An In-Operation Planning Tool Architecture for Flexgrid Network Re-Optimization

Lluís Gifre[1,2*], Alberto Castro[1], Marc Ruiz[1], Nacho Navarro[2], and Luis Velasco[1]

[1]*Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain*

[2]*Barcelona Supercomputing Center (BSC), Barcelona, Spain*

**e-mail: lgifre@ac.upc.edu*

ABSTRACT

In-operation network planning consists in re-optimizing a network, currently being operated to transport traffic, either minimizing resource utilization or maximizing the transported traffic. In the context of flexgrid optical networks re-optimization includes spectrum defragmentation, virtual topology reconfiguration, *etc*. To perform in-operation planning, a centralized element in the control plane is needed to efficiently coordinate the elements in the data plane. The active stateful Path Computation Element (PCE), able to update already established connections, can play the above role. Notwithstanding the active stateful PCE, to fulfil the required in-operation planning functionality a tool with a number of interfaces to populate and synchronize databases as well as to attend for re-optimization requests should be implemented. In this paper we propose an in-operation planning tool architecture specifically designed to be connected to the front-end PCE. To illustrate its operation, we study a specific use case: the defragmentation problem to reallocate already established connections so as to make enough room for an incoming request.

Keywords: in-operation network planning, network reconfiguration, high performance computing.

1. INTRODUCTION

In a scenario where connection provisioning can be automated, network resources can be made available by reconfiguring and/or re-optimising the network on demand and in real-time. Authors in [1] named that as in-operation network planning, and proposed to take advantage of novel network reconfiguration capabilities and new network management architectures to perform it, aiming at reducing network CAPEX by minimising the over-provisioning required in today's static network environments.

Significant academic research and standardisation effort has assisted in defining control plane architectures and protocols to automate connection provisioning. Starting from a distributed paradigm, control plane have lately moved towards a centralised one led by the development of the software-defined network (SDN) concept with the introduction of OpenFlow (Open Networking Foundation, https://www.opennetworking.org). IETF is also moving in that direction with the definition of the Application-Based Network Operations (ABNO) architecture [2] based on standard functional elements defined by the IETF like the active stateful Path Computation Element (PCE) [3].

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. A simplified view of the ABNO architecture is represented in Figure 1. It includes: a) The ABNO controller as the entrance point to the network for a Network Management System (NMS) or an Operations Support System (OSS) and the service layer for provisioning and advanced network coordination. It acts as a system orchestrator invoking its inner components accordingly to a specific workflow. b) The PCE defined as an entity to serve path computation requests. The PCE Communication Protocol (PCEP) is used to carry paths computation requests and PCE responses. Requests can be processed independently of each other or in groups, utilising a view of the network topology stored in the Traffic Engineering Database (TED) (stateless PCE) or considering as well information regarding Label Switched Paths (LSPs) that have been set-up in the network, stored on the LSP Database (LSP-DB) (stateful PCE). Finally, a PCE is said to be Active if it can modify in-place LSPs based on network trends. c) The Virtual Network Topology Manager (VNTM) coordinates Virtual Network Topology (VNT) configuration by setting up or tearing down lower-layer LSPs, and advertising the changes to higher-layer network entities. d) The Provisioning Manager (PM) is responsible for the establishment of LSPs. This can be done by interfacing the control plane or by directly programming the data path on individual network nodes using Network Configuration Protocol (NetConf) or acting as an OpenFlow controller. e) The Operations, Administration, and Maintenance (OAM) handler is responsible for detecting faults and taking actions to react to problems in the network. It interacts with the nodes to initiate OAM actions such as monitoring and testing new links.

Directly connected to the ABNO's front-end PCE (fPCE), the in-operation planning tool can be deployed as a dedicated back-end PCE (bPCE) for performance improvements and optimisations [4]. The back-end PCE is accessible via the PCEP interface, so the ABNO components can forward requests to the planning tool.

