ABNO: a feasible SDN approach for multi-vendor IP and optical networks

A. Aguado, V. López, J. Marhuenda, O. González de Dios and J.P. Fernández-Palacios
Telefónica I+D c/ Don Ramón de la Cruz, 82-84, 28006 Spain
jpfpg@tid.es

Abstract: ABNO architecture is proposed in IETF as a framework which enables network automation and programmability thanks to the utilization of standard protocols and components. This work not only justifies the architecture but also presents the first experimental demonstration.
OCIS codes: (060.0060) Fiber optics and optical communications; (060.4250) Networks.

1. Introduction

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

Software Defined Networking (SDN) and network programmability offer the ability to direct application service requests towards the IP/MPLS and optical network. SDN approach can help operators to reduce the CAPEX and OPEX in the networks, thanks to the optimization of the resources and the reduction of the complexity in the operation of the network. However, the proposed SDN controllers in the market are based on monolithic software, which are not adapted to current heterogeneous network environments. Such SDN solutions are like a "black-box" and its deployment leads to different problems for the operator: (1) vendor lock-in (solutions in the market are mono-vendor), (2) lack of support with non Open Flow (OF) networks and (3) problems to support E2E multi-domain path establishment in current networks.

Within the IETF, Application-based Network Operations (ABNO) is defined to provide a solution based on standard protocols and components [1]. The main component of the ABNO architecture is the Path Computation Element (PCE). The IETF ABNO architecture is based on existing standard blocks defined within the IETF (PCE, ALTO, VNTM...), which could be implemented either in a centralized or distributed way according to network operator requirements. Thanks to the modular nature of ABNO architecture, building blocks can be deployed by different vendors or third parties and even by a single provider. This modularity and the standard interfaces between the modules solve the problem of vendor lock-in for the operators. On the other hand, ABNO is specially adapted to multidomain and multivendor networks, enabling interoperability between control plane based and OF based domains. In this paper, we validate via experimentation the ABNO architecture for a multilayer use case in two scenarios with control plane and with OF in the optical layer.

2. Application-Based Network Operations architecture

The ABNO architecture combines a number of technology components, mechanisms and procedures. The ABNO architecture and subsequent components provide a variety of services by coordinating the components that operate and manage the network [1]. A detailed explanation of the ABNO architecture is done in [1], here we highlight the main components for the multi-layer use case (Fig. 1). The ABNO Controller is the main component of the architecture and is responsible of orchestrating the workflows, and invokes the necessary components in the right order. ABNO stores a repository of workflows with operations in the network (MPLS provisioning, L3VPNs, etc.). The Path Computation Element (PCE) is the unit that handles the path computation across the network graph. If the PCE is active, it can directly operate in the nodes to create LSPs. Coordination between multiple PCEs operating on different TEDs is also required to do path computation in multi-domain (for example, inter-AS) or multi-layer networks. For multi-layer scenarios, Virtual Network Topology Manager (VNTM) is a key element. VNTM is in charge of maintaining the topology of the upper layer by connections in the lower layer. This entity simplifies the upper-layer routing and traffic engineering decisions as it hides the optical connections set up by the LSP.

Topology Module (TM) has multiple databases: a view of each layer, an inter-layer relation and an inventory DB with the configuration parameters of each of the resources. One part of the module is devoted to getting and maintaining up to date the topology from the available sources (e.g. routing protocols, OF controllers) [2]. The other part of the module is devoted to providing the information to the requesting parties such as the Provisioning Manager, PCE or VTNM. Let us remark that this interface to disseminate the topology is under discussion in IETF.

