Highly Available SDN Control of Flexi-Grid Networks With Network Function Virtualization-Enabled Replication

Ramon Casellas, Ricard Vilalta, Ricardo Martínez, and Raül Muñoz

Abstract—New trends and emerging requirements have driven the development of extensions to the path computation element (PCE) architecture beyond the computation of a set of constrained routes and associated resources between endpoints, given a network topology. Such extensions involve the use of a PCE for the control of network services, in which deploying a PCE as a centralized network controller facilitates the adoption of software-defined networking (SDN) principles while allowing a progressive migration of already existing deployments. A key requirement for the adoption of centralized control solutions is the ability to deploy a resilient, secure, dynamically configurable, adaptive, and highly available (virtualized) infrastructure supporting end-to-end services, including critical and vertical ones. Part of this infrastructure is the control plane functional elements (e.g., controllers), and the use of network function virtualization (NFV) is an enabler for the high availability of such elements while additionally reducing OPEX and CAPEX. NFV provides a feature-complete framework for the replication of software components that is a straightforward and commonly adopted approach to address the aforementioned requirement, but it implies the need for timely synchronization of databases between replicas. In this paper we present, implement, and validate an architecture for PCE and SDN control high availability, combining the virtualization of the control function by means of dynamic replication and the timely synchronization of their internal state using the PCEP and BGP-LS protocols. We experimentally validate the approach with a testbed, including a GMPLS/PCE control plane, and a replica management system implemented following the ETSI NFV framework, using the OpenStack cloud management software.

Index Terms—Flexi-grid networks control and management; High availability; Network function virtualization (NFV); Path computation element (PCE); Replication; Software-defined networking (SDN).

I. INTRODUCTION

The path computation function is commonly accepted as an integral part of either a management or a control plane (either centralized or distributed). As such, the path computation element (PCE) architecture, developed within the IETF [1], defines a PCE as an entity capable of performing constrained path computation, along with a PCE communications protocol (PCEP, [2]). The PCEP protocol was initially specified to allow a path computation client (PCC) to request path computations, enabling a wide range of deployment scenarios and addressing specific problems such as path computation in multi-domain networks with limited topology visibility [3].

The ASON/GMPLS architecture remains a viable, mature approach for the provisioning of data channels, fitting of mature protocols, and the adoption of software-defined networking (SDN) is in part justified by the fact that business and application logic can be easily integrated into a control layer, while relegating, e.g., the GMPLS control plane, to an automation tool part of the provisioning process. With SDN, in view of programmability and the use of open interfaces, operators can provision new services efficiently.

Common deployments of PCEs are centralized, although this is not mandated by the architecture. This has driven the development of extensions to the PCE architecture beyond the original scope of computing constrained routes between endpoints, given a network topology. Such extensions involve the use of a PCE for the control of network services, driving the actual provisioning processes. A PCE can ease the adoption of SDN principles while allowing progressive migration of existing deployments, acting as a centralized entity where operator-defined algorithms and policies can be deployed while still driving distributed MPLS/GMPLS transport networks and other technologies.

In particular, the application-based network operations [4] architecture defines an SDN-like approach that can be used for the control of transport optical networks, including a stateful PCE (a PCE that takes into account both the network topology and connections database to perform path computation). Lately, PCEs are becoming more and more functionally equivalent to an SDN controller, and PCEP extensions are being developed to use a PCE with different south-bound interfaces, including the PCE-driven control and instantiation of label-switched paths (LSPs) in MPLS/GMPLS [5] or its use in segment routing networks.
[6], where a source node can choose a path without relying on hop-by-hop signaling protocols, such as RSVP-TE. Finally, efforts are ongoing to allow a PCE to have direct control over each node along the path, driving the setup and release of cross-connections and related forwarding operations [7].

Generically, a key requirement for the adoption of centralized control is the deployment of a resilient, secure, dynamically configurable, adaptive, and highly available (virtualized) infrastructure supporting end-to-end services, including critical and vertical ones. For the particular case of the PCF, network operators need to be able to upgrade different components without disrupting the existing network operation. This includes hot-swapping, software and hardware upgrades, policy changes, etc. Carrier-class solutions require reliable software components, with flexible upgrade/update cycles and redesigning of active-standby deployments, as well as innovative approaches and mechanisms dealing with unprecedented system complexity and service criticality (i.e., including environments supporting multi-tenancy).