Furthermore, in-operation network planning can only be achievable if planning tools are synchronised with the state of network resources, so new configurations can be computed with updated information, and those configurations can be easily deployed in the network. In the proposed architecture, the back-end PCE gathers network topology and current state of network resources, via the ABNO components, using protocols designed to convey link-state and traffic engineering information, such as BGP-LS [5].

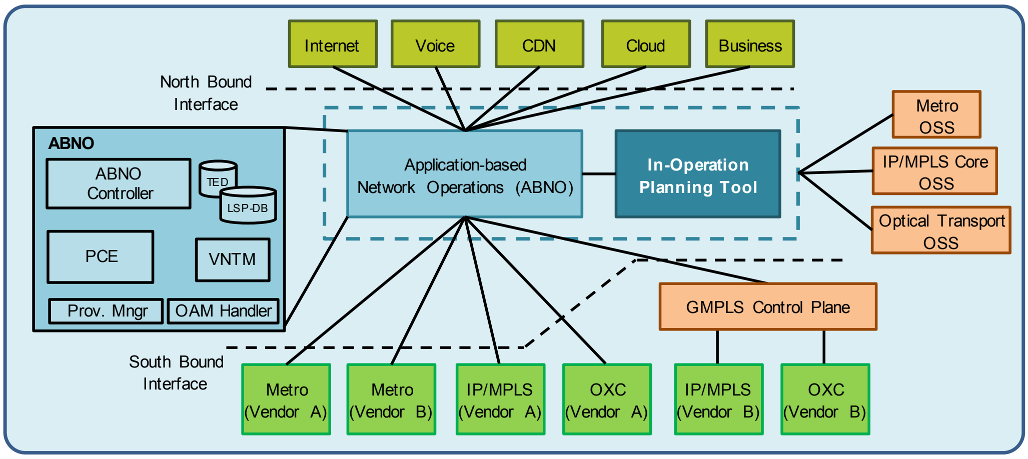


Figure 1. Future dynamic architecture based on ABNO enabling an in-operation planning tool.

In the next section, we present the architecture of our implementation of in-operation planning tool, named as Planning Tool for Optical Networks (PLATON).

2. PLATON ARCHITECTURE

PLATON's architecture is divided into 4 main modules: the communications module, the optimization algorithm framework, the databases (TED and LSP-DB), and the plugged algorithms.

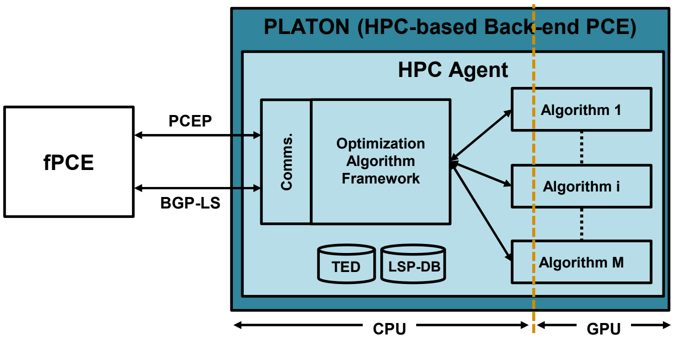


Figure 2. PLATON architecture.

Aiming at connecting PLATON to other modules in the control plane, the communications module must support standard protocols to keep updated its internal databases and to receive computation requests and return the computed solutions. In this regard, both, the BGP-LS and the PCEP protocol are available. BGP-LS Update and PCEP Report messages are used to synchronize the topology into the TED and LSPs into the LSP-DB, respectively. Whereas PCEP Request and Reply messages are used to receive computation requests and to reply with the results, respectively.

At starting time, PLATON dynamically loads the set of algorithms to be plugged; those algorithms will be run upon the reception of computation requests. When a request is received, the algorithm to be executed is selected based on a PCEP Request message's object named Objective Function (OF). Two families of algorithms can be plugged, those to be executed in the CPU and some other that lever on specialized many-core hardware to accelerate computations [6]. Every algorithm uses the internal databases as well as those data objects received in the incoming PCEP Request message and produces a solution that is coded into a PCEP Reply message.

