Coordinated Computation of Multi-layer Paths via Inter-layer PCE Communication: Standards, Interoperability and Deployment

M. Chamania*, O. González de Dios†, V. López†, M. Cuaresma †, M. Drogon*, A. Jukan*, X. Masip-Bruin‡, M. Yannuzzi‡
*TU Braunschweig, Germany, E-mail: {chamania, jukan, drogon}@ida.ing.tu-bs.de
†Telefónica I+D, Spain, E-mail: {ogondio, vlopez, mcs}@tid.es
‡Advanced Network Architectures Lab (CRAAX), Spain, E-mail: {xmasip, yannuzzi}@ac.upc.edu

***Abstract*—The Path Computation Element (PCE) is positioned nowadays as one of the solutions that almost every carrier will eventually deploy. The PCE architecture as well as a number of components, including the PCE Communication Protocol (PCEP), have been standardized by the IETF. However, a number of challenges remain to be solved on its way from standards to deployment. In particular, the existing proposals for multi-layer path computation within the PCE framework need to be further developed and tested, before considering their possible integration into operational networks. This is especially true for the interoperability of the various PCE implementations and the extensions such as the Virtual Network Topology Manager (VNTM) which cannot be taken for granted. This paper presents a functional implementation of coordinated computation of multi-layer paths supported through inter-layer PCE communication, where one PCE is developed by industry and the other as an open-source effort. To this end, we consider an IP/MPLS network deployed over a Wavelength Switched Optical Network (WSON), each of which deploying its own PCE, in an attempt to create an inter-operable multilayer solution. We discuss the key challenges that the research community will face in this area, which in turn will drive a considerable part of the upcoming efforts in terms of standardization.**

I. Introduction

In the last few years, carriers have experienced a significant increase in the number of customers requesting the provision of on-demand network services, especially to run applications with high Quality of Service (QoS) requirements. These demands are forcing network carriers to review their service provisioning and network planning practices. In current carrier networks, most customer services are deployed over an IP/MPLS infrastructure, given its flexibility in terms of connection and QoS provisioning. The IP/MPLS network is in turn deployed over an Optical Transport Network (OTN) infrastructure, which is used to provide connectivity between the IP/MPLS routers. Given the diversity in the operation and administration tasks of these networks, carriers typically have two different departments to manage their corresponding infrastructures. Moreover, to minimize the interaction between these departments (and thereby the operational costs), IP/MPLS networks are typically over-provisioned, to the extent that peak utilization may be limited in practice to 30-40% of the link capacity [1], [2]. However, the increasing demand for more bandwidth along with dynamic (and flexible) service offerings are forcing carriers to explore multi-layer provisioning techniques, which are expected to optimize network resources while providing better support for upcoming applications and services.

To make this possible, carriers are faced with the challenge of how to best automate the computation and provisioning of multi-layer service paths, which are problems that have been addressed within the Path Computation Element (PCE) charter at the IETF [3], [4]. The PCE is an entity that uses topology and capacity information available in its Traffic Engineering Database (TED) to compute constrained paths in the network, and is emerging as the de-facto path computation solution in carrier-grade networks. The existing proposals for multi-layer path computation within the PCE framework can be broadly categorized into two groups: i) the ones employing a single PCE to compute multi-layer paths, which entails that the PCE must have information about both network infrastructures; and ii) the ones proposing to use independent PCEs for the different network infrastructures. In this second group, the proposals can be in turn divided into two categories, one in which the computation of multi-layer paths is handled without explicit communication among the PCEs in different layers and another which supports inter-layer PCE communication. At present, this latter seems more likely to be employed, as it not only maintains the operational independence between the two network infrastructures within a carrier, but also facilitates deterministic multi-layer path computation.

However, the deployment of a PCE architecture targeting coordinated computation tasks between independent networks poses significant challenges both in terms of design and implementation. Different works have outlined specific difficulties in coordinated path computation using multiple PCEs, including issues such as signaling overhead and delay as well as the additional components and extended capabilities required at the network elements. To this end, many research challenges pertaining to actual system parameters such as topology update mechanisms, algorithmic complexity and performance, type and nature of multi-layer requests, etc., are still to be explored and evaluated, especially, in real settings.