The use of network function virtualization (NFV) [8], described later, partially addresses this requirement, additionally reducing OPEX and CAPEX. Its use for the deployment of control plane functions, including the PCE, has been recently considered [9,10]. The use of replication for software components is a straightforward approach to high availability, but it implies the need for distributed network databases and their timely synchronization. One of the missing aspects of previous work is related to the synchronization of PCE internal databases.

The GMPLS/PCE architecture conveys two main control plane databases: the traffic engineering database (TED), and the label-switched path database (LSPDB). While the use of general-purpose distributed databases is in scope, we still lack clear and standard information and data models to successfully model such databases, along with the actual reference points, protocol(s), and interfaces, especially in order to avoid vendor-specific solutions and scenarios, limiting interoperability. Alternatively, and as put forward in this work, the synchronization of TED and LSPDB between dynamically instantiated replicas is carried out by using existing, mature, open, and standard protocols, namely, PCEP and BGP-LS [11]. Consequently, the network (link and node) data and information models are implicit by the currently supported protocol information objects.

This paper is an extended version of our previous work published in [12], and it is structured as follows: after this introduction, we briefly present the main concepts behind the ETSI NFV (Section II) in view of its applicability for the virtualization of PCE replicas. In Section III, we detail our proposed control plane architecture and proposed functional entities, message exchanges, and workflows. In Section IV, we present the main components of our experimental testbed, and in Section V, we summarize the main results of our experimental evaluation. Finally, Section VI concludes the paper.

II. ETSI Network Function Virtualization

The ETSI NFV industry specification group addresses the dynamic deployment and operation of common network functions stored and executed in virtual computing instances, which are, in turn, typically running in commodity hardware. NFV defines the architecture and interfaces for the management and orchestration of such virtualized network functions (VNFs) and, amongst relevant aspects, the initial documents recognized the need for the arbitrary and flexible composition of VNFs into graphs, potentially spanning multiple domains. An end-to-end ETSI network service (NS) can be described by a network function (NF) forwarding graph of interconnected network functions and endpoints.

Notable functional elements of NFV management and orchestration are the NFV-orchestrator (NFV-O), which manages the lifecycle of ETSI network services, global resource allocation, and the validation and authorization of infrastructure resource requests, and the virtualized infrastructure manager (VIM), which controls and manages the compute, storage, and network resources within one operator infrastructure sub-domain. Multiple VIMs can be orchestrated by the NFV-O.

The concept of domain within the NFV is manifold. The architecture defines, notably, the concepts of VNF domain, infrastructure domain, and tenant domain, where multiple tenant domains can co-exist in a single infrastructure domain, separating domains associated with VNFs from domains associated with the NFV infrastructure (NFVI). Within the NFVI [13], the aspects of compute, hypervisors, and infrastructure networking are maintained as separate. Geographically speaking, an NFVI may have multiple points of presence (NFVI-PoP), defined as a single location with a set of deployed NFVI-Nodes. A given NFVI can be administratively split into NFVI domains, thus managed by one or more VIMs.

In this work, we are mostly concerned with a single VNF domain, potentially, although not necessarily, across multiple infrastructure domains. We consider PCE (or SDN controller) replicas as the VNFs, and it is thus the role of the NFV-O to orchestrate NFVI resources across one or multiple VIMs. We assume that a (private) NFVI is available for the network operator to deploy control plane functions. By operating this domain, multiple instances can be launched under the control of the operator (see Fig. 1).

III. Control Plane Architecture

In this section, we detail the major elements of the control plane architecture, focusing on the virtualization of PCE functions. PCE high availability relies on synchronized PCE replicas and is enabled by the combined use of cloud computing architectures (with the actual coordination of PCE instances under the responsibility of a dedicated cloud infrastructure controller) and entities that enable the database synchronization, avoiding complex state machines.
The proposed architecture (see Fig. 2) relies on the following main component concepts:

- A controlled transport network infrastructure. In this work, this network is assumed to be an optical transport network with flexi-grid optical spectrum switching, composed of flexi-grid ROADMs interconnected with optical fibers in an arbitrary mesh topology. For the experimental demonstration, this work assumes that PCEs act as SDN controllers, ultimately delegating establishment of LSPs to an underlying GMPLS control plane (without excluding other PCE south-bound interfaces not requiring GMPLS, including, e.g., OpenFlow [14,15] or PCEP for forwarding configuration [7]).

- A private ETSI NFVI (implemented in terms of a cloud infrastructure and OpenStack deployment). This, in turn, enables the deployment of multiple, dynamically allocated PCE replicas, understood as different instances of the same functional entity, which are themselves synchronized by means of the PCEP and BGP-LS protocols.

