# A Survey on the Contributions of Software-Defined Networking to Traffic Engineering

Alaitz Mendiola, Jasone Astorga, Eduardo Jacob, *Senior Member, IEEE*, and Marivi Higuero

***Abstract*—Since the appearance of OpenFlow back in 2008, software-defined networking (SDN) has gained momentum. Although there are some discrepancies between the standards developing organizations working with SDN about what SDN is and how it is defined, they all outline traffic engineering (TE) as a key application. One of the most common objectives of TE is the congestion minimization, where techniques such as traffic splitting among multiple paths or advanced reservation systems are used. In such a scenario, this manuscript surveys the role of a comprehensive list of SDN protocols in TE solutions, in order to assess how these protocols can benefit TE. The SDN protocols have been categorized using the SDN architecture proposed by the open networking foundation, which differentiates among data-controller plane interfaces, application-controller plane interfaces, and management interfaces, in order to state how the interface type in which they operate influences TE. In addition, the impact of the SDN protocols on TE has been evaluated by comparing them with the path computation element (PCE)-based architecture. The PCE-based architecture has been selected to measure the impact of SDN on TE because it is the most novel TE architecture until the date, and because it already defines a set of metrics to measure the performance of TE solutions. We conclude that using the three types of interfaces simultaneously will result in more powerful and enhanced TE solutions, since they benefit TE in complementary ways.**

***Index Terms*—Software-defined networking, traffic engineering, network resource optimisation, flow granularity.**

## I. Introduction

DURING the last decade, Software-Defined Networking (SDN) has emerged as a revolutionary networking paradigm, and has gained the attention of both the industry and the academia. Actually, SDN has been included in several reports as one of the most disruptive and interesting technologies in the networking area [1], [2]. Several factors have been decisive for the success of SDN. On the one hand, the vast range of SDN-enabled networking devices available in the market has been primordial. Both classical manufacturers such as Cisco [3], HP [4] or NEC [5] and novel manufacturers such as Corsa [6] are commercialising SDN products. On the other hand, the availability of open-source controllers with OpenFlow [7] support has fostered the implementation of SDN applications.

For the moment, there is no clear consensus about what SDN is and how it is defined. Most definitions agree on the availability of open programmable interfaces at networking devices and the separation of the control and forwarding planes. Nevertheless, most agents involved in the standardisation of SDN do agree on some possible applications. In addition to its utilisation in Data Center (DC) networks [8], campus networks [9], [10] and as an enabler for Network Functions Virtualisation (NFV) [11], SDN appears as a promising candidate to enhance current Traffic Engineering (TE).

TE has always been one of the most challenging topics in communication networks [12]. As stated by the Internet Engineering Task Force (IETF), TE deals with the performance optimisation of operational networks, and plays a key role in the provisioning of services with Quality of Service (QoS) [13]. Being aware of the benefits that SDN can bring to TE, telecom companies such as AT&T have started to work on SDN-based solutions [14]. Similarly, Internet Service Providers (ISP) and Research and Education Networks (REN) have also started to analyse the applicability of SDN to their transport networks.

A service typically provided by the ISPs and RENs to satisfy the increasing demand of users with short-term, high-capacity and high-availability demands is Bandwidth on Demand (BoD). Given the crucial role of TE for the provisioning of this type of service, many RENs have started to design and deploy SDN-based solutions. For instance, the Energy Sciences Network (ESnet) [15], the high-speed computer network serving the United States Department of Energy, is evolving the On-Demand Secure Circuits and Advance Reservation System (OSCARS) [16] from the Path Computation Element (PCE)-based architecture [17] towards SDN. Likewise, the pan-European research and education network Géant is following a similar approach to improve AutoBAHN, their BoD service provisioning tool [18].

SDN appears to RENs and ISPs as the enabler for next generation TE solutions thanks to its high network programmability and the possibility it offers to apply new and powerful TE strategies. For example, the logically centralised control plane of OpenFlow makes possible to use PCE-like dedicated elements. This allows to perform complex path

Manuscript received July 17, 2015; revised January 29, 2016, July 11, 2016, and October 20, 2016; accepted November 22, 2016. Date of publication December 1, 2016; date of current version May 31, 2017. This work was supported in part by the EC through the Horizon 2020 Research and Innovation Programme (GN4) under Grant 691567, in part by the Spanish Ministry of Economy and Competitiveness under the Secure Deployment of Services Over SDN and NFV-based Networks Project S&NSEC under Grant TEC2013-47960-C4-3-P, and in part by the UPV/EHU (Training and Research Unit) under Grant UFI11/16.

The authors are with the Department of Communications Engineering, Faculty of Engineering, University of the Basque Country UPV/EHU, 48013 Bilbao, Spain (e-mail: alaitz.mendiola@ehu.eus; jasone.astorga@ehu.eus; eduardo.jacob@ehu.eus; marivi.higuero@ehu.eus).

Digital Object Identifier 10.1109/COMST.2016.2633579

computations and to easily deploy novel advanced reservation mechanisms. Furthermore, it allows to extend current TE-dependant services to include fast failure recovery mechanisms (e.g., Géant's BoD service does not provide resilience) and enables the utilisation of more convenient flow-based TE strategies. In this regard, it is worth mentioning that most SDN technologies are flow-oriented, where a flow is defined as the sequence of packets identified by a set of common header fields. As a consequence, it is possible to perform per-flow operations to increase the network resource utilisation, such as *flow relocation* or *flow disaggregation*. In the former, a flow can be moved to an alternative path, whereas in the latter, a flow is divided into multiple sub-flows in order to accept new service demands or balance the load. Furthermore, mechanisms such as flow relocation and disaggregation can be applied taking into consideration the characteristics of the traffic being forwarded, which as stated in [19], improves considerably the QoS metrics performance and therefore, TE strategies' performance.

### A. Contributions

This manuscript provides a literature review of the SDN-based TE solutions published until 2015. Moreover, it also analyses the impact of SDN on TE, making special emphasis on its contribution to the optimisation of the network resource utilisation. Among the plethora of technologies usually included in the SDN environment, this survey analyses a comprehensive list of SDN protocols that impact TE. The protocols have been categorised taking into account the interface in which they operate, considering the SDN architecture proposed by the Open Networking Foundation (ONF). This architecture defines three interfaces: the Data-Controller Plane Interface (D-CPI), the Application-Controller Plane Interface (A-CPI) and the Management Interface (MI).

All in all, this survey provides a complete view of the impact of SDN on TE, in which SDN-based TE solutions applied to a variety of scopes are surveyed, such as Wide Area Networks (WAN), DC networks, or inter-DC networks. This paper concludes that the contributions to TE of the analysed SDN protocols is tightly coupled to the interface in which they operate. In this regard, the protocols operating at the D-CPI interface are the ones with a greater impact on TE, although the best course of action to provide enhanced TE in future networks is to use complete SDN frameworks with support for protocols operating at the three different interfaces.

But most importantly, this survey answers the question of how and why SDN can contribute to TE. In summary, SDN can benefit TE thanks to the higher granularity available at the forwarding devices, making possible the utilisation of flow disaggregation mechanisms to improve the network resource utilisation. On the other hand, the logically centralised controller plane allows the implementation of advanced path computation algorithms fed with up-to-date network state information. Furthermore, the high programmability and the logically centralised controller plane of SDN provides the means to react upon network failures.

### B. Related Work

As far as we know, this is the first survey about SDN entirely focused on its applicability to TE, which studies the impact of the different SDN interfaces on TE. Surveys like [20] and [21] provide a general overview of SDN, while other papers are focused on more specific topics like security [22]–[24], programmability [25], network virtualisation [26], the controller plane [27], or its application to other network types such as optical networks [28] or mobile networks [29]. Most surveys are focused entirely on OpenFlow and its applications [30]–[32], some of which deal briefly with TE [33], [34]. However, none of the surveys analyse how different SDN protocols impact TE, and how the impact depends on the interface at which the protocol operates.

### C. Structure of This Paper

This document is structured as follows. First, Section II provides some background information about TE in packet networks, identifying the most common performance objectives and techniques used today. Furthermore, it briefly reviews the evolution of TE, and summarises the limitations found in today's TE solutions that SDN-based approaches can solve. Then, Section III introduces SDN and the architectures proposed by the ONF and the Software-Defined Networking Research Group (SDNRG). This section also introduces the taxonomy used in this paper to categorise the SDN protocols, which is based on the interface types defined by the ONF's SDN architecture, namely D-CPI, A-CPI and MI. Section IV provides a review of the SDN protocols analysed in this paper, namely ForCES, OpenFlow, I2RS, BGPLS/PCEP, ALTO, OVSDB Management Protocol, NETCONF and OF-CONFIG. Later, Section V surveys current TE solutions where D-CPI protocols have been used, while Sections VI and VII do the same with A-CPI and MI protocols respectively. Then, Section VIII provides a qualitative evaluation of the analysed protocols taking as reference the PCE-based architecture. Section IX presents the lessons learnt in the form of a comprehensive list of future research areas. Finally, Section X summarises the conclusions. Table I presents the glossary used in this manuscript.

## II. Background: TE in Packet Networks

This section introduces TE in packet networks and presents a list of common TE performance objectives and the techniques used to achieve them. Moreover, a brief overview of the evolution of TE in packet networks until the appearance of the PCE-based architecture is included. Finally, the limitations of current TE solutions are described, in order to identify how SDN can improve TE.

### A. Definition of TE

In communication networks, TE consists in the application of strategies and scientific principles to optimise the performance of operational networks [13]. The general objective of TE is to route traffic in a data network so that traffic demands are met, by optimising a selected performance objective. This usually involves the computation of a path between a

TABLE I
GLOSSARY

| | | | |
|---|---|---|---|
| **CE** | Control Element | **OCS** | OpenFlow Capable Switch |
| **CLI** | Command Line Interface | **OLS** | OpenFlow Logical Switch |
| **CSPF** | Constraint-based Shortest Path First | **ONF** | Open Networking Foundation |
| **DC** | Data Center | **ONOS** | Open Network Operating System |
| **D-CPI** | Data-Controller Plane Interface | **OSCARS** | On-Demand Secure Circuits and Advance Reservation System |
| **E2E** | End-to-end | **OSPF** | Open Shortest Path First |
| **ERO** | Explicit Route Object | **OVS** | Open vSwitch |
| **FE** | Forwarding Element | **OVSDB** | Open vSwitch DataBase |
| **FIB** | Forwarding Information Base | **P2P** | Peer-to-Peer |
| **ForCES** | Forwarding and Control Element Separation | **PCC** | Path Computation Client |
| **GMPLS** | Generalized MPLS | **PCE** | Path Computation Element |
| **H-PCE** | Hierarchical Path Computation Element | **PCEP** | Path Computation Element Communication Protocol |
| **I2RS** | Interface to the Routing System | **QoE** | Quality of Experience |
| **IETF** | Internet Engineering Task Force | **QoS** | Quality of Service |
| **IGP** | Interior Gateway Protocol | **REN** | Research and Education Network |
| **ISP** | Internet Service Provider | **RIB** | Routing Information Base |
| **LFB** | Logical Forwarding Block | **RSVP** | Resource Reservation Protocol |
| **LLDP** | Link Layer Discovery Protocol | **SDN** | Software-Defined Networking |
| **LSP** | Lable Switched Path | **SDO** | Standards Development Organization |
| **M-PCE** | Multiple Path Computation Element | **TCAM** | Ternary Content-Addressable Memory |
| **MPLS** | Multi-Protocol Label Switch | **TE** | Traffic Engineering |
| **NE** | Network Element | **TED** | Traffic Engineering Database |
| **NSP** | Network Service Provider | **WAN** | Wide Area Network |
| **OCP** | OpenFlow Configuration Point | **WG** | Working Group |

given source-destination pair, or the computation of multiple paths to share the load according to specific traffic-splitting ratios.

### *B. TE Performance Objectives and Techniques*

The performance optimisation of a network is an iterative process in which new technologies and optimisation mechanisms are continuously required [12]. When a TE solution is designed, the performance objective must be selected carefully, since different performance objectives can be mutually exclusive. This section presents a comprehensive list of performance objectives and the techniques that are used for their optimisation.

*1) Congestion Minimisation:* In an operational IP context, congestion is one of the most important problems, since it affects delay, jitter and packet loss [12]. Therefore, it is one of the most critical performance objectives in current communication networks.

Congestion minimisation can be achieved using different techniques:

- Sharing the network resources by multiple traffic streams.
- Re-allocating network resources by redistributing the traffic over the infrastructure.
- Denying the access to congested resources. Once the congestion is detected, the TE system can only assign uncongested resources to new demands.