3. USE CASE: DEFRAGMENTATION

Algorithms in the control or management planes compute routes and find feasible spectrum allocations for connection requests taking into account the state of network resources at the time each connection is requested. Nonetheless, as a consequence of network dynamics, some resources may not be released so that better routes could be computed and thus, re-optimisation could not be applied to improve network efficiency. For example, imagine an optical connection that due network congestion, is required to circumnavigate optimal nodes and links, so that the end-to-end connection requires intermediate regenerators; at some point additional paths become available and the service could be rerouted to use the shorter route and eliminate regeneration. Additionally, other existing services could be rerouted to remove the bottlenecks and avoid network congestion. Or even allow some connections to increase their capacity when needed.

In this use case we study a specific problem that arises in flexgrid networks and where re-optimisation could bring clear benefits. In such networks, connections can be allocated using variable-sized frequency slots, whose width (usually a multiple of a basic width such as 12.5 GHz) is a function of the requested bit rate, FEC and modulation format. Such frequency slots must be contiguous in the spectrum and the same along the links in its route. As a consequence of the unavailability of spectrum converters, spectrum fragmentation appears increasing the blocking probability of connection requests, making worse the network grade of service.

Aiming at illustrating spectrum fragmentation, Figure 3 shows an example on the small network topology depicted in Figure 3a, where each node and link is labeled. The entire spectrum width consists of 16 slices. Figure 3b represents the utilization of each frequency slice in the network, where a number of optical connections are already established. In this scenario, the connection request between nodes 4 and 7 requesting 4 slices cannot be served. Notwithstanding, each link in the shortest route of the new optical connection *newP* (through links 4-5-6) has at least 4 free slices and then, the request could be established shifting some of the established connections. In the example, connections *p1*, *p3*, *p4*, *p5*, and *p6* are using one or more of the links in the computed shortest route, and thus can be considered as candidates to be part of the defragmentation process. Finally in Figure 3c, connections *p4* and *p5* have been shifted making enough room for *newP*.

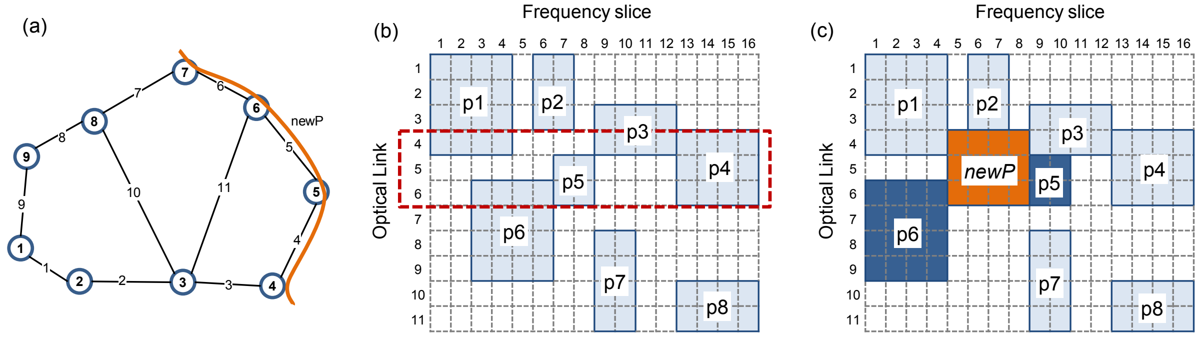


Figure 3. Example of spectrum reallocation.

Authors in [7] describe the SPRESSO algorithm to efficiently compute the set of connections to be reallocated. In [8] the SPRESSO algorithm was integrated into an active stateful PCE and reallocations were performed in a hitless manner by using the Push-Pull technique [9].

The main purpose of spectrum defragmentation is improving network resource utilization. From a control and management perspective, the defragmentation process maps into a set of state changes of active connections. Such state changes are reflected in the change of connections' attributes, in our case spectrum allocation by shifting the nominal central frequency of the slot allocated to a connection. Some other attributes, not covered in this work, can be updated, *e.g.* its allocated spectrum width.