- The use of a replica reflector, an entity acting conceptually as a BGP reflector [16], thus avoiding the full mesh between replicas and limiting control plane overhead. This PCEP/BGP-LS reflector acts as a bridge between the replicas and the underlying control plane, being a proxy for centralized LSP provisioning and path computation.

- The replica manager with a graphical user interface (GUI), both interacting with the operator's operation and business support systems and, at the same time, behaving like an NFV-O for the dynamic allocation of replicas and the coordination of the replica reflector.

There are several important considerations to note. First, although the straightforward implementation of the concept relies on homogeneous software images, diversity is not precluded (for example, to cover migration or software upgrades), as long as the different software images implement the synchronization protocols. Second, the use of a reflector raises the issue of high availability for the reflector itself. Even if the reflector is significantly simpler than the actual PCEs and not subject to updates, upgrades, and life cycles, multiple reflectors can potentially be deployed (e.g., in pairs and clustering), thus fulfilling the high-availability requirement (see Fig. 3).

A. Dynamic Operation and Procedures

We summarize here the main, simplified workflows and message exchanges for the system, with the help of Fig. 4. The NMS/Replica manager (NFV-O) uses the cloud controller (VIM) exported REST API, which enables the on-demand dynamic instantiation and deallocation of customized PCE instances, with varying capabilities in terms of memory, CPU, and deployed algorithms and policies (in the figure, Nova Instance Launch), retrieving its dynamic IP address. Once a new replica is instantiated, the reflector establishes both a BGP-LS and a PCEP session towards the new replica upon request from the manager (in the figure, Replica activation). After the PCEP and BGP-LS handshakes, the sessions are kept active for the purpose of continuous and dynamic synchronization.

The activation and de-activation of a replica is assumed to happen on a longer time scale than the on-demand provisioning of flexi-grid optical connections. This involves consuming the north-bound interface defined by any replica (in practice, this can be accomplished by the use of floating IP addresses or DNS round robin). When an instance receives an instantiation request, it sends a PCEP path initiate message (PCInit) to the reflector, which forwards it to the corresponding head end-node. Upon the successful establishment of the LSP, the head end-node sends a PCEP path computation report (PCRpt), which is forwarded to all the replicas. Figure 3 shows the architecture and the simplified flow of messages.
B. LSPDB Synchronization

The synchronization of the LSP database (LSPDB) (the set of active LSPs and their attributes) is done mainly by means of the PCEP stateful capabilities with instantiation protocol extensions and, in particular, the use of the PC initiate (PCInit) and PC report (PCRpt) messages. The PCInit message specifies that an LSP is to be instantiated (or released) in the network. The PCInit message includes, notably, the endpoint nodes, the path to use (in terms of the explicit route object, or ERO), and related objects to uniquely identify the LSP in the scope of the control domain (such as the LSP object and/or LSP symbolic name). For flexi-grid networks, additional parameters are included, such as the optical spectrum needed and allocated frequency slot. Likewise, the PC report (PCRpt) message is used to advertise the status of an LSP upon initiation or modification (commonly sent by the ingress PCC upon completion of the establishment procedure). It conveys the LSP operational status, LSP identifiers, and mapping with the GMPLS control plane constructs and other relevant information such as the detailed route and resources used. Consequently, forwarding or relaying the same PCRpt messages to multiple instances or replicas is an effective means to synchronize the LSPDB.

C. TED Synchronization

Topology synchronization happens at two different levels. At the lowest level, the PCEP/BGP-LS reflector is able to obtain an up-to-date, detailed view of the topology (TED) by passive inspection of OSPF-TE link state advertisements (LSAs). The TED can later be exported since, at the highest level, the synchronization between the reflector and the PCE replicas is done by means of the BGP-LS protocol with extensions for a flexi-grid. In short, BGP-LS refers to the extensions done to the well-known BGP protocol to support the exchange of link-state (topological) information between entities, and it is used to relay TE information, directly mapping OSPF-TE information objects to BGP-LS ones. From the perspective of protocol operation, the synchronization happens after the BGP-LS session has been established, where a BGP-LS peer can send UPDATE messages, including the MP_REACH attribute. The network layer reachability information contains the attributes of the network nodes and links: for a node, this is reflected in terms of IPv4 router ID, autonomous system (AS) identifiers, routing area, and other related properties. For a TE link, this means its source and destination node and the TE attributes. For this, the protocol uses the IPv4 addresses of the nodes, local node descriptors, and remote node descriptors. Additionally, in a flexi-grid network, unnumbered interfaces of the links as well as the maximum, unreserved, reservable bandwidths, the TE default metric, SRLGs, and a new bitmap reflecting the status of the different nominal central frequencies are also included. For further details please see, for example, [17].