Sharing the network resources by multiple traffic streams is of special relevance for the congestion minimisation, since it is a proactive technique aiming to avoid congestion. This is often achieved by minimising the links' utilisation solving a traditional optimisation problem known as the minimum cost multi-commodity flow problem [35]. This optimisation problem has been widely studied in the literature [36]–[39] and it is an ongoing research work. The main purpose of this approach is to balance the traffic load along the network, which results in a better network utilisation. This is achieved by splitting the traffic into a set of streams that are routed through multiple paths connecting the ingress-egress router pair. As a consequence, the load is balanced among a higher number of network resources, resulting in a smaller amount of packets queued at the forwarding devices and less occupied bandwidth at the links.

A common approach to solve the multi-commodity flow problem is the computation of an optimal splitting ratio for the incoming traffic demand. Traffic splitting can be achieved in different ways. On the one hand, the most simple mechanism to split the traffic is on a per-packet basis, for example in a round-robin fashion. On the other hand, it is also possible to split the traffic on a per-flow basis, by applying a hash function over a set of the packets' header fields. Current commercial routers can be configured to divide traffic based on the result of hashing different TCP/IP header fields.

As mentioned before, congestion can result in a higher end-to-end (E2E) delay and packet loss. In other words, congestion minimisation can be considered a general performance objective that has a direct impact on more specific performance objectives such as the E2E delay and packet loss minimisation. Therefore, the techniques used to minimise the congestion are also useful to minimise these two performance parameters. Notwithstanding, there are other factors besides congestion that can be the root cause of the E2E delay and the packet loss, requiring the utilisation of more specific techniques for their optimisation. As a consequence, the following sections present other techniques to deal with the minimisation of the E2E delay and the packet loss, where they are considered independent performance objectives.

*2) E2E Delay Minimisation:* A typical network-related performance objective that impacts QoS and Quality of Experience (QoE). The minimisation of the E2E delay is essential for critical real-time communications. It can be applied on a per-flow basis or as an overall objective that takes into account the E2E delay of all the packets transmitted in the network. One of the most common techniques to minimise the E2E delay is Constrained-Shortest Path First (CSPF), where the E2E delay is used as a constraint for the path selection [40].

*3) Packet Loss Minimisation:* Another typical network-related performance objective that can also be evaluated per-flow or network-wide. Besides congestion, packet loss can also be the result of failures in the network, such as forwarding devices and links, requiring additional techniques to increase the failure recovery capabilities of the network. This performance objective is usually tackled by over-provisioning the network to increase resilience [41] by means of redundant resources to be used in case of failure. In fact, if multiple paths are available to convey traffic between a given source-destination pair, traffic can be re-routed among the available paths when one of them suffers a disruption.

*4) Energy Consumption Minimisation:* This is a performance objective that does not necessarily match a network performance parameter. It is widely used in the scope of *green computing* [42], which aims to lower the environmental impact of Information and Communication Technologies (ICT). This performance objective is usually optimised either by adapting the rate of network operation to the offered workload or by reducing the amount of active resources [43]. In this last case, the energy consumption is reduced when traffic is gathered into a few paths and unused line cards can be powered down in the network equipment. This is a good example of how the different performance objectives can be mutually exclusive, since the minimisation of the energy consumption and the congestion minimisation cannot be achieved at the same time when this approach is followed.

*5) QoE Maximisation:* The QoE, as defined by the European Telecommunications Standards Institute (ETSI), is a parameter that measures the performance of using an ICT service or product taking into account objective technical parameters, like QoS, and subjective psychological parameters [44]. In other words, it is a parameter that gets affected by all the elements involved in the E2E transmission, including the end devices, environmental factors such as the light and the network performance. Therefore, the QoE maximisation also requires the optimization of the network performance, which is inside our scope of interest. Notwithstanding, maximising the QoE does not always imply the maximisation of the network throughput, and a correlation between the QoE criteria and the network-related performance parameters needs to be defined, as argued in [37].

*6) Resource Utilisation Optimisation:* The optimisation of the resource utilisation is another performance objective. For example, computation, buffer space and bandwidth are resources that need to be efficiently used, since they can impact congestion and other parameters. In addition, a good utilisation of the resources helps network operators to serve a higher number of service demands without increasing their costs. That is, a good utilisation of the network resources allows network operators to allocate a higher amount of traffic.

A common approach to optimise the network resource utilisation is to schedule well characterised data transfers. For instance, network operators can decide to transmit backup traffic between various data centers during the night hours, since more resources are available at that time. Advance reservation systems [45] are also used for the provisioning of the BoD service, since they are meant to optimise the bandwidth utilisation. Advance reservation systems allow to maintain a detailed inventory of the resource consumption over time and a better assignment of resources to satisfy new demands.

### *C. Evolution of TE*

According to Awduche [46], TE is considered a control issue where the element in charge of TE acts as a controller in an adaptive feedback control system. In this schema, available control actions must include the modification of traffic management parameters, the modification of parameters associated with routing and the modification of the attributes and constraints associated with resources. Over the years, TE in packet networks has been tackled using different approaches, as mentioned in the RFC 3272 [12]. However, first proposals were not appropriate for TE because they did not satisfy the aforementioned requirements posed in [46].

First routing protocols in the ARPANET were highly scalable and resilient distributed protocols but without the flexibility required by TE [47]. When the Internet became a reality, the adaptive routing protocols used in ARPANET were substituted by dynamic routing protocols. Though, the Interior Gateway Protocols (IGP) that run on the Internet were neither appropriate for TE, since the route selection was based on shortest path algorithms fed with additive link metrics and not on the resources available in the network.

As a first approach to take advantage of TE strategies in the Internet, overlay models were used, like IP over ATM [48]. By means of a secondary technology capable of establishing virtual circuits, point-to-point links between IP routers were served. This way, arbitrary virtual topologies were defined and superimposed onto the physical network topology that resulted on a much easier TE operation. Nevertheless, the use of overlay technologies increased the overall complexity of the network operation. In addition, these strategies were usually based on circuit pre-provisioning, given the lack of efficient mechanisms to create new circuits on demand.

Parallel in time, the Nimrod routing architecture was designed to provide service-specific routing taking into account multiple constraints [49]. Nimrod was based on the distribution of link-state maps that abstracted network connectivity and services information, and introduced the concept of explicit routing to allow the selection of paths at originating nodes. Even if this protocol was never deployed in the public Internet, it introduced some concepts adopted in more recent proposals, like explicit routing.

In the next iteration, Shortest Path First (SPF) algorithms that take into account the requested Type of Service (ToS) were proposed [50]. These approaches lead to an unfair usage of the network resources, where the shortest paths end up congested and other paths remain underutilised. Next, traffic splitting was introduced by means of the Equal Cost Multipath Protocol (ECMP), where traffic was split equally among all the available shortest paths [51]. Although the utilisation of traffic splitting mechanisms is not always optimal, which will be further explained in Section II-D, the use of ECMP is very extended, and many manufacturers support this protocol in their networking devices.

Later, the MPLS forwarding architecture emerged to provide flexibility and to increase the performance and scalability of the network layer routing [52]. In MPLS, packets are transmitted between the edge nodes of an MPLS domain using Label Switched Paths (LSP) and the forwarding decisions at each node are done based on previously assigned *labels*. This results in a higher network performance, since the forwarding decision is performed using a single header field. MPLS is useful for TE because it provides most of the functionalities available from the overlay model in an integrated manner and at a lower level [13]. It supports the creation of explicit LSPs that are not constrained by the destination-based forwarding paradigm, facilitating the multi-path routing. Furthermore, MPLS is appropriate for TE because it supports explicit routing and allows traffic aggregation and disaggregation, while the classical destination-based IP forwarding only supports aggregation based on IP subnetting. In addition, with MPLS it is relatively easy to integrate constraint-based routing frameworks.

Finally, the IETF proposed the PCE-based architecture for MPLS and Generalized MPLS (GMPLS) networks [17], which extends packet switching capabilities of MPLS to an open set of networking and switching methods. The PCE-based architecture proposed a dedicated element to be in charge of the path computation making possible the application of complex algorithms such as CSPF. In addition, it supports the instantiation of point-to-point and point-to-multipoint LSPs, which is known as explicit routing. This architecture can be used when the path computation is CPU-intensive or when there is no visibility of all the network elements involved. As a consequence, this architecture is being adopted for TE [53], and can be used in intra-domain, inter-domain and inter-layer contexts.

### D. Limitations of Latest TE Solutions

Since this survey is focused on the contributions of SDN to TE, it is important to identify the limitations of current TE solutions to determine in which areas SDN can pose an improvement. MPLS is the technology most widely used by network operators for TE, therefore, this section focuses on identifying the limitations found in MPLS-TE, including the limitations present in the PCE-based architecture, considered the most novel approach for TE in this type of networks. This section first identifies the limitations, and afterwards presents how SDN can improve each of them.

*1) Unrealistic Traffic Splitting Ratios:* Congestion minimisation is often achieved using multiple paths, but the current mechanisms to split the traffic present some limitations. On the one hand, per-packet traffic splitting results in an excessive packet reordering in the destination endpoint node, which is undesirable, especially for TCP applications. As explained in [54], packet-level multipath routing can entail TCP segments arriving out of order to the destination entity, triggering the TCP congestion avoidance mechanism unnecessarily and resulting in the application throughput and the whole network performance being degraded. In addition, jitter can occur, requiring large buffers to temporarily store the packets received out of order. On the other hand, per-flow traffic splitting allows individual TCP or UDP flows to be distinguished, avoiding the traffic reordering problem. Nevertheless, the traffic splitting granularity is determined by the forwarding element and the hash function that is used to split the traffic. This granularity does not necessarily need to be the same granularity demanded by the TE solution, resulting in the assignment of inappropriate traffic ratios to each path. As a result, the overall network performance and the capacity of the TE mechanism to deal with congestion may not be optimal. In addition, traffic splitting mechanisms, such as ECMP, may not take into account the potential congestion of the shortest paths used to balance the traffic, which results in a poor performance [55].

*SDN can help overcome this limitation thanks to the higher granularity available at the forwarding devices.*

*2) Unoptimal Path Computation Algorithms:* Path computation and the required resource handling in MPLS-TE present some limitations as well. Pathak *et al.* [56] detected that some of the links in an over-provisioned network were experiencing some latency. They analysed their MPLS-TE solution and deducted that latency inflation was a consequence of both the CSPF algorithm that they used and the continuous path changes that occur as a consequence of the *autobandwidth* algorithm, which is provided by many MPLS vendors to automatically adjust the reserved bandwidth of the LSPs depending on the traffic demand.

*SDN can help overcome this limitation thanks to the possibility that it provides to perform the path computation at a logically centralised controller.*

*3) TE Databases Do Not Reflect the Network State in Real-Time:* Although the PCE-based architecture can improve such limitations of classic MPLS-TE, it also presents some limitations on its own. In the PCE-based architecture, path computation is done using the TE information stored in the Traffic Engineering Database (TED). This database holds an inventory of the resources available in the network, information that is used by the PCE to compute the paths. Notwithstanding, according to the RFC 4655 [17], the TED does not always reflect the network state in real-time. When the TED is not properly synchronised with the network state, which can occur at specific times, the rate of wrong computed paths may increase.

*SDN can help overcome this limitation thanks to its logically centralised control plane, which is aware of the network state in real-time.*

*4) Long Convergence Times of Distributed Protocols:* Another important limitation that can be found in MPLS-TE and in the PCE-based architecture is their dependence on RSVP-TE. When a network device requests a path, the PCE replies with the computed path information. Then, the network device uses the distributed protocol RSVP-TE to inform the other nodes. As a consequence, the establishment of the path can take more time than the one that is required by a centralised control plane with out-of-band programming capabilities. This has direct impact on the network stability and will impact on already established data flows. This fact also affects scalability as defined by the RFC 4655, because RSVP-TE is an in-band signalling protocol. This is a limitation that most MPLS-TE based solutions present, as they all rely on RSVP-TE.

*SDN can help overcome this limitation thanks to the high network programmability that it provides, especially when the control of the network resources is done out-of-band.*

All in all, TE has significantly evolved in MPLS networks. From the early implementations with MPLS-TE to the more advanced PCE-based solutions, the use of network resources is increasingly better. However, these MPLS-based TE solutions still present some limitations, such as the problems with the network state representation or the time required by the Label Distribution Protocol (LDP) to setup the LSPs. As a consequence, new approaches are envisaged, being the ones based on SDN the natural path to follow.

## III. Software-Defined Networking

This section presents the fundamentals of SDN, including a brief introduction to the history of SDN, its definition and the proposed architectures. In addition, the taxonomy used to categorise the different SDN protocols is presented.

### A. Fundamentals of SDN

SDN is the result of three key research areas very popular since their inception in the mid 90s [57]. First, proposals like Open Signaling [58] and Active Networking [59] pushed in favor of *network programmability* by means of open interfaces and code piggibacked inside the user messages respectively. Second, the *control and data plane separation* brought to the fore the possibility to control the network from an external entity [60] and the transition towards a logically centralised control plane [61]. Finally, the appearance of *network operating systems* and the clean slate approach proposed at the 4D [62] project led to the release of the OpenFlow switch specification and the SDN revolution.