In this paper, we present the first implementation of inter-layer PCE coordination in a multi-layer network scenario and test its performance and interoperability. For the implementation described in this work, an IP/MPLS network is deployed over a Wavelength Switched Optical Network (WSON), each of which has its own PCE. More precisely, we describe coordinated inter-operation between a PCE developed by Telefonica I+D for WSON networks, and an open-source PCE emulator developed by TU Braunschweig for IP/MPLS networks [5], [6]. Both implementations are based on the current IETF standards. We have also developed a basic implementation of a Virtual Network Topology Manager (VNTM), which is used to provision server-layer connections in the WSON network and to announce them to the client IP/MPLS network layer. Based on these implementations, we present preliminary performance results and highlight the existing challenges toward future efforts in this area. It should be noted that most current PCE implementations are proprietary and vendor specific, and as such have not been tested on their interoperability. To the best of our knowledge this is the first work which test the PCE interoperability between different standardized PCE implementations, and that also in a multi-layer scenario.

The rest of the paper is organized as follows. In Section II, we give an overview of the current standards and related work. Our implementation is described in Section III, and an overview of the testbed setup and measurements are presented in Section IV. Finally, Section V outlines some of the important open problems and concludes the paper.

II. Standards and Related work

A description of the different multi-layer PCE configurations as well as some specialized components that can be used for facilitating inter-layer path provisioning can be found in [4]. There are three basic configurations, namely:

1) *Single multi-layer PCE:* In this configuration, a single PCE has the topology information for all the different layers in a multi-layer network, and uses this information to compute a multi-layer path.
2) *Multiple PCEs without inter-PCE communication:* In this case, each layer contains a single PCE but they do not interact directly with each other. The source node in the client layer (i.e., the IP/MPLS layer) asks the source PCE for a tentative path, which returns either a path to the destination or a loose path including a transit node v that is the entry point to the server layer (the WSON layer). The transit node v, upon receiving the path computation signaling information, communicates with the server layer PCE in order to compute a path in the lower layer and returns it to the start node. Then, the signaling process begins in the server layer, and once the circuit setup is completed, the signaling continues in the client layer.
3) *Multiple PCEs with inter-PCE communication:* In this third configuration, the difference is that when the client PCE cannot find a path in the client network, the client PCE initiates a connection with the server PCE to trigger the computation of a multi-layer path jointly. The computed path is then returned in the form of a *multi-layer Explicit Route Object* (ERO) to the client node.

Although the single PCE configuration can perform optimal multi-layer path computation—by virtue of having the complete multi-layer topology information at a centralized location—this scheme is generally not applicable to carrier networks. The limiting factors are scalability and the organizational separation of IP/MPLS and WSON networks, since complete topology information is typically not shared among these networks. In contrast, the multi-PCE model is perfectly aligned with the separation observed in carrier networks, and is therefore the approach chosen for our implementation. The use of multiple PCEs, however, does come with additional overhead, such as larger path computations and setup delays (see, e.g., the analysis presented by Gunreben et al. in [7]).

There have also been approaches such as [8], which propose to segment the path computation process horizontally rather than vertically. In this case, a PCE is used for a subset of client nodes and the nodes to which they are connected in the server layer, and the interaction between several such PCEs in the network can be used to compute end-to-end multi-layer paths. While shown to be efficient in terms of blocking, this model (like in the single PCE configuration) fails to fit into the administrative separation of current carrier networks, and is therefore not used in our implementation.

Within the multi-PCE model, we have implemented the third configuration, i.e., the one where PCEs in different layers communicate with each other to facilitate the multi-layer path computation. We have implemented a *Virtual Network Topology Manager* (VNTM) from that proposed in [4], [9], which is a management element with a twofold role: to trigger the signaling for setup/decommissioning of the client links in the server network, and to re-optimize the topology in the client network. For instance, to setup a new MPLS link, the VNTM is responsible for initiating signaling for the corresponding circuit in the WSON network. In our model, VNTM can commission/decommission links in the client network using the Explicit Route Object (ERO) provided during multi-layer path computation; but the re-optimization traffic within the network - as shown in [10], in which the VNTM can trigger automatic *switch-on* or *switch-off* of links in the client network, is not currently supported in our implementation.