Fig. 3. Basic message flow procedure to synchronize states. LSPDB synchronization is done mainly by means of the PCRpt messages, which are forwarded after a PCInit-driven successful provisioning of an LSP. TED synchronization is done by forwarding BGP-LS update messages (mapping OSPF-TE-inspected link state advertisements, LSAs).

Fig. 4. Summary of the workflow and message exchange showing the NMS/App and the different architecture components.
BGP-LS Updates as topology elements (networks and nodes), as shown in Fig. 4. This is the part that may present a bigger spike in control plane overhead, as we will see in Section V.

IV. TESTBED DEPLOYMENT

In order to validate the approach and architecture, we have deployed an experimental control plane testbed. In this section, we detail its different elements (see Fig. 5).

At the lowest level of the testbed, there is an optical transport network featuring a GMPLS control plane. The PCE acts as the centralized element behaving as an SDN controller, with or without delegation to the GMPLS. The control plane protocols PCEP, BGP-LS, OSPF-TE, and RSVP-TE have been extended to convey attributes such as the per-link nominal central frequencies (NCFs) and the availability of sliceable bandwidth variable transceivers (S-BVTs) [18].

The replica manager application is implemented in Python and executed in a dedicated GNU/Linux PC, presenting the system status to the operator (see Fig. 6). It is responsible for coordinating, authorizing, and reserving NFVI resources (in our case, within one single NFVI-PoP), consuming the north-bound APIs offered by the OpenStack controller. Other functions of the replica manager are to keep track of how many PCE replica instances have been allocated and their attributes, notifying the reflector of a new instantiated replica, and triggering the handshake and initial state synchronization. The consumed APIs concern the OpenStack keystone, glance, and nova services for identity, image, and computing resources management, respectively. There are no requirements for advanced networking services (that is, all instances are launched in the same private sub-network that is mapped to a physical network, providing connectivity between the replicas and physical systems). The management and orchestration application resides at host 10.1.6.202.

The deployed OpenStack setup includes an OpenStack controller node and one compute node, located at 10.1.6.207. Virtualized PCE instances (replicas) are all instantiated in the same administrative project, each one configured with only one network interface card, dynamically obtaining an IP address via an OpenStack managed DHCP agent, from a single, common, and private IP address pool (10.1.6.226; 227; …). The requirements of our implemented PCE/SDN controller are reasonable, so each instance can be launched using an OpenStack small flavor, which requires a single CPU core, 1 GB of RAM, and a small hard disk (20 GB). Finally, the GMPLS controllers in the aforementioned topology have a private IP address for management (from 10.1.6.101 to 10.1.6.128), while the controllers themselves use the node identifiers 172.16.102.101 to 172.16.102.114. The PCE reflector is itself located at the node 10.1.6.101. The reflectors and GMPLS controllers run on an Intel Core 2 Duo CPU at 2.66 GHz with 2–4 GB RAM.

In this paper, we have focused on the highly available features of the proposed architecture when a replica instance fails and how the PCEP and BGP-LS protocols are used to synchronize states between replicas. It is worth noting the high-availability requirements have further implications, such as accounting for the failure of compute nodes. While this can be, in part, mitigated by the operator by instantiating replicas in different compute nodes and different availability zones (making use of the cloud management capabilities), macroscopically, we leverage the benefits of the NFVI/cloud approach and existing mechanisms to move VMs and re-instantiate active ones in other compute hosts.

For our specific scenarios, the emulated data plane flexi-grid network topology (Fig. 7) represents a core Spanish network with 14 flexi-grid ROADM nodes and 14 client nodes, attached to each core node via a dedicated link and having a sliceable transceiver. There are 22 inter-ROADM bidirectional links and 14 attachment links. Each link has
128 nominal central frequencies. The ROADMs are assumed to be colorless, directionless, and contentionless, being able to switch any frequency slot from any port to any port.

V. EXPERIMENTAL PERFORMANCE EVALUATION

To carry out the performance evaluation, illustrate the main procedures, and obtain some meaningful performance indicators, we proceed with different experiments, instantiating up to two replicas (details of the replicas can be seen from the OpenStack Horizon web interface, Fig. 8).