According to the ONF [63], the changing traffic patterns within an enterprise DC, the need to accommodate the traffic of new personal devices in a fine-grained manner, the rise of cloud services and the associated increasing demand for network capacity are key computing and communication trends that require a new network paradigm such as SDN.

*1) Definition:* The ONF defines SDN as an emerging network architecture where the network is directly programmable and where the control and forwarding planes are decoupled. One of the main characteristics of SDN is that the intelligence is logically centralised in SDN controllers. Such controllers maintain a global view of the network, which results in the network appearing to the applications and policy engines as a single, logical switch [64].

With SDN, network design and operation are simplified because the entire network can be controlled from a logically centralised point using open interfaces. Network control becomes vendor-independent and the utilisation of simpler network devices is a real possibility, since the devices only need to understand the SDN technology that controls them.

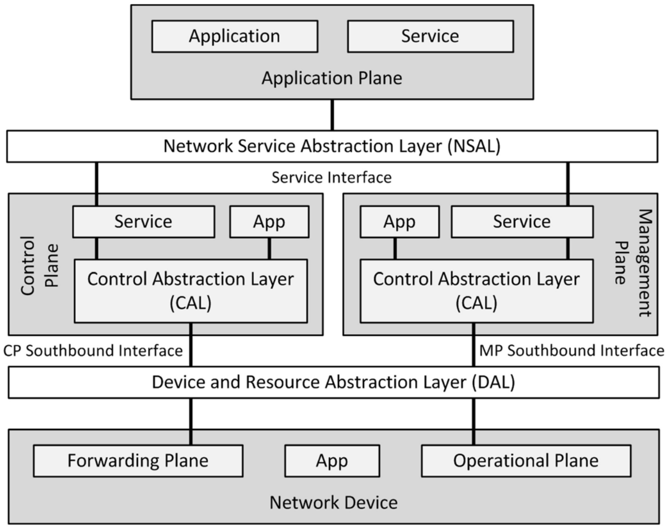

Fig. 1. SDN architecture proposed by the SDNRG that consist of five different planes: *forwarding plane*, *operational plane*, *control plane*, *management plane* and *application plane*, two abstraction layers: *Device and Resource Abstraction Layer (DAL)* and *Network Service Abstraction Layer (NSAL)* and two interfaces to communicate the control plane with the forwarding plane and the management plane with the operation plane: *CP Southbound interface* and *MP Southbound interface* respectively.

One of the main features of this new paradigm is that networks can be programmatically configured, making possible the management of the entire network through intelligent orchestration and provisioning systems. Besides, SDN architectures support a set of Application Programming Interfaces (API) that enable the implementation of common network services, custom tailored to meet business objectives.

*2) Architecture:* Nowadays, not only the ONF but many SDOs are dealing with SDN. For instance, the IETF has created an Internet Research Task Force research group focused on this trend, named the Software-Defined Networking Research Group (SDNRG). Both the ONF and the SDNRG have proposed different SDN architectures, which are described in this section.

*a) SDNRG architecture:* The architecture proposed by the SDNRG is depicted in Figure 1, and defines five different planes. Inside the network device, the *forwarding plane* is the one responsible for handling packets in the data path and it is often referred to as the data plane. Secondly, the *operational plane* is the plane responsible for managing the operational state of the network. On the one hand, the *control plane* is the one in charge of taking the decisions about how packets are forwarded at network devices, and it is also in charge of pushing such decisions down to network devices so that they are executed. On the other hand, the *management plane* is the one in charge of monitoring, configuring and maintaining the network devices. Finally, the *application plane* is where the applications that rely on the network to provide services for the end users and processes reside. In addition to these five planes, the SDNRG architecture also defines two abstraction layers: the *Device and Resource Abstraction Layer* and the *Network Service Abstraction Layer*. The first one abstracts the network devices' forwarding and operational planes and

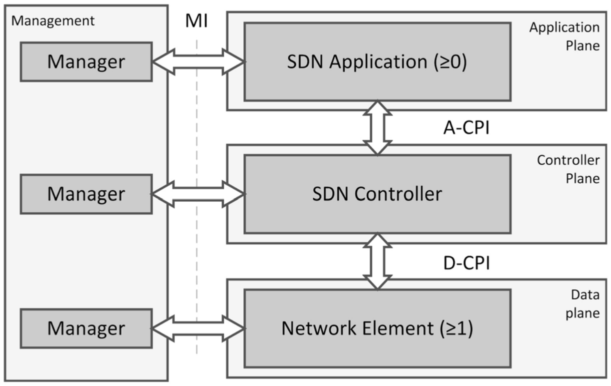

Fig. 2. SDN architecture proposed by the ONF that consists of a *data plane* that communicates with a *controller plane* using a *Data-Controller Plane Interface (D-CPI)*. The controller plane also communicates with the *application plane* through an *Application-Controller Plane Interface (A-CPI)*, and the three planes are managed using the *Management Interfaces (MI)*.

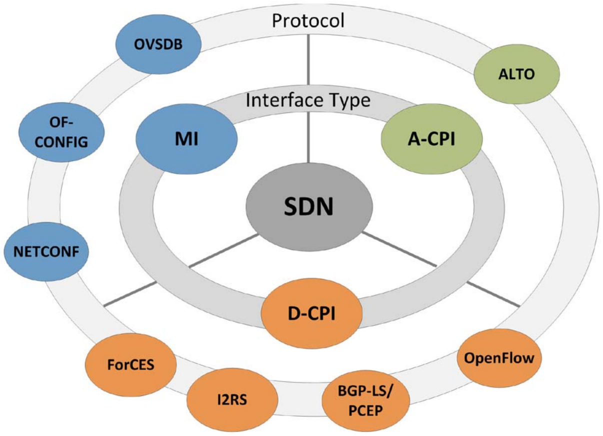

Fig. 3. Categorisation of SDN protocols depending on the interface at which they operate, namely *Data-Controller Plane Interface (D-CPI)*, *Application-Controller Plane Interface (A-CPI)*, and *Management Interface (MI)*.

connects to the control plane and management plane through the *Control Plane Southbound Interface* and the *Management Plane Southbound Interface* respectively. The second abstraction layer exposes the control and management planes through a *Northbound Interface* to the application plane.

*b) ONF architecture:* The ONF has also presented a reference architecture for SDN [65], which follows a three layer approach, as depicted in Figure 2. It has to be taken into account that the main goal of this architecture is to provide a high level overview of the reference points and open interfaces that should be present in every SDN deployment, in order to guarantee a minimum set of capabilities that would allow to control the connectivity provided by the network resources and the traffic flows through them.

The first layer is known as the *data plane* and it is the plane in which the network elements reside. The data plane uses the *Data-Controller Plane Interface (D-CPI)* to expose the network elements' capabilities to the second layer, that is, the *controller plane*. As its name suggests, the controller plane contains the SDN controller, which is the element in charge of controlling the network elements through the previously mentioned D-CPI. The controller plane also exposes services to the third layer, known as the *application layer*, through the *Application-Controller Plane Interface (A-CPI)*. This latter plane holds the applications that specify the behaviour of the network through the A-CPI. In addition to these two interfaces, this architecture also considers MI to configure and manage the three different planes.

### B. Taxonomy

The SDN protocols analysed in this paper will be categorised according to the interface defined by the ONF's SDN architecture at which they operate. The three categories used in this paper are listed below:

- *D-CPI protocols:* used to communicate the data plane with the controller plane. The D-CPI is aware of an instance of the data plane's informational model, that is, the set of resources on the data plane and the operations that can be performed on them. These protocols operate on an event timescale, that is, they are able to enable or disable circuits at the data plane within milliseconds.
- *A-CPI protocols:* protocols that are used to communicate the controller plane with the application plane. The protocols in this category can provide an abstraction of the network resources to the applications, or user-friendly and standardised mechanisms to program the network elements.
- *MI protocols:* the ONF's SDN architecture includes management technologies to operate over the three planes. However, given that currently there are not standardised technologies to manage the controller plane or the application plane, this survey focuses on management technologies that operate over the data plane. Therefore, the MIs surveyed in this paper are used to manage the network elements, and are in charge of tasks such as policy provisioning, port, queues or LSPs configuration and in some cases, even of failure detection. They operate on much slower timescale when compared with the D-CPI protocols, within minutes or hours.

According to this taxonomy, the SDN protocols that are analysed in this paper are classified as depicted in Figure 3. On the one hand, OpenFlow, ForCES, I2RS and BGP-LS/PCEP are D-CPI technologies. On the other hand, NETCONF, OVSDB Management Protocol and OF-CONFIG are MIs. Finally, ALTO is the only protocol identified as an A-CPI protocol.

## IV. SDN Protocols

This section briefly reviews a comprehensive list of SDN protocols of interest for TE, which have been categorised taken into account the taxonomy described in Section III-B.

### A. D-CPI Protocols

As mentioned before, the D-CPI protocols are the ones used to communicate the data plane with the controller plane. ForCES, OpenFlow, I2RS and BGP-LS/PCEP fall into this category.

*1) ForCES:* Back in 2003, the ForCES WG of the IETF presented the Forwarding and Control Element Separation (ForCES) framework [66], which enables the separation of the control and forwarding planes of the network elements. Although the framework and the homonym protocol were designed to easily add new functionalities to the forwarding plane, neither the industry nor the academia adopted the proposal. In fact, due to the lack of open implementations of the ForCES protocol, the ForCES framework was ostracised [67]. Currently, with SDN being a hot topic, the ForCES WG has resumed the standardisation process. As stated in the RFC 3746 [68], ForCES does not only define a framework, but also standardises all the associated protocols that make possible the information exchange between the control and the forwarding planes. It can be considered as a framework aiming to improve network programmability through an open interface. However, unlike other SDN technologies, the ForCES framework does not impose a centralised control plane, in fact, it can be used with legacy distributed control protocols.

In the ForCES framework, which is depicted in Figure 4, a Network Element (NE) consists of Forwarding Elements (FE) and Control Elements (CE). In short, the FEs are logical entities that use the underlying hardware to provide per-packet processing. They must support a minimal set of capabilities to be able to establish network connectivity. FEs are formed by Logical Functional Blocks (LFB), which are programmed by the CE by means of the ForCES protocol to implement a wide variety of logical functions, e.g., L3 forwarding, Firewall or Network Address Translation. The FEs reside inside Physical Forwarding Elements (PFE), whereas CEs do the same in the Physical Control Elements (PCEL).[1] Typically, the PFEs and the PCELs are placed in the same physical machine, although, they can also be located separately as specified by the RFC 6041 [69] by a single or multiple hops, as stated in RFC 6053 [70]. There are two operational phases identified in the ForCES framework. First, in the pre-association phase the *CE manager* and the *FE manager* decide whether the CEs and the FEs are part of the same NE. However, this operational phase is out of the scope of the ForCES protocol. Second, in the post-association phase, the FEs and the CEs use the ForCES protocol to associate and exchange information to facilitate packet processing.

The ForCES protocol supports CEs redundancy. Multiple CEs can operate over the same FE, though, the coordination between the CEs is out of the scope of the ForCES protocol. As a consequence, it is possible for different CEs to implement different routing or signalling protocols, where the FE acts as the entity in charge of redirecting the control packets to each one of the CEs according to some filtering rules. Similarly, the framework also supports the coexistence of multiple FEs, which imposes additional challenges. First, the functions that each one of the FEs implement must be very well defined, as it can affect the overall performance of the system. Furthermore, depending on the functions that each FE is in charge of, it may be necessary to perform multiple forwarding decisions in more than one equipment.

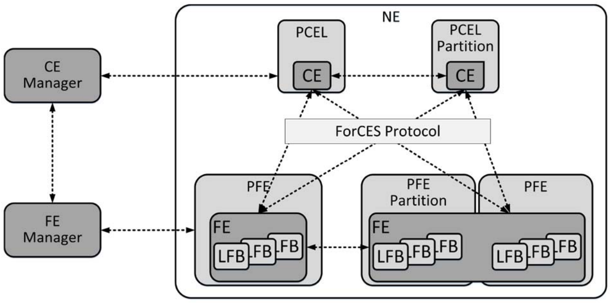

Fig. 4. ForCES framework elements where a Network Element (NE), composed of two Control Elements (CE), each of them residing on a Physical Control Element (PCEL), control two Forwarding Elements (FE) with multiple Logical Functional Blocks (LFB) that reside on one or multiple Physical Forwarding Elements (PFE) using the ForCES Protocol.

