components and interfaces which, when combined together, provide a method for controlling and operating the network. The ABNO controller is the entrance point to the network for NMS/OSS and the service layer for provisioning and advanced network coordination, and it is responsible for the establishment of LSPs.

The control plane relies on the GMPLS protocol suite extended for flexible networks and on an active stateful PCE system controlling the whole network and a set of PCCs in charge of local resources. Nodes use the RSVP-TE protocol for LSP signalling and the PCEP to communicate with the PCE.

The standardized PCE computes routes in response to path computation requests. It takes advantage of a traffic engineering database (TED) that is updated after network resources are effectively used or released. The PCE architecture is composed by an active stateful PCE system controlling the whole network and a set of PCCs in charge of local resources. Nodes use the RSVP-TE protocol for LSP signalling and the PCEP to communicate with the PCE.

To efficiently support the flexible optical network including SBVT implementation and functionalities, an enhanced control plane architecture has been designed within the REACTION project.

- Specifically designed and implemented control plane functionalities and routing and spectrum assignment (RSA) algorithms are designed and experimentally demonstrated. In particular, to operate flexible optical networks equipped with SBVTs.

  For provisioning, the PCE needs to solve the RSA problem, the counterpart of the routing and wavelength assignment (RWA) in traditional wavelength switched optical networks (WSONs). The allocated spectral resources must be, in absence of spectrum converters, the same along all the links belonging to the computed route (the continuity constraint). Moreover, the RSA problem adds new constraints to guarantee that allocated resources are also contiguous in the spectrum (the contiguity constraint).

  In REACTION, efficient methods to allow solving realistic problem instances in practical times are proposed and validated. The algorithm considers modulation formats, distance/impairment constraints, and guard bands. Having a finer granularity in the optical network might result in a higher variability of the data-rate offered to the optical layer along the day. Therefore, in addition to a first spectrum allocation, there is the specific requirement of algorithms for elastic operations used to dynamically increase and decrease the spectrum width assigned to a connection. In this regard, specific work is performed within the REACTION project.

- From a control plane perspective, REACTION proposes a PCE architecture consisting of an active stateful front-end PCE, in charge of computing RSA and elastic spectrum allocations or provisioning, and a back-end PCE responsible for performing complex network re-optimization actions.

- The North-Bound Distribution of Link-State and TE Information through BGP, known as BGP-LS, is here adopted to provide the PCE with adequate networking information.

This document is structured in the following way.

In chapter 2, the OPNET model developed within the REACTION project is detailed. The following implemented modules are described: RSVP-TE, PCEP, OSPF-TE, TCP, IP, and SBVT (modelled as Transmitter/Receiver couple representing the physical interfaces).

In chapter 3, the SBVT technology is described and applied under different transponder architectures and networking conditions. Specific control plane and RSA strategies are presented and evaluated through

simulations (performed through the implemented OPNET model), considering both provisioning and recovery scenarios.

In Chapter 4, a specific use case of bit-rate squeezing and multipath restoration is then considered. The proposed takes advantage of SBVTs. After the problem statement, ILP formulation and heuristic algorithms are proposed. In addition, a specifically designed Software Defined Networking (SDN) architecture is introduced and experimental validated.

In Chapter 5, an enhanced control plane architecture based on a proactive Hierarchical PCE using BGP-LS is proposed for Elastic Optical Networks. The solution is supported by extensive simulative results.

In Chapter 6, the use case of in-operation planning is considered as a comprehensive example of the whole set of REACTION innovations. The use case focuses on restoration after failure repair optimization, with multipath merging (MP-AFRO). After the problem statement, a MILP formulation and heuristic algorithm are proposed, together with an optimization workflow. The use case is then evaluated through simulations and experimentally assessed.

In chapter 7, the REACTION solutions are applied to the UNINET use case. In particular, the case of transparent ROADM-based network is considered. Then, the flexi grid technology is introduced.

Finally, conclusions and discussions on future works are provided.

2 OPNET Model

In this section, the most relevant functionalities of the simulation model implemented within the OPNET framework are reported.

The OPNET Modeler is used for developing the protocol stack used by the REACTION control plane architecture. In particular, the OPNET framework is used for model design, simulation, statistics data collection and analysis.

The following sections describe the major blocks that have been developed. In particular, starting with a brief description of the network topology under consideration, the implementation of the main modules composing the GMPLS suite and the path computation element will be described.

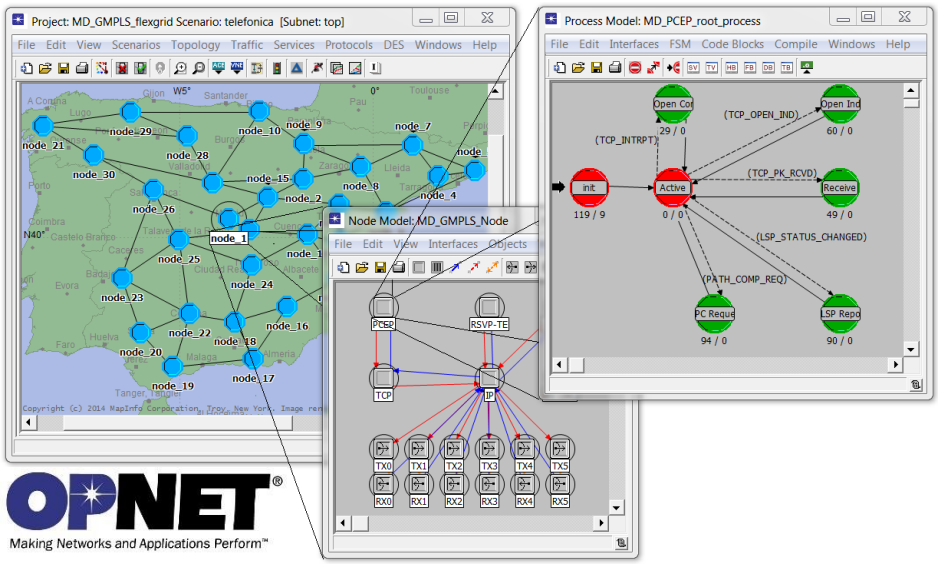

Figure 1 – OPNET Modeler

2.1 Control plane OPNET node model

Nodes on the control plane are connected through an out-of-band network composed of point to point full duplex links with a bit rate of 1Gbps (i.e. assuming a Gigabit Ethernet connection).

The model has been implemented to also account for the bit rate and the propagation delay of the control plane. This way, accurate time statistics can be collected during simulations. For example, simulations will enable the assessment of the backward blocking probability, which becomes a relevant aspect in case of high inter-arrival rate (i.e., when the inter-arrival time becomes lower than or comparable to the mean round-trip time, e.g. upon failure occurrence). In this study, the propagation delay of each control plane packet is based on the physical distance between the nodes (i.e. by the length of the links).

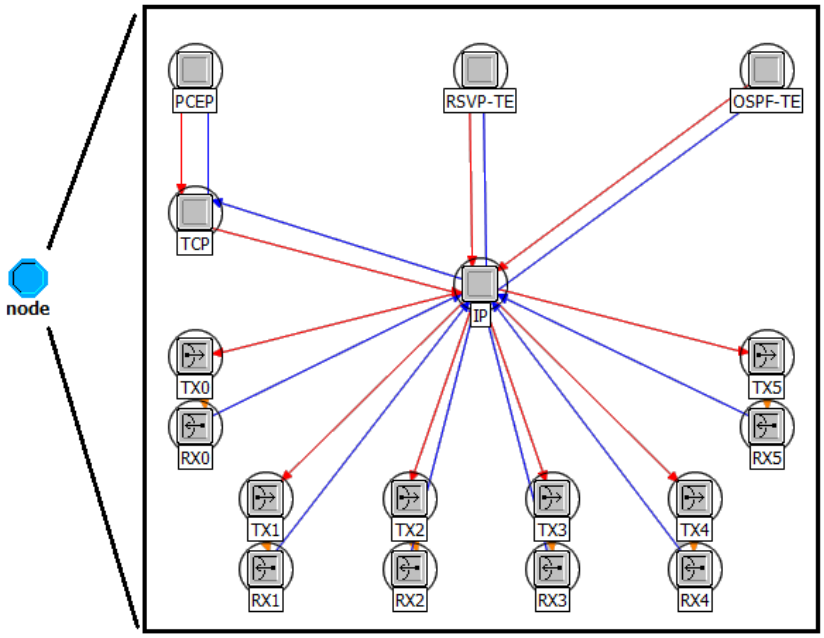

Figure 2 - Node Model

Nodes are multifunctional and able to play different roles in the network depending on the installed modules. The control plane node architecture is composed by the following modules:

- The RSVP-TE module.
- The PCEP module.
- The OSPF-TE module.
- The TCP module.
- The IP module.
- Multiple Sliceable Bandwidth Variable Transponders (SBVTs), modeled as Transmitter/Receiver couples representing the physical interfaces.

The implementation includes all the modules listed above, tested and validated with both distributed and centralized scenarios.

Node Input Attribute

Attribute name	Type	Description
Bandwidth In Frequency Slices	Integer	It represents the total amount of spectrum in terms of frequency slices (1 slice = 12.5 GHz). All the SBVTs installed on the node are assumed to work in the frequency range obtained by this parameter and considering 193.1THz as central frequency.
Transponders	Compound	An array of structure where each entry defines a group of transponders installed on the node. The structure is composed of: - Number of transponders - Type (MW-SBVT or ML-SBVT) - Number of carriers supported by the transponders - Minimum allowed spacing between the carriers - Maximum allowed spacing between the carriers
Traffic Requests	Compound	An array of structure where each entry describes a Poisson traffic arriving at the node. Each traffic request is described by: - Capacity in Gbps - Mean inter-arrival time - Mean service time

2.2 SBVT OPNET Model

Two different SBVT architectures have been considered and implemented in the simulator: Multi-laser SBVT (ML-SBVT) and Multi-wavelength SBVT (MW-SBVT).

As detailed in the next sections, the ML-SBVT uses an array of N tunable lasers, while the MW-SBVT uses a multi-wavelength source that consists in a single laser able to generate various carriers. The ML-SBVT does not introduce any constraint on the routing and spectrum assignment (RSA), guaranteeing total freedom in the tuning of the carriers. The MW-SBVT provides greater stability among the carriers thus better spectrum compactness. Moreover, adopting multi-wavelength transponders allows to reduce the cost of each node by decreasing the number of lasers and therefore the power consumption. As drawback, the MW-SBVT imposes new constraints on the tuning and spacing among the carriers that reduces the solution space of the RSA.

The implementation automatically discovers the number of SBVT configured during the initialization phase. For example, in Figure 2 - Node Model, six SBVT modules are present.

In order to solve the routing and spectrum assignment problem, the SBVT constraints have been mapped directly into the topology graph. The constraints become new virtual nodes and virtual links forming an extended version of the topology graph. The RSA algorithm is then applied to the extended graph.

More specifically, suppose we want to solve the path computation on the graph in Figure 3 for a source destination pair (S, D):

- The source node is split by creating a new virtual source node (SV).

- Each transponder (t) in the source node becomes a new node (TSt).
- Each TSt is connected with S through a new link (SLl).
- Each TSt is connected with SV through a new link (SVLl).
- The two links SLi and SVLi have the same spectral description, which is based on:
    - Type of transponder TSt
    - Transponder's state
    - Path computation parameters (number of carriers, carrier spacing, etc.)

Similarly for the destination node:

- The destination node is split by creating a new virtual destination node (DV).
- Each transponder (t) in the destination node becomes a new node (TDt).
- Each TDt is connected with S through a new link (DLl).
- Each TDt is connected with SV through a new link (DVLl).
- The new links DLi and DVLi have the same spectral description based on the parameter mentioned before.

The path computation is performed considering the new graph (Figure 4) and the new virtual source/destination couple (SV, DV)

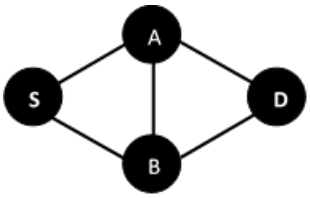

Figure 3 - Real topology graph

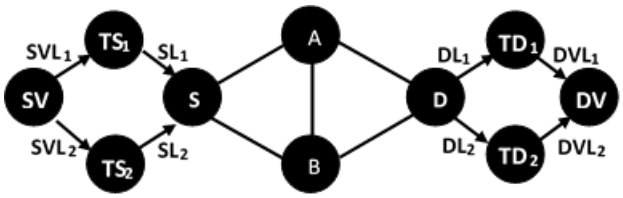

Figure 4 - Extended graph with virtual nodes and virtual links

2.3 OSPF-TE OPNET Module

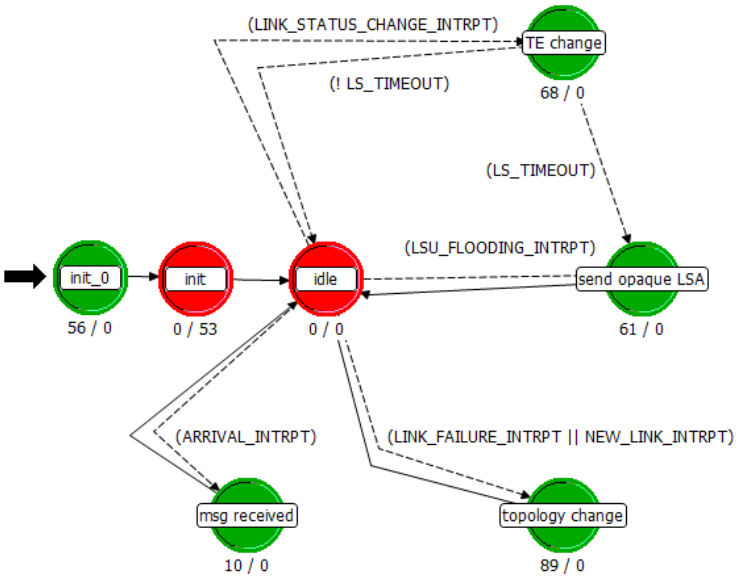

Figure 5 - OSPF-TE Process state machine

In this project, a simplified version of the OSPF-TE has been implemented to encompass all the required functionalities of interests for the REACTION architecture.

In particular, the LSA flooding is triggered by a specific event including link failure/restoration, a portion of spectrum reserved/released on a specific link or the initial neighbor discovery.

Neighbors discovery

The first task performed in the initialization state of the OSPF-TE process is the neighbor's discovery. In this phase, each node discovers all its neighbors by interrogating each output interface. If there is an active node at the other side of the link, a new adjacency is created. When the neighbor nodes is discovered, a new message containing a Router LSA with link state information is flooded.

Whenever the topology map changes the OSPF-TE computes the routing table and informs the IP module to update its own forwarding table based on the new information.

Spectrum availability update

Whenever a RESV message or PATH TEAR/RESV ERR messages cross the node, some spectrum on the transit link is either reserved or released in the Traffic Engineering Database (TED). In accordance with the time of generation of the last LSA and the value of the considered minLSInterval, the change on the link status induced by the reservation protocol triggers the flooding of the link state update messages performed by the OSPF-TE protocol to inform the rest of the network.

Figure 5 shows the state machine of the OSPF-TE protocol, in particular when the state of a link changes, two states are traversed: The "link update" state notifies that one or more links have undergone some changes,

typically an LSP passing from that link has been either established or released. The “send opaque LSA” state is where the link state update message containing the opaque LSA is effectively sent. The transition between the two states occurs, as already anticipated, only if the minLSInterval is already expired, on the contrary, if it is not expired, it means that the previous opaque LSA has been sent less than minLSIterval seconds ago, therefore the current opaque LSA is queued. When the timer expires, all the queued opaque LSAs are flushed and the timer is reset.

Link failure and restoration

When a link failure event occurs, the downlink node (i.e. the node that receives the light signal from that link) is triggered. The OSPF-TE, once identified the failed link, goes to the “link failure” state and immediately sends a router LSA containing the information of all the links excluded the one just failed. When the link is later restored a new router LSA is sent containing also the recovered link.

In this implementation, as for the standard OSPF-TE recommendations no timer or delay is present, and every change is notified as soon as possible. This enables a quick update of the topology interconnection (very important especially in case of failed links).

2.4 RSVP-TE OPNET Module

Most of the features are coded inside the GMPLS RSVP-TE module.

This module is divided into several parallel processes (child) coordinated by a single process (root) that acts as a dispatcher of the incoming requests.

Since each node can be traversed by many LSPs, the RSVP-TE module is designed to keep track and manage multiple LSPs at the same time (this is possible by exploiting child processes). Every time a new LSP request is generated, the root process of the RSVP-TE module creates a new dedicated child process to handle all the future messages related to the new LSP.

RSVP-TE Input Attributes

Attribute name	Type	Description
LSP Inter-arrival Time Distribution	String	The distribution describing the time between two consecutive LSP requests (by default a Poisson process is considered, therefore an exponential inter-arrival distribution is used).
LSP Mean Inter-arrival Time	Double	The mean value of the inter-arrival time distribution expressed in seconds.
LSP Inter-arrival Time Variance	Double	The variance of the inter-arrival distribution expressed in seconds.
LSP Holding Time Distribution	String	The distribution describing the lifetime of an established LSP (by default it is exponentially distributed).
LSP Mean Holding Time	Double	The mean value of the holding time distribution expressed in seconds.
LSP Holding Time Variance	Double	The variance of the holding time distribution expressed in seconds.
Network Load	Double	The load of the network expressed in Erlang, i.e. the service time divided by the inter-arrival time all multiplied by the number of nodes of the network.
Configuration File	String	It is the path of a formatted file containing the list of all the capacity demands that can be requested by an LSP.
Start LSP Generation Time	Double	It is possible to specify the simulation time from when the LSP generation

		process start working. By setting this parameter to “Infinity” the node do not generate any LSP request.
Stop LSP Generation Time	Double	It is possible to specify the simulation time when the LSP generation will be stopped. By setting this parameter to “Infinity” the node never stops to generate LSP requests.
Spectrum Assignment	Integer	Different algorithms can be used by the egress node to decide how to allocate the requested frequency slot given the label set received through the PATH message. Four algorithms have been implemented: Random, First Fit, Last Fit and Exact Fit.
Distributed Architecture	Toggle	Depending on this Boolean variable, the path computation will be afforded locally by the node or forwarded to the centralized PCE.

The following finite state machines (FSMs) describe the different processes of the RSVP-TE module:

- Root FSM
- Ingress root child FSM
- Ingress child FSM
- Intermediate child FSM
- Egress child FSM

2.4.1 Root process

The RSVP-TE module loads the root process when the simulation starts. The root process, based on the FSM in Figure 6 -RSVP-TE Root FSM, acts like a coordinator and a dispatcher of all the RSVP-TE messages generated by/traversing the node. In particular, for each RSVP-TE message, the root process recognizes the associated LSP and delivers the message to the dedicated child process.

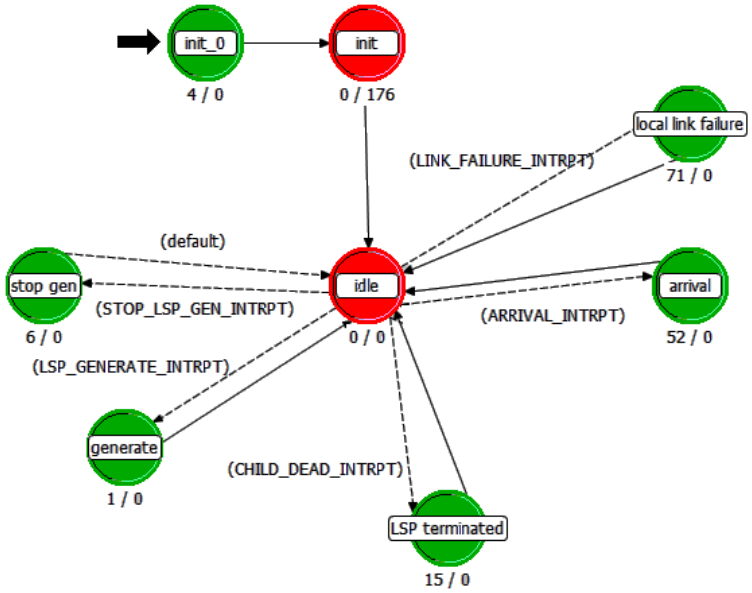

Figure 6 -RSVP-TE Root FSM

Initialization

Most of the data structures used in the RSVP-TE module are instantiated inside the initialization state. A description of the most relevant structures follows:

1. Child hash table: used to keep track of all the child processes associated to the LSPs (established or under provisioning) that traverse the node.

2. Shared data: It is a module wide structured memory shared among all the processes (root and child processes). It contains some common variables that have to be accessible at the root as well as at all the child processes. The most important variable in the shared structure is the hash table containing the list of all the links attached to the node. For each link the following information is stored:

- The minimum and the maximum frequency indexes delimiting the working spectrum range for that link.
- The total number of frequency slices (12.5GHz) not yet allocated.
- A vector of currently reserved frequency slots described by an upper and lower frequency indexes. This array is updated during the reservation and the tear down phases, and it is consulted during the provisioning phase by the correspondent child process.

LSP requests generation

The random arrival of a new LSP requests from the upper layer is emulated using a self-interrupt scheduled by the root process. The self-interrupt is scheduled according to the probability distribution given as input parameter. The interrupt triggers the root process into the generation state where a new LSP request is created. In particular, a destination node is randomly selected with a uniform distribution among all the existing nodes, and then the capacity required by the LSP is also randomly extracted from a set of admissible capacities given

as input. Once the destination node and the capacity are decided, a progressive tunnel ID is assigned to the new LSP and finally a new ingress child is created to handle it.

Arrival

When an RSVP-TE packet is received by the node, the root process state machine moves into the arrival state. In this state the session object is first read, based on it, the proper child is invoked passing the received packet. Finally the root process checks whether the child process is still alive after the invocation, if not, the latter is removed from the child hash table.

Local link failure

When a link failure event occurs, the RSVP-TE module in the downlink node receives an interrupt. Upon the interrupt arrival, the RSVP-TE has to notify each ingress nodes whose LSP connection has been disrupted due to the broken link. The root process iterates the list of all child processes informing those involved about the failure, therefore each child sends a NOTIFY message towards the respective ingress node. The notify message, unlike the PATH message, may also follow a different path than the one followed by the LSP, indeed in the IP header it present the ingress node address and a proper TTL value.

2.4.2 Child process

A node could be traversed by many ongoing LSPs at a time, therefore the root process must be capable of managing multiple child processes.

The root process exploits the RSVP-TE Session Object of the LSPs as unique key to store/retrieve the child processes to/from an hash table.

The session object is formed by the combination of two different addresses, one from the source (ingress) node, the second from the destination (egress) node, plus a tunnel-ID that is a counter used to distinguish different LSPs with the same source-destination pairs.

With the introduction of the slice-ability functionality, it is possible that a single capacity demand is split and served by many sub-LSPs. Rather than assigning a different tunnel-ID to each sub-LSP, we introduced a new field named “sub-LSP-ID” inside the session object. Even if sub-LSPs are treated independently from each other’s, thanks to the sub-LSP-ID, it is possible to keep track of the initial capacity request of the joint LSP and to understand whether it has been fully or partially established.

A child process can behave differently depending on the role of the node assumes for that specific LSP: ingress node, egress node or intermediate node. Therefore different child processes has been developed.

The role is determined when a new LSP request is generated by the node or when a message with a new session object is received. In the former case an ingress child process is created. In the latter case the destination address encapsulated inside the session object is read, if it matches the node address then an egress child process is created, otherwise an intermediate child is created.

The ingress child process is actually decomposed into different processes: one ingress root child process and a number of ingress sub-child processes. This separation is required with the introduction of the slice-ability functionality.

The processes that practically deal with a specific LSP are the ingress sub-child process, the intermediate child process and the egress child process.

These processes are responsible to interpret and elaborate all the RSVP-TE messages of the related LSPs; in practice they implement the RSVP-TE protocol.

The diagrams shown in Figure 8, Figure 9 and Figure 10 represent the lifetime of the associated LSP under the perspective of the node role.

2.4.3 Ingress root child process

The ingress root child process is created each time a new LSP request is generated by the node. In particular it is responsible to manage all the potential sub-LSPs associated to the original LSP request.

The state machine of the ingress child process (Figure 7) is composed by the states described below:

Initialization

In this state some variables are initialized, such as the hash table that will keep track the ingress sub-child processes, the total capacity demand of the whole LSP and the number of sub-LSPs that will serve it.

Path Computation Request

The first step after the initialization is to compute the path for the new LSP according to the selected scenario and the routing and spectrum assignment (RSA) scheme.

Two possible scenarios can be chosen before starting the simulation: a distributed scenario, where each node solves the RSA problem on its own, or a centralized scenario, where each node delegates the PCE to compute the RSA and send back a solution.