A first quantitative result involves the time it takes to instantiate a virtual machine (VM) for an image containing the PCE software. In short, the latency for instantiating a replica depends on several factors. First, the capabilities of hosting nodes (compute nodes for OpenStack) can be quite diverse in terms of processing power and memory, including whether the CPU has instructions supporting virtualization. Second, there are the parameters associated with the VM request itself, such as the VM image size (commonly a qcow file) and the requested memory and CPU for the VM. As a guideline, with PCE software running on an Ubuntu GNU/Linux OS below 3 GB, a given PCE replica is typically operative in between 10 and 60 s, measured since the use of the REST interface to allocate VMs, until the replica manager is able to retrieve the IP address allocated to the replica by actively polling for its state.

A second performance indicator is strongly tied to the initial synchronization. Even if a given replica can be instantiated when there are no active LSPs (empty LSPDB), the initial synchronization of the TED will always be required. In this case, it is also dependent on the actual TCP implementation (the BGP-LS protocol is implemented over TCP) and different options (MTU, loss rate) that define the TCP application throughput. In our specific case, where components run in a dedicated LAN, the initial TED synchronization between the reflector and replica 1 (address 10.1.6.226) is carried out in a few seconds (1.15 s in the iteration, for which we show the Wireshark capture in Fig. 9). This includes not only the Update messages (which can be packed in one or multiple TCP segments), but also the BGP handshake (including the Open and KeepAlive messages).

Next, we proceed with experiments varying the offered traffic load, requesting LSP connections following a Poisson arrival process with the inter-arrival time set to a negative exponential of average of 3 s, and varying the holding times, depending on the desired traffic load. Other relevant parameters for the connections include the random selection of source and destination endpoints and uniformly chosen amongst distinct transceiver pairs (from Fig. 7. Spanish topology with 14 core nodes and 22 links (client nodes not drawn). Each link has 128 NCFs.

Fig. 7. Spanish topology with 14 core nodes and 22 links (client nodes not drawn). Each link has 128 NCFs.

128 nominal central frequencies. The ROADMs are assumed to be colorless, directionless, and contentionless, being able to switch any frequency slot from any port to any port.

V. EXPERIMENTAL PERFORMANCE EVALUATION

To carry out the performance evaluation, illustrate the main procedures, and obtain some meaningful performance indicators, we proceed with different experiments, instantiating up to two replicas (details of the replicas can be seen from the OpenStack Horizon web interface, Fig. 8).

A first quantitative result involves the time it takes to instantiate a virtual machine (VM) for an image containing the PCE software. In short, the latency for instantiating a replica depends on several factors. First, the capabilities of hosting nodes (compute nodes for OpenStack) can be quite diverse in terms of processing power and memory, including whether the CPU has instructions supporting virtualization. Second, there are the parameters associated with the VM request itself, such as the VM image size (commonly a qcow file) and the requested memory and CPU for the VM. As a guideline, with PCE software running on an Ubuntu GNU/Linux OS below 3 GB, a given PCE replica is typically operative in between 10 and 60 s, measured since the use of the REST interface to allocate VMs, until the replica manager is able to retrieve the IP address allocated to the replica by actively polling for its state.

A second performance indicator is strongly tied to the initial synchronization. Even if a given replica can be instantiated when there are no active LSPs (empty LSPDB), the initial synchronization of the TED will always be required. In this case, it is also dependent on the actual TCP implementation (the BGP-LS protocol is implemented over TCP) and different options (MTU, loss rate) that define the TCP application throughput. In our specific case, where components run in a dedicated LAN, the initial TED synchronization between the reflector and replica 1 (address 10.1.6.226) is carried out in a few seconds (1.15 s in the iteration, for which we show the Wireshark capture in Fig. 9). This includes not only the Update messages (which can be packed in one or multiple TCP segments), but also the BGP handshake (including the Open and KeepAlive messages).

Next, we proceed with experiments varying the offered traffic load, requesting LSP connections following a Poisson arrival process with the inter-arrival time set to a negative exponential of average of 3 s, and varying the holding times, depending on the desired traffic load. Other relevant parameters for the connections include the random selection of source and destination endpoints and uniformly chosen amongst distinct transceiver pairs (from

Fig. 8. Replica manager GUI embedding OpenStack Horizon interface showing the two instantiated PCE replicas in an OpenStack compute node. Replicas have IP addresses 10.1.6.226 (replica 1) and 10.1.6.227 (replica 2) and have been instantiated with small flavor (1 GB RAM, 20 GB HDD).