ForCES is a master-slave protocol, with CEs acting as masters and FEs as slaves. The protocol provides the means to associate the different elements of the framework, so as to tear down such associations. It is also in charge of transmitting subscribed-to events from FEs to CEs and of responding to status requests issued from the CEs to the FEs. Additionally, it is used to configure the FEs and the associated LFBs' operational parameters, so as to activate or deactivate the FEs. In the end, the protocol manages the LFBs at the FEs, which are compliant with the FE model defined in the RFC 5812 [71].

As mentioned before, the FEs are composed of LFBs that are interconnected in a direct graph, and receive, process, modify, and transmit packets along with metadata. The FE model establishes a formal way to define the FE's LFBs using XML, while the configuration components, capabilities and associated events of the LFBs are defined when they are formally created. On the one hand, the FEs can be broadly defined by simply specifying their capabilities. For instance, FEs can be described in terms of IPv4 or IPv6 forwarding support or by the set of matching fields supported for the packet classification. On the other hand, the FE model can also be used to describe the FE state model, which presents a snapshot view of the FE to the CE. For each LFB, the number of inputs and outputs can be specified, as well as the packet types accepted in each of them and the routing criteria.

As stated in [72], the ForCES protocol is powerful enough to define other protocols. For instance, the authors of this paper state that both OpenFlow and NETCONF, which are later explained in this subsection, could be considered subsets of the ForCES protocol. Therefore, according to the ONF's SDN architecture ForCES could be considered both a D-CPI and an MI. Nevertheless, since the primary goal of the ForCES protocol is the communication of the CEs with the FEs, that is, the controller and data planes of this architecture, it is considered a D-CPI protocol within the evaluation.

*2) OpenFlow:* Back in 2008, the Stanford University released the first stable version of OpenFlow. Since then, the ONF has become the SDO in charge of the standardisation of the OpenFlow Switch Specification and its homonym protocol. The OpenFlow Switch Specification defines both the OpenFlow Logical Switch (OLS) and the OpenFlow

[1] To avoid confusion with the Path Computation Element (PCE)

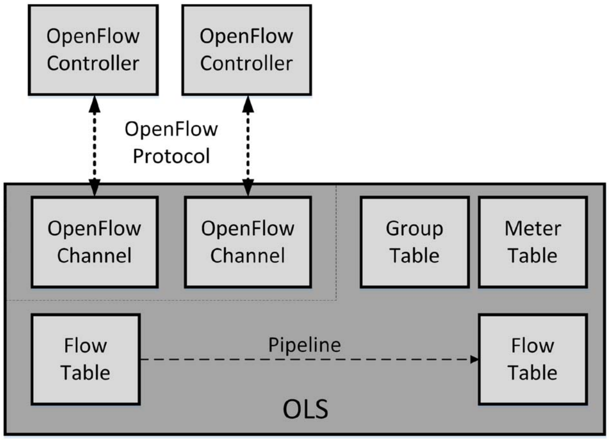

Fig. 5. Architecture on an OpenFlow Logical Switch (OLS), where at least one *OpenFlow Controller* communicates with one *OpenFlow Channel* using the *OpenFlow Protocol* to program the *Group Table*, the *Meter Table* and the *Flow Tables* on the switch.

protocol, used for the communication between the OLS and the OpenFlow controller.

As depicted in Figure 5, the OLS consist of one or more control channels and a datapath. The datapath is where the packet lookups and forwarding are performed, by means of one or more flow tables, a group table and a meter table. The OLS connects to the external controller through the OpenFlow channel, often referred to as the control channel, using the OpenFlow protocol.

Each OLS must have at least a flow table composed of flow entries, which are formed by the *match fields*, the *priority* that specifies the matching precedence of the entry, the *counters* that hold statistical information, the set of *instructions* that are applied to the matching packets, the *timeouts* and the *cookie* that unambiguously identifies the flow entry. Among the instructions that can be applied to packets, the ability to direct the packets to specific *meters* is of special relevance for TE. Meters are switch elements able to measure and control the rate of packets being forwarded; therefore, they play a key role in QoS enforcement. Instructions can also be used to apply a certain set of actions. Available actions include sending the packet to a queue or to an outport (output port), directing the packet to a group table or re-writing a specific field, to cite a few. Another interesting feature of OpenFlow is the fine-granularity that it supports for the matching of packets. OpenFlow takes into account at least the physical ingress port and additional Ethernet, IPv6, IPv4, TCP and UDP header fields. Moreover, additional header fields can be included thanks to the OpenFlow eXtensible Match (OXM), which is a very flexible model where new matching fields are defined as Type Length Values (TLV).

In a nutshell, the external controller populates the flow tables of the OLS with the flow entries that determine the behaviour of the traffic that matches them. Each OLS can contain more than one flow table with its corresponding flow entries. The OpenFlow pipeline process defines how packets interact with those flow tables. According to the latest OpenFlow Switch Specification [7], packets are always matched first against the first flow table and in the cases where there are multiple flow-tables, packets are forwarded to the subsequent ones. When a packet matches one or multiple flow entries, the instruction set associated to the entry with the highest priority is applied, which can include directing the packet to another flow table. Then, when the pipeline process finishes, either because there are not more redirections to subsequent flow tables or because it is the last flow table, all the associated actions are applied. It is worth mentioning that an OLS is able to handle flow miss-matches. Depending on the configuration, when a packet arrives that does not match any of the flow entries installed in a flow table, the OLS can specify how to process it. As a consequence, packets can be directed to another flow table or be sent to the controller. This feature makes possible to work reactively besides of proactively; that is, to act in response to packets that do not match any entry of a flow table.

There is a single group table per OLS, which makes possible to represent additional forwarding methods. For instance, in an OLS it is possible to flood packets creating a group that associates output actions to all the ports but the ingress port. Each entry at the group table is defined by a unique identifier, a set of counters, the action buckets (ordered list of actions to execute and the associated parameters) that must be applied to the packets and the *group type* they belong to. For the moment, OpenFlow defines four different group types: all, select, indirect and fast fail-over. The first one is characterised by applying all the action buckets defined for the group. The *select* group type uses just one of the action buckets associated to the group for each packet, e.g., in a round-robin fashion. Third, the *indirect* group type supports a single action bucket. Finally, the *fast fail-over* group applies one action bucket at each time, following the order in which they are configured.

Regarding the OpenFlow controllers, it is worth mentioning that network operators can choose between centralised (e.g., NOX [73] POX [74], Trema [75], Ryu [76], FloodLight [77], Beacon [78], Maestro [79], McNettle [80], Jaxon [81], Snac [82]) or distributed (e.g., Onix [83] and HyperFlow [84], Helios [85]) controllers. They can also select the programming language to use, being Java and Python the most popular ones. Furthermore, there are also available special purpose controllers, such as FlowVisor [86], Open Virtex [87], and AutoSlice [88], which make possible to virtualise OpenFlow-based networks by slicing the network resources and exposing the network control to other controllers transparently. For further information regarding network virtualisation with OpenFlow (see [26], [89], [90]). In addition, several frameworks have appeared recently that support a set of SDN protocols, including OpenFlow. This is the case of the Open Network Operating System (ONOS) [91], OpenDaylight Platform (ODP) [92] or Cisco Open Networking Environment (ONE) [93]. Further information about OpenFlow controllers can be found in [94].

Undoubtedly, the OpenFlow protocol is a D-CPI, used to communicate the OLSs that reside in the data plane with the OpenFlow controller placed in the homonym plane of the ONF's SDN architecture.

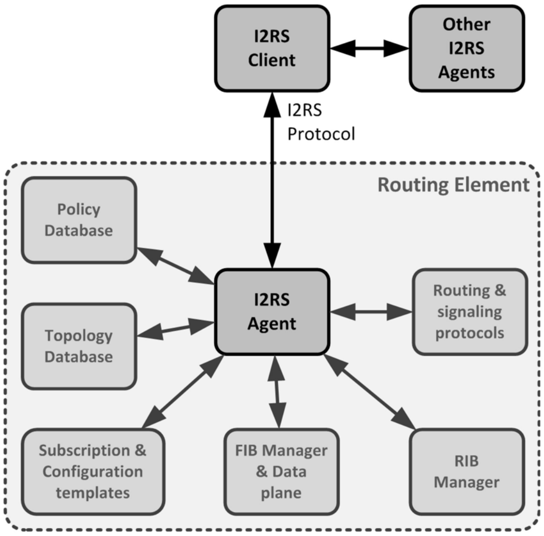

Fig. 6. Main components of the I2RS architecture (highlighted): the *I2RS Agent* that communicates with the *I2RS Client* using the *I2RS Protocol*, and the interactions between the I2RS components and the components of legacy routing elements.

*3) I2RS:* The Interface to the Routing System, known as I2RS, is an IETF WG created in late 2012 [95]. This WG is actively focused on the definition of the I2RS protocol, the high-level architecture for its application and the key use cases for its operational use. In a nutshell, the I2RS protocol is a protocol [96] for transferring state into and out of the routing system that exploits the operating system of the router itself, that is, I2RS allows forwarding elements to keep their routing logic.

The majority of commercial routers maintain a Routing Information Base (RIB) and implement routing protocols such as OSPF, IS-IS and Border Gateway Protocol (BGP). The routing protocols insert routes into the RIB through the RIB Manager north-bound interface, while the Forwarding Information Base (FIB) manager consults the RIB and programs the FIB of the hardware. The I2RS harnesses the mechanisms that the routing systems and their operating system offer and provides an interface to control the RIB. In other words, I2RS interacts directly with the L3 forwarding engine and routing protocols.

I2RS allows applications built on top of the network to access the dynamic information that routers already have about the topology of the network, events, traffic and status. Thanks to the information they have access to, these applications are able to observe the routing related state of the routing elements, which allows them to enhance the routing control processes [97]. The I2RS architecture must be able to ensure that the correct state is operative [98], and to be able to do that, it defines the following elements [98], [99], where the most representative ones are depicted in Figure 6:

- *I2RS Client:* the entity that communicates with the I2RS Agents through the I2RS protocol and uses the I2RS Services to accomplish a task. It is able to interact with the I2RS Agents both, to collect information from the routing and forwarding system so as to modify the state of the routing system to achieve operational goals. Moreover, it can interact with other elements of the policy, provisioning and configuration system. The I2RS Client can be part of one or more applications and it may or may not be co-located with them.
- *I2RS Agent:* the entity that provides the supported I2RS services from the routing sub-systems of the local system. It is able to communicate with I2RS Clients through the I2RS protocol and it is considered a part of the routing element. The I2RS Agent is in charge of collecting and delivering the data obtained from the routing element, data that can be stored in a routing device or in an external element. Furthermore, it applies changes to the system and maintains a log with information about the changes and the active subscriptions.
- *I2RS Service:* a set of related state access functions and the policies that control their usage. Services can be associated to routing and label information bases, IGP, BGP and Multicast protocols, MPLS and Policy and QoS mechanisms. In general, to each logical protocol or set of functionalities susceptible of being described by a separable data-model. Thus, each protocol or functionality will be represented by a data-model that defines the semantics of the information that can be written or read. Furthermore, the data-model describes the notifications available to I2RS Clients and a capability model that determines the parts of a service that are supported.
- *I2RS Protocol:* the protocol used between I2RS Clients and I2RS Agents to communicate.

In order to provide programmability to the solution, the I2RS WG has specified an information model for the RIB, which can be used to define a data-model able to program a routing element [100]. For instance, a route data-model consists of a set of route attributes, the match condition (IPv4, IPv6, MPLS, MAC and Interface) and the next hop. The I2RS protocol makes it possible to write and read from the RIB information. Besides, being a standardised information model, it would be possible to use it to program multi-vendor routing elements.

As stated in [97], both the protocol and the modelling architecture must be simple. I2RS data-models must be extensible and easy to integrate with other data-models. These data-models have to be able to model next hops and handle next-hop indirection and recursion, which allows flexibility and increases functionality. Besides, I2RS has to be able to handle different types of tunnelling and encapsulation methods. In addition, the solution is intended to support multiple simultaneous asynchronous operations, multi-headed control, high throughput, responsiveness, secure control and extensibility and interoperability, among other features.

The I2RS protocol is meant to track and control the dynamic state of networking elements such as routers and switches. According to the I2RS WG, the I2RS protocol presents some major benefits. Firstly, it provides high flexibility to network operators since they can adapt their legacy networking hardware to SDN principles by installing an I2RS agent. I2RS relies on already existing technologies and therefore, already existing networking elements can be extended to implement the I2RS protocol by means of a firmware update. Secondly, it will be applied in highly reliable scenarios. Currently, the ODP supports I2RS data-models, which are defined using the YANG [101] modelling language.

In a nutshell, I2RS enables the distributed control protocols to coexist with the centralised management and control aspects provided by I2RS. It clearly operates at a different layer compared to OpenFlow and ForCES, but it is still an interface that allows to externally control the routing elements. As a consequence, I2RS is considered a D-CPI protocol within this survey.

