iONE: A Workflow-Oriented ABNO Implementation

L. Velasco and Ll. Gifre
Optical Communications Group (GCO)
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain
lvelasco@ac.upc.edu

***Abstract*—An implementation of the IETF standardized ABNO architecture, named as iONE is presented. iONE consists in one single generic configurable module and a set of dynamically linkable workflows. The spectrum defragmentation use case is used to experimentally demonstrate iONE's functionalities.**

Keywords—ABNO, SDN, Optical networks

I. INTRODUCTION

The recently-standardized ABNO architecture [1] is specially intended to provide on-demand connectivity to network applications in multi layer / domain / vendor environments (Fig. 1). ABNO consists of a number of components, such as a Path Computation Element (PCE), plus a north-bound interface (NBI), a south-bound interface (SBI), and internal interfaces among modules.

The ABNO controller is the entrance point of ABNO; it implements the NBI for a Network Management System (NMS) / Operations Support Systems (OSS) to request connectivity services and network re-optimization. The SBI is used by the provisioning manager to configure network devices. To that end, any existing protocols can be used, including PCEP [2] and OpenFlow. ABNO includes two databases, the Traffic Engineering Database (TED) and the Label Switched Path Database (LSP-DB). The TED can be built, among others, using link-state distribution extensions to BGP (BGP-LS) [3].

A number of use cases are presented in [1] describing how ABNO can be applied to provide application-driven and NMS/OSS-driven network operations. Network operations include provisioning and in-operation planning [4]. The ABNO components need to work together to provide the requested network operation. That repeatable activity can be defined as a workflow among the components.

In this paper we present iONE, an implementation of the ABNO architecture specifically designed to facilitate workflow definition and deployment. iONE's functionalities are experimentally demonstrated through a use case for network spectrum defragmentation.

II. IONE ARCHITECTURE

iONE consists in one single generic software module developed in C/C++, which can be configured using an XML file to implement many of the ABNO modules. The iONE module's architecture, depicted in Fig. 2, consists of 5 components: the communication interfaces, the manager, the network databases, the algorithms framework, and the workflow engine. Furthermore, workflows can be defined through XML files, where the algorithms to be executed are dynamically loaded into the iONE module.

The communication interfaces are the gateway for all incoming and outgoing messages; these messages can be encoded using REST API, PCEP, and BGP-LS protocols. iONE's implementation of PCEP supports both active stateful [5] and LSP initiate capabilities. Besides, the REST API interfaces support both XML and JSON-encoded messages. An arbitrary number of interfaces for each protocol can be configured using a XML file.

The manager is responsible for configuring and coordinating the rest of components; it collects incoming messages received by any communication interface and either redirects them to the proper component or uses them to update databases.