Fig. 8. Replica manager GUI embedding OpenStack Horizon interface showing the two instantiated PCE replicas in an OpenStack compute node. Replicas have IP addresses 10.1.6.226 (replica 1) and 10.1.6.227 (replica 2) and have been instantiated with small flavor (1 GB RAM, 20 GB HDD).

Fig. 9. Wireshark capture of the initial synchronization of the TED, showing the different BGP-LS update messages. Several update messages can be included in a single TCP segment, and the initial synchronization between the reflector and a replica takes 1.15, including the handshake.
the set of client-facing interfaces). Each connection requested client data rate (bandwidth parameter) is selected randomly as 100, 200, and 500 Gbps, with each PCE performing routing and spectrum assignment allocating the required optical frequency slot parameters (see [19]). For the dynamic setup and release of connections, the average provisioning time, as seen from the NMS that performs the request, is 155 ms, with values ranging from a minimum of 89 ms to a maximum of 360 ms.

At a given time, we instantiate a second PCE replica, measuring the time it takes to synchronize databases. The sync time roughly increases with the number of active LSPs up to approximately 2.05 s, which is obtained at 30 Erlangs. Macroscopically, it is easy to see that this LSPDB latency will be determined by the maximum number of LSPs that can be active at a given time. In our specific case, it was easy to give this maximum, since we have a limited number of usable transceivers (14), which limits the number of active LSPs. Figure 10 shows the LSPDB of the replica at a given time.

Finally, a new experiment that removes the transceiver limitation (just provisioning flexi-grid media channels, without interacting with transceivers) is run, theoretically having a larger potential number of concurrently active LSPs and allowing us to better measure the control plane overhead. In this case, we were just dealing with requests for the optical spectrum, requesting values for $m = 1, \ldots, 5$, resulting in $m \times 12.5 \text{ GHz}$.

The main parameters that impact control plane overhead in a replication-enabled scenario are, a priori, i) the number of active replicas, ii) the redundancy in terms of reflectors, and iii) the traffic arrival rate. The first one means a given reflector will need to forward as many copies of a topology or LSPDB update to as many replicas as possible in order to keep the synchronization between replicas. The second factor, the number of reflectors (in case a reflector fails), when deployed in simple yet inefficient approaches will also increase the number of individual messages linearly, since the reflectors will forward copies that may have already been received by each replica. Finally, the traffic pattern itself will determine the arrival and departures of LSPs (thus generating PCRpts accordingly and topology changes for at least the number of traversed links that change states).

To provide some numerical values, with 100 Erlangs, the replica 2 initial synchronization of approximately 90 LSPs happened in around 2.8 s, with 39 captured packets with an average packet size of 845 bytes, thus requiring a throughput of 0.113 Mbit/s. Note that, in practice, the synchronization delay is not necessarily linear with the number of active LSPs, since the Linux kernel is able to pack multiple PCEP PCRpt and BGP-LS messages into a single TCP segment. As a main guideline, in a dynamic operation close to the expected production systems and a real operation, the main factor of control plane overhead will be synchronizing the TED, since a single LSP generates multiple OSPF-TE LSAs (per each crossed link) that are mapped into BGP-LS Update messages (in our measurements, of around 248–348 bytes). In the case where this presents a scalability problem, it can be mitigated by applying thresholds and policies, at the expense of a slightly outdated TED.

Finally, another experiment is set up to stress the system: we deploy two replicas, wait until the system
has converged after the initial TED synchronization, and sequentially launch 100 LSPs (a new LSP is set up when the previous one has been acknowledged as established), monitoring the real-time synchronization with the two replicas. In total, the LSPs are set up and the TED/LSPDBs are synchronized to the new state in less than 12 s, requiring, on average, 0.49 Mbit/s, as seen from the reflector (see Fig. 11).

VI. CONCLUSIONS

The successful deployment of centralized control plane functions (SDN controllers or specific functions, such as a PCE) is constrained by stringent requirements regarding not only dynamicity, performance, and cost efficiency, but also high availability, robustness, and fault tolerance. The ultimate adoption of this technology by carriers and operators is conditioned by the availability of “carrier-class” solutions and infrastructures that meet such requirements while still delivering the benefits associated with SDN. In this scope, the use of transport-NFV concepts can fulfill such requirements, enabling much-wanted features such as in-operation modifications, software image or policy upgrades, and hot-swapping.