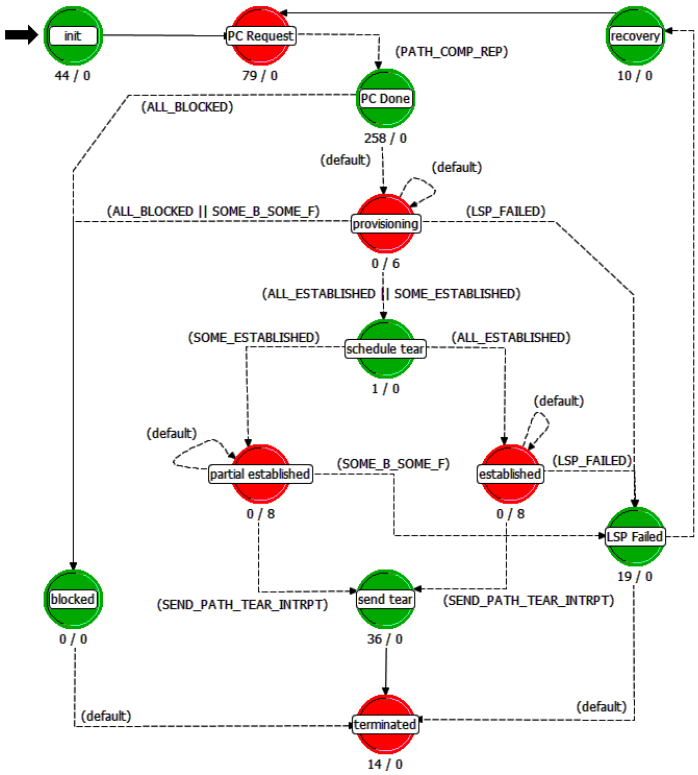

Figure 7 - Ingress root child process state machine

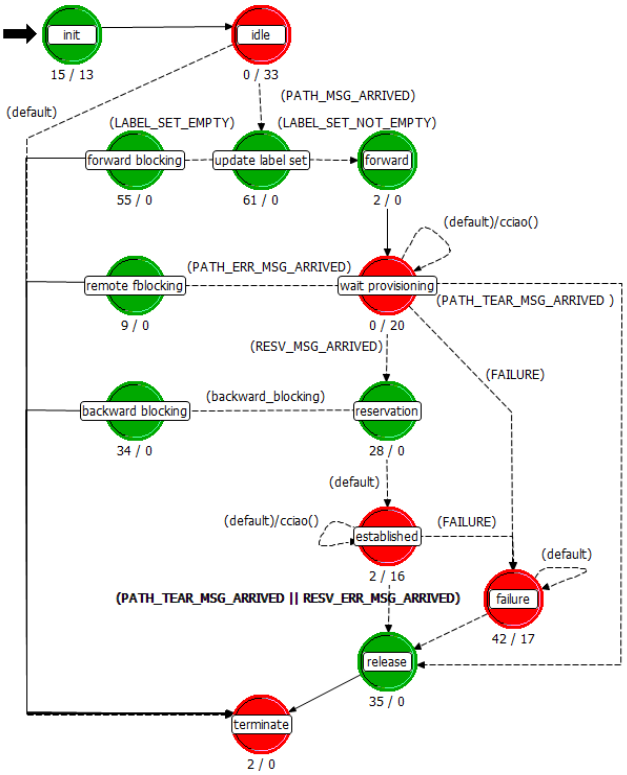

Figure 9 – Intermediate child process state machine

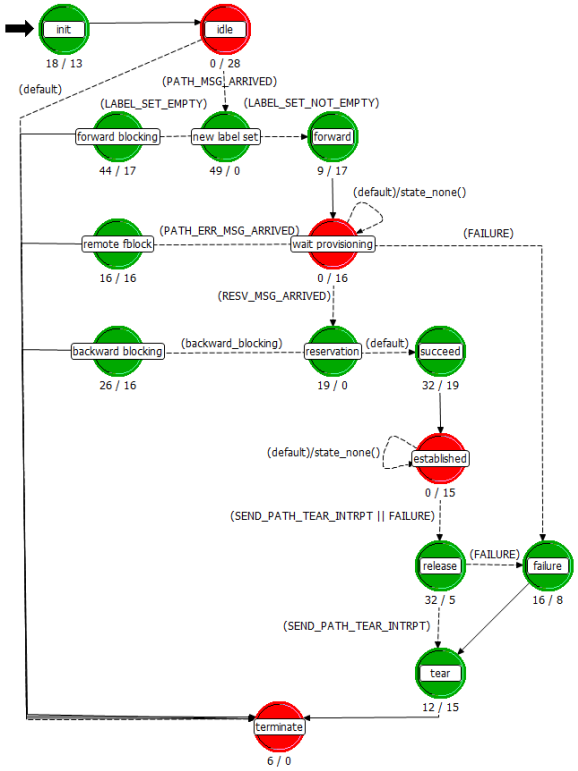

Figure 8 - Ingress sub-child process state machine

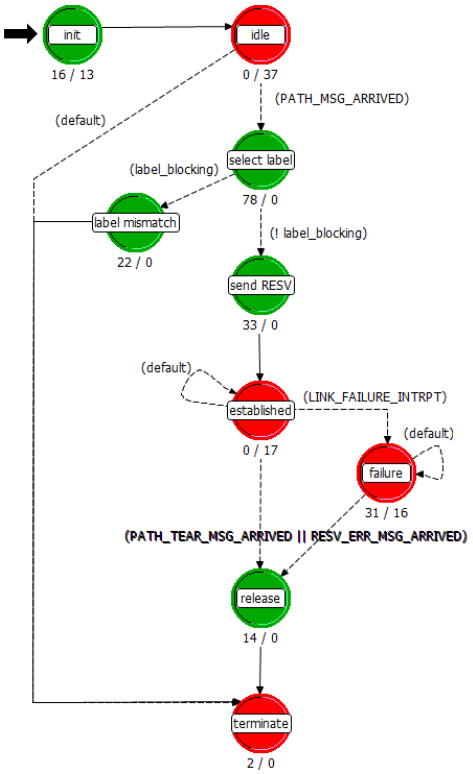

Figure 10 - Egress child process state machine

There are also different RSA schemes that can be adopted: shortest path or least congested path for the routing; first fit, random, last fit, exact fit for the spectrum assignment. Moreover the slice-ability functionality can be activated or not.

Regardless the scenario and the configuration, once the RSA is solved, an ERO object (or a list of EROs in case the slice-ability is exploited) with the associated required bandwidth is returned, then an equivalent number of ingress sub-child processes are generated and receives a couple ERO-bandwidth. Each ingress sub-child process is then responsible for starting the signaling procedure of its associated sub-LSP by sending the PATH message along the route specified by the ERO and requiring the aforementioned bandwidth.

In the case where the RSA solution does not give any ERO object we have routing blocking.

Provisioning

The LSP is considered to be under provisioning until there is still at least one sub-LSP under provisioning. In the meanwhile it is possible that some LSPs has already blocked or established, for this reason a counter for each possible condition has been implemented.

Once the summation of blocked and established sub-LSPs is equal to the total number of generated sub-LSPs, we move to the next state. In particular, if at least one sub-LSP is established, the process schedules the path tear; then it changes either to the established or to the partial established state depending whether all sub-LSPs were successful or not. If all the sub-LSPs stumble on a forward or backward blocking, the whole LSP is considered completely blocked and the process is terminated. If instead the whole LSP is blocked due to a link failure, the process moves into the failure state.

Established and Partial established

In these states the LSP has been partially or fully established and the process is waiting for the end of the connection marked by either the tear down interrupt or the unlucky failure of a link traversed the LSP(s). In both cases the LSP is released through the PATH TEAR message of the RSVP-TE protocol. Only in the latter the process moves into the failure state.

Failure

Whenever an LSP fails the ingress root child process decides whether to try the restoration or not. In this implementation we decided to restore only those LSPs that were not subdivided during the provisioning phase.

Recovery

The recovery procedure consists on trying to reserve recover the disrupted LSPs, hence the state jumps back to the generation step. The only difference is that a new routing and spectrum assignment and a different algorithm for the slice-ability can be adopted with respect to those used during the provisioning phase. Once the capacity is restored, the lifetime of the associated LSPs will be equal to the lifetime left before the failure occurrence.

2.4.4 Ingress sub-child process

The ingress sub-child process is the real manager of the LSP. In fact, it is the only process responsible for the lifecycle of the LSP: it starts the provisioning and it generates the message to tear down the connection.

The main functionalities performed by the ingress sub-child process are listed in the following table that describes the evolution of the state machine.

Event Table

State	Event	Action	Final State
Init	Power up	- Get the module wide shared memory.	Idle
Idle	PATH message arrived	- Generate the label set object based on the spectrum availability of output link.	New Label Set
New Label Set	Label set empty	- In case of centralized architecture, inform the PCE about the forward blocking.	Terminate
New Label Set	Label set NOT empty	- Insert the label set inside the PATH message and send it towards the next node of the ERO object.	Wait Provisioning
Wait Provisioning	PATH ERR message arrived	- Destroy the received message. - In case of centralized architecture, inform the PCE about the remote forward blocking.	Terminate
Wait Provisioning	RESV message arrived	- Destroy the received message. - Check if the selected label (central frequency) is still available.	Reservation
Wait Provisioning	Link failure NOTIFY message	- Destroy the received message. - Send a PATH TEAR message towards the egress node.	Terminate
Reservation	Label NOT available	- Send a RESV ERR message towards the egress node. - In case of centralized architecture, inform the PCE about the backward blocking.	Terminate
Reservation	Label available	- Reserve the bandwidth. - In case of centralized architecture, inform the PCE about the established LSP. - In case of distributed architecture, trigger the OSPF-TE protocol to schedule a new LSA flooding.	Established
Established	Send tear interrupt	- Release the reserved bandwidth. - Send PATH TEAR message towards the egress node. - In case of centralized architecture, inform the PCE about the released LSP. - In case of distributed architecture, trigger the OSPF-TE protocol to schedule a new LSA flooding.	Terminate
Established	Link failure notification	- Destroy the received message. - Release the reserved bandwidth. - Send PATH TEAR message towards the egress node. - In case of centralized architecture, inform the PCE about the released LSP. - In case of distributed architecture, trigger the OSPF-TE protocol to schedule a new LSA flooding.	Terminate

2.4.5 Intermediate child process

The main functionalities performed by the intermediate child process are listed in the following table.

Event Table

State	Event	Action	Final State
Init	Power up	- Get the module wide shared memory.	Idle
Idle	PATH message arrived	- Update the label set object based on the spectrum availability of output link.	Updated Label Set
Updated Label Set	Label set empty	- Destroy the received PATH message. - Send a PATH ERROR message towards the ingress node.	Terminate
Updated Label Set	Label set NOT empty	- Remove the first element from the ERO object. - Forward the PATH message to the next node in the ERO.	Wait Provisioning
Wait Provisioning	PATH ERR message arrived	- Forward the PATH ERR message towards the ingress node.	Terminate
Wait Provisioning	RESV message arrived	- Check if the selected label (central frequency) is still available.	Reservation
Wait Provisioning / Established	Link failure NOTIFY message	- Forward the NOTIFY message towards the ingress node	Failure
Wait Provisioning / Established	Link failure interrupt	- Send a NOTIFY message towards the ingress node.	Failure
Reservation	Label NOT available	- Send a RESV ERR message towards the egress node. - Send a PATH ERR message towards the ingress node. - Destroy the received RESV message.	Terminate
Reservation	Label available	- Reserve the bandwidth. - Forward the RESV message towards the ingress node. - In case of distributed architecture, trigger the OSPF-TE protocol to schedule a new LSA flooding.	Established
Established / Failure	PATH TEAR / RESV ERR message arrived	- Release the reserved bandwidth. - Forward the PATH TEAR / RESV ERR message towards the egress node. - In case of distributed architecture, trigger the OSPF-TE protocol to schedule a new LSA flooding.	Terminate

2.4.6 Egress child process

The main functionalities performed by the egress child process are listed in the following table.

Event Table

State	Event	Action	Final State
Init	Power up	- Get the module wide shared memory.	Idle
Idle	PATH message arrived	- Select a label from the label set giving priority to the suggested label.	Select Label
Select label	Label blocking	- Under centralized architecture, accept only the suggested label.	Terminate
Select label	Suitable label found	- Create the RESV message and send towards the ingress node.	Established
Established	Link failure interrupt	- Send a NOTIFY message towards the ingress node.	Failure
Established / Failure	PATH TEAR / RESV ERR message arrived	- Destroy the received message.	Terminate

2.5 PCEP OPNET Module

The PCEP module implements the PCEP protocol functionalities. Two different finite state machines (FSMs) describe the PCEP module. One root process (FSM in **Figure 7**Figure 11 –PCEP Root FSM) on the top, which interfaces with the TCP to handle new Session requests and dispatches messages to the active Sessions. Multiple peer processes (FSM in Figure 12 –PCEP Peer FSM), each handling a specific session between a PCE/PCC couple.

2.5.1 Root Process

The PCEP root process acts like a manager for the PCEP sessions.

Its main roles consist of:

- Issuing new PCEP sessions (when acting as PCC)
- Receiving and instantiating incoming sessions (when acting as PCE)
- Keeping track of active PCEP sessions

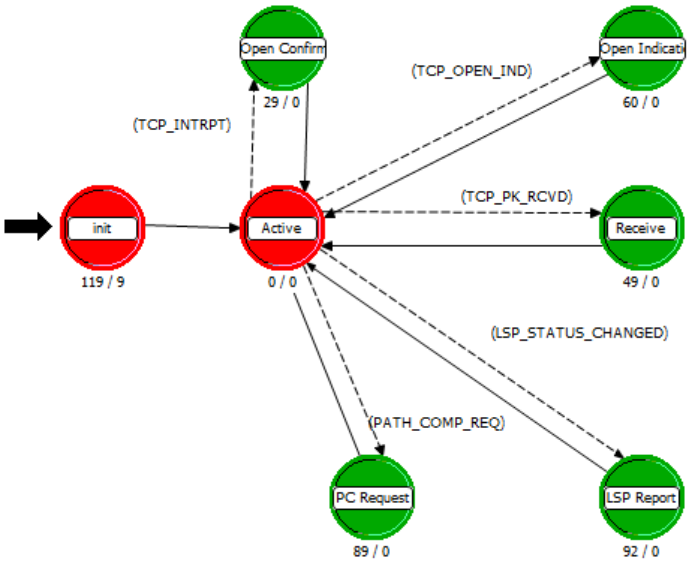

Figure 11 –PCEP Root FSM

In this implementation, PCEP Sessions are permanent, i.e. they remain active until the end of the simulation, and therefore subsequent path computation requests (PCReq) do not need to reestablish the Session.

The FSM of the Root process is described in the table below.

Event Table

State	Event	Action	Final State
Init	Power up	- Install the module memory shared with the various children processes. - Create an hash table to keep track of the PCEP peer processes (i.e. the PCEP sessions)	Active
Active	Tcp open indication	- Receives an indication from the TCP layer about a peer that is attempting to establish a new session. - Create and invoke a new peer process. - Return connection accept status to TCP layer.	Active
Active	Tcp packet received	- Pull the packet from the stream stack. - Invoke the peer process for which the packet was destined based on the TCP connection ID.	Active
Active	Lsp status changed	- Create PCEP Report message. - Invoke the peer process	Active
Active	Path computation request	- Create the PCEP Path Computation Request message. - Invoke the peer process.	Active

2.5.2 Peer Process

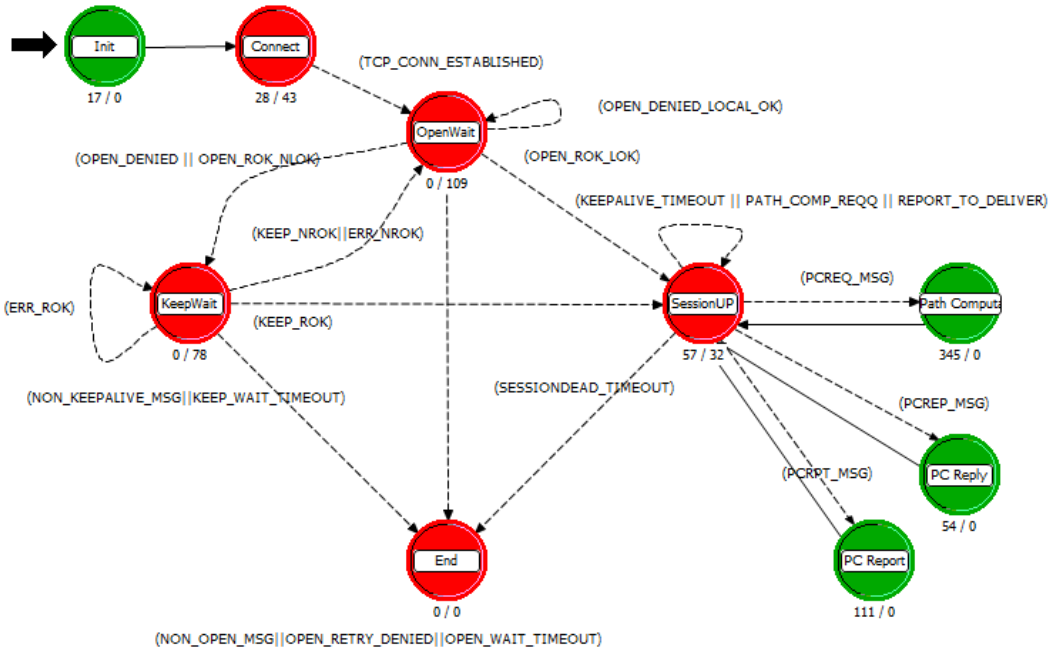

Figure 12 –PCEP Peer FSM

The peer FSM reflects the one described in rfc 5440.

Event Table

State	Event	Action	Final State
Init	Invokation from PCEP Root process	- Access the shared data between root and peer processes. - Access the parent memory passed. - Set the PCEP session as DOWN. - Save the handle to interface with TCP. IF acting as PCC: - Invoke TCP layer to open a new connection towards PCE. - Create a queue that will host the PCReq packets to send towards the PCE. Create a queue that will host the PCRpt packets to send towards the PCE	Connect
Connect	TCP Connection Established	- Set OpenWait self interrupt (1 minute). - Increment session id. - Create the open message to initialize the PCEP session. - Send the open message through the TCP connection. - Set the number of open trials to 0 since is the first open we send. - Initialize remote ok to false.	OpenWait.

		- Initialize local ok to false. - Initialize dead timer to 0 (disabled).	
OpenWait	Open Message Received AND acceptable session characteristics	- Clear the openwait self-interrupt. - Read the received characteristics and check if are acceptable. - Set remote_ok to true. - Set KeepWait self-interrupt (1 minute). IF local_ok: IF the dead timer received is enabled: - Schedule a SESSION DEAD self-interrupt after "dead timer" seconds. IF the keepalive parameter is enabled: - Schedule a KEEP ALIVE self-interrupt after "keepalive" seconds - Create Keepalive message and send through the TCP connection.	IF local_ok: SessionUP. ELSE: KeepWait.
OpenWait	Open Message Received AND NOT acceptable session characteristics	- Create PCErr message and send through the TCP connection. IF first open trial: IF local_ok: - Schedule OpenWait self-interrupt after 1 minute. ELSE: - Close TCP connection. - Increase number of open trials.	IF open trial >2: End. ELSE: IF local_ok: OpenWait. ELSE: KeepWait.
OpenWait	Non Open message received	- Create PCErr message and send through the TCP connection. - Close TCP connection.	End.
OpenWait	Openwait self-interrupt	- Create PCErr message and send through the TCP connection. - Close TCP connection.	End.
KeepWait	KeepAlive message received	- Clear the keepwait self-interrupt. - Set local_ok to true. IF remote_ok: IF the dead timer received is enabled: - Schedule a SESSION DEAD self-interrupt after "dead timer" seconds. IF the keepalive parameter is enabled: - Schedule a KEEP ALIVE self-interrupt after "keepalive" seconds. ELSE: - Restart the openwait self-interrupt (1 minute).	IF remote_ok: SessionUP. ELSE: OpenWait.
KeepWait	Non Keepalive message received	- Create PCErr message and send through the TCP connection. - Close TCP connection.	End.
KeepWait	Keepwait self-interrupt	- Create PCErr message and send through the TCP	End.

		connection. - Close TCP connection.	
SessionUP	Path Computation Request from upper layer.	- Create PCReq message and send through the TCP connection. - Save a reference to the upper layer process requesting the path computation.	SessionUP
SessionUP	PCReq Received	- Invoke the Path Computation Library. - Update the LSP-DB (stateful functionality). - Create PCRep message including the solution and send it through the TCP connection.	SessionUP
SessionUP	PCRep Received	- Update the LSP-DB.	SessionUP

Users interface allows to configure the following parameters:

Attribute name	Type	Description
PCE Address	String	Is the IP address of the PCE.
Keepalive	Double	Minimum period between the sending of PCEP messages (Keepalive, PCReq, PCRep, PCNtf) to a PCEP peer in seconds. By default is 30 seconds.
Deadtimer	Double	Period of time after which a PCEP peer declares the session down if no PCEP message has been received. By default is four times the Keepalive.
Stateful PCE	Boolean	When true the stateful functionalities are enabled.
Provisioning Path Computation Params	Struct	Parameters used by the path computation during provisioning.
Recovery Path Computation Params	Struct	Parameters used by the path computation during recovery.

Regarding the path computation request, the implemented PCReq contains:

- End-points object (source and destination addresses).
- RP Object
- Bandwidth Object
- PC Params Object

The bandwidth object in our case refers to the capacity in Gbps requested by the PCC.

The Path Computation (PC) Params Object contains the Metric and other parameters useful for the path computation that will be described later in the Path Computation Library chapter.

Regarding the path computation reply, the implemented PCRep contains:

- RP Object

- No Path Object
- List of LSP Objects

Instead of list of paths, a list of LSP objects is returned by the path computation in order to support the sliceable functionality. If multiple LSPs Objects are returned it means that the slice-ability has been applied, therefore the traffic demand will be served by multiple LSPs. To associate the path with the LSP, the ERO object is inserted inside the LSP Object.

The PCE has been designed to support different types of transponders and to distinguish among them. For this reason, we extended the LSP Object with some additional information:

- List of Carriers: The list of central frequencies each associated to a different carrier of the transponder.
- Ingress TX Transponder: Identifies the transponder to use as transmitter at the ingress node.
- Ingress RX Transponder: Identifies the transponder to use as receiver at the ingress node.
- Egress TX Transponder: Identifies the transponder to use as transmitter at the egress node.
- Egress RX Transponder: Identifies the transponder to use as receiver at the egress node.

2.6 Path Computation Library

The path computation algorithm depends on the parameters passed to the path computation function. The path computation parameters are the following:

- Metric: Could be “shortest path” or “least congested path among shortest + n”, where n is a configurable parameter representing the additional hops from the shortest.
- Slice-ability: Could be “disabled”, “max slice” where the traffic demand is subdivided into multiple LSPs based on the minimum traffic demand, or, “adaptive slice” where the traffic demand is recursively subdivided only if it cannot fit a single LSP. When the slice-ability is enabled, another parameter specifies whether partial traffic allocation is allowed or not.
- Suggested Label: Specifies the allocation policy used during the path computation. Could be “disabled”, “first fit”, “last fit”, or “random”.

Currently two different branches of algorithms are implemented that in the next future will be merged into one. Both branches supports different metrics and different allocation policies according to the previously defined parameters. The first branch supports slice-ability functionalities while it does not consider the transponders. The second branch supports the different types of transponders but does not support the slice-ability.

2.7 Traffic Engineering Database (TED) OPNET model

The Traffic Engineering Database implementation contains the topology information, i.e. nodes and edges composing the topology graph, plus additional information useful for traffic engineering purposes.

In particular, a node is characterized by:

- Router ID: An address associated to the node, usually the loopback IP address.
- List of Transponders: A list of all the transponders of the node.

For each transponder the following information are stored:

- ID: A unique identifier of the transponder.
- Type: Type of transponder. Up to now Multi-lasers SBVT, Multi-wavelengths SBVT with variable carrier spacing, and, Multi-wavelengths SBVT with fixed carrier spacing are supported.
- Number of carriers: The maximum number of carriers the transponder can generate.
- List of Carriers.
- Max spacing: the maximum spacing among carriers.
- Min spacing: the minimum spacing among carriers.

Edges are characterized by:

- List of Free Slots: The list of available spectrum slots described by lower and upper frequency indexes.
- Failure: Boolean indicating if the link is broken.

The TED is kept up to date through the OSPF-TE protocol.

2.8 LSP Database

When Stateful PCE functionalities are enabled, the LSP-DB stored inside the PCE node is used to keep track of the LSPs lifetime.

The information we store inside the LSP-DB are:

- Session Object: Identifies the LSP
- ERO Object: List of traversed nodes.
- Frequency Slot: The amount of spectrum occupied by the LSP described by lower and upper frequency indexes
- Carriers: List of carriers used by the LSP with the corresponding central frequency.
- Ingress TX Transponder: Identifies the transponder used as transmitter at the ingress node.

- Ingress RX Transponder: Identifies the transponder used as receiver at the ingress node.
- Egress TX Transponder: Identifies the transponder used as transmitter at the egress node.
- Egress RX Transponder: Identifies the transponder used as receiver at the egress node.

3 Sliceable Bandwidth Variable Transponder (SBVT)

In the REACTION project, a bandwidth variable transponder (BVT) architecture supporting super-channel transmission is considered. The BVT supports up to 1 Terabit/s multi-carrier transmission with coherent detection. The proposed BVT is derived from the implementation recently presented in the context of the EU IDEALIST project and enhanced to support the sliceable functionality. In particular, the considered BVT is a multi-flow transponder where each sub-carrier supports dynamic adaptation of modulation formats (e.g., DP-QPSK and DP-16QAM) and coding/FEC types. This way, the need for expensive regenerators will be strongly reduced, with significant simplification in network planning, provisioning, and control plane solutions. In addition, the sliceable functionality is supported: a sliceable BVT (SBVT) has the capability to independently configure and route each sub-carrier in the network.

SBVTs have the potential to combine the cost-reduction provided by the integration and sharing of multiple components in a single transponder with the flexibility guaranteed by the adaptation of each sub-carrier transmission parameter. This way, SBVTs can avoid the need to purchase many different transponders with fixed transmission parameters and rigid number of sub-carriers. Moreover, and differently with respect to previous solutions, the sliceable functionality allows a single SBVT to be connected to different destinations, each served by a sub-set of sub-carriers. This will enable the effective pay-as-you-growth strategy.

For example, as shown in Figure 13, a single SBVT supporting four sub-carriers can be used to connect four different sites at year 1, two different sites at year 2, and a single site with a fully co-routed (and more spectral efficient) super-channel at year 3. Moreover, sliceable transmission has the potential to improve the survivability of the network with respect to fully co-routed super-channels, given the smaller granularity to be considered during the rerouting upon failure. For example, also at year 3 the recovery of the super-channel could be performed by also considering the independent re-routing of each sub-carrier, thus potentially improving the capability to find available spare resources in the flexible network and, in turn, increase the possibilities to recover most or all sub-carriers.