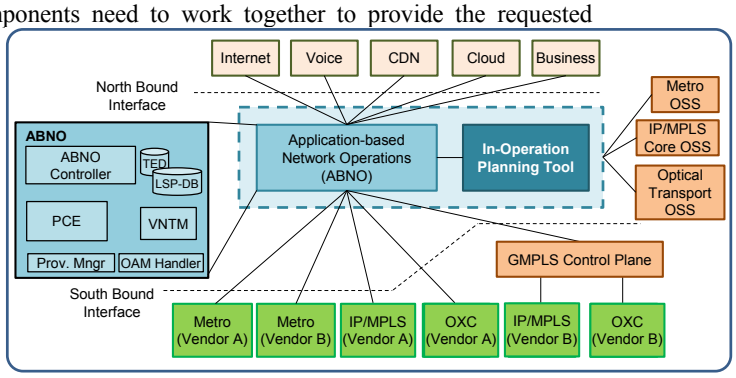


Fig. 1. ABNO-based control and management architecture

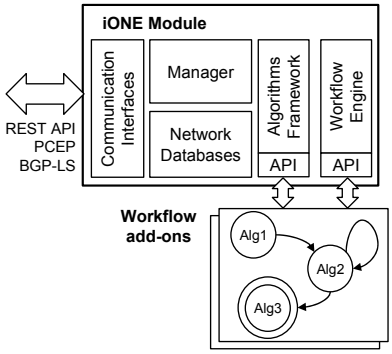


Fig. 2. Architecture of the iONE Module

978-1-4799-8821-1/15/$31.00 ©2015 IEEE

The network databases component provides in- memory storage for the TED and the LSP-DB. The databases can be pre-loaded by means of XML files and configured to synchronize their content through specific communication interfaces. For instance, a BGP-LS interface can be configured to handle BGP Update (*BGPUpd*) messages to synchronize the TED, while a PCEP interface can synchronize the state of the LSP-DB using asynchronous Path Computation Report (*PCRpt*) messages. The algorithms framework provides common functionalities for agile algorithm development, including routing algorithms, such as the Routing and Spectrum Allocation (RSA) [6] algorithm, optical spectrum handling functions, metaheuristic frameworks, and many more. Besides, protocol helper functions for rapid message handling are provided. All those functions are available through an API.

Finally, workflows are implemented as finite state machines (FSM). Two elements are needed to add a new workflow to an iONE module: a dynamically linkable file containing a number of algorithms and a XML file defining the workflow. Since workflows are implemented as FSMs, a number of states and transitions between states are defined in the XML file; a different algorithm is executed in each state of the workflow. Workflow state algorithms use the previously described algorithms API.

The workflow engine is responsible for initiating new workflow instances, and coordinating their execution subject to incoming messages. The engine contains a message table storing identifiers of messages pending to be received and the workflow instance awaiting each message.

Each workflow is configured through an XML file that contains: *i*) the triggering message to initiate the workflow, *ii*) the set of states, *iii*) the set of transitions between states based on incoming message types, *iv*) initial state of the FSM, *v*) the path to a dynamic linked library implementing the algorithms to be run on each state of the FSM, and *vi*) a workflow state algorithms configuration file to define parameters and constants. Workflows use the algorithms framework and workflow engine APIs to interact with the iONE module.

When a workflow is started, a workflow instance is created by the workflow engine containing the current state and internal workflow data, thus enabling workflow concurrently. Incoming *PCReq*, *PCUpd*, and *PCInitiate* messages trigger workflow instances creation. In the particular case of PCReq messages, Objective Function (`OF`) objects can be used to specify the desired workflow to be run; a default workflow must be defined for those messages without any `OF` object.

To select the appropriate function in the dynamic library implementing a given workflow state algorithm, its name must be the same as the name of the state. Workflow state algorithms can execute any generic algorithm and send messages to other modules using functions in the APIs. At the end of the state algorithm execution, the list of pending messages to be received is reported to the workflow engine, which updates the messages table with the expected identifiers in Request Parameters (`RP`) and Stateful PCE `RP` (`SRP`) objects. If no pending messages are reported, the workflow automatically finishes it execution.