While the concept and use of functional replication for high availability is quite well understood, the need for synchronization between databases is an issue to solve. On the one hand, the associated information and data models need to be clearly defined and, on the other hand, deployed solutions should not be exposed to vendor lock-in or proprietary products, for it should be possible to use implementations from different vendors and open solutions. We proposed an architecture and the use of existing open and standard PCEP and BGP-LS protocols for the synchronization of the main considered databases, namely, the traffic engineering (network topology) one and the LSP database (keeping a state of active connections), thus avoiding the aforementioned vendor lock-in.

The main performance considerations are related to the synchronization delays and control plane overhead. These performance indicators need to be addressed, keeping in mind the initial assumptions related to i) the dynamicity and associated timescales of traffic, which are the main sources of database changes, and ii) the availability of a deployed and dedicated control plane and management network in which control plane links have fairly consistent bandwidths and processing/transmission delays, along with the ability to deploy operator private clouds for the deployment of internal NFV services. Our experimental tests show that synchronization between replicas is of the order of a few seconds for the initial sync and the order of milliseconds for subsequent updates, with reasonable control plane overhead for the targeted deployment scenarios. Further work is still needed in heavily constrained scenarios in which the data communications network that supports the control plane may limit performance.

ACKNOWLEDGMENT

This work has been partially funded by the European Commission H2020 project 5G-Crosshaul project under grant agreement H2020-671598, as well as by the Spanish Ministry of Economy and Competitiveness (MINECO) project DESTELLO (TEC2015-69256-R).

REFERENCES

[1] A. Farrel, J.-P. Vasseur, and J. Ash, “A path computation element (PCE)-based architecture,” IETF RFC 4655, Aug. 2006 [Online]. Available: http://www.ietf.org/rfc/rfc4655.txt.
[2] J. Vasseur and J. L. Roux, “Path computation element (PCE) communication protocol (PCEP),” IETF RFC 5440, Mar. 2009 [Online]. Available: http://www.ietf.org/rfc/rfc5440.txt.
[3] J. Vasseur, R. Zhang, N. Bitar, and J. L. Roux, “A backward-recursive PCE-based computation (BRPC) procedure to compute shortest constrained inter-domain traffic engineering label switched paths,” IETF RFC 5441, Apr. 2009 [Online]. Available: http://www.ietf.org/rfc/rfc5441.txt.
[4] D. King and A. Farrel, “A PCE-based architecture for application-based network operations,” IETF RFC 7491, 2015.
[5] E. Crabbe, I. Minei, S. Sivabalan, and R. Varga, “PCE extensions for PCE-initiated LSP setup in a stateful PCE model,” IETF Internet Draft, Oct. 2015.
[6] S. Sivabalan, J. Medved, C. Filsfils, R. Raszuk, V. Lopez, J. Tantsura, W. Henderickx, and J. Hardwick, “PCE extensions for segment routing,” IETF Internet Draft, Mar. 2016.
[7] Q. Zhao, R. Li, K. Ke, L. Fang, C. Zhou, and B. Zhang, “Use cases for using PCE as the central controller (PCECC) of LSPs,” IETF Internet Draft, Mar. 2017.
[8] ETSI Group Specification, “Network function virtualization (NFV): Architectural framework,” ETSI GS NFV 002 v1.1.1, 2013.
[9] R. Vilalta, R. Munoz, R. Casellas, R. Martinez, V. Lopez, and D. Lopez, “Transport PCE network function virtualization,” in European Conf. on Optical Communication (ECOC), Cannes, France, Sept. 2014.
[10] R. Munoz, R. Vilalta, R. Casellas, R. Martinez, T. Szyrkowiec, A. Autenrieth, V. Lopez, and D. Lopez, “Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks,” J. Opt. Commun. Netw., vol. 7, no. 11, pp. B62–B70, 2015.
[11] H. Gredler, J. Medved, S. Previdi, A. Farrel, and S. Ray, “North-bound distribution of link-state and TE information using BGP,” IETF RFC 7752, May 2014.
[12] R. Casellas, R. Vilalta, R. Martinez, and R. Muñoz, “Active stateful PCE high-availability for the control of flexigrid networks with network function virtualization enabled replication,” II Optical Fiber Communication Conf. and Expo. (OFC), Anaheim, CA, Mar. 2016.
[13] ETSI Group Specification, “Network functions virtualisation (NFV); infrastructure overview,” ETSI GS NFV-INF 001 v1.1.1, 2015.
[14] Open Networking Foundation (ONF), “OpenFlow switch specification,” version 1.4 (Wire protocol 0 × 5), Oct. 2013 [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf.
[15] R. Casellas, R. Martínez, R. Muñoz, R. Vilalta, L. Liu, T. Tsuritani, and I. Morita, “Control and management of flexi-grid optical networks with an integrated stateful PCE and OpenFlow controller,” J. Opt. Commun. Netw., vol. 5, no. 10, pp. A57–A65, 2013.