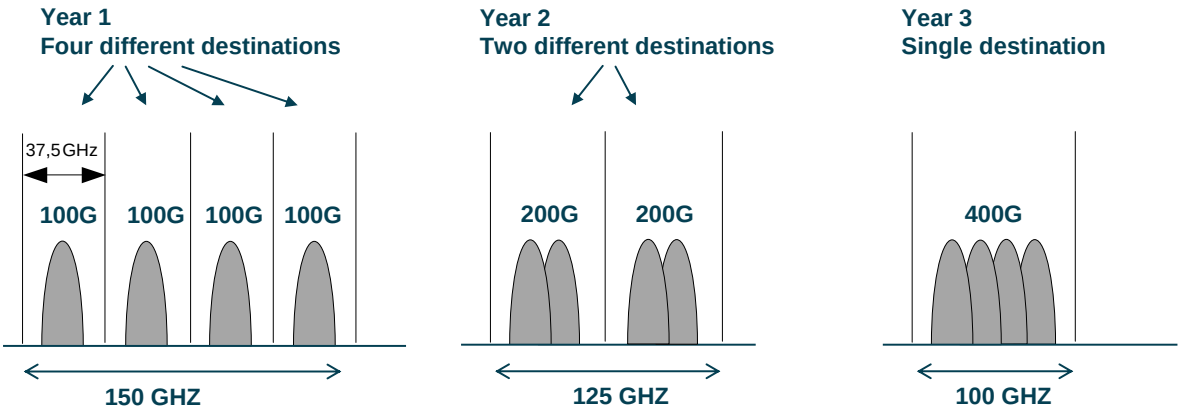

Figure 13 Sliceable functionality applied to a single four-carrier SBVT to cope with traffic increase (year 1, 2 and 3). Spectrum occupancy considered for DP-QPSK format enabling long-reach transmission (smaller occupancy can be achieved e.g. with PM-16QAM and shorter reaches).

In this chapter, the sliceable functionality is assessed in terms of networking performance through specific simulative studies using the REACTION OPNET simulator. In particular, both provisioning and restoration scenarios are considered.

3.1 Reference scenario

Optical transport networks are gradually evolving from the Wavelength Switched Optical Network (WSON) architecture, where all established optical connections (i.e., lightpaths) have to fit in a single channel of the fixed WDM grid, towards Elastic Optical Network (EON) architecture where the spectrum is exploited by means of a flexible grid, where each channel is adaptable to the effective bandwidth requirement.. Therefore, in EONs, high spectral efficiency is achieved because each lightpath can use a different amount of spectrum depending on the bit rate and the exploited modulation format [Jin-CM09, Ger-CM12].

The most important upgrades needed in the data plane for supporting the EON architecture are related to switching and transponder technologies. Nodes are required to support the switching of arbitrary portion of spectrum. Transponders are required to support the generation of multiple optical flow units (i.e., sub-carriers). Using such transponders sub-carriers can be merged in high-rate super-channels (i.e., single connections composed of multiple sub-carriers using a contiguous portion of spectrum), or can be *sliced*, i.e. independently routed along different paths towards the same or different destinations and, not necessary on contiguous portions of spectrum. Those transponders are typically named *multi-flow* or *sliceable bandwidth variable transponders (SBVTs)* [Jin-CM12].

Operators expect that the cost of an SBVT supporting *N* optical flows will be smaller than the cost of *N* BVTs [Lop-JOCN14]. The main reason is related to the production process, indeed the same functionality of *N* BVTs, in the case of an SBVT, can be built in a single integrated platform reducing production costs and transponder dimensions [Lop-JOCN14].

On the control plane, the needed GMPLS protocols extensions in support of flex-grid are under investigation [Cas-JSAC13]. Moreover, the Path Computation Element (PCE) is evolving from a pure state-less condition to an active stateful architecture in which lightpaths state information is stored and used to directly trigger

networking operations, e.g., defragmentation [Pao-SV13, Cas-ECOC13]. Several experimental demonstrations have been recently deployed showing the feasibility of EONs with a properly extended GMPLS/PCE control plane [Cug-JLT12, Cas-JSAC13].

Considering the huge amount of traffic traversing each link, in EONs, as in WSONs, it is fundamental to provide effective recovery mechanisms [Gio-JLT09]. Focusing on dynamic restoration, the EON scenario poses an additional important challenge related to spectrum fragmentation. Indeed, due to different spectrum occupation of established and released lightpaths, the spectrum of the links becomes typically fragmented in non-contiguous portions, thus preventing effective provisioning of new lightpaths and recovery of disrupted lightpaths. Since failures occur at not predictable time and disrupted traffic should be recovered as fast as possible, de-fragmentation methods, that may further delay the recovery process, are not an option to be applied upon failure [Tak-ECOC11]. Therefore, dynamic restoration schemes for EONs have to cope with highly fragmented spectrum. At this regard, the sliceable functionality provides an important opportunity: disrupted super-channels can be sliced to independently recover each sub-carrier, indeed the probability to find a number of narrow slots in a fragmented spectrum is typically higher than the probability of finding a single wide slot [Shi-TC13].

This study assesses the potential benefits of slice-ability during both provisioning and restoration in EONs in order to derive useful indications in support of the BVT implementation. Slice-ability benefits are not obvious because, on one hand, slicing sub-carriers increases the probability to find available spectrum resources, but on the other hand it requires a wider total spectrum with respect to the utilization of a single super-channel.

Specifically, EONs with a GMPLS control plane are considered in two different scenarios: a fully distributed scenario where each network node performs path computation for the locally originating lightpaths, and a centralized scenario where a PCE is used for routing purpose. Specifically, the considered slice-ability schemes allow the sub-carries to be established/recovered using a single super-channel or a number of independent sub-carriers routed along different paths or using not-contiguous spectrum along the same path.

3.2 Sliceable functionality

In this section, a possible network scenario is illustrated where the application of the sliceable functionality can provide benefits during restoration. Then, the most relevant practical implications for supporting such functionality in SBVTs are discussed.

An SBVT typically generates *N* sub-carriers. Let *Fs* be the ITU-T frequency slot occupied by a super-channel composed of *N* sub-carriers, i.e., the portion of spectrum with bandwidth |*F_s*| is expressed as an integer number of frequency slices of width 12.5 GHz. Alternatively, if sub-carriers are sliced, it is assumed that *Fi* is the frequency slot required by a single sub-carrier. Slicing sub-carriers typically introduces a spectrum overhead because, in general, |*Fs*| < *N* * |*Fi*|. As an example, a PM-QPSK super-channel composed of four sub-carriers can be accommodate in 8 slices, i.e. |*Fs*| = 100 GHz. However, if sub-carriers are sliced, each *Fi* needs 37.5 GHz to fit the ITU-T flex-grid.

Figure 14 shows how sliceability can be exploited during restoration for increasing the amount of recoverable traffic. A super-channel composed of *N*=4 sub-carriers is established along the path *s-b-d*, as shown in Figure 14a). Upon failure on link *b-d*, the super-channel cannot be recovered as a whole because no path has enough contiguous spectrum (see Figure 14b).If slice-ability is applied, the different spectrum slots available along each

path can be used to recover the four sub-carriers. In Figure 14(c), two sub-carriers are recovered using a 200 Gb/s super-channel along path *s-a-d*, one sub-carrier is recovered along path *s-c-e-d*, and the last sub-carrier is recovered along path *s-f-g-d*.

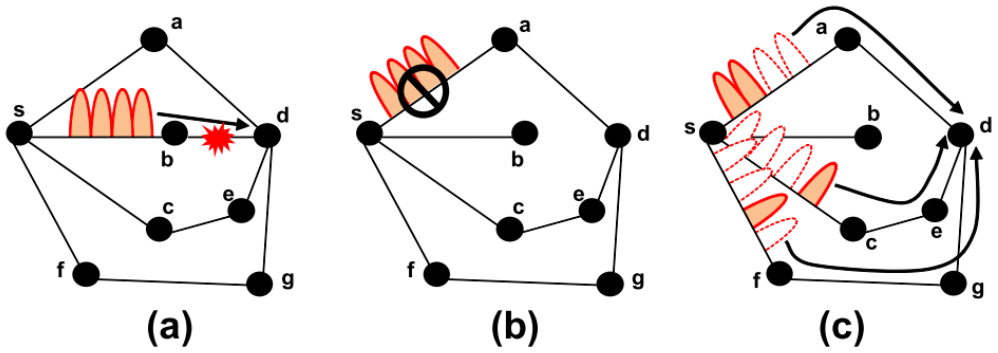

Figure 14: example of sliceable functionality applied during restoration

3.3 SBVT architecture

A typical SBVT architecture is illustrated in Figure 15. It includes three modules: the client module, the adaptation module, and the transmission module.

The client module receives the tributary traffic in the form of *N*M* flows and performs the required signal processing, e.g., Optical Transport Network (OTN) framing.

The adaptation module provides flexibility between the *N*M* client traffic flows and the *N* sub-carriers that will build up the super-channel generated by the SBVT. The utilization of this adaptation module can be particularly useful when slice-ability is used. For instance, if upon network failure only some sub-carriers are recovered, the client traffic flows with more stringent SLAs can be dynamically re-mapped to the recovered sub-carriers.

Finally, the transmission module is in charge of generation, modulation and aggregation of the optical signal. This module is composed of three sub-modules. The laser source sub-module generates *N* optical signals. The modulation sub-module is composed of *N* modulators (e.g., implemented using Photonic Integrated Circuits), each of them modulates one optical signal using *M* traffic tributaries coming from the adaptation module (e.g., *M*=4 in case of Polarization Multiplexed Quadrature Phase Shift Keying modulation, PM-QPSK). The aggregation sub-module multiplexes the *N* sub-carriers in a single output fibre.

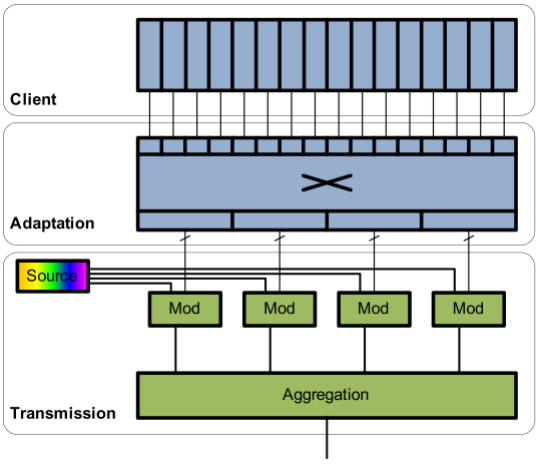

Figure 15: SBVT architecture

The laser source can be currently implemented using two alternative technologies. The first technology is a multi-wavelength source (i.e., a single source generating multiple sub-carriers with a single laser), the second one is the utilization of *N* independent laser sources. The former enables better frequency stability among sub-carriers thus reducing the risk of sub-carrier interference. Moreover, one multi-wavelength source is expected to be cheaper, and with lower foot print and energy consumption than *N* laser sources. The multi-wavelength source supports the tunability of the whole comb, however, the relative tunability among sub-carriers is typically not supported and the contiguity among sub-carriers is required, thus imposing a further constraint to the RSA. On the other hand, using *N* independent lasers do not introduce contiguity constraint, providing full and independent tunability of each subcarrier.

Two different control plane scenarios are considered for evaluating the effectiveness of slice-ability: the GMPLS scenario where path computation during both provisioning and restoration are locally performed at the network nodes, and the GMPLS/PCE scenario where a centralized PCE is employed to perform all the required path computations.

3.4 Control plane

Two different control plane scenarios are considered for evaluating the effectiveness of slice-ability: the GMPLS scenario where path computation during both provisioning and restoration are locally performed at the network nodes, and the GMPLS/PCE scenario where a centralized PCE is employed to perform all the required path computations.

3.4.1 Distributed scenario

In the distributed scenario, each network node stores its own Traffic Engineering Database (TED) including network topology and spectrum availability information. The TED is updated by means of Link State advertisement (LSA) information exchanged through OSPF-TE protocol. During provisioning, upon lightpath request, the source node computes a path and, using RSVP-TE, triggers the establishment of a Label Switched Path (LSP) for serving the specific request.

During restoration, upon failure, the detecting node sends an RSVP-TE Notify to the source node of each disrupted LSP and generates an OSPF-TE Router LSA identifying the failed link which is flooded on the network. A source node receiving the RSVP-TE Notify performs the following actions. First, it sends an RSVP-TE Tear message along the working path of the disrupted LSP, to release the utilized resources; then, it computes a new path by solving the routing and spectrum assignment (RSA) problem considering the locally stored TED and utilizing one of the slice-ability schemes detailed below; finally, when path computation is completed, the source node triggers RSVP-TE signaling to activate the computed backup path.

3.4.2 Centralized scenario

In the centralized scenario, besides the TED, the considered Stateful PCE also stores an LSP state database including information about all the LSPs currently established in the network. During provisioning, upon lightpath request, the source node uses the PCEP PCReq message for asking a path to the PCE. The PCE performs the path computation and replies with a PCEP PCRep message including the computed path and a suggested label indicating the spectrum slot to reserve. The source node will then trigger RSVP-TE to perform the LSP establishment.

Similarly to the distributed scenario, when a failure occurs, the detecting node sends an RSVP-TE Notify to the source node of each disrupted LSPs and floods an OSPF-TE Router LSA for advertising the failure. When the source node receives the Notify message, it sends a RSVP-TE Tear message to release the resources used by the disrupted LSPs. Then a PCEP PCReq message is sent to request the computation of a backup path. When the Tear message reaches the destination node a PCEP PCRpt message is sent to the PCE to report the released resources along the working path of the disrupted LSPs.

With respect to distributed scenario, the use of a PCE delays the recovery procedure. However, the PCE acts as an unique point that can effectively coordinate the recovery operations in order to avoid contentions among the several LSPs under recovery. Specifically, the PCE returns the computed backup path for each disrupted LSP accompanied by a suggested label indicating the spectrum slot to reserve. The PCE avoids blocking by keeping track of the already assigned spectrum slots.

3.4.3 Proposed RSA schemes using slice-ability

Three RSA schemes are applied in both the GMPLS and the GMPLS/PCE scenarios to evaluate the effectiveness of slice-ability:

- *NO slice* is the reference scheme where all lightpath requests are routed/recovered using a single LSP;
- *MAX slice* scheme applies slice-ability to all the lightpath requests slicing them in a number of LSPs using a single sub-carrier;
- *ADAPTIVE slice* scheme iteratively applies slice-ability only to those lightpaths that cannot be routed/recovered as a whole.

3.5 SBVT Performance evaluation

3.5.1 Simulation scenario

The described schemes are evaluated using the OPNET Modeler implemented within the REACTION project. As detailed in the previous section, the OPNET modeler includes an accurate implementation of the RSVP-TE, OSPF-TE and PCEP protocols, with all the extensions required for managing EONs supporting slice-ability.

Both provisioning and restoration are considered, in two network scenarios.

During provisioning, the network is loaded with 100 Gbps LSPs (requiring 3 slices) and 400 Gb/s LSPs (requiring 8 slices). An LSP between node pair *(s,d)* is routed along one of the pre-computed paths in the set $P_{s,d}$ including all the paths within one hop from the shortest path. In both network scenarios, using TED information, the path with the largest number of available frequency slots capable to accommodate the LSP is selected (i.e., least congested routing). 100 Gb/s LSPs are routed as a whole using a single sub-carrier, whereas 400 Gb/s LSPs can be routed as a whole using a single super-channel or divided in sub-LSPs of 200 Gb/s (requiring 5 frequency slices) and/or 100 Gb/s depending on the considered slice-ability scheme.

When restoration is considered a series of single failures of data plane bi-directional links is considered, assuming that the control plane remains fully operational. Specifically, the network is provisioned without applying slice-ability (i.e., NO slice scheme). Upon failure, disrupted 100 Gbps LSPs are restored as a whole using least congested routing on the set of paths $P_{s,d}$ computed considering the failure of link *i*. Conversely, 400 Gb/s LSPs can be recovered as a whole or sliced in sub-LSPs of 200 Gb/s and/or 100 Gb/s depending on the considered slice-ability scheme.

During both provisioning and restoration, using the ADAPTIVE slice scheme, if a 400 Gb/s LSP cannot be served as a whole, it is sliced in two sub-LSPs at 200 Gb/s, in turn if a 200 Gb/s sub-LSP cannot be served as a whole it is sliced in two 100 Gb/s LSPs. When the slice-ability is applied, a separate instance of the least congested routing is applied for each sub-LSP on the set of pre-computed paths so that load balancing is achieved. However, depending on the network load, different sub-LSPs can be routed also on the same path by using a different spectrum allocation.

In this simulative study, the considered test network is a Spanish topology, with 30 nodes and 56 bidirectional links with 256 frequency slices per direction. In the GMPLS/PCE scenario the PCE is located in Madrid. The traffic is uniformly distributed among node pairs and LSPs arrive following a Poisson process, the mean holding time is fixed to 1 hour. Spectrum assignment is first-fit. For each traffic load, a simulation has been performed for 1000 days of network operation where a single failure is generated per each day.

The time required to configure an OXC is considered to be 30 ms. The average path computation time has been estimated to be 0.15 ms (the considered routing algorithm runs on a Personal Computer Intel Core i7-4770 @3.4 GHz, RAM 16GB).

3.5.2 Simulation results: provisioning

During provisioning, the considered schemes are evaluated in terms of provisioning blocking probability (Pr^p) and provisioning time (T^p). Pr^p is defined as the ratio between the blocked traffic bandwidth and the overall requested traffic bandwidth. LSPs requests can be blocked during the path computation for lacking of resources (i.e., routing blocking), or during the signaling procedure for both lacking of resources (i.e., forward blocking) or resource contention (i.e., backward blocking) [Gio-JLT09]. T^p represents the time needed for the LSP setup, and it is designed as the time between the generation of the LSP request and the conclusion of the RSVP-TE signaling.

Figure 16 shows Pr^p as a function of the network load in the GMPLS scenario. The figure shows that applying slice-ability to all LSP requests (i.e., MAX slice scheme) degrades the blocking probability due to the introduced spectrum overhead. However, the figure shows that the ADAPTIVE scheme is able to reduce the blocking achieved by the NO-slice scheme. This proves that slice-ability can be effectively exploited to provision more traffic in the network.

Figure 17 depicts T^p as a function of the network load in the GMPLS scenario. This figure shows that the utilization of slice-ability implies an increase of the provisioning time. Indeed, if slice-ability is applied multiple RSVP-TE signaling are contemporarily triggered to establish the several sub-LSPs generating node configuration queueing. The figure also shows that using the ADAPTIVE scheme the provisioning time is kept at the level of the NO-slice scheme for low and medium network loads, while it gradually moves to the MAX slice scheme level for high loads when slice-ability is applied to the majority of LSP requests.

Figure 18 shows Pr^p as a function of the network load in the GMPLS/PCE scenario. With respect to Figure 16, the utilization of a PCE is able to considerably reduce the blocking probability of both NO-slice and ADAPTIVE schemes. Indeed, in the GMPLS scenario there is a backward blocking floor at about 10^{-4}, that is completely avoided if all the path computation are coordinated by the PCE.

Figure 19 shows T^p in the GMPLS/PCE scenario. With respect to the GMPLS scenario, there is only a negligible increase due to the PCEP communication between network nodes and the PCE.

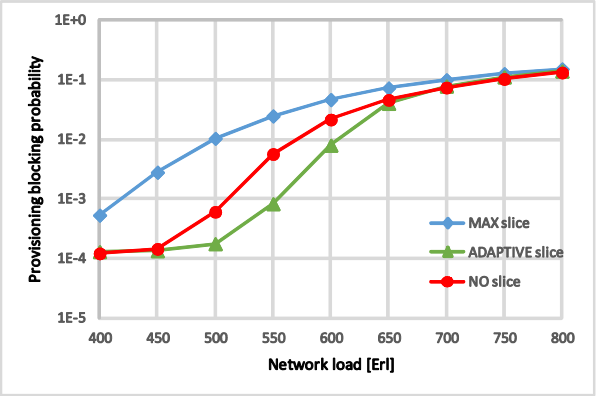

Figure 16: GMPLS scenario, provisioning blocking probability

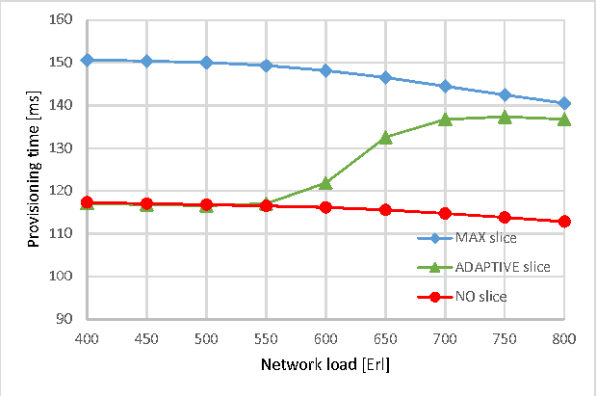

Figure 17: GMPLS scenario, provisioning time

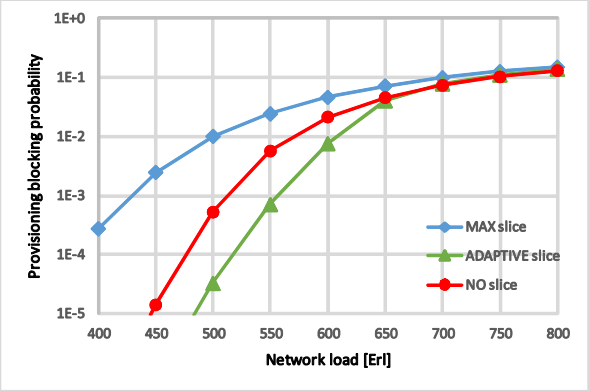

Figure 18: GMPLS/PCE scenario, provisioning blocking probability

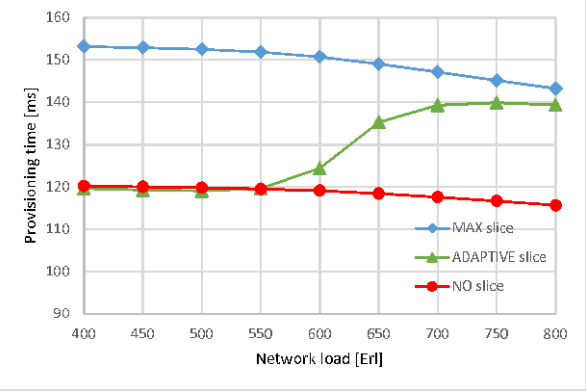

Figure 19: GMPLS/PCE scenario, provisioning time

3.5.3 Simulation results: recovery

During restoration, the considered schemes are evaluated in terms of restoration blocking probability Pr^r and recovery time T^r. Pr^r is defined as the ratio between the not recovered traffic bandwidth and the overall traffic bandwidth disrupted by the failure. T^r is defined only for effectively recovered LSPs as the time between the failure and the conclusion of the RSVP-TE signaling to establish the backup path.

Figure 20 depicts Pr^r as a function of the network load in the GMPLS scenario. The figure shows that the achieved blocking probability is not acceptable (i.e., higher than 20%). Indeed, during restoration, backward blocking is very likely because a high number of RSVP-TE instances are triggered almost simultaneously. In this case, multiple restoration attempts with crankback procedure are typically used to achieve acceptable restoration blocking probability, however only one restoration attempt is considered in the figure.

Figure 21 shows T^r as a function of the network load in the GMPLS scenario. Results show that, for all the schemes, T^r is significantly longer than T^r mainly due to high probability of node configuration contentions.

Figure 22 depicts Pr^r as a function of the network load in the GMPLS/PCE scenario. The figure shows that, at low and medium loads, the MAX slice scheme degrades the blocking achieved when slice-ability is not considered (i.e., NO slice scheme). This is because slicing an LSP when it could be recovered as a whole introduces a useless resource overbuild. Conversely, at high loads when spectrum is highly fragmented and recovering LSPs as a whole is very unlikely, MAX slice provides slight benefit with respect to NO-slice. Finally, the figure shows that independently on the network load the ADAPTIVE slice scheme significantly decreases the restoration blocking probability, i.e., blocking is decreased of 72.5% at 600 Erlang.

Figure 23 depicts T^r as a function of the network load in the GMPLS/PCE scenario. The figure shows that the recovery time is considerably increased if the slice-ability is applied to all disrupted LSPs (i.e. NO-slice scheme), however, the increase of recovery time is marginal with the utilization of ADAPTIVE slice scheme.

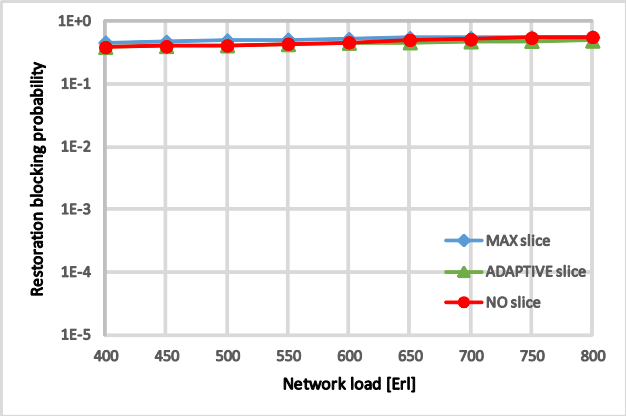

Figure 20: GMPLS scenario, restoration blocking probability

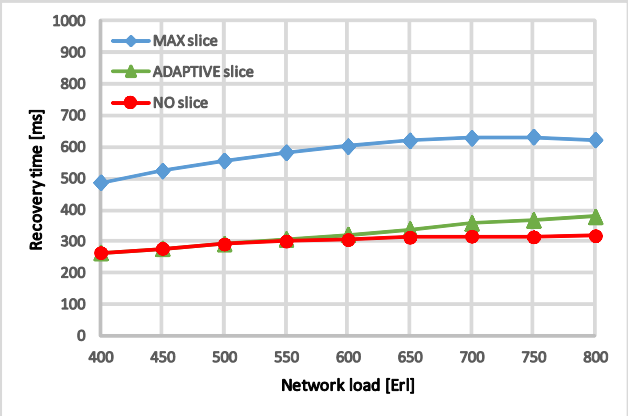

Figure 21: GMPLS scenario, recovery time

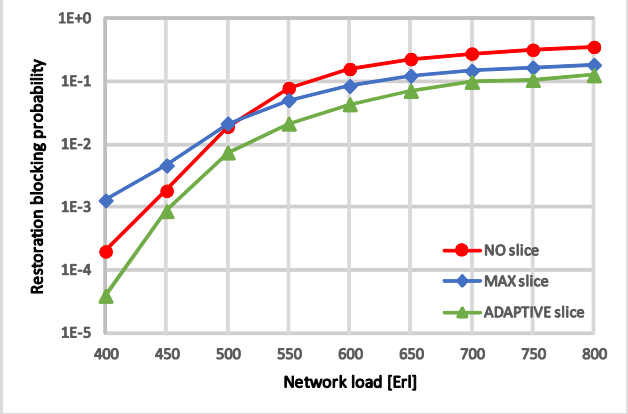

Figure 22: GMPLS/PCE scenario, restoration blocking probability

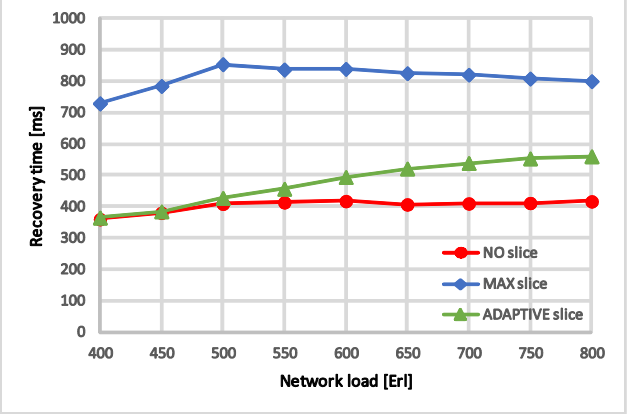

Figure 23: GMPLS/PCE scenario, recovery time

3.6 Concluding remarks

This study investigated the utilization of slice-ability in GMPLS-based EONs. Two network scenarios have been considered: a fully distributed GMPLS control plane, and a GMPLS/PCE control plane where a PCE is used for centralized path computation.