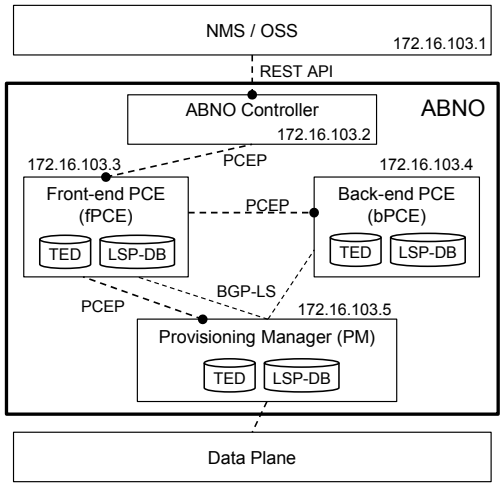


Fig. 3. Example of iONE set-up

III. Use Case: Spectrum Defragmentation

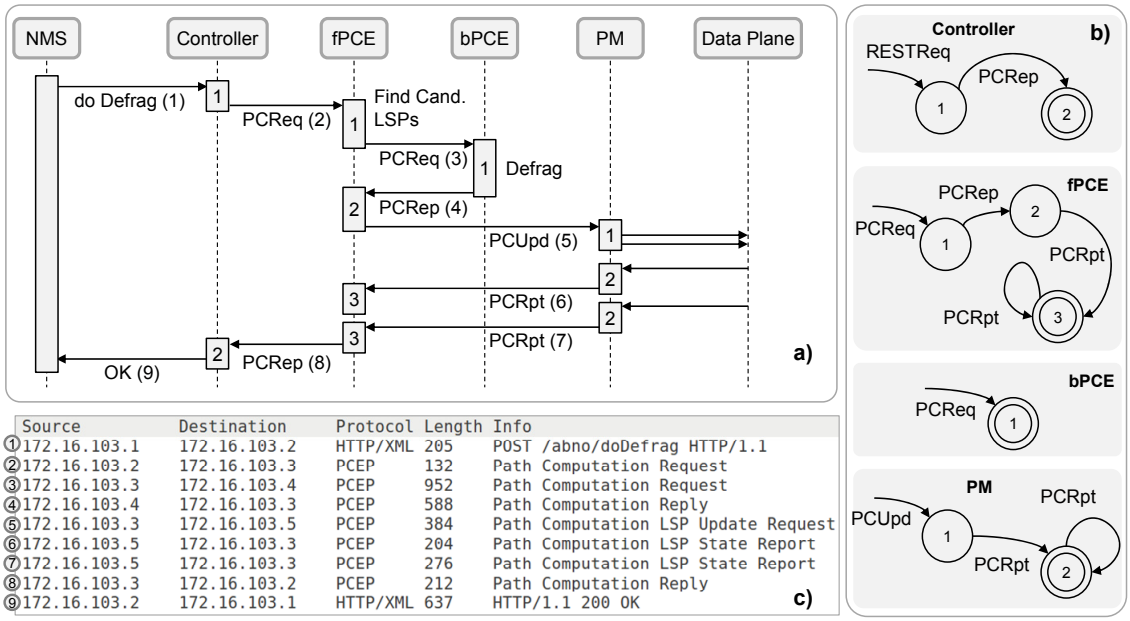
In this section a use case on spectrum defragmentation is used to illustrate how the iONE module can be configured to define complete workflows involving several ABNO components, including the ABNO controller, a front-end PCE (fPCE), a back-end PCE (bPCE), and a provisioning manager (PM) component (see Fig. 3). Each component is implemented using a single iONE module running in the depicted IP address. A configuration XML file is used to define per-module contained databases as well as the interfaces and the connectivity to other modules.

In our example, the NMS is connected to the ABNO controller through a REST API for requesting connections. Internally, ABNO components use PCEP and BGP-LS interfaces to communicate among them. For instance, the fPCE needs to implement three PCEP interfaces, one for the ABNO controller, another for offloading computations to the bPCE, and a third one for LSP operations with the PM.

PCEP is used for LSP-DB synchronization by means of PCRpt messages. The PM module generates PCRpt messages towards the fPCE, which updates its local LSP-DB and generates PCRpt messages towards the bPCE module. In addition, BGP-LS is used to synchronize the TED from PM to both fPCE and bPCE.

Specifically for the spectrum defragmentation use case, Fig. 4a presents the sequence diagram to be implemented. For this use case we assume that the NMS issues a spectrum defragmentation request on a selected optical link based on their specific policies and metrics. Each workflow algorithm has been numbered to facilitate its identification.

When the ABNO controller receives a spectrum defragmentation request through the REST API interface (message 1), it issues a PCReq message (2) to the fPCE containing an `OF` object defining the re-optimization algorithm to be executed and the specific optical link to be defragmented. The fPCE finds the candidate LSPs, those using the selected optical link, and offloads the defragmentation computation to