[16] T. Bates, E. Chen, and R. Chandra, “BGP route reflection—An alternative to full mesh IBGP,” IETF RFC 4456, Apr. 2006 [Online]. Available: http://www.ietf.org/rfc/rfc4456.txt.

[17] O. González de Dios, R. Casellas, R. Morro, F. Paolucci, V. López, R. Martínez, R. Muñoz, R. Vilalta, and P. Castoldi, “Multipartner demonstration of BGP-LS-enabled multidomain EON control and instantiation with H-PCE [Invited],” J. Opt. Commun. Netw., vol. 7, no. 12, pp. B153–B162, 2015.

[18] O. González de Dios, R. Casellas, F. Paolucci, A. Napoli, L. Gifre, A. Dupas, E. Hugues-Salas, R. Morro, S. Belotti, G. Meloni, T. Rahman, V. López, R. Martínez, F. Fresi, M. Bohn, S. Yan, L. Velasco, P. Layec, and J.-P. Fernández Palacios, “Experimental demonstration of multivendor and multidomain EON with data and control interoperability over a Pan-European test bed,” J. Lightwave Technol., vol. 34, no. 7, pp. 1610–1617, 2016.

[19] R. Martínez, R. Casellas, R. Vilalta, and R. Muñoz, “GMPLS/PCE-controlled multi-flow optical transponders in elastic optical networks,” J. Opt. Commun. Netw., vol. 7, no. 11, pp. B71–B80, 2015.

Ramon Casellas (IEEE SM’12) graduated from the telecommunications engineering programs in 1999 from both the UPC-BarcelonaTech University and ENST Telecom ParisTech in an Erasmus/Socrates double-degree program. After working as an undergraduate researcher at both France Telecom R&D and British Telecom Labs, he completed a Ph.D. in 2002. He worked as an associate professor in the Networks and Computer Science Department of the ENST (Paris) and joined the CCTC Optical Networking Area in 2006, with a Torres Quevedo research grant. He is currently a senior research associate and the coordinator of the ADRENALINE testbed. He has been involved in several international R&D and technology transfer projects. His research interests include network control and management, GMPLS/PCE architecture and protocols, software-defined networking, and traffic engineering mechanisms. He contributes to IETF standardization within the CCAMP and PCE working groups. He is a member of the IEEE Communications Society and a member of the Internet Society.

Ricard Vilalta graduated in telecommunications engineering in 2007 and received a Ph.D. in telecommunications in 2013, both from the Universitat Politècnica de Catalunya (UPC), Spain. He also studied audiovisual communication at the UOC (Open University of Catalunya) and has a master’s degree in technology-based business innovation and administration from Barcelona University (UB). Since 2010, Ricard Vilalta has been a researcher at CTTC, in the Optical Networks and Systems Department. His research is focused on optical network virtualization and optical OpenFlow. He is currently a Research Associate at the Open Networking Foundation.

Ricardo Martínez (IEEE SM’14) graduated in telecommunications engineering from UPC-BarcelonaTech in 2002 and obtained a Ph.D. in 2007. He has been actively involved in several publicly funded (national and EU) R&D as well as industrial technology transfer projects. Since 2013, he has been a Senior Researcher in the Communication Networks Division (CND) at the CTTC. His research interests include control and network management architectures, protocols and traffic engineering mechanisms for next-generation packets, and optical transport networks within aggregation/metro and core segments.

Raul Muñoz (IEEE SM’12) graduated in telecommunications engineering in 2001 and received a Ph.D. in telecommunications in 2005, both from the Universitat Politècnica de Catalunya (UPC), Spain. After working as an undergraduate researcher at Telecom Italia Lab (Turin, Italy) in 2000 and as an assistant professor at the UPC in 2001, he joined the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) in 2002. Currently, he is a Senior Researcher, Head of the Optical Network and System Department, and Manager of the Communication Networks Division. Since 2000, he has participated in several R&D projects funded by the European Commission’s Framework Programmes (FP7, FP6, and FP5) and the Spanish Ministries, as well as technology transfer projects. He has led several Spanish research projects and currently leads the European Consortium of the EU-Japan project STRAUSS. His research interests include control and management architectures, and protocols and traffic engineering algorithms for future optical transport networks.