Three slice-ability schemes have been compared during both provisioning and restoration.

Simulation results showed that slice-ability can provide some benefits in terms of provisioning and restoration blocking probability if it is used only when there is no possibility to serve the traffic request with a single super-channel. In terms of provisioning and recovery time, the utilization of slice-ability introduces only a marginal degradation. Finally, it is proved that, during restoration, slice-ability can be effectively exploited only if a PCE is used to coordinate the recovery operations.

4 Taking Advantage of SBVTs: Recovery

In this section we focus on the SDN architecture where solutions are proposed, designed and implemented in the context of a restoration use case applying bitrate squeezing and multipath restoration.

4.1 Bit-rate squeezing and multipath restoration

The contribution of this work is two-fold: first, bitrate squeezing and multipath restoration (BATIDO) problem is formally stated and then modelled using an Integer Linear Programming (ILP) formulation. The BATIDO problem aims at maximizing the amount of restored bitrate by exploiting the available spectrum resources also along multiple routes. As a result of the stringent time to computing a solution, a heuristic algorithm providing better trade-off between optimality and complexity is proposed to solve the problem. Second, a SDN architecture is introduced to support the configuration of SBVTs in EONs. OpenFlow extensions are presented and implemented to control the specific EON transmission parameters.

4.1.1 Overview

To illustrate the restoration schemes that can be applied to the set of connections affected by a failure in the context of EONs, Figure 24 shows a simple network topology were a lightpath is set-up between nodes *s* and *t*. In normal conditions (a), let us assume that the lightpath uses a slot consisting of 16 frequency slices to convey the requested bitrate, in our example 400 Gb/s.

Let us imagine that a link failure occurs and the lightpath is affected. If a restoration lightpath, including route and slot, can be found in the network for the required 16 slices, the disrupted lightpath is obviously restored using the restoration path. This is the normal restoration scheme that has been traditionally used in optical networking; we call this scheme as single path restoration (Figure 24b).

However, in contrast to protection schemes, the availability of 16 contiguous frequency slices at failure time is not generally guaranteed in restoration schemes. In such case, the disrupted lightpath can be restored in part. This case, named single path restoration with bitrate squeezing in this paper, is illustrated in Figure 24c. Note that the restoration lightpath uses just 10 frequency slices and thus, the conveyed bitrate has been squeezed to 200Gb/s.

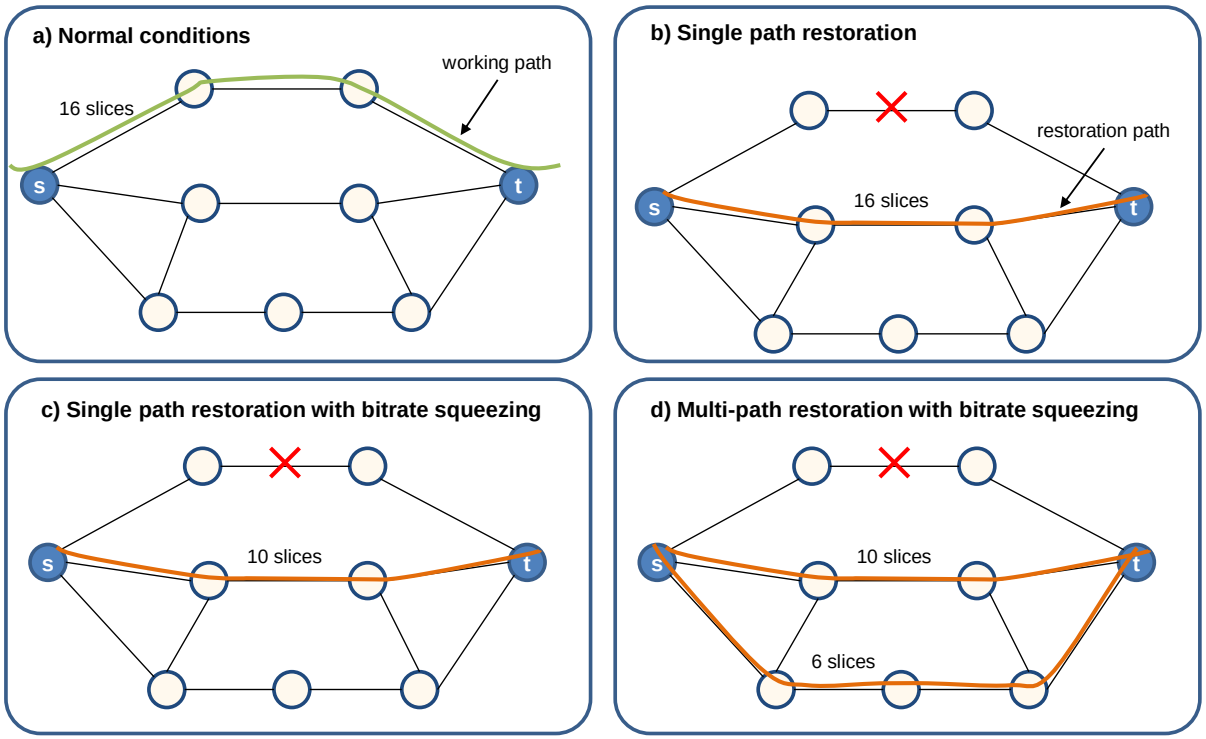

Figure 24: Bitrate squeezing and multipath restoration.

Another possibility is to use several parallel lightpaths, each conveying part of the total bitrate, so restore the original bitrate of the disrupted lightpath (d). Note that in this case, although restoration lightpaths use 16 frequency slices, the total bitrate cannot be recovered, since 200Gb/s can be conveyed within 10 slices and only 100Gb/s within 6 slices. This illustrates the fact that spectral efficiency decreases when multiple lightpaths are used. Although for that very reason network operators prefer not using multipath for provisioning, it can be exploited to improve restorability, provided that the number of parallel lightpaths is kept limited.

After stating the bitrate squeezing and multipath restoration problem, next subsections present two alternative ways to solve it.

4.1.2 The Bitrate Squeezing and Multipath Restoration (BATIDO) problem statement

The problem can be formally stated as follows:

Given:

- a network topology represented by a graph G(*N*, *E*), where *N* is the set of optical nodes and *E* is the set of fiber links connecting two optical nodes,
- a set *S* of available frequency slices of a given spectral width in each fiber link in *E*,
- a set *D* of failed demands to be restored, each requesting a fixed bitrate b^d.

Output: the routing and spectrum assignment for each restored demand $d \in D$.

Objective: maximize the total restored bitrate.

As previously discussed, the BATIDO problem can be faced using three different approaches: *i*) single path restoration, where the total requested bitrate is either restored using a single lightpath or blocked; *ii*) bitrate squeezing restoration, where part of the originally requested bitrate is restored whilst the rest is blocked; and *iii*) multipath restoration with bitrate squeezing, where several restoration lightpaths can be established to recover partially or totally one single demand.

A ILP-based model, which includes the above schemes, is presented next.

4.1.3 ILP formulation

The ILP model is based on the *link-path* formulation for RSA in [Vel-JLT14], where a set of routes are computed beforehand for each demand (excluding the failed fiber link). It is worth highlighting that the term lightpath includes both, the physical route and the allocated slot. Pre-computed slots are used to ensure frequency slice contiguity in the input data, thereby alleviating to some extent the problem complexity. The characteristics of the considered modulation format are also embedded in the input data.

The following sets and parameters have been defined:

Topology:

N	Set of optical nodes, index n.
E	Set of fiber links, index e.
K	Set of pre-computed routes excluding the failed fiber link, index k.
P	Set of SBVTs, index p.
$N(k)$	Subset of optical nodes that are source or destination of route k.
$K(n)$	Subset of routes which source or destination optical node is n.
$P(n)$	Subset of SBVTs of optical node n.
h^k_e	Equal to 1 if route k uses link e, 0 otherwise.
b^k	Maximum bitrate (in Gb/s) that route k can convey (due to physical impairments).
fF^n_p	Number of free flows in SBVT p of optical node n. If $p \notin P(n)$, fF^n_p is equal to 0.
b^n_p	Unreserved capacity of SBVT p of optical node n in Gb/s.

Spectrum:

S	Set of frequency slices available in each link, index s.
u_{es}	Equal to 1 if the slice s in fiber link e is free, 0 otherwise. To compute this parameter, only non-failed lightpaths are considered.
C	Set of slots, index c. Each slot c contains a subset of contiguous slices.

l_{cs} Equal to 1 if slot *c* uses slice *s*, 0 otherwise.

b_c Bitrate capacity of slot *c* in Gb/s.

Failed demands:

D Set of failed optical demands to be restored, index *d*.

b^d Bitrate requested by demand *d* in Gb/s.

K(*d*) Subset of routes for demand *d* (includes reach constraint).

C(*d*) Subset of feasible slots for demand *d* so to restore some amount of bitrate in the range (0, b^d].

sQ Equal to 1 if squeezing is used, 0 otherwise.

mP Equal to 1 if multipath approach is used, 0 otherwise.

mk^d Maximum number of lightpaths that can be used to restore demand *d*, when multipath approach is selected.

The decision variables are:

x^{dk}_c Binary. Equal to 1 if demand *d* uses route *k* and slot *c* for restoration, 0 otherwise.

x^{kn}_{cp} Binary. Equal to 1 if restoration lightpath with route *k* and slot *c* uses SBVT *p* in optical node *n*, 0 otherwise.

y^d Positive real. Restored bitrate for demand *d*.

y^k_c Positive real. Bitrate conveyed by restoration lightpath with route *k* and slot *c*.

y^{kn}_{cp} Positive real. Bitrate conveyed by restoration lightpath with route *k* and slot *c* using SBVT *p* in optical node *n*.

w^d Binary. Equal to 1 if demand *d* is restored (total or partially), 0 otherwise.

z^d Positive integer accounting for the number of lightpaths used to restore demand *d*.

The ILP formulation for the BATIDO problem is as follows:

$$(\text{BATIDO}) \quad Max \quad \Phi = \sum_{d \in D} y^d \tag{1}$$

subject to:

$$\sum_{k \in K(d)} \sum_{c \in C(d)} x_c^{dk} \le mP \cdot \left(mk^d - 1\right) + 1 \quad \forall d \in D \tag{2}$$

$$\sum_{k \in K(d)} \sum_{c \in C(d)} x_c^{dk} = z^d \quad \forall d \in D \tag{3}$$

$$z^d \le mk^d \cdot w^d \quad \forall d \in D \tag{4}$$

$$y^d \le \sum_{k \in K(d)} \sum_{c \in C(d)} b_c \cdot x_c^{dk} \quad \forall d \in D \tag{5}$$

$$y^d \le b^d \quad \forall d \in D \tag{6}$$

$$B \cdot sQ + y^d \ge b^d \cdot w^d \quad \forall d \in D \tag{7}$$

$$\sum_{d \in D} x_c^{dk} \leq \sum_{p \in P(n)} x_{cp}^{kn} \quad \forall k \in K, c \in C, n \in N(k) \tag{8}$$

$$\sum_{p \in P(n)} x_{cp}^{kn} \leq 1 \quad \forall k \in K, c \in C, n \in N(k) \tag{9}$$

$$\sum_{k \in K} \sum_{c \in C} x_{cp}^{kn} \leq fF_p^n \quad \forall n \in N, p \in P(n) \tag{10}$$

$$\sum_{d \in D} b_c \cdot x_c^{dk} \leq y_c^k \quad \forall k \in K, c \in C \tag{11}$$

$$y_c^k \leq b^k \quad \forall k \in K, c \in C \tag{12}$$

$$y_{cp}^{kn} \leq b^k \cdot x_{cp}^{kn} \quad \forall n \in N, p \in P(n), k \in K(n), c \in C \tag{13}$$

$$y_{cp}^{kn} \leq y_c^k \quad \forall n \in N, p \in P(n), k \in K(n), c \in C \tag{14}$$

$$y_{cp}^{kn} \geq y_c^k - b^k \cdot \left(1 - x_{cp}^{kn}\right) \quad \forall n \in N, p \in P(n), k \in K(n), c \in C \tag{15}$$

$$\sum_{k \in K} \sum_{c \in C} y_{cp}^{kn} \leq b_p^n \quad \forall n \in N, p \in P(n) \tag{16}$$

$$\sum_{d \in D} \sum_{k \in K(d)} \sum_{c \in C(d)} h_e^k \cdot l_{cs} \cdot x_c^{dk} \leq u_{es} \quad \forall e \in E, s \in S \tag{17}$$

Equation (1) maximizes the total restored bitrate Φ. Constraint (2) accounts the number of lightpaths assigned to each failed demand. Constraint (3) stores the number of lightpaths used to restore each demand, and constraint (4) limits that number to the maximum value allowed while keeping track of the restored demands. Constraint (5) accounts the restored bitrate of each demand, which is limited by demand's bitrate in constraint (6). In case bitrate squeezing is not allowed, constraint (7) ensures that all its bitrate is restored or blocked. Parameter *B* represents a large integer number that can be computed as the larger bitrate of all demands.

Constraints (8)-(17) deals with lightpath allocation. Constraints (8) and (9) assign one SBVT at each end of the lightpath, whilst constraint (10) ensures that the available number of flows in each SBVT is not exceeded. Constraint (11) accounts the bitrate conveyed by a lightpath, whereas constraint (12) makes sure that the maximum bitrate for the assigned route *k* (b^k) is not exceeded. Constraints (13)-(15) force y^{kn}_{cp} to take the value of $x^{kn}_{cp} \cdot y^k_c$, and constraint (16) limits the total bitrate that can be allocated in each SBVT. Constraint (17) guarantees that at most by one lightpath uses each slice in each link.

Regarding the complexity of the BATIDO problem, it is *NP*-hard since simpler network routing problems have been proved to be *NP*-hard. Regarding its size, the number of variables is O(|*D*|·|*K*|·|*C*| + |*N*|·|*P*|·|*K*|·|*C*|) and the number of constraints is O(|*D*| + |*N*|·|*P*|·|*K*|·|*C*| + |*E*|·|*S*|).

Although the ILP model can be solved for small instances, its exact solving becomes impractical for realistic backbone networks under appreciable load, even using commercial solvers. Hence, aiming at providing near-optimal solutions for the BATIDO problem within the stringent required times (e.g., hundreds of milliseconds), we present next a heuristic algorithm to solve the problem.

4.1.4 Heuristic algorithm

In To cope with the required computation time constraints, we propose a very simple but efficient heuristic algorithm that generates a number of randomized solutions. The algorithm consists in performing a fixed

number of iterations (*maxIterations*) in which one solution is built from a randomly sorted set of demands (lines 2-5 in Table 1). Next, *k* shortest routes are computed for each demand (line 6-7), which are afterwards sorted in decreasing slot width order (line 8). For each route, the larger available slot (considering both continuous free slices and available resources at end nodes) is selected and the restored bitrate is updated (lines 10-13).

In case that the demand can be totally restored, the corresponding lightpath is allocated on graph *G* to prevent that these resources are used by subsequent restoration lightpaths. The lightpath is finally added to the solution being built (lines 14-18) and the algorithm continues with the next demand. In case that the demand can be only partially restored, we need to consider two options. Firstly, if multipath is allowed, the lightpath is allocated, added to the solution, and new lightpaths are considered, provided that the number of lightpaths already used does not exceeded the given limit (lines 19-25). Secondly, if bitrate squeezing is allowed, the lightpath is allocated and added to the solution (lines 26-30). Otherwise, the demand cannot be restored and it is blocked (lines 31-32).

Table 1: Heuristic Algorithm pseudo-code.

```
IN: N, E, D, mP, sQ, maxIterations
OUT: Φ, Sol
1:  Φ ← 0; Sol ← ∅
2:  for i = 1..maxIterations do
3:      tempΦ ← 0
4:      tempSol ← ∅
5:      randomSort(D)
6:      for each d ∈ D do
7:          Kd ← kShortestRoutes(N, E, d)
8:          sort(Kd, DEC_SLOT_WIDTH)
9:          z^d ← 0; tempB ← 0
10:         for each k ∈ Kd do
11:           z^d ++
12:           c ← getLargerSlot(k, b^d)
13:           tempB ← tempB + getBitrate(c)
14:           if tempB ≥ b^d then
15:             allocate(k, c)
16:             tempΦ ← tempΦ + b^d
17:             tempSol ← tempSol ∪ {d, k, c}
18:             break
19:           else
20:             if mP then
21:               allocate(k, c)
22:               tempΦ ← tempΦ + tempB
23:               tempSol ← tempSol ∪ {d, k, c}
24:               if z^d = mk^d then
25:                 break
26:             else if sQ then
27:               allocate(k, c)
28:               tempΦ ← tempΦ + tempB
29:               tempSol ← tempSol ∪ {d, k, c}
30:               break
31:             else
32:               break
33:     if tempΦ > Φ then
34:         Φ ← tempΦ
```

```
35:          Sol ← tempSol
36:      resetAllocations(tempSol)
37:  return {Φ, Sol}
```

Once a solution is built, its total restored bitrate is compared against the best solution obtained so far, which it is updated as long as Φ is improved (lines 33-35). Finally, before building new solutions, all the allocated resources are released from graph *G* (line 36). The best solution is eventually returned (line 37).

The proposed heuristic performs a constant number of iterations (*maxIterations*); on each iteration one permutation, as well as one k-shortest paths computation, based on the Yen's algorithm [Yen-QAM70], and one sorting for each demand, are performed. Therefore, the time complexity of the heuristic is polynomial and can be expressed as the $O(maxIterations \cdot |D| \cdot |K(d)| \cdot |N| \cdot (|E|+|N| \cdot \log(|N|)))$.

The performance of the proposed heuristic was compared against the optimal solution obtained from solving the ILP model for small instances. In all the tests performed, the optimal solution was found within running times of few milliseconds, in contrast to just above 1 hour needed to find the optimal solution with CPLEX. Consequently, we use the proposed algorithm to solve the instances presented below.

4.2 Extensions to OpenFlow

The support of multipath restoration and bitrate squeezing is here provided through a SDN architecture enhanced to operate over an EON. In particular, the OpenFlow protocol has been extended to enable the configuration of flexgrid lightpaths, i.e. to configure flexible transmitters, receivers and intermediate BV-OXCs. Three novel messages are hereafter proposed: the LIGHTPATH_IN message (extended from standard PACKET_IN message), the LIGHTPATH_OUT message, and the FLOW_ACK message. Moreover, the existing FLOW_MOD message is extended to support the configuration of the spectrum-flow entry.

The spectrum-flow entry stores the currently active cross-connections of the switch (input port, output port), along with the related reserved spectrum, expressed as the tuple {central frequency, channel width in terms of number frequency slices}, i.e., *n* and *m* values of the ITU-T flexible grid label. The structure of the extended OpenFlow messages is depicted in Figure 25.

The LIGHTPATH_IN message is sent by the source node to the controller requesting the provisioning or restoration of a lightpath. It is inherited by the PACKET_IN message and includes the most utilized parameters of a flexible lightpath, such as the end-points and the requested bitrate. The optional duration time is included in the message. The lightpath parameters are inherited from our implementation for the path computation element protocol (PCEP). The extended FLOW_MOD message is sent by the controller to the switches in order to set up a new flow entry. The novel *com* field specifies the kind of operation (i.e., new flow, modification of existing flow, deletion) and includes the cross-connections indications (i.e., output port, input port), the flexible grid frequency slots (i.e., grid/CS/identifier, *n* and *m*), the modulation format (MF) and forward error correction (FEC) indication. The latter parameters are utilized in the case of source or destination node, indicated by the source/destination/transit (SDT) flag. Moreover, the information bitrate and optional optical channel specification (e.g., in case of multiple sub-carriers the number of sub-carriers and their displacement in terms of central frequency and width) are carried out. The novel FLOW_ACK message is sent by the switch upon the reception of the related FLOW_MOD message, when the data-plane cross-connection configuration is performed. The result of such configuration is reported in this message, specifying the state of the related flow-

entry (i.e., enabled/disabled) and the possible reason of a failed configuration (e.g., software error, hardware error). The LIGHTPATH_OUT message reports the outcome of the lightpath setup and the actual configuration parameters to the source node. In the case of successful setup, its reception triggers the data plane to start data transmission onto the active lightpath.

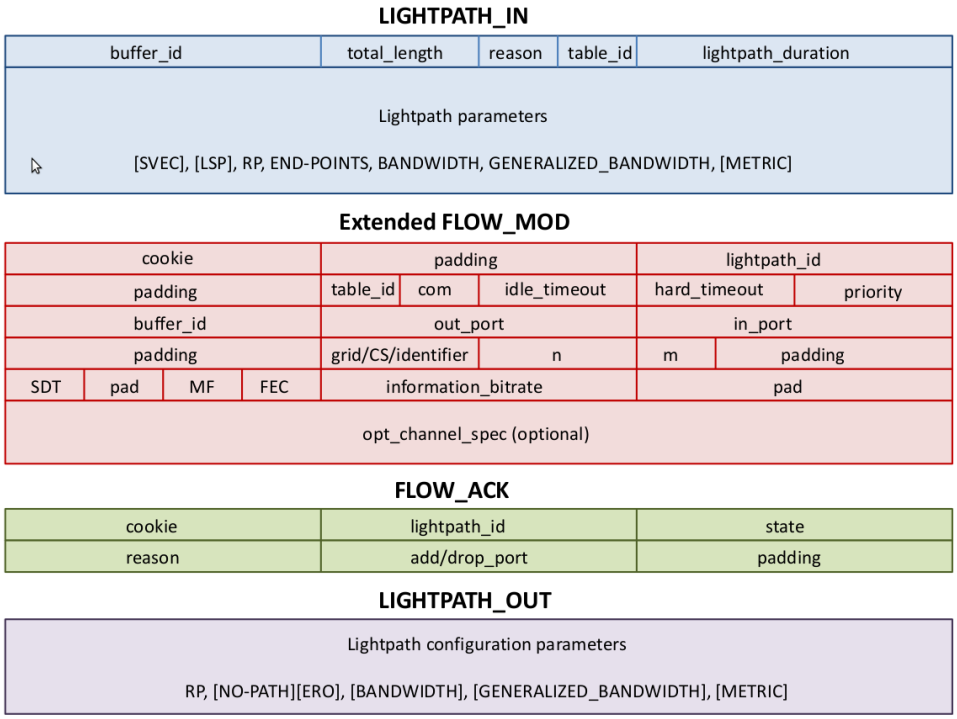

Figure 25: OpenFlow messages extensions.

The proposed extensions are experimentally validated in the next section in terms of provisioning time, the key point to reduce total restoration time in case of failure.

4.3 Experimental validation

In this section, we first assess the proposed OpenFlow extensions on an experimental test-bed. Because of the limited resources available in the test-bed, the performance of the proposed restoration schemes is compared using simulation under real-size core network topologies.

The SDN control plane solution has been developed and experimentally evaluated in a flexgrid optical network test-bed including configurable BV-OXC based on the liquid crystal on silicon (LCOS) technology. In particular, C++-based software implementations of OpenFlow controller and OpenFlow switch, called FlexController and FlexSwitch, respectively, have been developed and tested. The test-bed is depicted in Figure 26 and includes four flexgrid optical nodes, four co-located mini PCs (Intel Atom, CPU 1.60GHz, RAM 1GB) running FlexSwitch and controlling the optical node through an USB interface. All mini PCs are connected through Gigabit Ethernet interfaces to a central controller (Intel Xeon, CPU 3.40GHz, RAM 4GB) running FlexController.

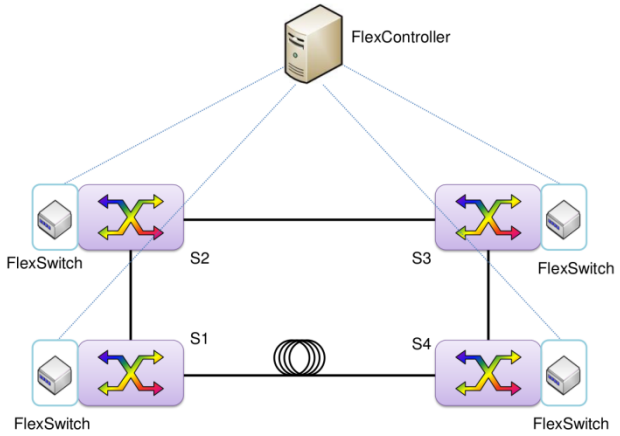

Figure 26: Experimental test-bed used to validate the proposed OpenFlow extensions

The FlexController software tool architecture is depicted in Figure 27 and is based on three main modules: the Controller Handler, the Flex Path Solver and the OpenFlow Interface. The Handler is the main responsible of the controller orchestration, including the path solver trigger, the Label Switched Path (LSP) database update, the OpenFlow event handling and all the session operation involving the controlled nodes. In particular, it stores the active LSPs currently installed and the related OpenFlow entry installed in each switch. Moreover, it implements the mechanism to support the FLOW_ACK message. The Flex Path Solver is responsible for path computation and resorts to the traffic engineering database (TED) updated by the Handler upon lightpath setup events or lightpath duration timer expiration. For each requested bit rate, the Solver performs joint path computation and spectrum assignment based on specific impairment model. The two modules communicate through an internal socket. In addition, the OpenFlow interface maintains the communication with the switches and implements the OpenFlow protocol with the aforementioned extensions.