Let us assume that the defragmentation procedure is triggered after the front-end PCE fails to find a suitable route for a provisioning request. The request for a new connection is originally issued by the NMS and received by the ABNO controller through the north-bound interface. In such case, the ABNO controller is responsible for coordinating connection set-up, which composes and sends a specific request towards the front-end PCE, in charge of computing and finally coordinate connection establishment.

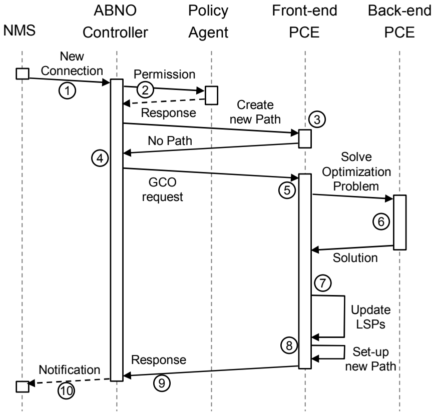
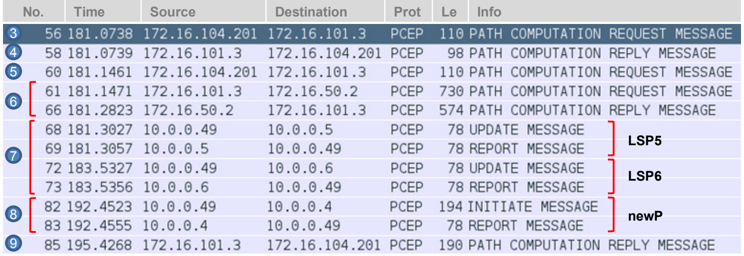


Figure 4. Defragmentation workflow



No.	Time	Source	Destination	Prot	Le	Info	
③ 56	181.0738	172.16.104.201	172.16.101.3	PCEP	110	PATH COMPUTATION REQUEST MESSAGE	
④ 58	181.0739	172.16.101.3	172.16.104.201	PCEP	98	PATH COMPUTATION REPLY MESSAGE	
⑤ 60	181.1461	172.16.104.201	172.16.101.3	PCEP	110	PATH COMPUTATION REQUEST MESSAGE	
⑥ 61	181.1471	172.16.101.3	172.16.50.2	PCEP	730	PATH COMPUTATION REQUEST MESSAGE	
66	181.2823	172.16.50.2	172.16.101.3	PCEP	574	PATH COMPUTATION REPLY MESSAGE	
⑦ 68	181.3027	10.0.0.49	10.0.0.5	PCEP	78	UPDATE MESSAGE	**LSP5**
69	181.3057	10.0.0.5	10.0.0.49	PCEP	78	REPORT MESSAGE	
72	183.5327	10.0.0.49	10.0.0.6	PCEP	78	UPDATE MESSAGE	**LSP6**
73	183.5356	10.0.0.6	10.0.0.49	PCEP	78	REPORT MESSAGE	
⑧ 82	192.4523	10.0.0.49	10.0.0.4	PCEP	194	INITIATE MESSAGE	**newP**
83	192.4555	10.0.0.4	10.0.0.49	PCEP	78	REPORT MESSAGE	
⑨ 85	195.4268	172.16.101.3	172.16.104.201	PCEP	190	PATH COMPUTATION REPLY MESSAGE	

Figure 5. PCEP messages exchanged.

The workflow that represents the provisioning-triggered defragmentation use case is detailed in [10]. As already introduced, it starts with a network operator requesting a new connection provisioning through the NMS. Establishing a flexible optical connection includes computing and provisioning a continuous slot between two nodes in the data plane. The request is received by the ABNO controller via its north-bound interface (step 1 in Figure 4). When the ABNO controller receives the request it asks the policy agent to check about rights of the received request (2). If access is granted, the ABNO controller requests the front-end PCE to compute the route and eventually set up the optical connection (3).