*4) BGP-LS/PCEP:* Defined by the Inter-domain Router (IDR) IETF WG [102], BGP-LS is a protocol used to collect and share information about link state and TE [103]. By means of a set of extensions added to the BGP routing protocol [104], BGP-LS retrieves the topological information from the Link State Databases and distributes it to a consumer both directly or through a BGP Speaker or a Route Reflector. A BGP speaker exchanges network reachability information with other BGP speakers, including the intermediate ASs that the traffic must transit to reach destinations, whereas a Route Reflector is mostly used as a concentrator for multiple BGP speakers inside an IGP area. Taking into consideration that BGP is an inter-AS technology, with BGP-LS it is possible to provide information about other IGP areas to the external components. Although it can work independently, BGP-LS is a mechanism that can also be used by multiple applications, such as PCE and ALTO. For example, PCE performs path computation using TE information, and TE information is never exchanged across different network domains. Since BGP-LS can be used to exchange TE information between different IGP areas and network domains, BGP-LS makes a PCE capable of computing E2E paths across different IGP areas. Thus, BGP-LS is a mechanism that can improve actual TE solutions such as the PCE-based architecture.

BGP-LS can be used to provide information about the maximum bandwidth, the maximum reservable bandwidth or the unreserved bandwidth on a given link. It can also be used to inform about the default TE metric. That is, to inform about the objective function of the TE strategy, such as the minimisation of the delay or of the link utilisation.

Many vendors have started to include BPG-LS support in their devices (i.e., Cisco or Juniper), so as many SDN platforms like the aforementioned ODP, ONOS and Cisco ONE [105]. BGP-LS by itself cannot be considered a full D-CPI technology, since it is only valid to exchange topological information among network elements and does not provide network programmability. However, most SDN controllers use BGP-LS together with the PCEP protocol, which is the reason why these two protocols working together are considered another D-CPI solution in this paper.

As mentioned before, the main characteristic of the PCE-based architecture is that the path computation is performed in a dedicated element. In this architecture, a Path Computation Client (PCC) requests a path, which is computed by the PCE using the TE information stored in a TED. In order to fulfil its intended objective, the PCE-based architecture relies on two key protocols: Path Computation Element Communication Protocol (PCEP) [106] and Resource Reservation Protocol (RSVP)-TE, defined in RFC 4657 [106] and RFC 3209 [107] respectively.

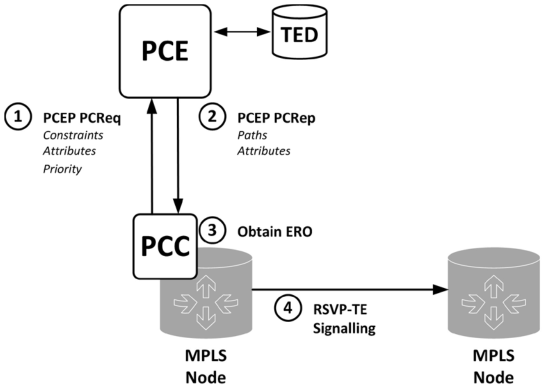

Fig. 7. Main components of the Path Computation Element (PCE)-based architecture and Path Computation Element Communication Protocol (PCEP) message exchange between the Path Computation Client (PCC) and the PCE.

As stated in the RFC 4657 [106], the PCEP protocol is used for the communication between PCCs and PCEs, so as for the inter-PCE communication. Figure 7 depicts how a PCC communicates with a PCE to request a path computation. (1) the PCC sends PCEP *PCReq* messages to the PCE when it wants a path to be computed for one or more TE LSPs. Using the same message, it sends the set of constraints and attributes that the PCE requires to compute the path and a priority number to indicate the urgency of the request. When the PCE has finished computing the path, it replies (2) with a PCEP *PCRep* message, which can be a negative message indicating the reason why the computation has failed or a positive one. In the latter case, the response includes the set of computed paths and the sets of attributes associated with them, such as the path costs (e.g., cumulative link TE metrics and cumulative link IGP metrics) and the computed bandwidth. In order to avoid negative messages, the PCE can notify PCCs that it is unable to satisfy certain requests or that it has been experiencing unacceptable delays. This way, since the PCE-based architecture supports multiple PCEs in the same network domain, the PCC has the opportunity to send its *PCReq* to another PCE.

Using the PCEP protocol, (3) the PCEs send explicit paths to the PCCs specified by means of Explicit Route Objects (ERO). These EROs are used for the (4) establishment of the LSPs through RSVP-TE in MPLS and GMPLS networks. They consist of sets of IPv4/v6 prefixes and Autonomous System (AS) numbers, among other possible parameters. Hence, the computation of the paths must support everything that can be expressed in an ERO, like the degree of paths disjointness or the maximum hop count among others. It is worth mentioning that PCEP includes support for load-balancing. The PCC can indicate the support for load-balancing and the number of paths that can be included in the balancing group. This is a very interesting feature for multi-path communications and the minimisation of the link load, because the more paths to split the traffic, the lower will be the load on each of them.

On the one hand, BGP-LS is able to retrieve topological and link state information but lacks the necessary mechanisms to program the network elements. On the other hand, PCEP is able to program the network elements but lacks the necessary mechanisms to retrieve information from the network resources. However, the two protocols complement each other, and working together compose another D-CPI technology to be taken into account. Working in conjunction with BGP-LS can lead the PCE-based architecture to a whole new level, since it can be useful to solve many of the limitations found regarding the retrieval of TE information to store it in the TED.

### B. A-CPI Protocols

Currently only one of the analysed protocols lies in this category, the ALTO protocol.

The Application Layer Traffic Optimization (ALTO) IETF working group [108] is in charge of the standardisation of the ALTO protocol since 2008, which is defined in RFC 7285 [109]. As a brief summary, the ALTO protocol provides information about the state of the network that allows to improve both applications' and network's performance at the same time. The optimisation can be done taking into account different criteria: operator's policies, geographical location, etc. The information provided by ALTO about the state of the network is neither granular nor in real-time, as it operates on a large time scale. According to the RFC 7285, the applications that use the information provided by the ALTO protocol can take better TE decisions. For example, an overlay application can use information provided by the ALTO protocol to avoid the links that impose higher delays than others. As a consequence, there have been some efforts from the IETF ALTO WG to integrate ALTO within SDN.

ALTO aims to improve traffic pattern distribution in cases where MPLS-TE or Diffserv do not provide any benefit. This is the case of distributed applications, such as peer-to-peer (P2P) communications, file sharing, cache/mirror selection, live media streaming, distributed hash tables or real-time communications. As depicted in Figure 8, in the ALTO architecture the following elements are differentiated [110], [111]:

- *ALTO Service:* when the same resource can be provided by different providers, it tells the requester which one must be selected in order to optimise both, the Quality of Experience (QoE) and the resource consumption in the underlying network infrastructure.
- *ALTO Server:* logical entity that provides interfaces for the queries to the ALTO service.
- *ALTO Client:* logical entity that sends ALTO queries.
- *ALTO Protocol:* used for sending ALTO queries and ALTO replies between an ALTO client and an ALTO server.
- *Provisioning Protocol:* used for populating the ALTO server with information.
- *ALTO Information:* a generic term referring to the network information sent by an ALTO Server.
- *ALTO Information Base:* internal representation of ALTO Information maintained by an ALTO Server.

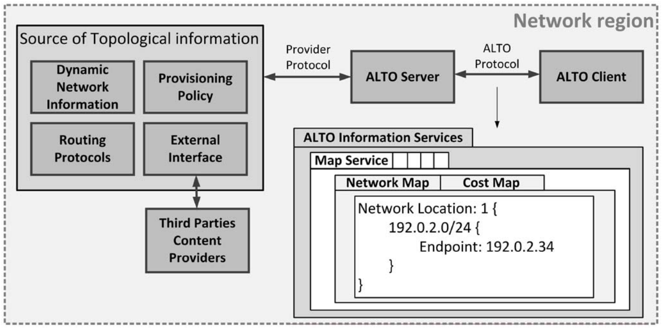

Fig. 8. Main components of the ALTO architecture and details about how the network topology is presented to the ALTO Client by the ALTO Server.

- *Endpoint:* an application or host that is capable of communicating (sending and/or receiving messages) on a network.
- *Network Location:* represents one or more endpoints.

It is worth mentioning that the ALTO Server aggregates the information of multiple systems and provides it to the application on a more useful and unified way. Figure 8 shows how an ALTO Server is able to receive information from multiple sources, e.g., static network configuration databases, dynamic network information, routing protocols and provisioning policies. Furthermore, it is also capable of retrieving information from third party content providers using an external interface. Each of these sources can provide a variety of network state related information with different purposes and different levels of detail. By combining them, the ALTO server is able to provide aggregated network state information, which represents network state more accurately.

The ALTO protocol follows a RESTful design and it is based on JSON over HTTP. Its main goal is to provide basic network location information and preferences of network paths in order to improve applications performance, while resource consumption is also enhanced. In other words, the ALTO protocol is an interface that networks can use to publish heterogeneous information such as network locations, costs among them at configurable granularities, and endpoint properties to network applications. To be able to do that, ALTO exposes abstract maps of the network that provide a more simplified view of the network to the applications. On the one hand, the *network map* provides a full set of Network Location groupings defined by the ALTO Server and the Endpoints contained within each grouping. On the other hand, the *cost map* defines the *path costs* pairwise for a given network map, that is, the E2E cost when a unit of traffic goes from the source to the destination among sets of source and destination Network Locations. Precisely, these cost maps are the elements that make possible for ALTO Servers to indicate preferences among Network Locations. Although the RFC 7285 specifies that the granularity is configurable, the Endpoints can only be defined with IPv4 or IPv6 addresses (prefixes are also supported) at the moment of writing this paper.

As proposed in [112], the ALTO server can be implemented as an SDN application on top of an SDN controller where the ALTO client resides. In such a scenario, the ALTO protocol which is in charge of communicating both entities will behave as an A-CPI as proposed by the ONF's SDN architecture. At the moment of writing this manuscript there are not many solutions for ALTO, especially in the SDN environment, while the ALTO project in the ODP represents the most successful initiative until today.

### C. MI Protocols

The present section reviews a comprehensive list of SDN protocols that operate at the MI: OVSDB Management Protocol, NETCONF and OF-CONFIG.

*1) OVSDB Management Protocol:* In brief, Open vSwitch DataBase (OVSDB) Management Protocol is a protocol that makes possible to manage the resources in an Open vSwitch (OVS) and it is defined in RFC 7047 [113]. OVS is an open-source multi-layer software switch released under Apache 2.0 license [114]. It was created in 2009 as a result of a collaborative project between Nicira Networks and members of the Computer Science Division of the University of California, Berkeley. Due to its good performance, the OVS was committed to the Linux Kernel in its 3.3 release [115]. It was originally built for its application in virtual environments, more specifically, to be in charge of the inter-VM and intra-VM connectivity. Nevertheless, it has evolved and now it is used in a variety of environments [116]. For instance, it is possible to use it as the control stack of hardware switches [117].

Even that OVSDB Management Protocol can only be used with OVSs, it has been included in this survey as it is a very relevant SDN technology. According to [118], OVS exposes two well-defined interfaces; OpenFlow for the control of the forwarding behaviour and OVSDB for the configuration of the switch. The first interface has been analysed in Section IV-A2, whereas this subsection is focused on OVSDB.

As depicted in Figure 9, OVS runs both in the kernel and the user space. The *ovsdb* is the OVS database that stores the configuration information of the switch, which is precisely the database that is manipulated by the OVSDB Management Protocol. The information stored at the ovsdb is retrieved by the *ovs-vswitchd* daemon at startup time, and this information is later used for setting up the configuration of the switch and the corresponding datapaths. It is worth mentioning that when a change occurs in the ovsdb, the ovs-vswitchd automatically updates the switches' configuration accordingly. Broadly speaking, the ovsdb is a database that holds the configuration used by the vswitch daemon [119]. The configuration information is held in well-defined tables that store specific information about bridges or ports, to cite a few.

The architectural components of the OVS are organised in two clusters: the management cluster and the control cluster. The former one encompasses the managers that use the OVSDB Management Protocol to manage the OVS instances, where there is at least one manager per OVS instance. On the other hand, the latter one encompasses the controllers that use the OpenFlow protocol to install the forwarding state into the OpenFlow switches, where there is at least one controller per OpenFlow bridge or logical datapath. Further information about these components can be found on the OVS website [120].