Before presenting the details of our implementation, we proceed to formalize the concept of a "multi-layer path". In the PCE framework, a "path" is represented as an ERO, which in the client IP/MPLS network consists of a series of IPv4 address objects, whereas in the server WSON layer, the ERO contains the list of unnumbered interfaces along the computed path as well as the address of the destination node. In order to help the network elements differentiate between paths computed in the client and the server layer, the SERVER_LAYER_INFO sub-object defined in [11] acts as a delimiter, to identify the start and the end of each new WSON circuit resulting from the path computation process.

III. Implementation Overview

The implementation overview is presented in Fig. 1, which contains four major components: 1) The client (node R1 in

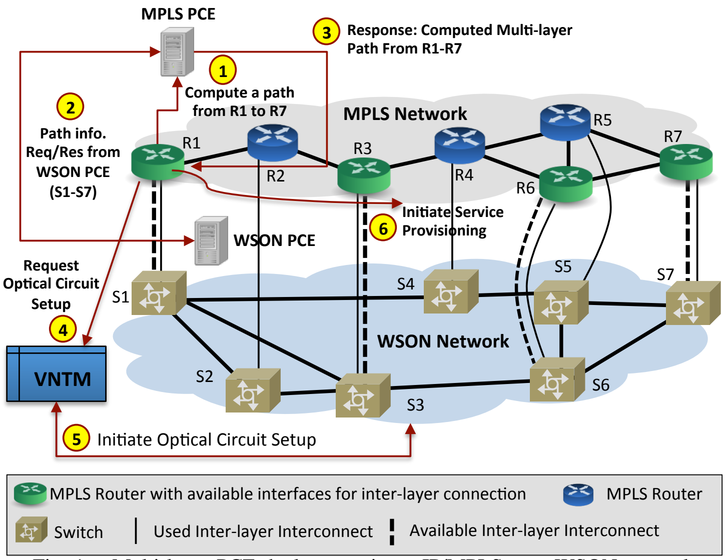

Fig. 1. Multi-layer PCE deployment in an IP/MPLS over WSON networks

this example), 2) the IP/MPLS PCE, 3) the WSON PCE and 4) the VNTM. As mentioned above, the IP/MPLS PCE is implemented using an open-source PCE emulator [6], while the WSON PCE has been developed at Telefonica I+D. For the preliminary tests presented here, the functionalities of the client and the VNTM are fairly basic. A typical path computation and provisioning workflow can be summarized as follows: The client requests the IP/MPLS PCE for a path (1), where the client (i.e., the Path Computation Client (PCC)) could be an MPLS Labeled Switch Router, such as R1, or it could be a Network Management System (NMS). Once the PCE receives the request, it searches for a path within the IP/MPLS layer, and in case sufficient resources exist, it returns the computed path to the client. In case that the resources available in the IP/MPLS network are insufficient, the PCE requests the WSON PCE for additional resources in the form of a set of computation requests for a circuit (2). The number of path computation requests made by the IP/MPLS PCE to the WSON PCE is typically more than one, given the uncertainty in the required path (recall that full topology is not known). Upon receiving an answer from the WSON PCE, the IP/MPLS PCE chooses a suitable circuit computed in the WSON layer, embeds the computed path in the Explicit Route Object (ERO) and returns the answer to the client (3). In case a multi-layer path is computed, the client forwards the ERO to the VNTM (4), which initiates circuit setup in the WSON layer (5) and signals the creation of a new link to IP/MPLS PCE. In our implementation, the circuit setup implies an update of the available link capacity values within the WSON PCE TED, while a link setup is indicated as a