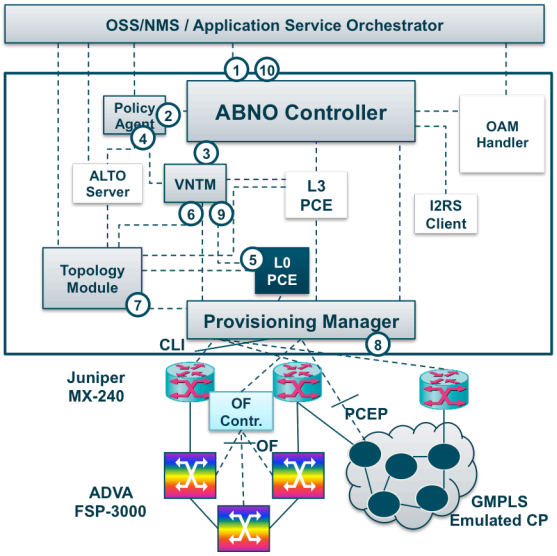

Fig. 1. ABNO architecture and its interactions for IP Link provisioning use case

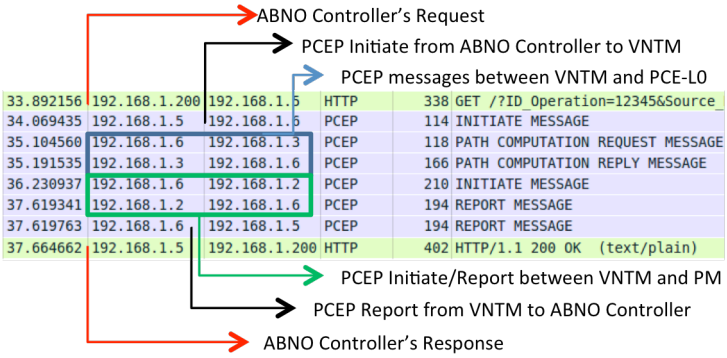

Fig. 2. Message flow in IP Link provisioning use case.

Finally, the Provisioning Manager (PM) is the unit in charge of configuring the network elements. It can do so both by configuring the resources in each node (CLI, OF or NetConf) or by triggering a set of actions to the control plane via PCEP [3].

3. PCEP to remotely initiate GMPLS LSPs

The extensions described in [3] propose that the PCE protocol can be used to set-up GMPLS LSPs remotely. The idea behind is to enable an active PCE to set-up GMPLS LSPs. However, these extensions can be used to request a path setup from any element in the network. This feature within ABNO framework facilitates the interaction between multiple of the modules in the architecture. It is clear that all modules that talk with PCE can use this interface to request for a new path. However, there are other modules that can use this interface for their interactions. For instance, ABNO controller can use this interface to request the VNTM to create a new path [3,4].

PM is other module that can take advantage of PCEP. When a PCInitiate message is received by the PM it asks the TM for information about each node. The TM gives for each IP in the ERO: the layer of the node, the configuration protocol (CLI, NetConf, OF or PCEP) and the specific parameters to configure the node. The main advantage of PCEP at this point is that the PM does not have to perform any operation since it would have just to forward the PCInitiate message to the network element. For the equipment that does not support PCEP, PM has to know the CLI commands or XML configuration for network in the case of NetConf.

Another option to configure the nodes is OF. Although OF is a very promising technology for L2 technologies, there are not yet deployments in real core networks. Some vendors have done implementations for transport networks [2, 5], but the OF protocol has specific vendor extensions. This is the reason why, when our PM receives information about the configuration of one node via OF, it relies in an external OF controller. This OF controller deals with the specific implementation for optical networks. In the future, OF can be supported within the PM for optical and packet technologies.

Thanks to the utilization of PCEP, most of the entities in the ABNO architecture are technology agnostic and only the PM and the Topology Module have to know which are the protocols the network elements are using. We think that this fact a great advantage of the proposal in [3].

4. Experimental validation of the ABNO architecture for IP link provisioning

In this work, we implement the ABNO architecture to support an IP link provisioning workflow. We want to demonstrate the capabilities of the architecture not only to support multi-layer scenarios, but also to operate in OF and GMPLS enabled networks. To do so, we have used three Juniper MX-240 routers, four ADVA FSP 3000 optical nodes and five GMPLS nodes in a CP emulator developed in TID. ABNO architecture modules are virtual machines deployed on a server with two processor Intel Xeon E5-2630 2.30GHz (6 cores each) and 192 GB RAM. Similarly, each GMPLS node emulates the control plane using a VM.

Fig. 1 presents the interactions of the modules to provision an IP Link, while Fig. 2 shows the PCEP and HTTP messages exchange. Some messages (like KeepAlive and Open messages) are omitted to improve readibilitty. Each step is following explained:

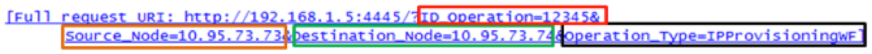

Fig. 3 HTTP Request to ABNO Controller.

```
Line-based text data: text/plain
  <html><body><p>Code :{"Error_Code":"NO_ERROR","Source_Node":"10.95.73.73","Result":"IP_LINK_CONFIGURED",
   "Operation_Type":"IP_LINK_PROVISIONING","ID_Operation":"12345","Destination_Node":"10.95.73.74"} </p></body></html>\n
```