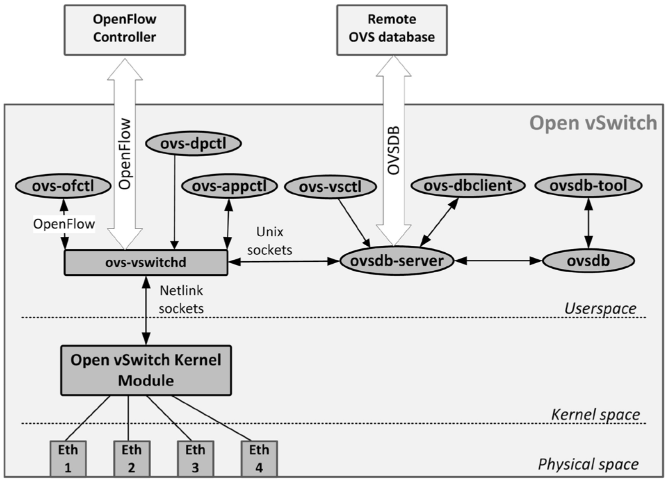

Fig. 9. Main components of the Open vSwitch (OVS) architecture placed in the physical, kernel or user space. OVS exposes two interfaces to external components: OpenFlow and OVSDB Management Protocol.

As defined in [113], the OVSDB Management Protocol is based on JSON Remote Procedure Call (RPC) version 1.0 [121] and its purpose is to operate on the OVS instance. Through the OVSDB Management Protocol it is possible to create, modify and delete OpenFlow datapaths. Furthermore, it provides the means to configure these OpenFlow datapaths. For instance, it supports the creation, modification and deletion of ports, tunnels and queues, so as the configuration of QoS policies and the attachment of those policies to the queues. It is also able to handle the configuration of the set of controllers to which an OpenFlow datapath should connect and collect statistical information. All in all, OVSDB is an MI protocol that allows to remotely configure OVSs.

*2) NETCONF:* The main goal of the NETCONF protocol is to provide a unified, cross-vendor and inter-operable management interface for automated control of network equipment. This feature makes NETCONF a very powerful tool for implementing the network management model required by programmable networks [122]. NETCONF has been widely adopted by network equipment vendors. Among others, Cisco [123], Juniper [124] or NEC [125] support NETCONF in their commercial products. In a nutshell, the NETCONF protocol exposes an API that external applications can use to manage network devices, it follows a RPC paradigm and it is defined by means of a XML schema. The protocol is maintained by the IETF Network Configuration working group [126], which since the first release of the protocol back in 2006 has published more than 10 RFCs.

Through this protocol, applications and users are able to access the syntactic and semantic content of the device's native user interfaces. Furthermore, it allows to discover the set of protocol extensions supported by network devices. It is often said that NETCONF is focused on the information required to get the device into its desired running state. When talking about NETCONF, the following terminology is used in RFC 6241 [127]:

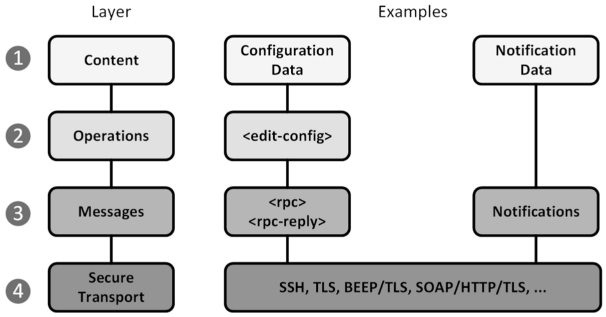

Fig. 10. Layers where the NETCONF protocol operate: secure transport layer, messages layer, operations layer and content layer.

- *Client:* the element that invokes the protocol operation on the server, usually an application or a script running in the Network Management System (NMS). It can also receive notifications from a server.
- *Server:* the element that executes the protocol operations invoked by a client, it is usually the network device itself. It can also send notifications to a client.
- *Configuration data:* the set of writable data that is required to transform a system from its initial default state into its current state.
- *State data:* additional data on a system that is not configuration data such as read-only status information and collected statistics.
- *Configuration datastore:* the datastore holding the complete set of configuration data that is required to get the device from its initial default state into a desired operational state.

The configuration model followed by NETCONF is characterised by the definition of one or more configuration datastores that support a well-known set of operations. For instance, the running-configuration datastore holds the active configuration of the network device. Each device has one and only one running-configuration datastore and it is always present in the base NETCONF model. In order to add further configuration datastores, the NETCONF protocol has to be used, as it supports the definition of additional datastores defined by certain capabilities (available only on devices that advertise the capabilities). The NETCONF protocol operates into four well differentiated layers, which are depicted in Figure 10: secure transport layer, messages layer, operations layer and content layer. The secure transport layer is in charge of the communication between the client and the server, and it can be a protocol with a minimum set of capabilities as defined in the RFC 6241 [127]. The messages layer provides a framing mechanism to encode the RPCs and the notifications defined at the operations layer. More in detail, the operations layer is in charge of the definition of the base protocol operations that are invoked as RPC methods where the parameters are encoded using XML. Finally, at the content layer the NETCONF data-model is specified. Please note that the YANG data-model used to monitor the NETCONF protocol, which is described in the RFC 6022 [128], covers the third and fourth layers.

In order to support the addition of new sets of functionalities to the base NETCONF specification, NETCONF defines *capabilities*. These capabilities augment the basic operation of the devices and describe the additional operations allowed, so as the content which is allowed inside these operations. They are usually described in external documents and identified by Uniform Resource Identifiers (URI), that is, by means of a string of characters. On the one hand, capabilities can be proprietary, meaning that they are valid for certain devices, as it happens with the Extended NETCONF Operations for Cisco devices [123]. On the other hand, capabilities can also be standardised by an SDO. In any case, the set of capabilities supported by a device are advertised when the session between the server and the client is established.

As in the case of the OVSDB Management Protocol, NETCONF clearly fits as a management technology in the SDN Architecture proposed by the ONF.

*3) OF-CONFIG:* The OpenFlow Configuration (OF-CONFIG) is a protocol that complements OpenFlow with the purpose of configuring and managing the operational context in which OpenFlow switches reside. More specifically, it aims to configure the OpenFlow datapaths on a physical or virtual platform and is characterised by a much slower operational time-scale than the OpenFlow protocol [129]. Now in its 1.2 version, the OF-CONFIG protocol appeared in 2011 to complement the OpenFlow protocol, which lacks the mechanisms to manage and configure the OpenFlow switches. Although it is also an effort of the ONF, it has not achieved the implementation level that the OpenFlow protocol has. The OpenFlow environment has also been extended with the OpenFlow Notifications Frameworks [130], to allow notifications and alerts regarding OpenFlow and OF-CONFIG. The OF-CONFIG protocol handles the following terminology, and some of the elements are depicted in Figure 11 [129]:

- *OpenFlow Logical Switch (OLS):* an abstraction that represents an OpenFlow switch, also referred to as datapath. In other words, it is a set of ports that belong to an OpenFlow Capable Switch associated to an OpenFlow controller.
- *OpenFlow Capable Switch (OCS):* operating context that contains one or more OLS. Each OCS can be configured by multiple OpenFlow Configuration Points and it can be a physical switch or a virtual network environment that hosts one or more OLSs. In the last case, the OCS handles the association of the OLS resources to the OpenFlow related resources.
- *OpenFlow Configuration Point (OCP):* service that uses the OF-CONFIG protocol to configure the OCS that it handles. It is worth mentioning that a single OCP is able to manage multiple OCSs. It can reside in an OpenFlow controller or as a service inside a Network Management Framework (NMF).
- *OpenFlow resource:* a resource associated with an OCS or an OLS. For instance, an OpenFlow queue, which is a queuing resource of an OLS or an OpenFlow port, that is, a forwarding interface of an OLS.
- *OpenFlow controller:* software that controls OLSs via OpenFlow.

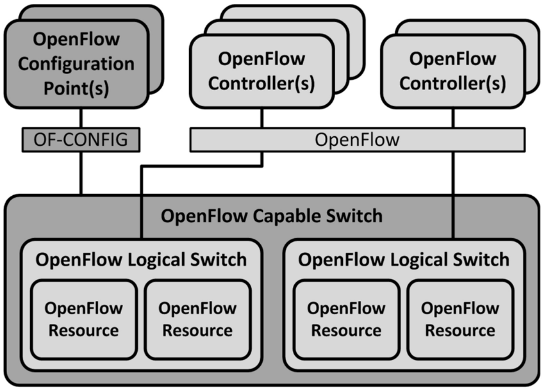

Fig. 11. Main components of OF-CONFIG, highlighted to differentiate from OpenFlow components. The *OF-CONFIG* protocol is used for the communication between the *OpenFlow Configuration Points (OCP)* and the *OpenFlow Capable Switch (OCS)*.

- *Negotiable Datapath Model (NDM):* abstract switch model that describes the specific switch forwarding behaviour that can be controlled via the OpenFlow protocol.

As previously stated, the OF-CONFIG protocol is focused on the configuration of the OLSs. With OF-CONFIG it is possible to assign the controllers to the OpenFlow data planes, to configure queues and ports and even change some aspects of the ports. The protocol also supports the configuration of the certificates that enable the establishment of a secure communication between the OLSs and the OpenFlow controllers. In addition, OF-CONFIG is able to instantiate OpenFlow data planes, so as to discover the capabilities and assign resources of an OCS to them. Besides, a very interesting feature is that it is built upon the NETCONF protocol and uses it as the transport protocol. This has been decided because the NETCONF protocol meets the OF-CONFIG requirements for communication, e.g., partial switch configuration or retrieval of configuration data. As a consequence, it is mandatory for devices that implement OF-CONFIG to support the NETCONF protocol.

The current schema of the OF-CONFIG Protocol covers basic configuration elements and it is encoded using XML. The data-model that describes the protocol allows the establishment of parameters such as the maximum number of packets that can be buffered at the logical switch and the number of flow tables or ports that the switch supports. It also makes possible to retrieve statistical information regarding flow tables, ports, flows, queues or even groups. Furthermore, it allows to configure the flow tables themselves, specifying the instructions that they support, the next flow table in the processing pipeline, etc. Some SDN controllers and Network Operating Systems have started to include OF-CONFIG support, like the ODP and Ryu.

When it comes to TE, OF-CONFIG complements the OpenFlow protocol by adding support for the management and configuration of ports and queues. There is, though, a difference between port configuration and queue configuration. The OpenFlow protocol is not able to configure queues, whereas it is able to configure ports. On the one hand, the OF-CONFIG protocol increases the configurability of ports available in OpenFlow by adding the possibility to configure additional parameters. For instance, it makes possible to set up the administrative state of the port, so as to specify if it allows or not the reception, forwarding or redirection of packets to the controller. In addition to this, OF-CONFIG provides the means to configure the advertised features of the ports, such as the speed or the ones related to the auto-negotiation, to cite a few. On the other hand, the OF-CONFIG protocol is able to configure the minimum and the maximum rate of a queue, so as additional parameters thanks to the *experimenter* extensions. As such, OF-CONFIG lies in the MI protocol category according to the ONF's SDN architecture.

## V. TE Solutions Based on D-CPI Protocols

This section provides an overview of the TE capabilities and literature review of the previously reviewed D-CPI protocols. These protocols are the ones with the capacity to impact TE more profoundly, especially the ones that do not only provide an interface to increase the programmability, but also a higher granularity at the forwarding plane compared to the legacy solutions.

### A. ForCES

As mentioned in Section IV-A1, the ForCES protocol operates at the D-CPI interface. It is helpful for TE because it can be used with legacy protocols, supports CE redundancy and it can be used to define other protocols such as OpenFlow and NETCONF. The ForCES-based proposals surveyed in this section have been categorised depending on their TE performance objective: optimisation of the network resource utilisation or the packet loss minimisation. Table II summarises TE-related proposals where the ForCES protocol is used, where the scope of the solutions is also identified. Regarding the scope of the solutions, it is distinguished between solutions applied at ForCES router level, or network level. In ForCES, a ForCES router consists of multiple FEs and CEs, where the inter-FE topology represents how the FEs are interconnected in a single NE. Therefore, since a ForCES router is a set of FEs, i.e., a set of switches, TE solutions can be applied to increase a ForCES Router's performance or to increase networks' performance.

*1) Resource Utilisation Optimisation:* This section surveys the ForCES proposals where the resource utilisation is optimised. The solutions have been classified depending on the resource type being optimised.

*a) Logical function blocks:* The solution presented in [131] applies the same principles that are used in Cloud Computing to assign on-demand computational resources to increase ForCES router's performance. The solution is based on a resource scheduling algorithm based on an economic model that allows to select resources in a programmable and scalable fashion. The objective function of the algorithm is to optimise resource utilisation, which in this case are the LFBs of the ForCES router. The algorithm selects the LFBs taking into account QoS and pricing objectives and the real-time node computing resource utilisation.