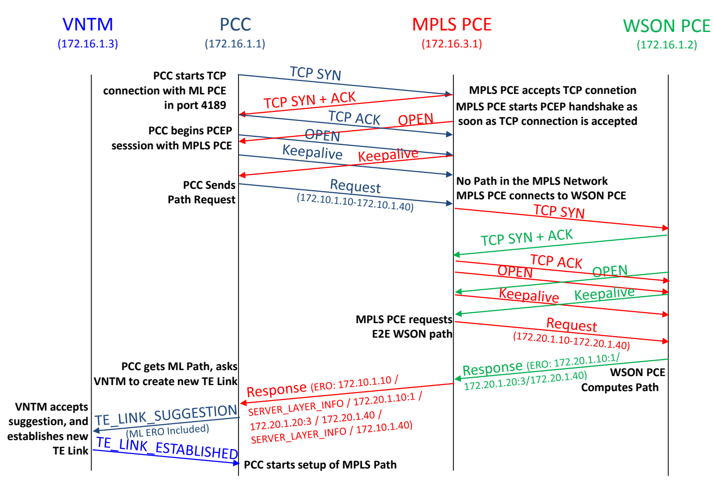

Fig. 2. Multi-PCE/VNTM signaling for coordinated multi-layer path computation. Note that in this example, the IP addresses of the PCC and the source node within the path computation request differ. As mentioned above, this is because the PCC could be an NMS. In the next section, we will examine this case in further detail.

change in the IP/MPLS topology within the IP/MPLS PCE TED. After the completion of this signaling, the client then initiates the resource reservation in the IP/MPLS network (6), which is again indicated as an update of the available link capacity information in the IP/MPLS PCE TED.

A. Detailed Inter-layer PCE Signaling

The detailed signaling interactions among the four major components in the architecture are shown in Fig. 2. The PCEP signaling used for the communication between the client and the MPLS PCE as well as the MPLS and the WSON PCE follow the specifications standardized in RFC 5440 [12]. As the first step, the client establishes a PCEP session with the MPLS PCE, which involves the exchange of the *Open* and the *Keepalive* messages between them. The client then sends the *Path Computation Request* to the MPLS PCE, which contains information about the service endpoints and additional constraints such as bandwidth and delay to compute the path. The MPLS PCE attempts to compute the path, and if a path is found, it returns the description of the path as an Explicit Route Object within the *Path Computation Response Message*. However, if a path is not found in the MPLS network, the MPLS PCE then established a PCEP session with the WSON PCE, and based on the algorithms used for multi-layer path computation, requests the computation of one or more paths in the WSON network. In order to establish the session, the MPLS and WSON PCE exchange *Open* and *Keepalive* messages before the MPLS PCE sends a path computation request to the WSON PCE.

Given that neither layer has information about the topology of the other layer, at least one of the two PCEs should have information about the inter-layer interconnects available in the network. This is necessary to match a path requirement in the MPLS layer to the corresponding optical switches and ports in the lower layer. In our implementation, we assume that the MPLS PCE has information about the corresponding switch in the lower layer, as well as the number of free interfaces available on the MPLS router, so that it can translate a demand for a link between two MPLS routers to a corresponding demand for a circuit between two WDM switches in the transport network. As an example, based on the topology in Fig. 1, the MPLS PCE has information that routers $R1$, $R3$, $R6$ and $R7$ are connected to the optical switches $S1$, $S3$, $S6$ and $S7$ respectively, and translates a request for a new link between $R1$ and $R7$ to a request for a circuit between $S1$ and $S7$, which is then sent to the WSON PCE.

The WSON PCE, upon successful computation of the circuit, returns the description as an ERO with unnumbered interfaces and the endpoint for the destination switch (as described in the previous section). The ERO may represent a segment used by the service within the MPLS network and as a result is included inside the ERO computed within the MPLS PCE. To indicate that the ERO contains a multi-layer path, the *SERVER_LAYER_INFO* sub-object is used as demarcation, to indicate the start and the end of the path in the server layer. The complete ERO is then included in a *Path Computation Response* message and is sent to the client.

Note that it may be necessary for some services to use multiple lower layer segments in the transport network to provision a service, which might require the computation of multiple non-overlapping paths in the transport network. In such a scenario, the Synchronized Vector List [12] can be used by the MPLS PCE to request paths to the WSON PCE. In our current implementation, however, we only compute end-to-end paths in the server network, i.e. if a request from $R1$ to $R7$ cannot be satisfied in the client network, the MPLS PCE looks for a circuit from $S1$ to $S7$ in the server network, and rejects the request if the circuit cannot be established in the server network.