Fig. 4 HTTP Response from ABNO Controller

1. The NMS sends a request to the ABNO controller using a HTTP Message with the operation type or workflow (IP Provisioning), its ID and the parameters for this operation (source and destination). The format of this request is shown in Fig. 3. The IP addresses for the interfaces and the IGP can be added in the request.
2. ABNO asks the policy agent to check whether it has sufficient rights to make the request.
3. Once the ABNO Controller has checked the permission, it tells the VNTM to create a link between the two routers. The VNTM is the module with the inter-layer information so it knows to which optical nodes each router is attached to. This interface uses PCInitiate message with the extensions of (Fig. 2) [3]. The PCInitiate includes the source and destination IP address in the End-Points object.
4. The VTNM also needs to verify with the Policy Agent if it can provide the service.
5. The VNTM module sends a PCRequest message to the L0-PCE for a path between two optical nodes, and L0-PCE sends a PCResponse with the computation (Fig. 2). In this experiment, the L0-PCE is stateless and does not have instantiation capabilities. The VNTM uses the L0-PCE just to get the path in the optical layer.
6. The VNTM has the interlayer information, L0 path and end points. This information is sent to the PM to configure the nodes using a PCInitiate message (Fig. 2) [3]. The information in this request is similar to the PCInitiate message from the ABNO controller to the VNTM, but adding the ERO obtained from the L0 PCE.
7. The PM queries the TM for the description of each node. Based on the response, the PM knows the layer and the technology for each node, and, consequently, that a new link is created in the optical layer, because there are hops in the IP and the optical layer.
8. Depending on the technology and configuration mode, PM selects a different protocol to complete the request. First the optical layer is configured and, secondly, the IP nodes. To configure the IP layer there are some options: PCEP, CLI or NetConf. As PCEP is not yet supported, we have used CLI in this experiment. To configure the optical layer, PCEP, OF or even UNI from the router can be used. We have tested two operations mode here:
   a. To configure the ADVA nodes, an OF controller is used with extensions for wavelength switching [4]. A web service interface is used to trigger ADVA's controller.
   b. For GMPLS CP, we have implemented the extensions in [3] to support PCInitiate forwarding to RSVP in TID control plane nodes (Fig. 5). The RSVP message contains the information of the PCInitiate (Fig. 6).
9. Once the network elements are configured, PM notifies the VTNM using a PCRpt message (Fig. 2).
10. Similarly, the VNTM advertises to ABNO controller, which sends a HTTP Reply to the NMS (Fig. 4). This response has the ID of the operation, the operation type, an error code, the source, the destination and the result of the operation. To support any correlation in the NMS.

5. Conclusions

ABNO is a framework with a pragmatic approach towards Software Define Networks because of the utilization of modules and protocols already under standardization. The main contribution of this work is the experimental demonstration of the ABNO architecture using commercial equipment (ADVA and Juniper nodes) and the validation of the PCEP extensions to support remote GMPLS LSP set-up. Thanks to this work, we have showed that the operational complexity of using multiple configuration technologies (like CLI, GMPLS and Openflow) and different layers (IP and optical) can be highly reduced thanks to the ABNO architecture.

6. Acknowledgements

The research is funded by the European Community's under grant agreement no. 317999 IDEALIST project.

7. References

[1] D. King et al., draft-farrkingel-pce-abno-architecture-05.
[2] Z. Ali et al., IETF Draft, draft-ali-pce-remote-initiated-gmpls-lsp-01.
[3] O. Gonzalez de Dios et al., Mo.4.E.2, ECOC 2013.
[4] O. Gonzalez de Dios et al., IPOP 2012.
[5] A. Autenrieth, et al. OFELIA Workshop, ECOC 2011.

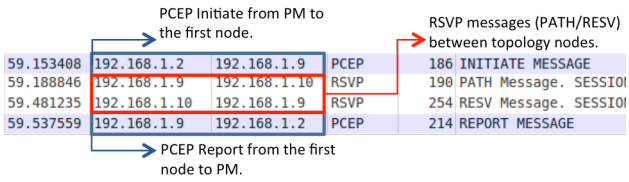

Fig. 5 PCInitiate and RSVP messages in edge node

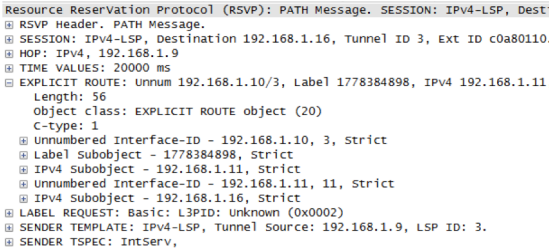

Fig. 6 RSVP PATH message in edge node