Let us assume that, as a result of spectrum fragmentation, no end-to-end continuous slot is found (4). In that case, the ABNO controller may autonomously decide to perform a defragmentation process and sends a message to the front-end PCE (5). When the front-end PCE receives the request for defragmentation, it checks its feasibility and gathers information to create a GCO request that is sent towards the back-end PCE to solve the optimization problem (6). When the back-end PCE ends, it sends back the solution found. The front-end PCE proceeds then to execute the defragmentation that consists in shifting some of the candidate LSPs (7) and finally, when every LSP has been updated, the front-end PCE proceeds to establish the requested connection (8). Upon its completion, the front-end PCE notifies the ABNO controller (9), which in turn notifies the NMS (10).

Obviously, if the request in (3) finds a feasible route and spectrum allocation, the front-end PCE proceeds with step (8) to establish the connection.

All interactions between ABNO and PCEs are done by exchanging PCE protocol (PCEP) messages. In particular, Path Computation Request (PCReq) and Path Computation Reply (PCRep) messages are exchanged between front-end and back-end PCEs (step 6 in Figure 4). Next, we analyze up to what extent current standards can deal with the use case presented.

4. CONCLUDING REMARKS

In this work, the ability of ABNO architecture to deal with the defragmentation use case, which arises when operating dynamic flexgrid networks, has been experimentally assessed. To that end, the provisioning-triggered spectrum defragmentation use case was firstly investigated. The use case, an example of in-operation planning, starts when a connection request cannot be served as a consequence of spectrum fragmentation in the links of the network. As subset of already established connections are candidate to be part of the defragmentation problem, so that by reallocating some of them enough room is made to serve the new connection requested. Aiming at evaluating the feasibility of the network control plane to deal with such use case, ABNO, the latest architecture of control plane currently under standardization in the IETF, has been evaluated. In particular, a front-end/back-end PCE architecture was considered. The defragmentation use case was modeled as a workflow and the relation among ABNO modules was examined. Specifically, the possibility of using the standard PCEP protocol to convey complex requests and responses between front-end to back-end PCEs, was analyzed.

Our proposal for provisioning-triggered defragmentation was presented afterwards. The proposed workflow, in the form of distributed algorithm running in several modules of the ABNO architecture, was detailed and the PCEP messages exchanged between front-end and back-end PCEs were specified. Finally, the feasibility of the ABNO architecture to deal with the defragmentation use case was experimentally demonstrated; the relevant PCEP messages were shown and its contents analyzed.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o} 238875 GÉANT (open call REACTION), from the Spanish Science Ministry through ELASTIC (TEC2011-27310) project, and from the Barcelona Supercomputing Center NVIDIA CUDA Center of Excellence.

REFERENCES

[1] L. Velasco *et al.*, "In-operation network planning," *IEEE Communications Magazine*, vol. 51, pp. 52-60, 2014.
[2] D. King and A. Farrel, "A PCE-based architecture for application-based network operations", IETF draft, work in progress, 2014.
[3] E. Crabbe *et al.*, "PCEP extensions for stateful PCE," IETF draft, work in progress, 2014.
[4] L. Velasco *et al.*, "Solving routing and spectrum allocation related optimization problems: From off-line to in-operation flexgrid network planning," (Invited Paper) *IEEE/OSA Journal of Lightwave Technology*, 2014.
[5] H. Gredler *et al.*, "North-bound distribution of link-state and TE information using BGP," IETF draft work in progress, 2013.
[6] Ll. Gifre *et al.*, "Experimental assessment of a high performance back-end PCE for flexgrid optical network re-optimization," in *Proc. OSA Optical Fiber Communication Conference (OFC)*, 2014.
[7] A. Castro *et al.*, "Dynamic routing and spectrum (re)allocation in future flexgrid optical networks", *Comp. Networks*, vol. 56, pp. 2869-2883, 2012.
[8] A. Castro *et al.*, "Experimental demonstration of an active stateful PCE performing elastic operations and hitless defragmentation," in *Proc. European Conference on Optical Communication (ECOC)*, 2013
[9] F. Cugini *et al.*, "Push-pull defragmentation without traffic disruption in flexible grid optical networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 31, pp. 125-133, 2013.
[10] Ll. Gifre *et al.*, "Experimental assessment of in-operation spectrum defragmentation," *Springer Photonic Network Communications*, 2014.