As shown in Figs. 1 and 2, in our implementation, to setup a connection, the client sends the ERO to the VNTM, which establishes the path in the server network and then instructs the client to reserve resources in the MPLS network.

IV. Testbed Setup and Measurements

As mentioned in the previous section, we used two PCEs, namely, the open-source PCE for the MPLS network [6] and the PCE developed at Telefonica I+D for the WSON network. In the current implementation, there was no real MPLS or WSON network in the setup, and all reservations and releases were emulated by updating the TEDs in the two PCEs.

In our setup, the PCC and the VNTM were co-located on a single machine (IP 172.16.1.1) at Telefonica Premises in Madrid, so the Round Trip Time (RTT) between them was negligible. The WSON PCE (IP 172.16.1.2) was also deployed at Telefonica premises in Madrid while the MPLS PCE (IP 172.16.3.1) was deployed at TU Braunschweig in Germany, and the average RTTs (measured using ping traces) from the PCC to the MPLS PCE, and from the MPLS PCE to the WSON PCE, were found to be 44.29 ms for both measurements (averaged over 1000 RTT's with a standard deviation $<$ 0.15 ms). In the topology used, all MPLS routers were addressed in the the subnet 172.10.1.0/24, while the WSON switches were addressed in the subnet 172.20.1.0/24, with co-located routers/switches having the same post-fix. So for example, in Fig. 1, if R1 had IP address 172.10.1.10, the address of S1 was 172.20.1.10.

The typical message exchange indicating the computation of a multi-layer path using the two PCEs is shown in Fig. 3. The PCC on 172.16.1.1 initiates a PCEP session with the MPLS PCE (172.16.3.1), involving the exchange of the Open and Keepalive messages, after which the PCC sends the Path Computation Request to the MPLS PCE. The MPLS PCE is unable to compute the entire path, so it opens a connection with the WSON PCE (172.16.1.2), again involving the exchange of Open and Keepalive messages, before the PCEP request can be forwarded to the WSON PCE. The WSON PCE then replies with a PCEP Response message, containing the ERO for the computed circuit. Upon receiving this information, the MPLS PCE computes the path and sends the response back to the PCC.

Fig. 4 presents the Wireshark traces for the path computation request and response messages exchanged between the PCC and the MPLS PCE, as well as the between the MPLS PCE and the WSON PCE. In Fig. 4(a), we see a request for a path from 172.10.1.130 to 172.10.1.70 sent from the PCC to the MPLS PCE. The MPLS PCE is responsible for translating the incoming MPLS request to a request for a circuit in the WSON network, and as a result, we see a change in the end-points in Fig. 4(b), where a WSON circuit is requested from 172.20.1.130 to 172.20.1.70. The response from the WSON PCE is shown in Fig. 4(c), and it contains the ERO with two unnumbered interfaces (172.20.1.130:1, 172.20.1.140:1) and the WSON endpoint 172.20.1.70, which is then included in the response from the MPLS PCE in Fig. 4(d). The *SERVER_LAYER_INFO* sub-object currently does not have a type defined in the standards and in this setup we set the type of this object as 40. The ERO for the multi-layer path indicates the path to be an end-to-end circuit, with the computed WSON path ERO enclosed within two *SERVER_LAYER_INFO* sub-objects with the MPLS endpoints 172.10.1.130 and 172.10.1.70 indicating the beginning and the end of the computed MPLS path.

Using this implementation, we performed a set of tests in order to measure the total delay in the path computation process. In the setup, both the MPLS and the WSON networks used the Atlanta network topology [13], with 14 nodes and 22 bi-directional links, and we forced all path computation requests to be multi-layer requests by setting the available link capacities to zero for all MPLS links. We also re-used the established PCEP sessions between the PCC and the MPLS PCE as well as the MPLS PCE and the WSON PCE, so that the signaling only included the exchange of the Path Computation Request and Response messages. We made 1000 measurements and found the total time required for the average multi-layer path computation to be 97.84 ms with a standard deviation of 1.02 ms. The signaling includes two round-trip times of 44.29 ms, one between the PCC and the MPLS PCE and another between the MPLS PCE and the WSON PCE. As all links were set with no remaining capacity in the MPLS network, the path computation time in the MPLS PCE was negligible ($<$ 1 ms) [5], so the total time was primarily affected by the path computation times in the WSON PCE.