	Source	Destination	Protocol	Length	Info
①	172.16.103.1	172.16.103.2	HTTP/XML	205	POST /abno/doDefrag HTTP/1.1
②	172.16.103.2	172.16.103.3	PCEP	132	Path Computation Request
③	172.16.103.3	172.16.103.4	PCEP	952	Path Computation Request
④	172.16.103.4	172.16.103.3	PCEP	588	Path Computation Reply
⑤	172.16.103.3	172.16.103.5	PCEP	384	Path Computation LSP Update Request
⑥	172.16.103.5	172.16.103.3	PCEP	204	Path Computation LSP State Report
⑦	172.16.103.5	172.16.103.3	PCEP	276	Path Computation LSP State Report
⑧	172.16.103.3	172.16.103.2	PCEP	212	Path Computation Reply
⑨	172.16.103.2	172.16.103.1	HTTP/XML	637	HTTP/1.1 200 OK

c)

Fig. 4. a) Spectrum Defragmentation Sequence Diagram. b) FSM definition for each iONE module. c) Messages exchanged.

the bPCE (UPC's in-operation network planning tool named as PLATON [4]). The PCReq message (3) contains an `OF` object so the bPCE identifies the algorithm to be executed, runs it and replies the fPCE with the solution in a PCRep message (4). The fPCE processes the received message and requests LSP updates to the PM by means of a PCUpd message (5). The fPCE collects all the replied PCRpt messages (6-7) and eventually replies to the ABNO controller using a PCRep message (8), which in turn replies to the NMS (9).

It is clear that every ABNO component needs to do a number of actions that are triggered by the received messages. As explained above, we model the actions to be done by each module as a FSM. The FSMs for each iONE module are represented in Fig. 4b, where each state is identified using the same identifier used in Fig. 4a; error states were omitted for simplicity.

The FSM for the ABNO controller consists of two states; the workflow is triggered by a REST Request (*RESTReq*) message and the algorithm for state (1) is run, and upon the reception of a PCReq message the algorithm for state (2) is run. The FSM for the fPCE is slightly more complex since it involves iteratively executing state 3 until all PCRpt messages are collected. The FSM for the bPCE is the simplest one since its operations (process incoming PCReq message, execute spectrum defragmentation algorithm, and reply solution) are executed strictly sequentially, thus allowing to be encapsulated in a single state.

Finally, Fig. 4c shows the exchanged REST and PCEP messages captured in our test-bed.

IV. Conclusions

An ABNO implementation, named as iONE, was presented; it consists of one single generic module that can be configured to act as almost any ABNO module. Workflows are easily configured by defining FSMs in each module, where FSMs' states are deployed to be dynamically linked. iONE's functionalities were experimentally demonstrated through a use case for optical spectrum defragmentation.

Additional use cases using PLATON (iONE module configured as bPCE) for re-optimization [7] and multicast provisioning [8] have been recently experimentally assessed. There, a similar FSM to that in Fig. 4b was implemented.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 317999 IDEALIST and from the Spanish MINECO SYNERGY project (TEC2014-59995-R).

References

[1] D. King, A. Farrel, "A PCE-Based Architecture for App.-Based Netw. Operations," IETF RFC 7491, 2015.

[2] JP. Vasseur et al., "Path Computation Element (PCE) Comm. Protocol (PCEP)," IETF RFC 5440, 2009.

[3] H. Gredler et al., "North-Bound Distribution of LS and TE Information using BGP," IETF work-in-progress, 2015.

[4] L. Velasco et al., "In-Operation Network Planning," IEEE Communications Magazine, vol. 52, pp. 52-60, 2014.

[5] E. Crabbe et al., "PCEP Extensions for Stateful PCE", IETF draft-ietf-pce-stateful-pce-11, 2015.

[6] L. Velasco et al., "Solving Routing and Spectrum Allocation Related Optimization Problems: from Off-Line to In-Operation Flexgrid Network Planning," J. of Lightwave Tech (JLT), vol. 32, pp. 2780-2795, 2014.

[7] Ll. Gifre et al. "First Experimental Assessment of ABNO-driven In-Operation Flexgrid Network Re-Optimization," J. of Lightwave Tech. (JLT), vol. 33, pp. 618-624, 2015.

[8] Ll. Gifre et al. "Experimental Assessment of ABNO-driven Multicast Connectivity in Flexgrid Networks," J. of Lightwave Tech. (JLT), vol. 33, pp. 1549-1556, 2015.