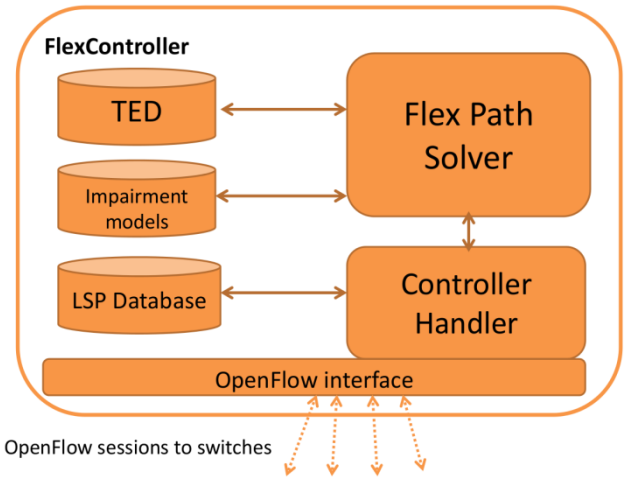

Figure 27: FlexController architecture

The FlexSwitch software implements the OpenFlow switch for flexgrid optical nodes. It includes the OpenFlow session operation, the flow entries management and enforcement, the automatic BV-OXC filter configuration and monitoring. The main modules are shown in Figure 28. In particular, the Switch Handler orchestrates the

switch operation by populating and maintaining the flow entry database and the switch state database (i.e., enclosing the switch abstraction in terms of input and output port, their status and currently reserved spectrum). The Switch Handler exploits XML-based local communication and triggers BV-OXC configuration. The Device Interface is responsible for mapping the flow entry specifications (i.e., cross-connection identified by the couple {in_port, out_port} and reserved spectrum expressed in terms of central frequency and number of frequency slices) into a list of configuration commands to properly set the BV-OXC.

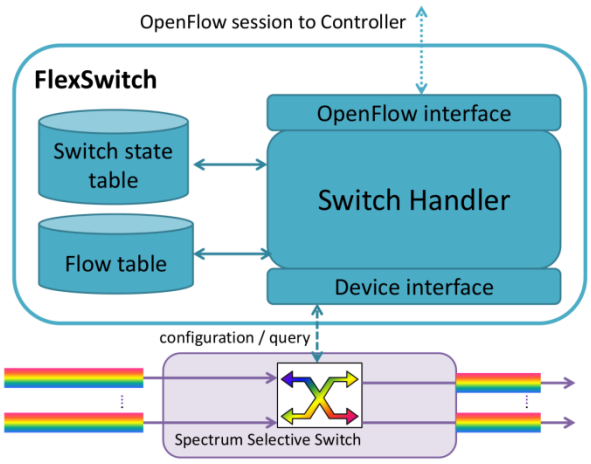

Figure 28: FlexSwitch architecture

The SDN architecture extended for flex-grid has been considered in the case of bitrate squeezing and multipath restoration. In particular, the main objective of this experimental evaluation is to assess the overall time required to apply the configuration computed by the restoration algorithm. To this extent, the SDN FlexController is employed to operate on the simultaneous configuration of different FlexSwitches, which in turn have to trigger the setup of the controlled BV-OXCs. The experiment, repeated 1500 times, consists in the setup of a lightpath request, as being elaborated to restore disrupted connections. In the considered test-bed, the lightpath setup involves three switches, from S1 to S3 are considered. The setup time is computed as the time elapsed between the generation of the first OpenFlow message and the reception of the last one (i.e., not including the path computation time, which remains bounded to less than 100ms when the heuristic algorithm is employed, as described above).

In Figure 29 the setup time distribution is reported, showing that it is in the range of 1.6-1.7 s. It has to be noted that most of the setup time is required to configure the BV-OXC. This time is mainly due to the firmware version of the proprietary software tool, while the actual switching time of the device is around 40 ms. The SDN-based control plane contribution to the setup time is in the range of 30-40 ms. This is mainly due (around 83%) to the FlexSwitch procedures upon the reception of a FLOW_MOD message and, in particular, to the proprietary BV-OXC large configuration file writing procedure performed in the device interface module. Thus, the net contribution of the OpenFlow control plane (i.e., TED update, message exchange, flow entry check and update) is in the order of 5 ms.

Additional experiments have been performed considering a different amount of involved FlexSwitches. In the considered test-bed scenario, given the adopted parallel configuration mechanism, no significant variations have been measured.

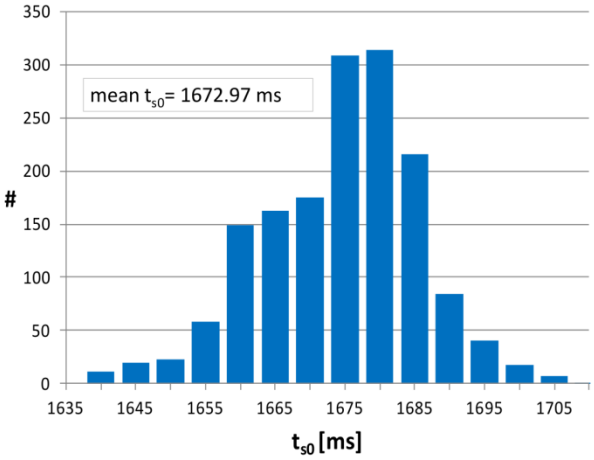

Figure 29: Configuration time.

4.4 Restoration Schemes Evaluation

Once the feasibility of the proposed SDN extensions have been experimentally validated to be used in the help of reducing restoration times, we focus on studying the performance of the restoration schemes. To that end, we consider larger network topologies: the 30-node 56-link Spanish Telefónica (TEL) and the 22-node 35-link British Telecom (BT) topologies; each node location is equipped with one single BV-OXC.

A dynamic network environment was simulated where incoming connection requests arrive to the system following a Poisson process and are sequentially served without prior knowledge of future incoming connection requests. To compute the routing and spectrum allocation of the lightpaths, we used the algorithm described in [Cast-CN12]. The holding time of the connection requests is exponentially distributed with a mean value equal to 2 hours. Source/destination pairs are randomly chosen with equal probability (uniform distribution) among all nodes. Different values of the offered network load are created by changing the inter-arrival rate while keeping the mean holding time constant. We assume that no retrial is performed; if a request cannot be served, it is immediately blocked. Regarding the optical spectrum, the total width was fixed to 4 THz and the slice width to 6.25 GHz.

Besides incoming connection requests, optical link failures follow a Poisson process with a mean time to failure (MTTF) equal to 50 hours, and link failures are randomly chosen with equal probability. We consider that the link is repaired immediately after the restoration process has ended.

In our experiments, the bitrate of each connection request was selected considering 80% of the connections being 100 Gb/s and the other 20% of 400Gb/s. To convert bitrate into spectrum width, we use the correspondence in the table below. Finally, note that each point in the results is the average of 10 runs with 150000 connection requests each.

Table 2: Bitrate-Spectrum width

Bitrate (Gb/s)	Bandwidth (GHz)	#Slices (6.25GHz)
100	37.5	6

200	62.5	10
400	100	16

Figure 30a plots blocking probability as a function of the offered load for the restoration approaches and network topologies considered. Note that offered loads have been limited to those unleashing blocking probability in the meaningful range [0.1%-5%]. For the sake of comparability between topologies, offered loads have been normalized to the value of the highest load.

As shown, all 3 approaches behave similarly, i.e. whatever the restoration approach is selected the blocking probability for provisioning remains unchanged. Note that the considered MTTF value is larger enough to reduce the probability of two lightpaths being affected by two consecutive failures, and hence virtually all the restored lightpaths have been torn down when a new failure occurs.

When we analyze the results for aggregated restorability, i.e. combined values for 100Gb/s and 400Gb/s lightpaths, (Figure 30b), we observe that multipath and bitrate squeezing approaches restore almost all the failed bitrate, in contrast to the non-split one. Certainly, aggregated restorability using multipath and bitrate squeezing is better than 95%, even for the highest considered loads, remarkably higher than that obtained using the non-split approach, which values range in the noticeable poor interval [80%-60%].

To get insight into the performance of the different approaches, Figure 31 focuses restorability for 400 Gb/s lightpaths. The performance of multipath and bitrate squeezing is noticeably divergent: the multipath approach shows significantly better performance than that of the bitrate squeezing. The rationale behind that behavior is that the multipath approach includes the bitrate squeezing one and additionally adds the ability to use several lightpaths to restore one single demand.

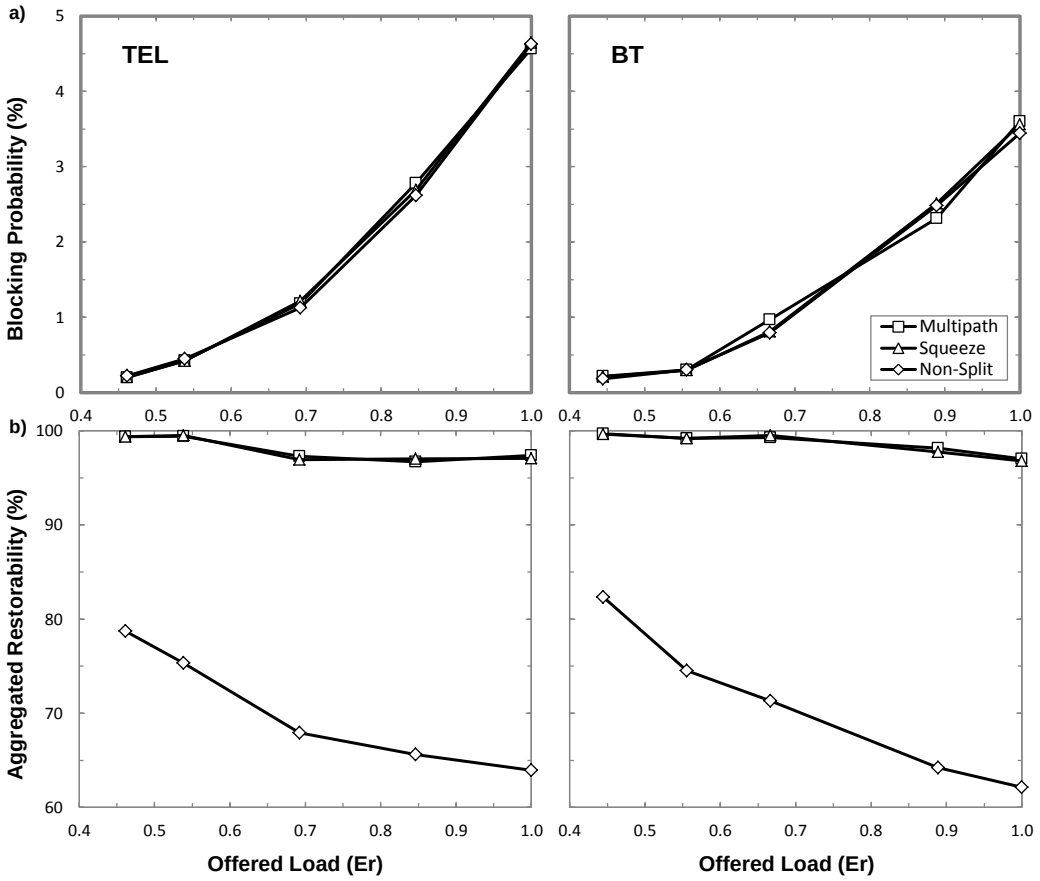

Figure 30: Performance results for the TEL (left) and BT (right) network topologies. Blocking probability (a) and aggregated restorability (b) against offered load.

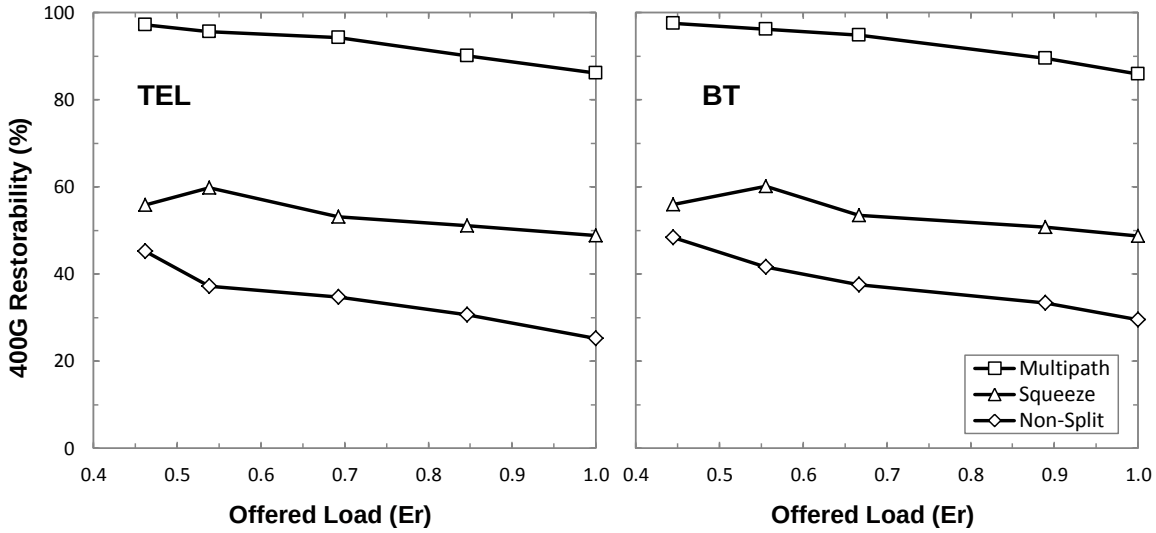

Figure 31: 400Gb/s connections restorability against offered load for the TEL (left) and BT (right) topologies.

To appreciate the way the multipath approach works, let us evaluate the distribution of lightpaths actually used for restoration. Note that since 400 Gb/s demands can be restored using any combination of 400Gb/s, 200Gb/s

and 100Gb/s lightpaths. Thus, 1, 2, 3 or 4 different lightpaths can be used by the restoration approach. Then, Figure 32 depicts number of demands restored using z^d lightpaths and its average value as a function of the offered load.

Figure 32 shows an upwards trend of z^d average that clearly is as a consequence of under heavier loads the probability of finding a single lightpath with enough spectral resources for each demand decreases. That can be clearly observed analyzing the distribution of z^d values; the number of demands restored using one single lightpath decreases as the offered load increases, whereas the demands restored using more than one lightpath increases with the load.

Consequently, as the offered load grows the multipath approach takes advantage from using multiple lightpaths to maximize the total restored bitrate.

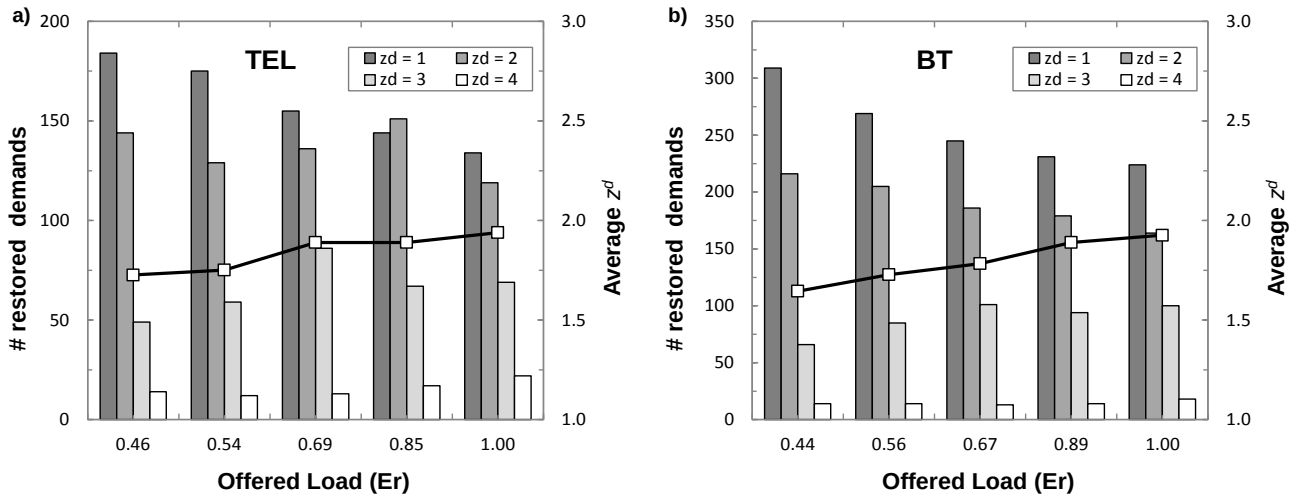

Figure 32: Distribution and average z^d values for restored 400Gb/s demands using the multipath approach against offered load for the TEL (left) and BT (right) topologies.

4.5 Concluding remarks

In this study, the REACTION solutions have been presented and applied on a use case of network restoration. In particular, a technique enabling multipath recovery and bitrate squeezing in elastic optical networks has been proposed, implemented and evaluated. The techniques exploits the advanced flexible capabilities provided by sliceable bandwidth variable transponders (SBVTs), which support the adaptation of connection parameters in terms of number of sub-carriers, bitrate, transmission parameters and reserved spectrum resources.

To efficiently recover network failures by exploiting limited portions of spectrum resources along multiple routes, the BATIDO problem is stated and an ILP model and heuristic algorithm are proposed.

A SDN architecture is also introduced to adequately support the SBVT configuration. The SDN architecture is utilized to experimentally assess the overall re-configuration time upon failure detection, which includes up to 100ms for path computation, around 30-40 ms for OpenFlow communications and around 1.6 seconds to apply the configuration on the BV-OCXs.

Finally, illustrative results obtained by simulation showed that the proposed multipath recovery and bitrate squeezing can even double the percentage of restored bitrate with respect to technique where no squeezing or multipath are exploited.

5 Proactive Hierarchical PCE based on BGP-LS for Elastic Optical Networks

5.1 Introduction

In the context of multi-domain EONs, the Hierarchical Path Computation Element (HPCE) architecture has been proposed to perform effective end-to-end path computation [HPCE]. In the HPCE architecture a single *parent* PCE (pPCE) is responsible for inter-domain path computations, while in each domain a local *child* PCE (cPCE) performs intra-domain path computations. In this scenario, effective inter-domain path computation is achieved only if detailed and updated intra-domain TE information (i.e., spectrum slices availability in EONs) are available or retrievable by pPCE.

The recent introduction of Link State extensions to Border Gateway Protocol (BGP-LS) opened new possibilities to update the pPCE that can improve scalability and effectiveness of the HPCE architecture [BGPLS].

In this study, BGP-LS updates are triggered at the cPCE when by the reception of local interior gateway protocol (IGP) updates. In particular, this study proposes a proactive scheme to update the pPCE, in which BGP-LS updates are not dependent on the interior gateway protocol (IGP) updates, but automatically triggered by path computation requests. The proposed scheme is compared with the most effective PCEP method proposed in [Gio-OFC12] and with the traditional BGP-LS method based on IGP trigger.

5.2 Proposed BGP-LS schemes for Hierarchical PCE update in multi-domain EONs

In multi-domain EONs a separate instance of an Interior Gateway Protocol (IGP), e.g., OSPF-TE, runs in each domain advertising the available spectrum slices of each link. Therefore, each cPCE resorts to a local TE Database (TED) dynamically updated by the IGP for routing and spectrum assignment of intra-domain LSPs considering the spectrum continuity constraint. Conversely, cPCEs resort to pPCE for computing the path of inter-domain LSPs.

Specifically, the pPCE locally stores a Hierarchical TED (i.e., H-TED) that can include a network topology abstraction or a representation of the whole network topology with detailed spectrum availability information. In the former case the pPCE, upon inter-domain path computation request, uses the PCEP protocol to

dynamically retrieve the required intra-domain spectrum availability information from cPCEs. In the latter case BGP-LS is used among cPCEs and pPCE to periodically update the spectrum availability information stored in the H-TED. In both cases, upon reception of an inter-domain LSP request the pPCE is able to use the H-TED for computing an end-to-end path considering spectrum continuity constraint.

Figure 33 and Figure 34 illustrate the path computation procedure for both intra-domain and inter-domain LSPs using two BGP-LS schemes. Figure 33 considers the standard scheme (i.e., BGP-LS IGP), where BGP-LS is triggered at the cPCE by the reception of IGP Link State Advertisements (LSAs). Figure 34 considers the proposed scheme (i.e., BGP-LS PROACTIVE) where BGP-LS is triggered at the cPCE by reception of intra-domain path computation requests.

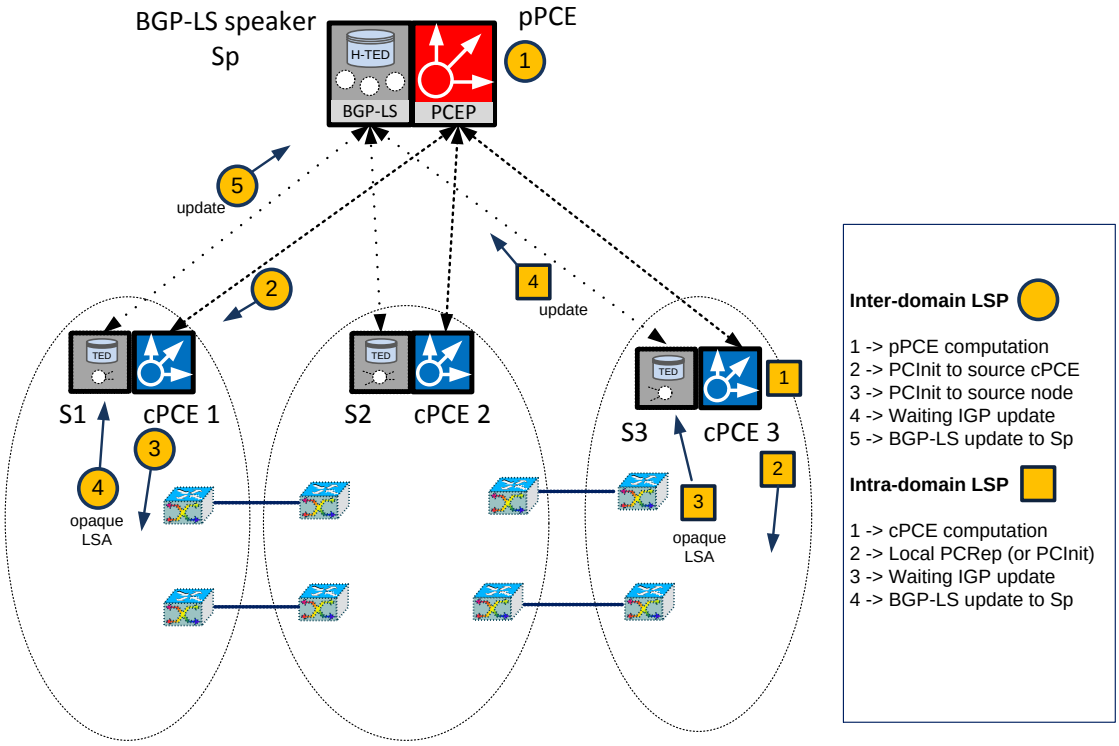

Figure 33: Path computation procedure using the standard BGP-LS IGP scheme to update the H-TED.

BGP-LS IGP scheme: BGP-LS updates are triggered at the cPCEs every time a new IGP LSA is locally received. Specifically, when an inter-domain LSP has to be established (i.e., yellow circles in the figures) the pPCE performs the path computation using the locally stored H-TED and uses a PCEP PCInit message for communicating the computed path to the proper cPCE. In turn, the cPCE communicates the computed path to the LSP source node inside the domain that triggers the RSVP-TE signalling to actually establish the LSP. When resources are effectively reserved by the signalling protocol, traversed nodes generate IGP LSAs in according to the local IGP timers. When the cPCE receives a valid LSA it forwards it to the pPCE using BGP-LS. In case of intra-domain LSPs (yellow squares in the figures) the procedure is the same but the path is locally computed at the cPCE. This mechanism automatically updates the pPCE when an LSP is released.

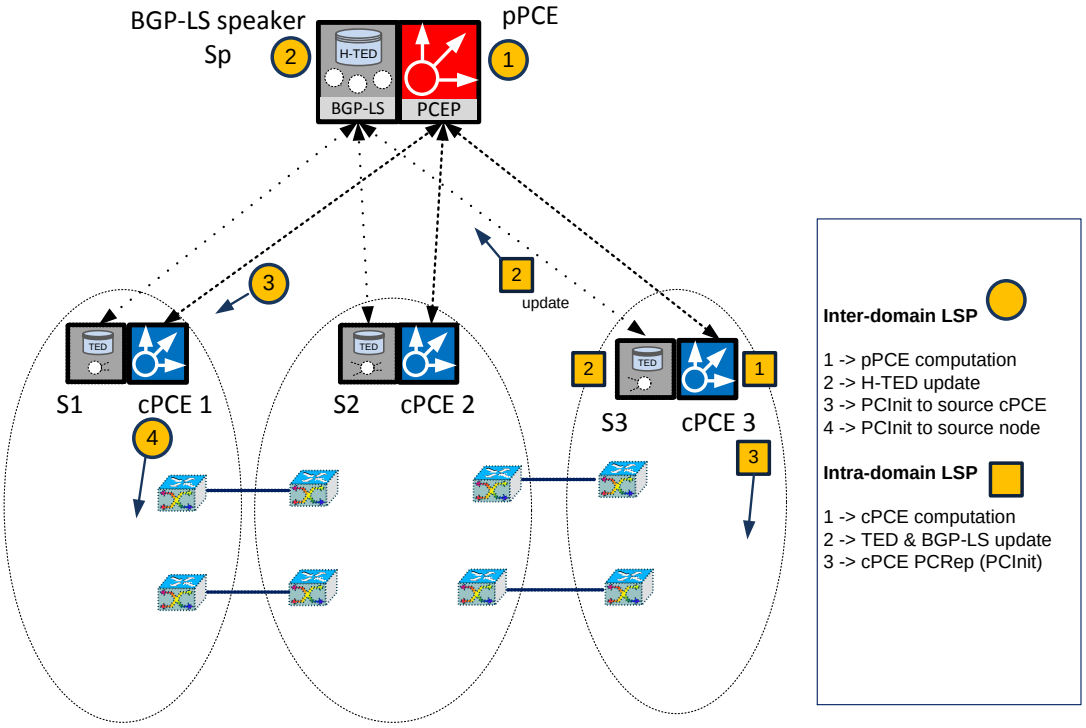

Figure 34: Path computation procedure using the proposed BGP-LS PROACTIVE scheme to update the H-TED.