In our implementation, topology updates were initiated by the PCC or the VNTM to emulate setup of an MPLS connection or a circuit in the WSON network respectively. This explicit update mechanism meant that the TEDs in both networks were updated as quickly as possible. The updates were sent as an link update for each individual link affected in the network. The RTT between the VNTM and the WSON PCE was $<$ 1 ms, and therefore the total update times were negligible. However, owing to the larger RTTs between the PCC and the MPLS PCE, an average delay of about 68.5 ms *per link* was observed when updating the MPLS topology. This delay may lead to a race condition, especially, when the arrival rate of path computation requests is high. This issue is further discussed in Section V-B.

Time	Source	Destination	Protocol	Length	Info
0.000000	172.16.1.1	172.16.3.1	TCP	76	38517 > pcep [SYN] Seq=0 Win=5744 Len=0
0.045879	172.16.3.1	172.16.1.1	TCP	76	pcep > 38517 [SYN, ACK] Seq=0 Ack=1 Win=
0.045904	172.16.1.1	172.16.3.1	TCP	68	38517 > pcep [ACK] Seq=1 Ack=1 Win=5760
0.047013	172.16.1.1	172.16.3.1	PCEP	80	OPEN MESSAGE
0.116876	172.16.3.1	172.16.1.1	PCEP	80	OPEN MESSAGE
0.118012	172.16.1.1	172.16.3.1	PCEP	72	KEEPALIVE MESSAGE
0.161777	172.16.3.1	172.16.1.1	PCEP	72	KEEPALIVE MESSAGE
2.008948	172.16.1.1	172.16.3.1	PCEP	96	PATH COMPUTATION REQUEST MESSAGE
2.078247	172.16.3.1	172.16.1.2	TCP	76	54497 > pcep [SYN] Seq=0 Win=5744 Len=0
2.081970	172.16.1.2	172.16.3.1	TCP	76	pcep > 54497 [SYN, ACK] Seq=0 Ack=1 Win=
2.126310	172.16.3.1	172.16.1.2	TCP	68	54497 > pcep [ACK] Seq=1 Ack=1 Win=5760
2.127371	172.16.1.2	172.16.3.1	PCEP	80	OPEN MESSAGE
2.131504	172.16.3.1	172.16.1.2	PCEP	80	OPEN MESSAGE
2.171718	172.16.1.2	172.16.3.1	PCEP	72	KEEPALIVE MESSAGE
2.176935	172.16.3.1	172.16.1.2	PCEP	72	KEEPALIVE MESSAGE
2.224110	172.16.3.1	172.16.1.2	PCEP	96	PATH COMPUTATION REQUEST MESSAGE
2.226090	172.16.1.2	172.16.3.1	PCEP	120	PATH COMPUTATION REPLY MESSAGE
2.282304	172.16.3.1	172.16.1.1	PCEP	152	PATH COMPUTATION REPLY MESSAGE

Fig. 3. Wireshark captures of PCEP signaling for multi-layer path computation.

```
Path Computation Element communication Protocol
⊞ PATH COMPUTATION REQUEST MESSAGE Header
⊞ RP object
⊟ END-POINT object
    Object Class: END-POINT OBJECT (4)
    Object Type: 1
  ⊞ Flags
    Object Length: 12
    Source IPv4 Address: 172.10.1.130
    Destination IPv4 Address: 172.10.1.70
```

(a) Path computation request from PCC to MPLS PCE.

```
Path Computation Element communication Protocol
⊞ PATH COMPUTATION REQUEST MESSAGE Header
⊞ RP object
⊟ END-POINT object
    Object Class: END-POINT OBJECT (4)
    Object Type: 1
  ⊞ Flags
    Object Length: 12
    Source IPv4 Address: 172.20.1.130
    Destination IPv4 Address: 172.20.1.70
```