*b) Data channel:* Also related to the optimisation of resources, [132] presents a OSPF routing optimisation scheme

TABLE II
SUMMARY OF TE-RELATED RESEARCH WITH FORCES. *NA* STANDS FOR "NOT APPLICABLE"

| **Authors** | **Summary** | **Performance Objective** | **Resource** | **Scope** |
|---|---|---|---|---|
| Bin *et al.* [131] | Scheduling mechanism based on an economic model to select LFBs. | Resources utilisation optimisation | LFB | ForCES router |
| Wang *et al.* [132] | OSPF routing optimisation scheme that minimises the maximum link utilisation. | Resources utilisation optimisation | Data channel | ForCES router |
| Jiang *et al.* [133] | Load balancing mechanism to balance BGP updates among multiple CEs. | Resources utilisation optimisation | Control channel | ForCES router |
| Chen et al [134]. | Bandwidth allocation mechanism to guarantee congestion free traffic between FE and CE. | Resources utilisation optimisation | Control channel | ForCES router |
| Li *et al.* [135] | Scheduling model to increase the communication performance between FEs and CEs. | Resources utilisation optimisation | Control channel | ForCES router |
| Luo *et al.* [136] | Multicast protocols to communicate high number of FEs with the same CE. | Resources utilisation optimisation | Control channel | ForCES router |
| Tarnaras *et al.* [137] | LLDP implemented as an LFB at the FEs to reduce the control traffic. | Resources utilisation optimisation | Control channel | ForCES router |
| Zhong *et al.* [138] | Click-based FE redundancy using cloned virtual machines and VRRP. | Packet loss minimisation | NA | ForCES router |
| Jeong *et al.* [139] | Multi-layer fast fault detection mechanism. | Packet loss minimisation | NA | Network |
| Yoon *et al.* [140] | BFD/OAM mechanism to detect link failures in Diffserv-aware networks. | Packet loss minimisation | NA | Network |

that minimises the maximum link utilisation between the FEs that compose a ForCES router. This approach aims to improve the performance of the ForCES router by reducing the congestion of the traffic inside the router. The Weight-Smart-OSPF algorithm that they propose in their work balances traffic among multiple paths with a certain weight. They demonstrate an improvement of up to 55% in the maximum utilisation of the links compared to pure OSPF.

*c) Control channel:* Some proposals deal with the improvement of the communication between the CEs and FEs. Li *et al.* [135] present a scheduling model between the CE and FE to increase the communication performance between them, whereas Chen *et al.* [134] adopt bandwidth allocation mechanisms between the CE and the FEs in an effort to guarantee congestion free traffic exchange between both elements. Even multicast communication protocols are proposed to cope with the scenario of a high number of FEs per CE [136]. The optimisation of the communication between FEs and CEs is of significant importance, since it can directly affect the overall performance of the ForCES router or network.

One of the most interesting features of ForCES is that it supports multiple CEs working concurrently. This allows to balance the work load among the multiple CEs, which is precisely what Jiang et al propose [133]. In their paper, they present an algorithm to distribute BGP updates among multiple CEs. This approach allows to distribute the computational load of each CE while it minimises invalid route computation. The BGP update messages are processed by the CEs depending on the prefix of the destinations.

Finally, the work proposed by Tarnaras *et al.* [137] deals with the automatic discovery of the network resources, which has also a direct impact on TE. They propose an algorithm to automatically discover the network topology by implementing the Link Layer Discovery Protocol (LLDP) as a LFB directly at the FEs. Topology discovery is tightly coupled to TE, since it is necessary for path computation. The FEs advertise the topology information to the CE when a change occurs. The network discovery is not implemented in the controller, which saves computational resources at the controller since less packets must be processed and reduces the overhead introduced by the control traffic at the control channel since less packets are sent to the controller. This proposal does not require any modifications in the LLDP protocol, as it happens with OpenFlow, and reduces the time required to detect a new device at the controller by an order of magnitude.

*2) Packet Loss Minimisation:* Several proposals deal with fault restoration both in the case of FE failure or link failure. On the one hand, Zhong *et al.* [138] propose a mechanism to obtain FE redundancy based on cloned virtual machines. The forwarding plane of the router is based on two identical FEs based on Click [141] which communicate with each other using Virtual Router Redundancy Protocol (VRRP) and a gateway that communicates with the CE through the ForCES protocol. On the other hand, Jeong *et al.* [139] demonstrate that fault restoration in ForCES networks is possible. Their proposal is a scheme that relies on the fault restoration capabilities of different layers. It uses the fast fault detection mechanisms of the physical layer, some of the classic TE strategies available at the MPLS layer and the hierarchical priority-based resource sharing in IP layer.

In addition, in [140] the implementation of Bidirectional Forwarding Detection / Operations Administration and Management (BFD/OAM) functions in DiffServ-aware MPLS networks using the ForCES architecture for broadband real-time service provisioning with QoS is proposed. BFP/OAM is used to detect link failures and make performance measurements, and it is controlled by the ForCES control plane. The ForCES control plane activates the BFD/OAM for each

TABLE III

SUMMARY OF TE-RELATED RESEARCH WITH OPENFLOW. *Flow Aggreg* STANDS FOR "FLOW AGGREGATION", *Flow Disagg* FOR "FLOW DISAGGREGATION", *Flow Reloc* FOR "FLOW RELOCATION" AND *APC* FOR "ADVANCED PATH COMPUTATION CAPABILITIES". *Inter-DC* STANDS FOR "INTER-DATA CENTER NETWORK", *WAN* FOR "WIDE AREA NETWORK", *REN* FOR "RESEARCH AND EDUCATION NETWORK", *DC* FOR "DATA CENTER" AND *Campus* FOR "CAMPUS NETWORKS." THE SYMBOL ✓ INDICATES THAT THE CAPABILITY IS PRESENT IN THE SOLUTION

| **Authors** | **Performance Objective** | **Scope** | **Controller** | **Techniques** | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | **Flow Aggreg** | **Flow Disagg** | **Load balance** | **Flow reloc** | **Resilience** | **APC** | **Monitoring** |
| Jain *et al.* [38] | Resource utilisation optimisation | Inter-DC | ONIX | | | ✓ | ✓ | ✓ | ✓ | |
| Hong *et al.* [142] | Resource utilisation optimisation | Inter-DC | FloodLight | | ✓ | | ✓ | ✓ | ✓ | ✓ |
| Van deer Pol *et al.* [143] | Resource utilisation optimisation | WAN | - | | | ✓ | | | ✓ | ✓ |
| Mendiola *et al.* [144] | Resource utilisation optimisation | WAN / REN | ONOS | | ✓ | | ✓ | ✓ | ✓ | |
| ESNet [16] | Resource utilisation optimisation | REN | FloodLight | | | | | ✓ | ✓ | |
| Bredel *et al.* [145] | Resource utilisation optimisation | WAN / REN | ODP / FloodLight | | | ✓ | | | ✓ | |
| Das *et al.* [146] | Resource utilisation optimisation | WAN | NOX | | | | | | ✓ | |
| Das *et al.* [147] | Resource utilisation optimisation | WAN | NOX | ✓ | | | | ✓ | ✓ | ✓ |
| Agarwal *et al.* [148] | Resource utilisation optimisation | Hybrid SDN | - | | | ✓ | | | ✓ | |
| Koerner *et al.* [149] | Resource utilisation optimisation | Campus | NOX | | | ✓ | | | ✓ | ✓ |
| Gharbaoui *et al.* [150] | Resource utilisation optimisation | DC | Custom (OVFN) | | | | | | ✓ | ✓ |
| Wang *et al.* [151] | Resource utilisation optimisation | DC | - | | | | | | ✓ | |
| Huang *et al.* [152] | Congestion minimisation | WAN | Trema | | | ✓ | | | ✓ | |
| Braun *et al.* [153] | Congestion minimisation | WAN | - | | | ✓ | | ✓ | ✓ | ✓ |
| Li *et al.* [154] | Congestion minimisation | WAN | POX | | | | | | ✓ | ✓ |
| Tso *et al.* [155] | Congestion minimisation | DC | - | | | | | | ✓ | ✓ |
| Trestian *et al.* [156] | Congestion minimisation | DC | - | ✓ | ✓ | ✓ | | | ✓ | ✓ |
| Kassler *et al.* [37] | QoE maximisation | WAN | - | | | | ✓ | | ✓ | |
| Phemius *et al.* [157] | Packet loss minimisation | WAN | Floodlight | | | | ✓ | ✓ | ✓ | ✓ |
| Nguyen *et al.* [158] | Packet loss minimisation | WAN | - | | | | ✓ | ✓ | ✓ | ✓ |
| Pisa *et al.* [159] | Packet loss minimisation | DC | NOX | | | | ✓ | | | |

TE-LSP and receives information about the link failures and the measured performance in return. This information is later used to compute the paths in the CSPF.

### *B. OpenFlow*

As stated in Section IV-A2, OpenFlow has a number of mechanisms that can benefit TE. Firstly, it provides the means to add, delete and modify meters for traffic shaping. Secondly, its high-granularity makes it ideal to implement flow disaggregation strategies. Thirdly, it supports alternative forwarding methods through the group table, such as load balancing or fast failover. The OpenFlow-based proposals surveyed in this section have been categorised depending on their TE performance objectives, namely *resource utilisation optimisation*, *congestion minimisation*, *QoE maximisation* and *packet loss minimisation*, and the scope (network type) in which the solution is applied. Table III provides a quick reference to the TE solutions where the OpenFlow protocol is used, identifying the scope of the solution, the techniques used to achieve the performance objective and the controller.

*1) Resource Utilisation Optimisation:* This section surveys the OpenFlow-based solutions dealing with the optimisation of the network resource utilisation. In addition, the proposals have been further categorised depending on the network type in which they are applied.

*a) Inter-data center wide area networks:* The implementation of TE strategies based on OpenFlow has become a hot topic since the announcement that Google uses SDN and OpenFlow to optimise the links utilisation in B4 [38], one of its internal Wide Area Networks (WAN), which uses SDN principles and the OpenFlow protocol for the control of the switches. This network supports simultaneously standard routing protocols and a centralised TE solution implemented as an SDN application. According to Google researchers, their solution aims to deliver max-min fair allocation to applications allowing to maximise the utilisation of the network. This solution requires prior knowledge about the network, meaning that it is only valid in networks where the traffic demand and pattern is previously known. Google uses the statistical information they collect about the network usage to optimise it. As a consequence, this solution is entirely customised to fit Google's needs and it would not be possible to apply it directly to control other networks without a similar statistical analysis.

Similar to B4, Software-Driven WAN (SWAN) [142], is a system for inter-DC WANs that improves the network resource utilisation coordinating the sending rates of the different services and centrally allocating network paths. This solution differentiates three priority classes: interactive, elastic and background, being interactive the highest priority class and background the lowest one. SWAN reserves the shortest paths for higher classes' services and allocates bandwidth for these services in strict precedence taking into account their class priority. In this solution, the SDN controller is the one that computes how much traffic each service can send and the network paths that can accommodate that traffic. In order to maximise the network resource utilisation, they consider flow relocation and disaggregation techniques, for which they reserve a certain amount of link capacity and flow entries in the OpenFlow switches to minimise congestion and packet loss during the transitions. They use network agents to obtain information about the network topology, and they inform the controller about link or node failures immediately, which triggers a process at the controller to relocate the services. These network agents also gather information about the resources consumed by each service demand every five minutes, which is later used by the controller to re-compute the service allocations. Through these advanced TE strategies, they are able to carry up to a 60% of additional traffic compared to MPLS-TE.

*b) Research and education networks and wide area networks:* One of the most popular techniques to optimise the network utilisation is load balancing. In OpenFlow networks, load balancing algorithms can benefit from the logically centralised controller plane. In [143], Multi-Path TCP (MPTCP) is used to distribute the traffic across multiple paths in an OpenFlow controlled WAN. On the one hand, MPTCP has the ability to adapt the load balancing according to the load of other traffic flows on the path. On the other hand, the OpenFlow application computes the optimal paths that the different TCP sub-flows can use and commits them to the OpenFlow switches. One of the most interesting features of their proposal is that the path computation is done at the OpenFlow controller, following the PCE-based architecture principles. This allows them to easily implement and test different path optimisation algorithms such as the Edmonds-Karp maximum flow algorithm [160].

As mentioned before, OpenFlow can improve the network resource utilisation through flow relocation and disaggregation mechanisms. On the one hand, the Dynamic Path Computation (DynPaC) Framework [144] provides resilient E2E L2 circuits with bandwidth guarantees in OpenFlow-enabled domains, and it is currently being integrated with AutoBAHN [18], the multi-domain BoD service provisioning tool of the pan-European REN Géant. The framework consists of a stateful PCE that computes the shortest path that satisfies the bandwidth constraints for a given period of time, in which the network resources already consumed by other services are taken into account. Furthermore, the PCE provides pairs of primary and backup paths to guarantee the service provisioning even in case of failure, and allows re-locating and disaggregating already installed flows in real-time to satisfy new service demands that otherwise would be rejected by the system.