BGP-LS PROACTIVE scheme: for intra-domain LSPs the pPCE is updated by means of BGP-LS updates triggered by intra-domain path computation requests received at the cPCEs; for inter-domain LSPs the H-TED is automatically updated immediately after the path computation performed at the pPCE. Specifically, when an inter-domain LSP as to be established (i.e., yellow circles in Figure 34) the pPCE performs the path computation using the locally stored H-TED. After path computation it immediately updates the H-TED assuming that the computed path will be shortly established. Then pPCE uses a PCEP PCInit to communicate the computed path to the proper cPCE, then the LSP source node triggers the RSVP-TE signalling. The cPCEs update their TED waiting for the IGP LSAs as in the BGP-LS IGP scheme. In case of intra-domain LSPs (i.e., yellow squares in the figure), the cPCE locally performs the path computation; after path computation it sends to the pPCE a BGP-LS update including the computed path. Upon reception, the pPCE updates the H-TED considering that the computed path will be shortly established. Since the update of the H-TED is done proactively without waiting for confirmation of a successful signalling, in case of signalling errors a communication is required to align the H-TED to the real network status. Specifically, when signalling is blocked or an LSP is established using a different spectrum slot with respect to the one suggested by the PCE the source node has to notify its local cPCE that will inform the pPCE using BGP-LS. Finally, in case of release of an intra-domain LSP a BGP-LS update is also required to the pPCE.

5.3 Simulation results

Schemes are evaluated using a custom built event-driven C++ simulator. The considered multi-domain EON has 75 nodes and 145 bidirectional links with 256 frequency slots per direction covering the whole C-band. The network is divided in 9 domains. Each cPCE is co-located within a domain node, the pPCE is co-located with the cPCE of a central domain. Traffic is uniformly distributed among node pairs, LSPs arrive following a Poisson process, mean holding time is fixed to 1 hours. Spectrum assignment is first-fit. Two LSP granularities are considered with the same generation probability: 100 Gbps LSPs require 3 spectrum slices; 400 Gbps LSP require 9 spectrum slices. OSPF-TE LSA generation rate is set to the minimum value allowed by the standard (i.e., 5 s).

The proposed BGP-LS schemes are compared against the most effective PCEP-based solution proposed in [Gio-OFC12] (i.e., PCEP LABELSET scheme). In this case the H-TED includes an abstraction of the network topology with edge nodes, inter-domain links, and detailed spectrum availability information of inter-domain links. Upon reception of an inter-domain LSP request, the pPCE communicates with cPCEs requiring path computation of specific edge-to-edge segments, the cPCEs reply with the computed edge-to-edge segment and the list of slices that are available on all the links belonging to the segment. Upon reception of all replies, the pPCE selects the path that can accommodate the largest number LSPs considering spectrum continuity.

Figure 35 shows the received control packets at the pPCE (RSVP-TE, OSPF-TE, PCEP and BGP-LS messages are considered). The proposed BGP-LS PROACTIVE scheme generates the lowest control traffic thus guaranteeing an increased scalability of the controller also in case of dynamic traffic. Conversely the PCEP LABELSET scheme generates the highest control traffic because, for each path computation, a high number of PCEP messages between pPCE and cPCEs are exchanged.

Figure 36 shows the mean LSP setup time. Simulations consider message propagation, transmission and queuing times, typical processing time of control messages (i.e., 10 □s for packets that are just forwarded, 2 ms for packets requiring a local processing), typical SSS cross connection time in EONs (i.e., 100 ms), and typical path computation time (i.e., 10 ms). The figure shows that the two schemes based on BGP-LS achieve the same result and reduce the PCEP LABELSET LSP setup time. Indeed, using BGP-LS the pPCE immediately performs the path computation upon reception of the request without requiring additional PCEP communication with cPCEs.

Figure 37 shows the LSP blocking probability. At low loads, blocking during backward RSVP-TE signalling phase dominates, in this phase the BGP-LS PROACTIVE scheme significantly reduces the blocking. Indeed, by proactively updating the H-TED, the BGP-LS PROACTIVE scheme reduces the probability of resource contentions during the RSVP-TE signalling. At higher loads, the three schemes achieve similar blocking.

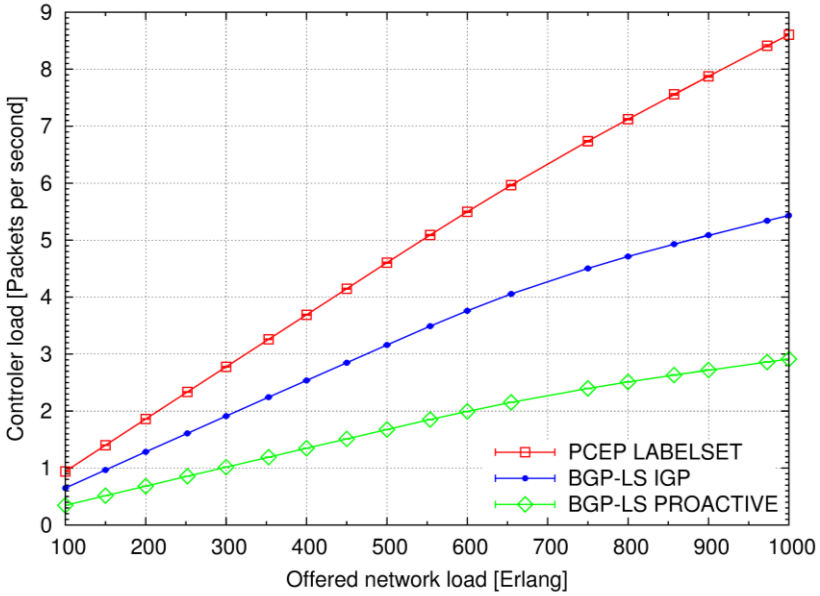

Figure 35: Controller load.

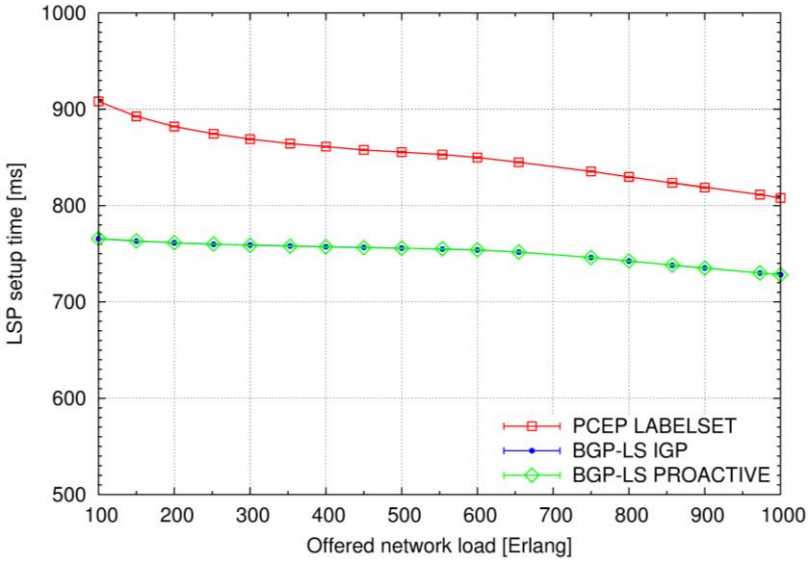

Figure 36: LSP setup time.

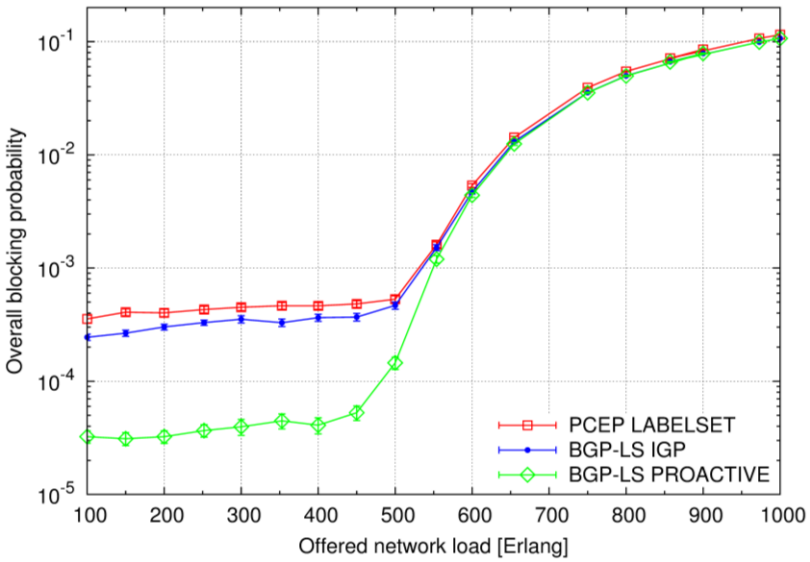

Figure 37: LSP blocking probability.

5.4 Concluding remarks

In this section, the use of the Hierarchical PCE architecture in EONs with a GMPLS/PCE control plane is considered.

A novel scheme to update the H-TED using BGP-LS is proposed.

Simulations have been performed to evaluate the pPCE control load, the LSP setup time, and the LSP blocking probability. Results have shown that the proposed scheme reduces the pPCE control load and achieves lower blocking with respect to standard PCEP and BGP-LS schemes.

6 In-Operation planning

In this section we use the ABNO-based control plane architecture, where an advanced and innovative PCE architecture composed by an active stateful PCE system has been designed and implemented in REACTION. It consists of an active stateful front-end PCE spitted into two elements: the front-end PCE is in charge of

computing RSA and elastic spectrum allocations or provisioning, while the back-end PCE is responsible for performing complex network re-optimization actions, e.g. for de-fragmentation purposes (see Figure 38). BGP-LS and PCEP protocols are used to synchronize TED and LSP-DB, respectively.

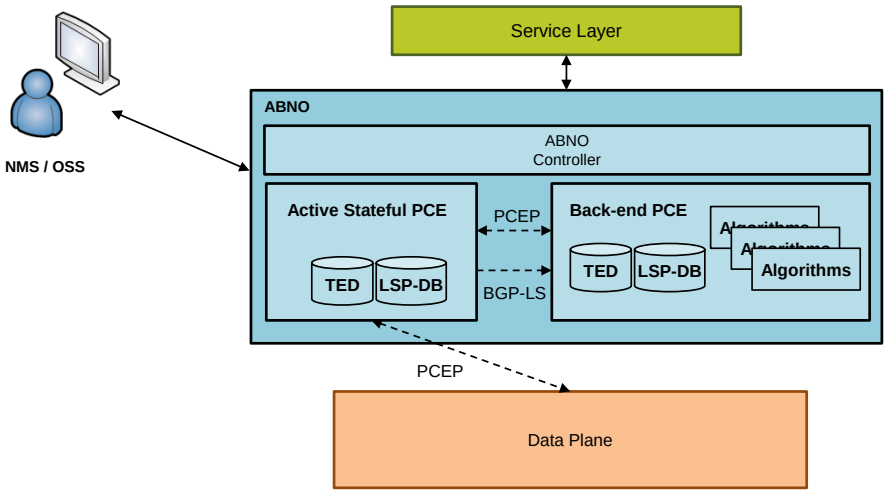

Figure 38: Architecture for In-Operation Planning.

In the event of an incoming request arriving to the front-end PCE, it decides whether the request can be computed using one of the local algorithms or it needs to be redirected towards a back-end PCE running specialized algorithms; in the latter, the front-end PCE relays on the parent PCE who looks for the most appropriated back-end PCE to redirect the request. PCEP is used for the communication between front and back end PCEs. The back-end PCE includes an active solver capable of solving a number of complex network optimization algorithms involving a number of different media channels, from which some could need to be set-up/teardown/modify.

The REACTION control plane architecture is here applied in the context of the in-operation network planning use case. In-operation network planning consists in making network resources available by reconfiguring and/or re-optimizing the flexgrid network on demand and in real-time [Vel-CM14].

6.1 After failure repair optimization

For illustrative purposes, Figure 39 reproduces a small flexgrid network topology where several connections are currently established; in particular, the route of connections P1 and P2 is shown. The spectrum usage is also provided in the figure, where the spectrum allocation for all five established connections is specified.

Three snapshots with the state of the network are shown: the link 6-7 has failed in Figure 39a; multipath restoration has been applied in Figure 39b, and connection P2 has been split into two parallel sub-connections P2a and P2b squeezing the total conveyed bitrate to fit into the available free spectrum slots; failed link 6-7 has been repaired in Figure 39c, and re-optimization has been performed by solving multipath after failure repair optimization (MP-AFRO), so that sub-connections have been merged back and bitrates have been expanded to its originally requested values.

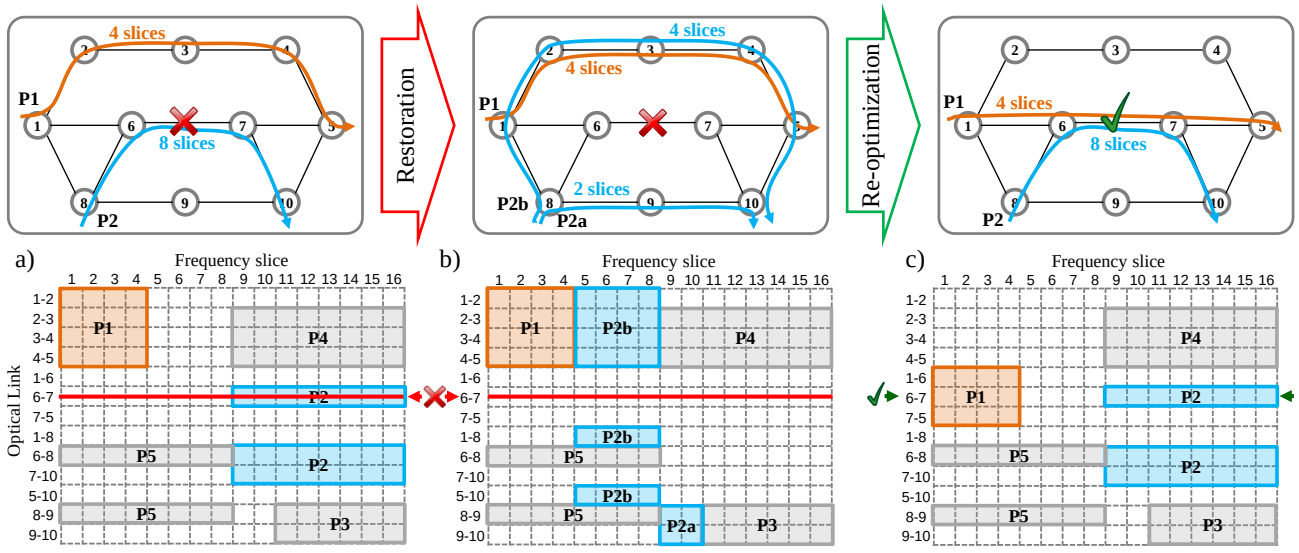

Figure 39: An example of multi-path restoration and after failure repair optimization. 6.25 GHz frequency slices are used.

In view of the example, it is clear that multipath restoration allows increasing restorability, in particular when no enough contiguous spectrum can be found along a single path, as happened when restoring P2. Nonetheless, this benefit is at the cost of an increased resource usage, not only as a result of using (not shortest) parallel routes, and squeezing the total conveyed connection's bitrate, but also because the spectral efficiency is degraded when connections are split. For instance, a 400 Gb/s aggregated flow can be conveyed on one single 100 GHz connection or on four parallel 37.5 GHz sub-connections, therefore using 50% more spectral resources even in the case of being routed through same links.

For this reason, resource utilization can be improved by applying MP-AFRO, i.e. by re-routing established connections on shorter routes, by merging parallel sub-connections to achieve better spectrum efficiency, and expanding the conveyed connections' bitrates to its original ones. Figure 39c illustrates an example of such re-optimization, where connection P1 has been rerouted using a shorter route that includes the repaired link, whilst sub-connections P2a and P2b have been merged on a single connection conveying the originally requested bandwidth. Note however that although those operations might entail short traffic disruption, it can be minimized using the standardized make-before-break rerouting technique [RFC3209], as revealed in [Rui-OFC14].

6.1.1 After failure repair optimization with Multipath merging (MP-AFRO)

In this section the MP-AFRO problem is formally stated and a MILP formulation to solve it is presented. After analyzing the complexity of the MILP model, a heuristic algorithm is presented to provide near-optimal solutions in the required computation times.

Problem statement

The MP-AFRO problem can be formally stated as follows:

- Given: a) a network topology G(N, E) defined as a set of optical nodes N and a set of fiber links E; b) an optical spectrum divided into frequency slices of a given width; c) the set of slices used by non-selected connections; d) a set of demands candidate for re-optimization, D.

- Find: the route and spectrum allocation for demands in D, merging sub-connections serving each original connection.

- Objective: maximize the bitrate served while minimizing the total number of sub-connections used to convey the traffic served.

An MILP-based model is presented in next section.

MILP Formulation

The MILP model is based in a link-path formulation for RSA [Vel-JLT14] where a set of routes is computed for each of the candidate demands to be re-optimized.

The following sets and parameters have been defined:

E	Set of network links, index e.
S	Set of slices in the spectrum of each link, index s.
D	Set of candidate demands, index d. Each demand d represented by tuple $<s_d, t_d, b_d>$, where s_d is the source node, t_d is the target node, and b_d is the requested bitrate.
$P(d)$	Set of sub-connections being used to serve d.
$K(d)$	Set of pre-computed routes for demand d, index k.
$C(d)$	Set of slots for demand d, index c.
R	Set of pairs reach-modulation format, index r.
δ_{ke}	1 if route k contains link e; 0 otherwise.
α_{es}	1 if slice s in link e is used by any already established connection not in D; 0 otherwise.
γ_{cs}	1 if slot c contains slice s; 0 otherwise.
β	Objective function weight.
$b(c, r)$	Maximum bitrate that can be conveyed using slot c with the modulation format given by pair r.
$b(k, c)$	Maximum bitrate that can be conveyed through route k using slot c.
$len(r)$	Reachability of pair r (in Km).
$len(k)$	Length of route k (in Km).

The decision variables are:

x_{dkc}	Binary. 1 if route k and slot c are used to serve demand d; 0 otherwise.
y_{dkr}	Binary. 1 if pair r is used for demand d in route k; 0 otherwise.
z_{dkc}	Positive Real. Served bitrate for demand d through route k and slot c.
w_d	Positive Integer. Total number of sub-connections used to serve demand d.

The $K(d)$ set is computed using eq. (18) as the union between the already-in-use paths $P(d)$ and the set of all shortest paths between s_d and t_d of the same length in hops as the shortest one, using the repaired link e.

$$K(d) = P(d) \cup \left\{ \begin{matrix} r \in KSP(s_d, t_d), e \in r \wedge \\ |r| = |SP(s_d, t_d)| \end{matrix} \right\} \quad \forall d \in D \tag{18}$$

The MILP formulation for the MP-AFRO is as follows:

$$Max \quad \Phi = \sum_{d \in D} \left(\frac{\beta}{b_d} \sum_{k \in K(d)} \sum_{c \in C(d)} z_{dkc} - \frac{1}{|P(d)|} w_d \right) \tag{19}$$

subject to:

$$\sum_{k \in K(d)} \sum_{c \in C(d)} x_{dkc} = w_d \quad \forall d \in D \tag{20}$$

$$\sum_{d \in D} \sum_{k \in K(d)} \sum_{c \in C(d)} \delta_{ke} \cdot \gamma_{cs} \cdot x_{dkc} \leq (1 - \alpha_{es}) \quad \forall e \in E, s \in S \tag{21}$$

$$w_d \leq |P(d)| \quad \forall d \in D \tag{22}$$

$$\sum_{k \in K(d)} \sum_{c \in C(d)} z_{dkc} \leq b_d \quad \forall d \in D \tag{23}$$

$$z_{dkc} \leq b_d \cdot x_{dkc} \quad \forall d \in D, k \in K(d), c \in C(d) \tag{24}$$

$$z_{dkc} \leq \sum_{r \in R} b(c, r) \cdot y_{dkr} \quad \forall d \in D, k \in K(d), c \in C(d) \tag{25}$$

$$\sum_{c \in C(d)} len(k) \cdot x_{dkc} \leq \sum_{r \in R} len(r) \cdot y_{dkr} \quad \forall d \in D, k \in K(d) \tag{26}$$

$$\sum_{r \in R} y_{dkr} = 1 \quad \forall d \in D, k \in K(d) \tag{27}$$

Objective function in eq. (19) maximizes the total served bitrate while minimizing the amount of sub-connections used to serve that bitrate. Constraint (20) accounts for the number of sub-connections used to serve each demand. Constraint (21) ensures that any single slice is used to convey one sub-connection at last, provided that it is not already used by other demand not in D. Constraint (22) guarantees that sub-connections count to serve any specific demand is not increased. Constraint (23) assures that served bitrate does not exceeds demand's requested bitrate. Constraint (24) sets to zero bitrate of unused sub-connections. Constraint (25) limits the bitrate conveyed by any specific sub-connection to the maximum associated to pair r, whereas constraint (26) limits the length of each used sub-connection to the reachability associated to pair r. Finally, constraint (27) selects one pair r for each sub-connections.

Regarding complexity, the MP-AFRO problem is NP-hard since simpler network routing problems have been proved to be NP-hard. Regarding its size, the number of variables is $O(|D| \cdot |K(d)| \cdot (|C(d)| + |R|))$ and the number of constraints is $O(|D| \cdot |K(d)| \cdot |C(d)| + |E| \cdot |S|)$.

Although the MILP model can be solved in a short period of time, e.g. minutes using a standard solver such as CPLEX or dozens of seconds if a column generation algorithm is used, during MP-AFRO computation new connections could arrive, which need to be queued and delayed until the MP-AFRO solution is implemented in the network. Aiming at reducing provisioning delay, providing a good trade-off between complexity and

optimality, a heuristic algorithm is presented next to solve the MP-AFRO in the stringent required times (e.g. < 1s.).

Heuristic Algorithm

The MP-AFRO algorithm is shown in Table 3. The algorithm maximizes the amount of bitrate that is served (note that only part of the original bitrate could be restored), while minimizing the number of sub-connections per demand being re-optimized.

The algorithm receives as input the network's TED, i.e. *G*(*N*, *E*), the candidate list of demands to be re-optimized *D,* and the maximum number of iterations *maxIter*; it returns the ordered set *bestS* containing the best solution found, where the list of sub-connections to serve each demand is specified.

All the sub-connections of each demand are first de-allocated from of the original TED in *G* (line 2) and then, the algorithm performs a number of iterations (*maxIter*), where the set *D* is served in a random order (lines 3-5). At each iteration, a copy of the original TED is stored into an auxiliary TED *G'*, so that every subsequent operation is performed over *G'*. Each iteration consists of two steps performed sequentially. In the first step (lines 6-12), the set of demands is served ensuring the currently served bitrate in the hope of finding a shorter route. The *getMP_RSA* function computes the set of sub-connections consisting of the routes with the highest available capacity, using eq. (28), and minimum cost and selects those to serve the required bitrate (line 8). A solution is feasible only if every demand obtain at least the same bitrate than the current allocation so in case the required bitrate cannot be served, the solution is discarded (lines 9-10), otherwise the resources selected are reserved in the auxiliary TED *G'* (line 12). In the second step, the bitrate of the demands is increased firstly by allocating wider slots along the same routes (lines 17-21) and then by adding more sub-connections.

$$c_k(d) = \left\{ c^* \;\middle|\; \begin{array}{l} b(k,c^*) \geq b(k,c)\ \forall c \in C(d) \\ \alpha_{es} = 0\ \forall e \in k, s \in S(c) \end{array} \right\} \tag{28}$$

Table 3: MP-AFRO Heuristic algorithm

IN: *G*(*N*, *E*), *D*, *maxIter*	
OUT: *bestS*	
1:	*bestS* ← *D*
2:	**for each** *d* **in** *D* **do** deallocate(*G*, *d.SC*)
3:	**for** *i* = 1..*maxIter* **do**
4:	*S* ← Ø; *G'* ← *G; feasible* ← true
5:	sort(*D*, random)
6:	**for each** *d* **in** *D* **do**
7:	*d.SC'* ← Ø; *d.reoptBw* ← 0
8:	{<*k*, c_k>} ← getMP_RSA(*d*, *d.servedBw*)
9:	**if** b({<*k*, c_k>}) < *d.servedBw* **then**
10:	*feasible* ← false; **break**
11:	*d.reoptBw* ← b({<*k*, c_k>}); *d.SC'* ← {<*k*, c_k>}
12:	allocate(*G'*, *d.SC'*); *S* ← *S* ∪ {*d*}
13:	**if not** *feasible* **then**
14:	**for each** *d* **in** *S* **do** deallocate(*G'*, *d.SC'*)
15:	**continue**
16:	**for each** *d* **in** *D* **do**
17:	**if** *d.reoptBw* < *d.bw* **then**

```
18:          {<k, c_k>} ← expandSlots (d, d.bw, G')
19:          if b({<k, c_k>}) > d.reoptBw then
20:             d.reoptBw ← b({<k, c_k>})
21:             d.SC' ← {<k, c_k>}; continue
22:          if d.reoptBw<d.bw & |d.SC'|< |d.SC| then
23:             {<k,c_k>}←getMP_RSA(d, d.bw-d.reoptBw)
24:             if b({<k, c_k>}) = 0 then break
25:             d.reoptBw ← d.reoptBw + b({<k, c_k>})
26:             d.SC' ← d.SC' ∪ {<k, c_k>}
27:             allocate(G', d.SC')
28:       Compute Φ(S)
29:       for each d in D do deallocate(G', d.SC')
30:       if Φ(S) > Φ(bestS) then bestS ← S
31:    for each d in D do allocate(G, d.SC)
32:    return bestS
```

After a complete solution is obtained, its fitness value is computed (line 28). Next, the used resources in the solution are released from the auxiliary TED (line 29). The fitness value of the just obtained solution is compared to that of the best solution found so far (line 30) and stored provided that it is feasible and its fitness value is better than that of the incumbent. The state of *G* is restored and the best solution is eventually returned (lines 31-32).