(b) Path computation request from MPLS PCE to WSON PCE.

```
Path Computation Element communication Protocol
⊞ PATH COMPUTATION REPLY MESSAGE Header
⊞ RP object
⊟ EXPLICIT ROUTE object (ERO)
    Object Class: EXPLICIT ROUTE OBJECT (ERO) (7)
    Object Type: 1
  ⊞ Flags
    Object Length: 36
  ⊞ SUBOBJECT: Unnumbered Interface ID: 172.20.1.130:1
  ⊞ SUBOBJECT: Unnumbered Interface ID: 172.20.1.140:1
  ⊞ SUBOBJECT: IPv4 Prefix: 172.20.1.70/32
```

(c) Path computation response from WSON PCE to MPLS PCE.

```
Path Computation Element communication Protocol
⊞ PATH COMPUTATION REPLY MESSAGE Header
⊞ RP object
⊟ EXPLICIT ROUTE object (ERO)
    Object Class: EXPLICIT ROUTE OBJECT (ERO) (7)
    Object Type: 1
  ⊞ Flags
    Object Length: 68
  ⊞ SUBOBJECT: IPv4 Prefix: 172.10.1.130/32
    Non defined subobject (40)
  ⊞ SUBOBJECT: Unnumbered Interface ID: 172.20.1.130:1
  ⊞ SUBOBJECT: Unnumbered Interface ID: 172.20.1.140:1
  ⊞ SUBOBJECT: IPv4 Prefix: 172.20.1.70/32
    Non defined subobject (40)
  ⊞ SUBOBJECT: IPv4 Prefix: 172.10.1.70/32
```

(d) Path computation response from MPLS PCE to PCC.

Fig. 4. Components of the path computation request and response messages within the PCEP signaling. Source IPv4 address (172.10.1.130) is different from that of the PCC (172.16.1.1). This is because in this example, the PCC is in fact an IP NMS.

V. Challenges and Future Work

In the coming years, we will observe considerable advances in the computation and provisioning of multi-layer paths supported through PCE-based architectures. This work shows a functional implementation of cooperating PCEs and a VNTM. In future work we will evaluate the performance of coordination mechanisms in realistic network scenarios. However, there are some open issues regarding implementation before assessing the performance in real networks. From our perspective, the main challenges are basically the following.

A. Topology Description, Update and Provisioning

Mechanisms for topology discovery and information update have been widely studied both in the management ecosystem and in the context of the PCE architecture. However, most of the efforts up to date have mainly focused on the discovery of topologies within a single layer. Given the diverse nature of technologies and vendors in the different layers, a primary challenge in current networks is how to automatically discover the inter-layer interconnects. Currently, only vendor-specific (proprietary) solutions exist to determine these interconnects, so in multi-vendor setting are maintained manually and are therefore error-prone. As up-to-date information on the inter-layer interconnects is critical to the performance of the multi-layer PCE architecture, it is necessary to devise accurate mechanisms to gather and update this information.

For provisioning, the major challenge faced was the description of the muti-layer ERO. The extensions to support the multi-layer ERO has been proposed in [11] and validated in this paper. However, this multi-layer ERO clearly defined the routers from which the path is initiated and terminated in the WSON layer, but the ports to be used in the MPLS network and the corresponding client ports to be used in the WSON network were not identified. Given that typically both network topologies are not aware of the interconnects, i.e. which port in an MPLS router is connected to which a client port in a WSON switch, provisioning of connections requires additional information to be provided during setup.

B. Managing Race Conditions in Stateless PCEs

In our performance evaluations, we observed that requests were well-spaced in time, allowing the provision of new links and the update of the MPLS PCE TED to finish before processing of a new request. However, in case of higher frequencies of incoming requests, there is the possibility of race conditions, which can lead to the creation of unnecessary links in the MPLS network. For example, consider a scenario where a request A *triggers* a new link to be provisioned, but before the provisioning of the link can finish, a new request B arrives, which, once processed according to the current TED information, it can trigger the provisioning of an additional link in the network. It could be perfectly the case that the provisioning of the second link would not have been triggered by request B if request A was completed before its arrival. Overall, the stateless PCE model does not keep track of the computed requests, and simply relies on the current information in its TED to compute the paths. Therefore, additional mechanisms will be needed to *police* the process of creation of new links in the network.