In the next section, we focus on devising the workflow that needs to be carried out in the control plane to be able to request solving the MP-AFRO problem and deploy the obtained solution.

6.2 Proposed optimization workflow

To deal with network re-optimization, we consider a control plane based on the ABNO architecture [ABNO, Agu-OFC14], which includes an fPCE responsible for computing provisioning requests and dealing with network data plane, and a bPCE capable of performing complex computations to solve optimization problems.

We assume that an operator in the Network Management System (NMS) triggers the MP-AFRO workflow after a link has been repaired. To that end, the NMS issues a service request towards the ABNO Controller. Figure 40 reproduces the re-optimization sequence diagram involving ABNO components and its execution flow diagram. When the request from the NMS arrives at the ABNO controller, re-optimization is requested by sending a PCReq message (labeled as 1 in Figure 40) to the fPCE. Upon receiving the request, the fPCE collects relevant data to be sent to the bPCE in the form of a PCReq message containing a Global Concurrent Optimization (GCO) request (2).

In light of the MP-AFRO problem statement, and assuming that the network topology and the current state of the resources has been synchronized, the information to be included in the GCO request is the set of connections candidate for re-optimization, D. Therefore, an algorithm to find the candidate connections in the LSP-DB is needed.

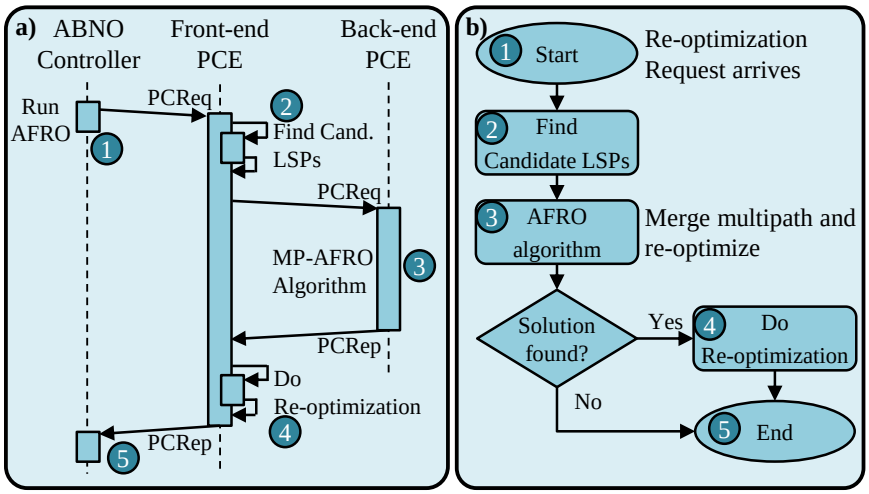

Figure 40: Re-optimization sequence (a) and flow (b) diagrams

The algorithm, presented in Table 4, receives as input the network's TED in G, the set L with the LSP-DB, and the repaired link e. The connections whose shortest path traverses the repaired link are selected and the candidate list of demands D to be re-optimized is eventually returned.

Table 4: Candidate demands algorithm

IN: *G(N, E)*, *L*, *e*	
OUT: *D*	
1:	$D \leftarrow \emptyset$
2:	**for each** $d \in L$ **do**
3:	$\{SP\} \leftarrow$ KSP $(d.s, d.t, G)$
4:	**for each** $SP \in \{SP\}$ **do**
5:	**if** $e \in SP$ **then** $D \leftarrow D \cup \{d\}$
6:	**return** D

Information regarding the candidate connections is sent in a PCReq message containing the GCO request. Each candidate connection is sent as an individual request, identified by a RP object; the end points and original bandwidth are also included using the END-POINTS and the BANDWIDTH objects, respectively. In addition, for each individual sub-connection related to the original connection, its current route and spectrum allocation are specified using a RRO (Record Route Object) object. The RRO object interleaves the route's nodes with its spectrum allocation. The sequence of nodes is encoded using the Unnumbered Interface sub-objects encoding each node and the corresponding outgoing port identifiers to reach the next node; and an IPv4 Prefix sub-object to identify the destination node.

The spectrum allocation is encoded using the Label Control sub-object that contains the tuple {n, m} that unambiguously defines any frequency slot, where n is the central frequency index (positive, negative or 0) of the selected frequency slot from a reference frequency (193.1 THz), whereas m is the slot width in number of slices at each side of the central frequency. Aiming at finding an optimal solution for the entire problem, individual requests are grouped together using a SVEC object.

The desired network-wide GCO related criterion, such as “MP-AFRO”, is specified by means of the OF object. Finally, the repaired link should/must be used for re-optimization. To that end, we extended current standards adding an IRO (Include Route Object) object that identifies those repaired link that should be included into the new routes in a symmetric way with respect to the XRO (Exclude Route Object) object that specifies the links that should be excluded from routed being computed.

Upon receiving the PCReq, the bPCE runs the specified heuristic algorithm. The solution of the MP-AFRO problem is encoded in a PCRep (labeled as 3 in Figure 40); each individual request is replied specifying the bitrate that could be served and the route and spectrum allocation of the connections related to each request in a list of ERO objects. Note that the solution might entail merging several existing sub-connections to create one or more new connections. The Order TLV is included in RP objects to indicate the order in which the solution needs to be implemented in the data plane.

Upon receiving the PCRep message with the solution from the bPCE, the fPCE runs the Do-Optimization algorithm (labeled as 4 in Figure 40), listed in Table III, to update the connections in the network’s data plane. The algorithm sorts D in the same order than S (line 1 in Table 3). Next, each demand d in D is sequentially processed. The possibly-updated demand ds in S corresponding to the current demand d in D is taken (line 3) by matching them using the identifier stored in its related RP object. If d differs from ds, the bPCE has found a better set of sub-connections for the demand, so the fPCE should send the corresponding PCUpd/PCInit messages towards source GMPLS controllers (line 4) to update/implement it [St-PCE, St-PCEP].

When the solution has been completely implemented i.e., when the corresponding PCRpt messages are received, the fPCE replies the completion of the requested operation to the ABNO controller using a PCRep message (labeled as 5 in Figure 40), which eventually informs the NMS.

Table 5: Re-optimization algorithm

IN: *D, S, order*	
1:	sort(*D*, *S.order*)
2:	**for each** *d* **in** *D* **do**
3:	*ds* ← getDemand(*S*, *d*)
4:	**if** *d* ≠ *ds* **then** update(*d*, *ds*)

Table 6: Bit-rate spectrum width

Bitrate (Gb/s)	Bandwidth (GHz)	#Slices
100	37.5	6
200	62.5	10
400	100	16

6.3 Performance evaluation

To evaluate the performance of MP-AFRO, we use two representative core network topologies: the 30-node Spanish Telefonica (TEL) and the 28-node European (EON) networks. We consider the fibre links with the spectrum width equal to 2 THz, divided into frequency slices of 12.5 GHz and granularity 6.25 GHz.

The MP-AFRO heuristic was developed in C++ and integrated in an ad-hoc event driven simulator based on OMNET++. Connection requests are generated following a Poisson process and are torn down after an exponentially distributed holding time with mean equal to 6 months. The source and destination nodes are randomly chosen using the uniform distribution.

The bitrate of each connection request was randomly selected considering 80% of the connections being 100 Gb/s and the other 20% of 400 Gb/s. To convert bitrate into spectrum width, we use the correspondence in Table IV. Finally, note that each point in the results is the average of 10 runs with 100,000 connection requests each. Finally, we assume a link failure rate of 2.72 10^{-3} per km per year [Gro-04] and consider that the link is repaired immediately after the restoration process has ended.

Aiming at comparing the performance when the MP-AFRO is used, Figure 41 plots the blocking probability (Pb) against the offered load with and without applying MP-AFRO. Pb is weighted using connections' bitrate and the load is normalized to the largest load unleashing Pb < 4%. Although the actual load gain depends, among other factors, on the characteristics of network topology considered, gains above 13% can be obtained using MP-AFRO.

Once the performance of MP-AFRO has been evaluated, the proposed workflow is experimentally validated next.

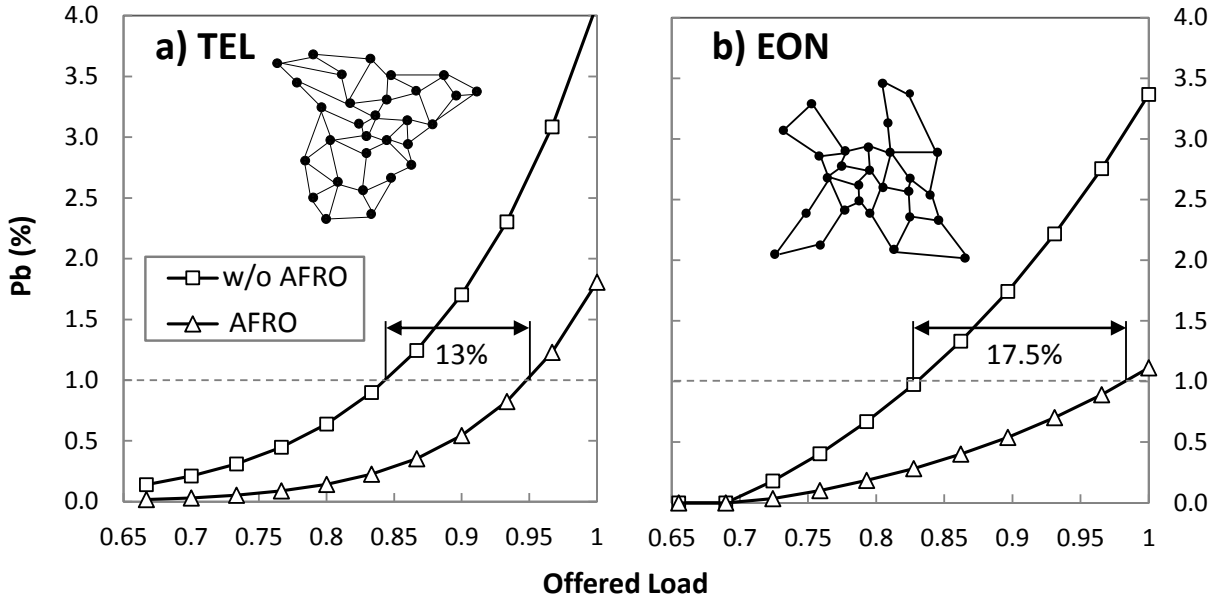

Figure 41: Blocking probability against normalized offered load for the TEL (a) and EON (b) network topologies.

6.4 Experimental assessment

For the experiments, we consider the network topology illustrated in Figure 39. The data plane includes programmable spectrum selective switches (SSS) and node emulators are additionally deployed to complete the topology. Nodes are handled by co-located controllers running RSVP-TE with flexgrid extensions. The controllers run a proprietary configuration tool for automatic filter re-shaping with a resolution of 1GHz. Controllers communicate with the fPCE by means of PCEP through Gigabit Ethernet interfaces.

Figure 42 depicts the distributed field trial set-up connecting premises in Telefonica (Madrid, Spain), CNIT (Pisa, Italy), and UPC (Barcelona, Spain) through IPSec tunnels where experiments have been carried out. Telefonica's ABNO controller was implemented in Java, CNIT's fPCE as well as UPC's bPCE were implemented in C++ for Linux. All components communicate by exchanging PCEP messages.

A link failure triggered a multipath restoration algorithm that rerouted the failed connections splitting them into multiple sub-connections per demand reducing its spectral efficiency and consuming higher amount of optical resources. After the failed link has been repaired, the NMS operator decided to trigger the MP-AFRO algorithm to restore the network efficiency, so the NMS requests the computation to the ABNO controller (IP: 172.16.104.2), who in turn sends the PCReq message (message 1 in Figure 42a) to the fPCE (172.16.101.3) containing the failed link identification by means of an IRO object, and an OF object to specify the algorithm to be executed, in this experiment the MP-AFRO.

When the fPCE receives the ABNO request, it runs the Find Candidate Demands algorithm using as repaired link the one encoded into the received IRO object, to obtain the candidate list of demands to be re-optimized, and a PCReq to be sent to the bPCE (IP: 172.16.50.2) is composed.

This PCReq message encodes the candidate demands using RP objects to identify them, an END-POINTS object defining its source and target nodes, a BANDWIDTH object to specify the requested bandwidth, and one RRO object for each sub-connection being used by the demand. Included candidate demands are grouped by means of a SVEC object so that they are re-optimized jointly. The received OF and IRO objects are also included in the PCReq message to inform to the bPCE on which algorithm should be executed and which link has been repaired respectively (message 2).

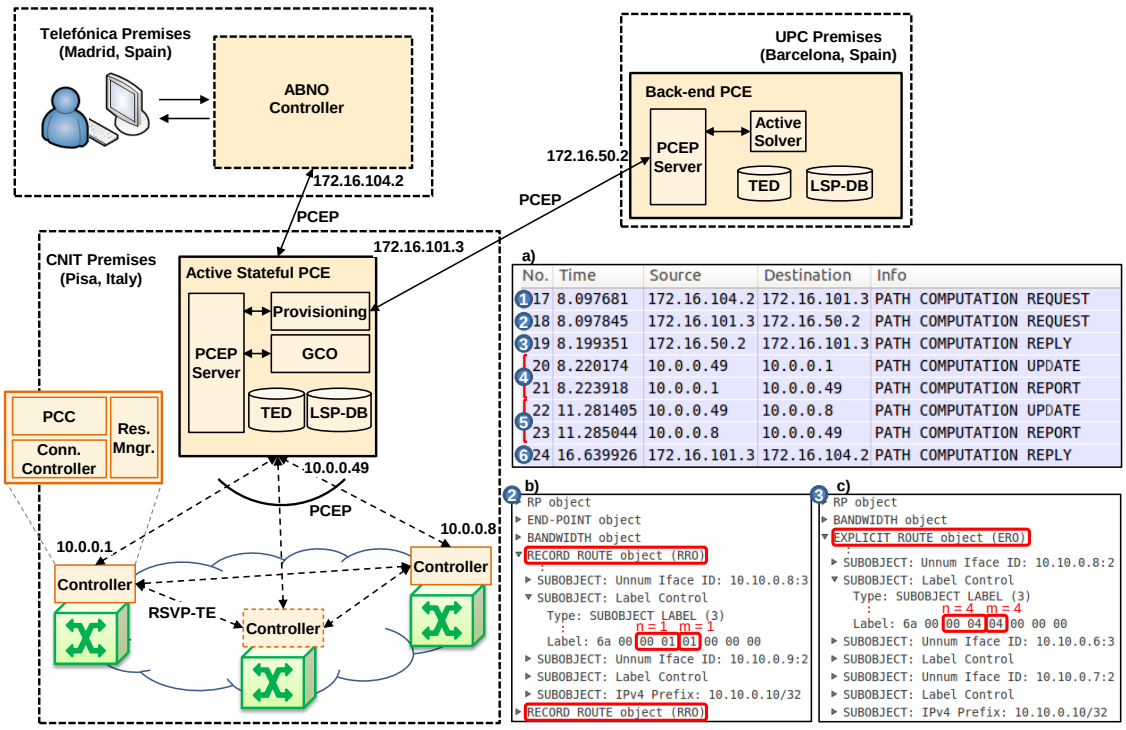

Figure 42: Distributed field trial set-up and exchanged messages.

Figure 42b details the PCReq message received by the bPCE highlighting the RRO objects encoding the original sub-connections. The first RRO object and one of the Label Control sub-objects has been expanded to identify its original sub-connection's route and frequency slot. Objects in Figure 42b correspond to demand P2a and P2b in Figure 39b, composed by two sub-connections. P2a traverses nodes 8-9-10 using frequency slot <n=-2, m=2>, and P2b traverses nodes 8-1-2-3-4-5-10 using frequency slot <n=1, m=1>.

After receiving the PCReq message, the bPCE computes the MP-AFRO algorithm and responds using a PCRep message (message 3) containing the solution found. For each demand, the received RP object is duplicated and included into the PCRep message, adding also a BANDWIDTH object specifying the served bandwidth, and one ERO object for each sub-connection to be established to serve the demand. The ERO objects have the same format than the RRO objects.

Figure 42c details the PCRep message replied by the bPCE highlighting the ERO objects encoding the re-optimized sub-connections. In contrast to Figure 42b, objects in Figure 42c correspond to demand P2, the result of merging P2a and P2b, in Figure 39b. The new computed connection traverses nodes 8-6-7-10 using frequency slot <n=-4, m=4>. The resulting connection has restored the initially failed connection using the repaired link.

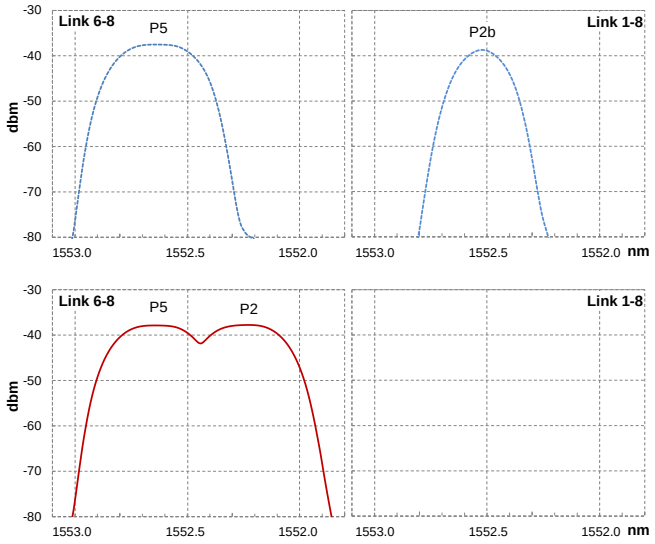

Figure 43: Optical spectrum before (top) and after MP-AFRO (bottom) in links 6-8 and 1-8.

Upon PCRep message arrival, the fPCE starts updating connections by sending PCUpd messages towards the involved GMPLS controllers (IPs: 10.0.0.X). Data plane nodes are located in network 10.10.0.X. The controllers operate on a flexgrid test-bed derived from [Cug-JLT12], including 100 Gb/s polarization multiplexed quadrature phase shift keying (PM-QPSK) optical signals and bandwidth variable cross-connects. Figure 43 shows the optical spectrum on links 6-8 and 1-8 before and after the re-optimization is performed. PCRpt responses are generated when the requested actions have been performed (messages 4-5). When the solution computed by the bPCE has been completely deployed, the fPCE confirms the ABNO controller the end of the requested re-optimization by sending a PCRep message (6).

The overall re-optimization was successfully completed in around eight seconds, including circa 150 ms of pure control plane contribution (i.e, messages exchange and path computation algorithms at fPCE and bPCE). Note that around 1 s is required to tune each SSS.

6.5 Concluding remarks

In this study, the REACTION solutions have been presented and applied on a use case of in-operation network planning. When a link fails, multipath restoration can be used to increase restorability of affected connections at the cost of worse resource utilization and spectral efficiency. After the link is repaired, the multipath after failure repair optimization (MP-AFRO) problem can be used to aggregate multiple sub-connections serving a single demand using shorter routes, thus releasing spectrum resources that now can be used to convey new connection requests. The MP-AFRO was modelled as using a MILP formulation and a heuristic algorithm was devised to find good feasible solutions in practical computation times.

After evaluating its performance on an ad-hoc network simulator, this use case of in-operation network planning was experimentally demonstrated for the first time on a distributed test-bed connecting premises in Telefonica, CNIT, and UPC. After a link was repaired, network re-optimization was requested from the NMS.

The ABNO architecture controlled a flexgrid-based optical network, where the PCE architecture consisted of an fPCE and a bPCE. The ABNO controller is in charge of initiating the MP-AFRO workflow, requesting re-optimization to the fPCE, which delegates complex computations to the bPCE. The relevant PCEP messages were shown and its contents analysed. Note that since network dynamicity frequently derives into not optimal use of resources, this use case can easily be extended to be triggered by any other event.

7 Use case of NREN evolution from fixed to flexi-grid networks

7.1 Introduction

In this section, the use case of the network evolution of the UNINETT NREN is considered.

The current network topology is shown in Figure 44.

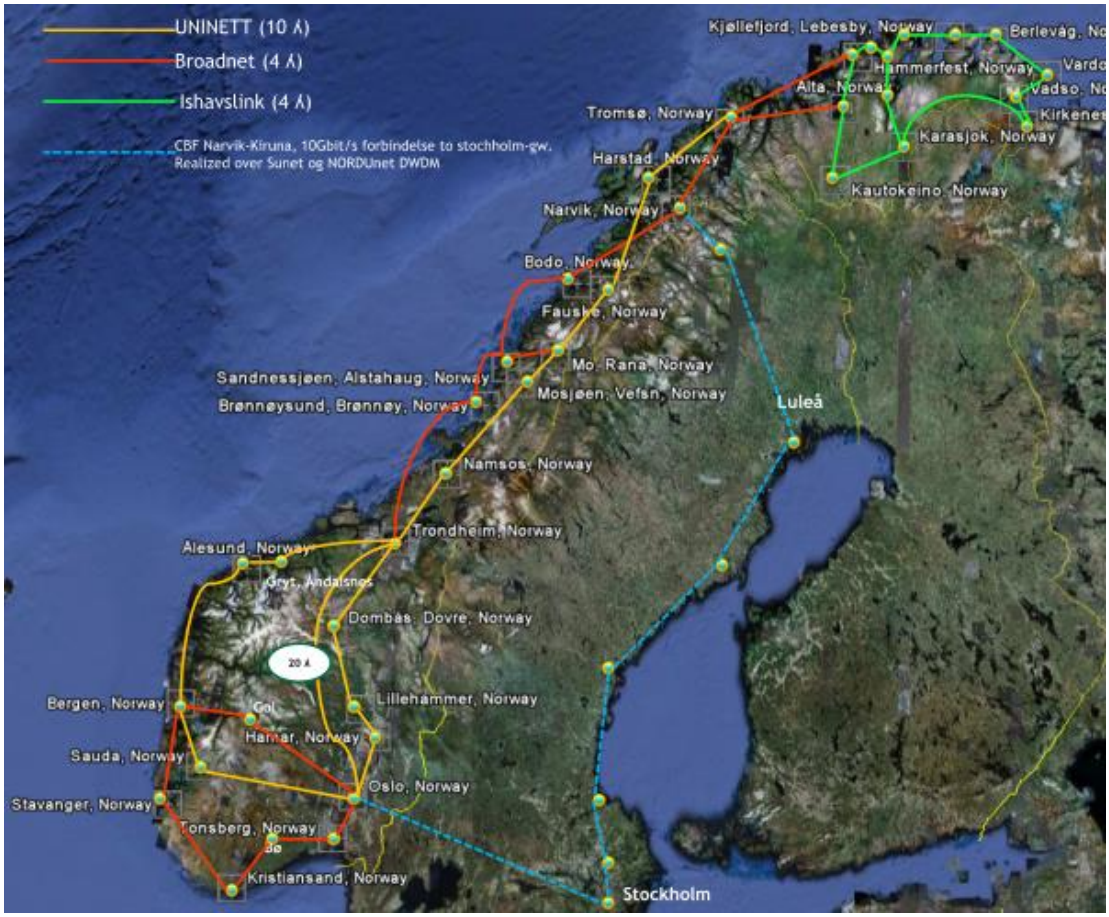

Figure 44: Current UNINETT Network

The current network includes point-to-point WDM links. Traffic is typically electronically terminated in the most relevant network nodes while nodes introducing a limited amount of traffic are typically equipped with Fixed optical add-drop multiplexers (OADM). The current status of the network shows a network utilization of up to 40 wavelengths, each operated at 10Gb/s. A traffic growth of around 30% per year has been experienced in the last years and it I expected to continue in the near future.

Given this pace, the 10Gb/s based WDM technology will soon exhaust the available spectrum resources. For this reason, 100Gb/s line cards are considered for provisioning new traffic requests (the setup of the first 100Gb/s lightpath has been recently completed).

In addition, the introduction of ROADM technologies where optical bypass is experienced in intermediate nodes is also considered.

In this study, the UNINETT network is fully re-designed, considering 100Gb/s ROADM-based technologies. In particular, the scalability performance is assessed by evaluating the fiber exhaustion time, in the cases where either the fixed and flexible grids are applied.

7.2 Network scenario

To assess the evolution of the UNINETT network, a scenario focusing on the most relevant network nodes and traffic demands has been considered.

The considered network topology is shown in Figure 45, while Figure 46 reports the traffic matrix derived from the current network resource utilization.

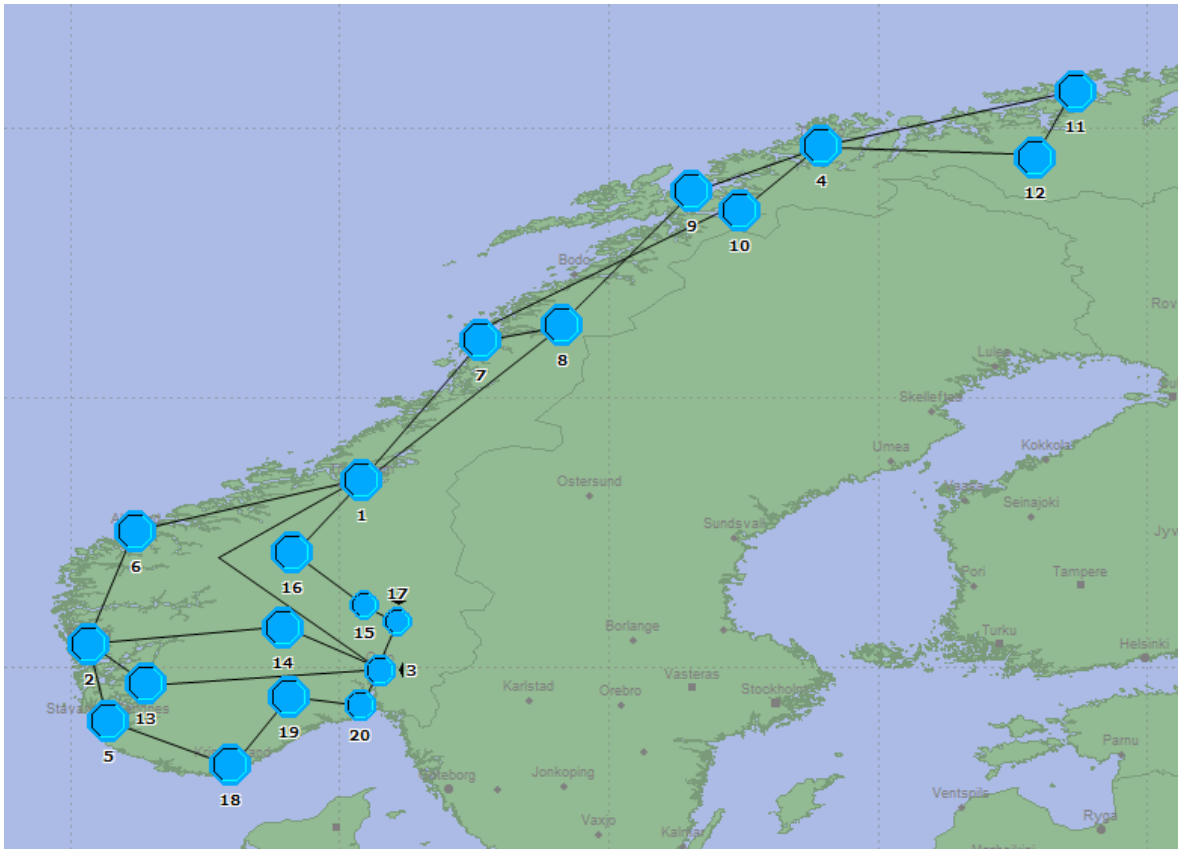

Figure 45: Considered UNINETT network topology

TRAFFIC MATRIX (Gbps)																					
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM / TO
1	0	175	490	98	49	1.8	1.8	1.8	1.8	1.8	3	4.4	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1
2	175	0	196	98	49	1.8	1.8	1.8	1.8	1.8	3	4.4	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	2
3	490	196	0	98	49	1.8	1.8	1.8	1.8	1.8	3	4.4	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	3
4	98	98	98	0	49	1.8	1.8	1.8	1.8	1.8	3	4.4	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	4
5	49	49	49	49	0	1.8	1.8	1.8	1.8	1.8	3	4.4	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	5
6	1.84	1.84	1.84	1.84	1.8	0	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	6
7	1.84	1.84	1.84	1.84	1.8	1.8	0	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	7
8	1.84	1.84	1.84	1.84	1.8	1.8	1.8	0	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	8
9	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	0	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	9
10	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	0	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	10
11	3.01	3.01	3.01	3.01	3	1.8	1.8	1.8	1.8	1.8	0	3	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	11
12	4.41	4.41	4.41	4.41	4.4	1.8	1.8	1.8	1.8	1.8	3	0	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	12
13	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	0	1.8	1.8	1.8	1.8	1.8	1.8	1.8	13
14	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	0	1.8	1.8	1.8	1.8	1.8	1.8	14
15	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	0	1.8	1.8	1.8	1.8	1.8	15
16	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	0	1.8	1.8	1.8	1.8	16
17	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	0	1.8	1.8	1.8	17
18	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	0	1.8	1.8	18
19	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	0	1.8	19
20	1.84	1.84	1.84	1.84	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	0	20
FROM / TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO

Figure 46: considered traffic matrix in Gb/s at year 0

An overall amount of spectrum resources per link equal to 2THz is assumed to route the considered and future traffic demands. The remaining portion of the spectrum resources (where electronic grooming may also be exploited) are devoted to low rate traffic demands injected from/to less relevant network nodes. This portion of resources also accounts for possible unexpected changes in the traffic evolution.

7.3 Network evolution: 100Gb/s ROADM-based over fixed grid

The network scenario detailed in the previous section is considered in the design of the new generation of the UNINETT NREN.

ROADM technology is assumed, including transparent optical pass-through at intermediate nodes, given the constraint of optical reach and wavelength continuity. ROADMs are equipped with line cards of 100Gb/s over a fixed 50GHz grid spacing.

Figure 47 shows the percentage of occupied link spectrum resources at year 0.

LINK OCCUPATION PERCENTAGE																					
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM / TO
1			18			15	18	13								10					1
2					10	18							10	2.5							2
3	18												13	2.5			7.5			5	3
4									18	13	13	13									4
5		10																5			5
6	18	15																			6
7	13							0		18											7
8	18						0		13												8
9				13				18													9
10				18			13														10
11				13								2.5									11
12				13							2.5										12
13		13	10																		13
14		2.5	2.5																		14
15																7.5	10				15
16	7.5														10						16
17			10												7.5						17
18					5														5		18
19																		5		5	19
20			5																5		20
FROM / TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO

Figure 47: percentage of link utilization at year 0

Figure 48 shows the expected percentage of occupied link spectrum resources at year 5, considering the aforementioned traffic growth of 30% per year.

LINK OCCUPATION PERCENTAGE																					
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM / TO
1			45			48	35	20								33					1
2					18	48							38	7.5							2
3	45												38	7.5			33			7.5	3
4									35	20	13	13									4
5		18																7.5			5
6	48	48																			6
7	20							0		35											7
8	35						0		20												8
9				20				35													9
10				35			20														10
11				13								2.5									11
12				13							2.5										12
13		38	38																		13
14		7.5	7.5																		14
15																33	33				15
16	33														33						16
17			33												33						17
18					7.5														7.5		18
19																		7.5		7.5	19
20			7.5																7.5		20
FROM / TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO

Figure 48: percentage of link utilization at year 5

TRAFFIC MATRIX (Gbps)																					
FROM / TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM / TO
1	0	1207	3379	676	338	13	13	13	13	13	21	30	13	13	13	13	13	13	13	13	1
2	1207	0	1352	676	338	13	13	13	13	13	21	30	13	13	13	13	13	13	13	13	2
3	3379	1352	0	676	338	13	13	13	13	13	21	30	13	13	13	13	13	13	13	13	3
4	676	676	676	0	338	13	13	13	13	13	21	30	13	13	13	13	13	13	13	13	4
5	338	338	338	338	0	13	13	13	13	13	21	30	13	13	13	13	13	13	13	13	5
6	12.7	12.7	12.7	12.7	13	0	13	13	13	13	13	13	13	13	13	13	13	13	13	13	6
7	12.7	12.7	12.7	12.7	13	13	0	13	13	13	13	13	13	13	13	13	13	13	13	13	7
8	12.7	12.7	12.7	12.7	13	13	13	0	13	13	13	13	13	13	13	13	13	13	13	13	8
9	12.7	12.7	12.7	12.7	13	13	13	13	0	13	13	13	13	13	13	13	13	13	13	13	9
10	12.7	12.7	12.7	12.7	13	13	13	13	13	0	13	13	13	13	13	13	13	13	13	13	10
11	20.7	20.7	20.7	20.7	21	13	13	13	13	13	0	21	13	13	13	13	13	13	13	13	11
12	30.4	30.4	30.4	30.4	30	13	13	13	13	13	21	0	13	13	13	13	13	13	13	13	12
13	12.7	12.7	12.7	12.7	13	13	13	13	13	13	13	13	0	13	13	13	13	13	13	13	13
14	12.7	12.7	12.7	12.7	13	13	13	13	13	13	13	13	13	0	13	13	13	13	13	13	14
15	12.7	12.7	12.7	12.7	13	13	13	13	13	13	13	13	13	13	0	13	13	13	13	13	15
16	12.7	12.7	12.7	12.7	13	13	13	13	13	13	13	13	13	13	13	0	13	13	13	13	16
17	12.7	12.7	12.7	12.7	13	13	13	13	13	13	13	13	13	13	13	13	0	13	13	13	17
18	12.7	12.7	12.7	12.7	13	13	13	13	13	13	13	13	13	13	13	13	13	0	13	13	18
19	12.7	12.7	12.7	12.7	13	13	13	13	13	13	13	13	13	13	13	13	13	13	0	13	19
20	12.7	12.7	12.7	12.7	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	0	20
FROM / TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM / TO

Figure 49: considered traffic matrix in Gb/s at year 7

Figure 49 shows the considered traffic matrix at year 7. The figure shows that most of the traffic demands still refer to a relatively low amount of data traffic. On the other hand, there are several traffic demands that refer to a huge amount of traffic. if these demands are routed through a point-to-point WDM system traversing electronic routers in intermediate nodes, the throughput requirements, and in turn the cost of these routers, may be extremely high. On the other hand, if these demands are routed through a ROADM-based WDM infrastructure enabling optical bypass in intermediate nodes, a significant reduction in terms of electronic processing can be achieved. The latter case is investigated below.

Figure 50 shows the expected percentage of occupied link spectrum resources at year 7, considering the aforementioned traffic growth of 30% per year.

In this case, specific simulative studies have been also conducted to evaluate the cases of single link failures. At year 7, the overall amount of provisioned traffic is always successfully recovered.

Figure 51 shows the example of expected percentage of occupied link spectrum resources at year 7 upon the recovery process has been completed. In the figure, the recovery of failure affecting link 1-3 is reported.

LINK OCCUPATION PERCENTAGE																					
From \To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	From \To
1			75			75	50	33								58					1
2					25	75							63	13							2
3	75												63	13			58			20	3
4									50	33	13	13									4
5		25																20			5
6	75	75																			6
7	33							0		50											7
8	50						0		33												8
9				33				50													9
10				50			33														10
11				13								2.5									11
12				13							2.5										12
13		63	63																		13
14		13	13																		14
15																58	58				15
16	58														58						16
17			58												58						17
18					20														20		18
19																		20		20	19
20			20																20		20
From \To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	From \To

Figure 50: percentage of link utilization at year 7

LINK OCCUPATION PERCENTAGE																					
From \To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	From \To
1			0			90	65	18								98					1
2					33	90							50	25							2
3	0												45	30			98			15	3
4									60	23	13	13									4
5		33																15			5
6	90	90																			6
7	23							0		65											7
8	60						0		18												8
9				18				60													9
10				65			23														10
11				13								2.5									11
12				13							2.5										12
13		45	50																		13
14		30	25																		14
15																98	98				15
16	98														98						16
17			98												98						17
18					15														15		18
19																		15		15	19
20			15																15		20
From \To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	From \To

Figure 51: percentage of link utilization at year 7, upon failure recovery (failed link node1-node3)

As shown in Figure 51, the percentage of link occupation is extremely high on the most congested links.

At year 8, the further increase in traffic demands does not allow, in case of link failure, to fully recover the amount of provisioned traffic.

7.4 Network evolution: 100Gb/s ROADM-based over flexible grid

In this section, the design of the new generation of the UNINETT NREN is performed by considering the availability of the flexible grid. In this case, a 100Gb/s lightpath is assumed to require an amount of spectrum resources equal to 37.5GHz. Smaller frequency requirements may be achieved, for example, by applying high order modulation formats. However, the considered network is not expected to significantly exploit more spectrally-efficient transmission technique, given the typical large distances among network nodes.

LINK OCCUPATION PERCENTAGE																					
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO
1			56			56	38	24								43					1
2					19	56							47	9.4							2
3	56												47	9.4			43			15	3
4									38	24	9.4	9.4									4
5		19																15			5
6	56	56																			6
7	24							0		38											7
8	38						0		24												8
9				24				38													9
10				38			24														10
11				9.4								1.9									11
12				9.4							1.9										12
13		47	47																		13
14		9.4	9.4																		14
15																43	43				15
16	43														43						16
17			43												43						17
18					15														15		18
19																		15		15	19
20			15																15		20
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO

Figure 52: percentage of link utilization at year 7 in the case of flexible grid network

Figure 52 shows the percentage of link utilization at year 7 in the case of flexible grid. With respect to Figure 50 it is possible to notice the lower percentage of utilized resources. This can be noticed also in the case of recovery, as shown in Figure 53.

LINK OCCUPATION PERCENTAGE																					
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO
1			0			68	49	13								73					1
2					24	68							38	19							2
3	0												34	23			73			11	3
4									45	17	9.4	9.4									4
5		24																11			5
6	68	68																			6
7	17							0		49											7
8	45						0		13												8
9				13				45													9
10				49			17														10
11				9.4								1.9									11
12				9.4							1.9										12
13		34	38																		13
14		23	19																		14
15																73	73				15
16	73														73						16
17			73												73						17
18					11														11		18
19																		11		11	19
20			11																11		20
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO

Figure 53: percentage of link utilization at year 7, upon failure recovery (failed link node1-node3) in the case of flexible grid network

Figure 54 shows the percentage of utilized resources at year 8 and Figure 55 shows the related scenario when network failure is considered.

Results show that, differently with respect to the case of fixed grid, at year 8 the whole amount of provisioned traffic is successfully recovered.

LINK OCCUPATION PERCENTAGE																					
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO
1			71			73	47	28								54					1
2					24	73							60	11							2
3	71												60	11			54			17	3
4									47	28	9.4	9.4									4
5		24																17			5
6	73	73																			6
7	28							0		47											7
8	47						0		28												8
9				28				47													9
10				47			28														10
11				9.4								1.9									11
12				9.4							1.9										12
13		60	60																		13
14		11	11																		14
15																54	54				15
16	54														54						16
17			54												54						17
18					17														17		18
19																		17		17	19
20			17																17		20
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO

Figure 54: percentage of link utilization at year 8 in the case of flexible grid network

LINK OCCUPATION PERCENTAGE																					
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO
1			0			84	58	17								92					1
2					30	83							47	23							2
3	0												43	28			94			11	3
4									56	19	11	7.5									4
5		26																15			5
6	83	84																			6
7	19							0		58											7
8	56						0		17												8
9				17				56													9
10				58			19														10
11				9.4								3.8									11
12				9.4							1.9										12
13		43	47																		13
14		28	23																		14
15																94	92				15
16	94														92						16
17			92												94						17
18					11														15		18
19																		11		15	19
20			15																11		20
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO

Figure 55: percentage of link utilization at year 8, upon failure recovery (failed link node1-node3) in the case of flexible grid network

A further use case is addressed Figure 56. In this case, the design of the new generation of the UNINETT NREN is performed by considering, in addition to the availability of the flexible grid and of 100Gb/s transponders, the presence of transponders supporting super-channels at 400Gb/s. Lightpaths at 400Gb/s are assumed to require an amount of spectrum resources equal to 100Ghz (i.e., lower than four times 37.5GHz). The improved spectral efficiency of super-channels with respect to a combination of independent single-carrier transmissions is due to the absence of guard-bands and filtering effects among co-routed and contiguous sub-super-channel carriers.

Results show that, differently with respect to the case of fixed grid employing only 100Gb/s transponders, also at year 9 the whole amount of provisioned traffic is successfully recovered.

LINK OCCUPATION PERCENTAGE																					
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO
1			0			88	53	23								88					1
2					39	88							43	32							2
3	0												38	39			88			13	3
4									38	38	9.4	9.4									4
5		38																15			5
6	88	88																			6
7	38							0		53											7
8	38						0		23												8
9				23				38													9
10				53			38														10
11				5.6								7.5									11
12				13							3.8										12
13		38	43																		13
14		39	32																		14
15																88	88				15
16	88														88						16
17			88												88						17
18					13														15		18
19																		13		15	19
20			15																13		20
FROM \ TO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	FROM \ TO

Figure 56: percentage of link utilization at year 9, upon failure recovery (failed link node1-node3) in the case of flexible grid network exploiting 400Gb/s super-channels

7.5 Concluding remarks

The benefits of ROADM-based solutions exploiting 100Gb/s transmission systems (and beyond) have been presented in the context of the UNINETT NREN network. Both provisioning and recovery scenarios have been evaluated, considering the expected evolution of the NREN traffic matrix.

Results show that fiber exhaustion will occur after around 7 years from now in the case of fixed grid technologies, further postponed in the case of flexi-grid networks.

8 Conclusions

The REACTION project has designed and validated a flexible optical network scenario enabling software-controlled super-channel transmission.

Relevant enhancements have been introduced in the context of data plane architectures by considering the novel sliceable functionality, control plane and routing and spectrum assignment algorithms. In particular, the sliceable functionality has been carefully investigated, showing that, despite the introduced spectrum overutilization, a properly designed routing strategy can successfully increase the amount of established/recovered traffic. An innovative back-end/front-end PCE architecture has been then investigated, showing significant benefits in terms of network programmability and efficiency in the use of spectrum resources. Finally, advanced RSA algorithm and routing strategies have been proposed and successfully validated through both simulations and experiments.

Selected use cases have been specifically considered, covering multipath recovery with bitrate squeezing and in-operation network planning achieved through multipath re-routing after failure repair optimization. Simulative and experimental validations have shown that relevant improvements in terms of network resource utilization and network programmability can be achieved by applying the proposed REACTION solutions.

The expected evolution of the UNINETT NREN network has also been considered, showing through simulations that fiber exhaustion will occur after around 7 years from now, further postponed in the case of flexi-grid networks.

The REACTION results have been presented in three highly-ranked peer-reviewed international journals and five conferences.

References

REACTION References

[Dal-OFC14] M. Dallaglio et al, “Impact of slice-ability on dynamic restoration in GMPLS-based Flexible Optical Networks”, OFC Conf., *Top scored*, March 2014.

[Dal-JOCN15] M. Dallaglio et al, “Provisioning and Restoration with Slice-ability in GMPLS-based Elastic Optical Networks [Invited]”, **Journal of Optical Communications and Networking (JOCN)**, Feb 2015

[Vel-OFC-PDP14] L. Velasco et al, “First experimental demonstration of ABNO-driven in-operation flexgrid network re-optimization”, OFC Conf., *Post-deadline Paper*, March 2014

[Gif-JLT14] Ll. Gifre et al, “First Experimental Assessment of ABNO-driven In-Operation Flexgrid Network Re-Optimization”, **Journal of Lightwave Technology (JLT)**, 2014.

[Pao-PN14] F. Paolucci et al, “Multipath restoration and bitrate squeezing in SDN-based elastic optical networks [Invited]”, **Journal of Photonic Network Communications**, May 2014

[Gif-IC14] Ll. Gifre, A. Castro, M. Ruiz, N. Navarro, L. Velasco, “An in-operation planning tool architecture for flexgrid network re-optimization”, ICTON Conf., July 2014

[Dal-ECOC14] M. Dallaglio et al, “Impact of SBVTs based on Multi-wavelength Source During Provisioning and Restoration in Elastic Optical Networks”, ECOC Conf. Sept. 2014

[Gio-OFC15] A. Giorgetti, F. Paolucci,F. Cugini, P. Castoldi, “Proactive Hierarchical PCE based on BGP-LS for Elastic Optical Networks”, OFC 2015

Additional References

[ABNO] D. King, and A. Farrel, “A PCE-based Architecture for Application-based Network Operations,” IETF draft, work in progress, 2014.

[Agu-OFC14] A. Aguado et al., “ABNO: a feasible SDN approach for multi-vendor IP and optical networks,” in Proc. OFC, 2014.

[Alv-NOF11] D. Alvarez, V. Lopez, J. Anamuro, J. de Vergara, O. de Dios, and J. Aracil, “ Utilization of temporary reservation of path computed resources for multi-domain path computation element protocols in WDM networks, ” in Proc. NOF, Nov. 2011.

[BGPLS] M. Cuaresma, et al, "Experimental demonstration of H-PCE with BPG-LS in Elastic Optical Networks," paper We.4.E.3, ECOC 2013.

[BGPLS] H. Gredler, J. Medved, S. Previdi, A. Farrel, S. Ray, “North-Bound Distribution of Link-State and TE Information using BGP,” IETF draft, work in progress, 2014.

[Cas-ECOC13] R. Casellas, R. Martinez, R. Munoz, L. Liu, T. Tsuritani, and I. Morita, “ Dynamic provisioning via a stateful PCE withinstantiation capabilities in GMPLS-controlled flexi-grid DWDM networks, ” in Proc. ECOC, Sept. 2013.

[Cas-JSAC13] R. Casellas, R. Munoz, J. Fabrega, M. Moreolo, R. Martinez, L.Liu, T. Tsuritani, and I. Morita, “ Design and experimentalvalidation of a GMPLS/PCE control plane for elastic CO-OFDM optical networks, ” IEEE J. Sel. Areas Commun., vol. 31, no. 1, pp. 49 – 61, Jan. 2013.

[Cast-CN12] A. Castro, L. Velasco, M. Ruiz, M. Klinkowski, J. P. Fernández-Palacios, and D. Careglio, “Dynamic Routing and Spectrum (Re)Allocation in Future Flexgrid Optical Networks,” Elsevier Computers Networks, vol. 56, pp. 2869-2883, 2012.

[Cug-JLT12] F. Cugini, G. Meloni, F. Paolucci, N. Sambo, M. Secondini, L. Gerardi, L. Poti, and P. Castoldi, “ Demonstration of flexible optical network based on path computation element”, J. Lightwave Technol., vol. 30, no. 5, pp. 727 – 733, Mar. 2012.

[Cug-JLT13] F. Cugini, F. Paolucci, G. Meloni, G. Berrettini, M. Secondini, F. Fresi, N. Sambo, L. Poti, and P. Castoldi, “ Push-pull defragmentation without traffic disruption in flexible grid optical networks, ” J. Lightwave Technol., vol. 31, no. 1, pp. 125 –133, Jan. 2013.

[Ger-CM12] O. Gerstel, M. Jinno, A. Lord, and S. Yoo, “ Elastic optical networking: A new dawn for the optical layer? ” IEEE Commun.Mag., vol. 50, no. 2, pp. s12 – s20, Feb. 2012.

[Gio-CL10] A. Giorgetti, F. Cugini, N. Sambo, F. Paolucci, N. Andriolli, P. Castoldi, “Path state-Based Update of PCE Traffic Engineering Database in Wavelength Switched Optical Networks”, IEEE Comm. Lett., 2010.

[Gio-JLT09] A. Giorgetti, N. Sambo, I. Cerutti, N. Andriolli, and P. Castoldi, “ Label preference schemes for lightpath provisioning and restoration in distributed GMPLS networks, ” J. Lightwave Technol., vol. 27, no. 6, pp. 688 – 697, Mar. 2009.

[Gio-OFC14] A. Giorgetti, et al., “Impact of Intra-domain Information in GMPLS-based WSONs with Hierarchical PCE,” OFC 2012.

[Gro-04] W. D. Grover, “Mesh-Based Survivable Networks,” Prentice Hall PTR, New Jersey, 2004.

[Jin-CM09] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, “Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies,” IEEE Commun Mag., vol. 47, pp. 66-73, 2009.

[Jin-CM12] M. Jinno, H. Takara, Y. Sone, K. Yonenaga, A. Hirano, “Multiflow Optical Transponder for Efficient Multilayer Optical Networking,” IEEE Commun Mag., vol. 50, pp. 56-65, 2012.

[Lop-JOCN14] V. Lopez, B. de la Cruz, O. G. de Dios, O. Gerstel, N. Amaya,G. Zervas, D. Simeonidou, and J. P. Fernandez-Palacios,“ Finding the target cost for sliceable bandwidth variable transponders, ” J. Opt. Commun. Netw., vol. 6, no. 5, pp. 476 – 485, May 2014.

[Pao-SV13] F. Paolucci, F. Cugini, A. Giorgetti, N. Sambo, and P. Castoldi, “ A survey on the path Computation element (PCE) architecture, ” IEEE Commun. Surv. Tutorials, vol. 15, no. 4, pp. 1819 – 1841, 2013.

[PCE] A. Farrel, J. Vasseur, J. Ash, “A Path Computation Element (PCE)-Based Architecture,” IETF RFC 4655, 2006.

[PCEP] JP. Vasseur, and JL. Le Roux, “Path Computation Element (PCE) Communication Protocol (PCEP),” IETF RFC 5440, 2009.

[RFC3209] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, “Extensions to RSVP for LSP Tunnels,” IETF RFC 3209, 2001.

[RSVP] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, “Extensions to RSVP for LSP Tunnels,” IETF RFC 3209, 2001.

[Rui-OFC14] M. Ruiz, M. Zotkiewicz, A. Castro, M. Klinkowski, L. Velasco, and M. Pioro, “After Failure Repair Optimization in Dynamic Flexgrid Optical Networks,” in Proc. OFC, 2014.

[Shi-TC13] W. Shi, Z. Zhu, M. Zhang, and N. Ansari, “ On the effect of bandwidth fragmentation on blocking probability in elastic optical networks, ” IEEE Trans. Commun., vol. 61, no. 7, pp. 2970 – 2978, July 2013.

[St-PCE] E. Crabbe, J. Medved, I. Minei, and R. Varga, “PCEP Extensions for Stateful PCE,” IETF draft, work in progress, 2014.

[St-PCEP] E. Crabbe, I. Minei, S. Sivabalan, and R. Varga, “PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model,” IETF draft, work in progress, 2014.

[Takara-ECOC11] H. Takara, T. Goh, K. Shibahara, K. Yonenaga, S. Kawai, and M. Jinno, “ Experimental demonstration of 400 Gb/s multiflow, multi-rate, multi-reach optical transmitter for efficient elastic spectral routing, ” in Proc. ECOC, Sept. 2011.

[Tak-ECOC11] T. Takagi, H. Hasegawa, K. Sato, Y. Sone, A. Hirano, and M. Jinno, “ Disruption minimized spectrum defragmentation in elastic optical path networks that adopt distance adaptive modulation, ” in Proc. ECOC, Sept. 2011.

[Vel-CM14] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López, “In-Operation Network Planning,” IEEE Comm Mag., vol. 52, pp. 52-60, 2014.

[Vel-JLT14] L. Velasco, A. Castro, M. Ruiz, and G. Junyent, “Solving Routing and Spectrum Allocation Related Optimization Problems: from Off-Line to In-Operation Flexgrid Network Planning,” IEEE J. of Lightwave Technol., vol. 32, pp. 2780-2795, 2014.

[Vel-JOCN13] L. Velasco, P. Wright, A. Lord, and G. Junyent, “Saving CAPEX by Extending Flexgrid-based Core Optical Networks towards the Edges,” (Invited Paper) IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 5, pp. A171-A183, 2013.

[Yen-QAM70] J. Yen, “An algorithm for finding shortest routes from all source nodes to a given destination in general networks”, Quarterly of Applied Mathematics, vol. 27, pp. 526–530, 1970.

Glossary

ABNO	Application-Based Network Operations
BVT	Bandwidth Variable Transponder
ERO	Explicit Route Obljject
LSP	Label Switched Path
OSPF-TE	Open shortest path first protocol with Traffic Engineering extensions
PCE	Path Computation Element
PCEP	Path Computation Element Protocol
RSA	Routing and Spectrum Assignment
RSVP-TE	Reservation protocol with Traffic Engineering extensions
SBVT	Sliceable Bandwidth Variable Transponder