C. Decommissioning of established circuits

In multi-layer path computations, there is a bandwidth mismatch between the request from the client layer and the circuit granularities available in the server layer. As a result, the circuit in the server layer will usually have much more capacity than that requested by the client PCE, and thus the same circuit can be used to provision other connections. This, however, makes de-commissioning a link a challenge, as a link established in the client layer to setup a connection (C_1) may also be used by other connections (C_2, C_3), and as a result cannot be torn down when the original requesting connection (C_1) is decommissioned.

In our implementation, we handled the creation of new links in the IP network in response to demands from the client. However, once services are decommissioned from the network, the additional circuits established in the MPLS network should also be decommissioned. This problem is typically referred to as *traffic uploading* and analytical models for the same have been proposed in [14]. However, frameworks to facilitate IP uploading, including monitoring and re-routing of existing traffic and decommissioning of optical circuits have not been proposed to date. When employing a multi-layer PCE solution in a dynamic network scenario, it is especially important to employ mechanisms to facilitate the same.

References

[1] FP7 Project ONE (INFSO-ICT-258300): Towards Automated Interactions between the Internet and the Carrier-Grade Management Ecosystems. http://www.ict-one.eu/

[2] Internet2 Headroom Practice. [Online]. Available: https://wiki.internet2.edu/confluence/download/attachments/17383/Internet2+Headroom+Practice+8-14-08.pdf

[3] A. Farrel, J. P. Vasseur, and J. Ash, “A Path Computation Element-Based Architecture,” *IETF RFC 4655*, August 2006. http://tools.ietf.org/html/rfc4655.

[4] E. Oki, T. Takeda, J.L. le Roux, and A. Farrel, “Framework for PCE-based inter-layer MPLS and GMPLS traffic engineering,” *IETF RFC 5623*, September 2009. http://tools.ietf.org/html/rfc5623.

[5] M. Chamania, M. Drogon, and A. Jukan, “An Open-Source Path Computation Element (PCE): Design, Implementation and Performance,” *IEEE Journal of Lightwave Technology 2011*, 2011.

[6] Open Source Path Computation Element Emulator. [Online]. Available: http://pce.ida-cns-group.net

[7] S. Gunreben and F. Rambach, “Assessment and performance evaluation of PCE-based inter-layer traffic engineering,” in *Optical Network Design and Modeling, 2008. International Conference on*, march 2008, pp. 1–6.

[8] F. Cugini et al., “Multiple path computation element (PCE) cooperation for multi-layer traffic engineering,” in *Proc. OFC 2007*, paper OWK5, March 2007, Anaheim, CA, USA.

[9] E. Oki, I. Inoue, and K. Shiomoto, “Path computation element (PCE)-based traffic engineering in MPLS and GMPLS networks,” in *Sarnoff Symposium, 2007 IEEE*, 30 2007-may 2 2007, pp. 1 –5.

[10] T. Miyamura et al. “Demonstration of pce-controlled dynamic traffic engineering for gmpls-based multilayer service network,” *IEE Conference Publications*, vol. 2005, no. CP502, pp. v2–269–v2–269, 2005. [Online]. Available: http://link.aip.org/link/abstract/IEECPS/v2005/iCP502/pv2-269/s1

[11] E. O. et al., “Extensions to the Path Computation Element communication Protocol (PCEP) for Inter-Layer MPLS and GMPLS Traffic Engineering,” *IETF draft-ietf-pce-inter-layer-ext-06.txt*, January 2012. http://tools.ietf.org/html/draft-ietf-pce-inter-layer-ext-06.

[12] J. Vasseur and J. L. Roux, “Path Computation Element Communication Protocol,” *IETF RFC 5440*, March 2009. http://tools.ietf.org/html/rfc5440.

[13] Atlanta Network Problem Instance, SNDlib–Library of test instances for Survivable fixed telecommunication Network Design. [Online]. Available: http://sndlib.zib.de/

[14] M. Chamania, M. Caria, and A. Jukan, “A comparative performance analysis of IP traffic offloading schemes over Dynamic Circuits,” in *INFOCOM, 2011 Proceedings IEEE*, april 2011, pp. 136 –140.