Similarly, the OSCARS software framework is the one in charge of managing and automating the network operations based on user-specified requirements to provide multi-domain BoD services in the ESnet. OSCARS provides multi-domain, high-bandwidth virtual circuits that guarantee E2E network data transfers [161]. It is based on the PCE-based architecture and it is currently operated over an MPLS network. However, the OSCARS framework lacks the mechanisms to obtain the topological information automatically, as the network is described by means of a static XML document. Furthermore, it lacks the resources to adapt the routing to topological changes and does not provide any mechanism to monitor or analyse the traffic. In such a scenario, being a production level solution with the code publicly available, there have been some efforts to use the OSCARS software framework over an OpenFlow network to tackle such limitations. For instance, ESnet developed a Path Setup Subsystem (PSS) for OpenFlow-enabled networks based on NOX, which was later adapted to the FloodLight controller [16]. In addition to adapt the OSCARS software framework to OpenFlow, their solution leverages the topology discovery capabilities available at the FloodLight controller, thus adding the possibility to react against network topological changes. However, they have not implemented the entire OpenFlow message and features set, such as the statistical messages and some field rewriting features, and the OpenFlow granularity is not reflected in the GUI, where it is not possible to specify all the matching fields available at OpenFlow. Similar efforts have been made at the OLiMPS project, where they have focused their efforts on integrating the OSCARS software framework with the FloodLight controller [145], and more recently with ODP.

The solution proposed by Das *et al.* [146] seeks to exploit both the advantages of OpenFlow and the highly efficient forwarding of MPLS. The solution is based on the utilisation of the Open Programmable Extensible Networks (OPEN) control plane to implement MPLS-TE based on OpenFlow. They apply the OPEN control plane to an MPLS based data plane implemented using modified Open vSwitches that perform MPLS data plane functionalities and a NOX controller modified to work with some MPLS extensions added to the OpenFlow protocol. However, this solution is merely focused on the application of OpenFlow as an alternative MPLS control plane, leaving other functionalities such as resilience or dynamic LSPs establishment aside. Moreover, in OpenFlow 1.1 the protocol was extended to support MPLS labels as matching field, and even to push and pop MPLS labels. The same authors implement TE techniques in a NOX controller to optimise several services in OpenFlow networks. The optimisation depends on different parameters gathered with real-time monitoring tools at the edge devices of a WAN [147]. Finally, Agarwal *et al.* [148] demonstrate that even in hybrid domains, where OpenFlow devices coexist with non-OpenFlow devices, it is possible to obtain benefits for TE with OpenFlow.

*c) Campus networks:* Koerner and Kao [149] integrate load-balancing functionalities in the existing OpenFlow devices deployed in a campus network. This approach allows to eliminate the need of having additional hardware, and being based on the distribution of the load-balancing functionality among the different switches, they overcome the problem of having a single point of failure. In their approach, they use multiple controllers, one per service, where specific load balance strategies are implemented and then enforced at the OpenFlow switches through L2/L3 address rewriting actions. Furthermore, thanks to the logically centralised controllers aware of the state of the entire network, they are able to apply more advanced balancing policies that take into account the load of the network to improve the workload and network performance.

*d) Data center networks:* As stated by Gharbaoui *et al.* [150], although virtualisation provides important advantages to DC networks, it also raises new challenges regarding DC infrastructure management, since operations such as VM provisioning and reconfiguration occur much more frequently than in legacy DCs. They present an approach that takes into account the current occupation of DC network links when selecting the server in which to allocate a virtual machine. In fact, in a DC network with a typical tree-like topology, the selection of a server and the selection of the corresponding network path are very tightly connected. That is, the selection of one of these parameters heavily constraints the options available for the selection of the other. For this reason, the authors present two selection algorithms, which basically differ in the order in which resources are selected, that is, in one of the algorithms the server is selected first, whereas in the other algorithm the network path is selected first. For each type of resource, the authors propose three selection policies: select the first one with enough available resources, select the most unloaded one or select the most loaded one with enough available resources. In order to implement this approach, the authors propose to use OpenFlow and the flow-level information obtained through statistics messages. The results obtained through simulation show that the proposed approach provides a lower blocking probability than a solution that allocates servers in a random fashion without taking into account network conditions.

Wang *et al.* [151] have designed an SDN controller able to dynamically reconfigure optical circuits to meet the connectivity requirements of big data applications at run-time. The SDN controller provides an interface towards the big data application's master node, which is used by the latter to send traffic demand matrices describing the volume and policy requirements of the traffic between the Top of Rack switches of the DC. This solution allows to quickly allocate and schedule high-bandwidth services to satisfy the connectivity demand. In order to minimise the number of flow rules in the OpenFlow switches and therefore, reduce the network reconfiguration time, they use VLAN tags to differentiate the services.

*2) Congestion Minimisation:* This section surveys the OpenFlow-based solutions dealing with congestion minimisation. In addition, the proposals have been further categorised depending on the network type in which they are applied.

*a) Wide area networks:* Huang *et al.* [152] proposed GridFTP, which relies on the OpenFlow protocol to route different TCP streams along different paths between the given endpoints. More specifically, the authors propose to build an OpenFlow controller which dynamically calculates a fix number of available paths between the source and destination nodes using breadth-first search. Then, the OpenFlow controller installs the appropriate flow entries in the OpenFlow switches in order to divide the TCP streams uniformly through the previously computed paths. With this approach, GridFTP is able to improve the data transmission time by using multiple parallel TCP streams.

Braun and Menth [153] propose a dynamic load balancing approach aimed at dealing with temporary network overloads, restricting the difficult network reconfiguration process only to long-term overloads. The authors assume that every flow has a primary path and a backup path available and that in a normal network condition, all the traffic is transmitted through the primary path. In the case of a network failure or overload, traffic is distributed, if possible, among the primary and backup paths to minimise the congestion. For this aim, three policies are proposed: to apply multipath routing to all the traffic, to apply multipath routing only to the excess traffic or to directly redirect all the excess traffic to the backup path. In order to implement the proposed strategy, the authors make use of the monitoring and fast fail-over mechanisms provided by OpenFlow. By means of an analytical model, the authors demonstrate that the proposed load-dependent flow splitting mechanism reduces the required network capacity and maximum link capacity, compared to traffic-agnostic mechanisms such as single-shortest-path and 2-shortest-paths mechanism, especially in the case of simultaneous network failures and overloads.

A different approach is followed by Li *et al.* [154], in which congestion minimisation is directly handled by applying a CSPF algorithm. They propose to use OpenFlow border routers in order to connect IPv4 and IPv6 islands in an efficient manner. The OpenFlow routers are connected to a centralised controller, which has an overall view of the network topology and state. For each incoming flow, the OpenFlow routers contact the centralised controller to compute the optimum path according to the current network conditions. Once the path is computed, the controller programs the OpenFlow routers so that the traffic is forwarded along the computed path, encapsulating IPv4 packets in IPv6 or vice-versa as needed. This way, the controller is able to select paths that reduce the E2E delay and the network congestion, compared to alternatives that implement TE mechanisms using dual stack routers or pre-configured static IP tunnels.

*b) Data center networks:* The logically centralised control plane of the OpenFlow switch allows to make more accurate decisions based on real-time information. For instance, the *BaatDaaT* [155] flow scheduling algorithm reduces congestion in DC networks based on real-time measurements of network utilisation, and the use of non-SPF algorithm to schedule traffic flows. The support of non-SPF algorithms for the selection of the paths is of interest to this topic because there are cases in which the optimisation function is not significantly affected by the number of the traversed hops.

OpenFlow can also be useful to enhance the utilisation of the DC network resources. The most common approach to apply TE in DC networks is to distinguish between long-lived flows, known as elephant flows, and short-lived flows (mice flows). Then, TE strategies are only applied to elephant flows, while mice flows are routed according to baseline routing methods. Although this approach facilitates the scalability of TE strategies, it might also cause congestion to mice flows, which can correspond to critical network traffic. The authors of this paper propose MiceTrap [156], an approach to extend TE to mice flows without hindering scalability. The main idea behind MiceTrap is to leverage the flow aggregation capacity provided by OpenFlow to handle a number of mice flows together and to apply a weighted routing algorithm to achieve improved load balancing of mice flows. The ratios used to split the traffic among the multiple paths are dynamically computed based on link utilisation.

*3) QoE Maximisation:* Kassler *et al.* [37] propose an architecture for service negotiation and path optimization in SDNs that seeks to maximise the QoE. It is based on two key enablers: QoS Matching and Optimisation Function (QMOF) and Path Assignment Function (PAF). The former one resides in the application plane, whereas the latter one resides at the OpenFlow controller and maintains updated information about the flows installed in the network. It also holds a topology database populated with the topological information provided by OpenFlow. The result is an architecture able to reassign paths in order to admit new services.

*4) Packet Loss Minimisation:* This section surveys the OpenFlow-based solutions dealing with packet loss minimisation. In addition, the proposals have been further categorised depending on the network type in which they are applied.

*a) Wide area networks:* When it comes to the minimisation of packet loss, monitoring the network state and the capability to react upon network failures are of uttermost importance. Some proposals rely on the OpenFlow protocol to gather statistical information from the network devices and to monitor the network by computing some QoS parameters at the controller. For instance, this is the approach followed by [157], where the solution monitors the occupied bandwidth at each link. Later, that information is used to relocate the traffic into less occupied paths. This solution also provides resilience, by detecting link failures and re-directing the traffic to alternative paths. However, the monitoring through OpenFlow mechanisms imposes some challenges that have not been solved yet. For instance, there must be a perfect synchronisation between all the elements of the network, including the controller, which is not trivial, since the latency at the control channel can affect the accuracy of the retrieved network state information. There are other SDN based TE solutions aiming to increase network resilience. Nguyen *et al.* [158] propose a mechanism to improve network resilience at WANs in case of natural disasters such as earthquakes or tsunamis, which also follows an approach based on flow relocations.

*b) Data center networks:* There are also solutions dealing with the minimisation of packet loss in DC networks based on OpenFlow. For example, Pisa *et al.* [159] propose an algorithm to migrate the virtual resources from one virtual network to another virtual network that minimises service disruption, and therefore, packet loss. The solution leverages the forwarding and control plane separation of OpenFlow to rearrange the virtual network topology seamlessly, by reconfiguring forwarding tables to re-route with minimum losses.

## C. I2RS

One of the main benefits of the I2RS protocol for TE is that it exploits the operating system of the router. As a consequence, the I2RS protocol can be supported in legacy routing elements by installing an I2RS agent. This characteristic of I2RS allows to SDN-ise a network without imposing a head to tail restructuring of it. In addition, TE solutions can benefit from an easier access to dynamic information regarding the topology, events and traffic that the router elements already have.

Due to the recent publication of I2RS, at the moment of writing this paper we have identified a single proposal based on this protocol focused on TE. Sgambellur *et al.* [162] propose a generalised SDN controller to provide E2E QoS and TE in access, metro and core networks. Their solution is based on the PCEP protocol to provide guaranteed bandwidth circuits in the IP/MPLS core network and I2RS to control the Passive Optical Network (PON) access network and the OpenFlow-enabled metro networks. It is worth mentioning that the solution considers that I2RS is implemented using the OpenFlow protocol, while it has not been decided yet if I2RS will be implemented using an existing technology such as OpenFlow or a new protocol.

## D. BGP-LS/PCEP

This section reviews TE solutions in which the BGP-LS protocol is used in conjunction with PCEP. With BGP-LS/PCEP, legacy MPLS networks can benefit from the high programmability and logically centralised control plane of SDN. Moreover, given that the PCE-based architecture supports the utilisation of multiple PCEs, TE solutions can span multiple domains. All the solutions surveyed in this section aim to optimise the network resource utilisation. Therefore, the solutions have been classified taking into account if they also try to minimise packet loss or not. Table IV summarises the surveyed proposals, specifying if they provide restoration or not, the PCE type used and whether the solution is applied in a multi-domain scenario or not.

It is worth noting that there are multiple PCE types, depending on the number of PCEs involved or the computational model they follow. Among all the possibilities, the Centralised PCE is of special relevance for this survey as it can be considered as a predecessor of current SDNs. PCEs can be stateful or stateless depending on how they manage the network state. On the one hand, a stateful PCE is aware of both, the network state (links state, bandwidth, etc) and the set of already computed paths and reserved resources in the network. The stateful PCE requires reliable state synchronisation mechanisms, which can result in control plane overhead. On the contrary, a stateless PCE has knowledge about the network topology (nodes, links, bandwidth, etc), information that it uses for the path computation, but it does not take into account the amount of resources that are already used or reserved in the network (e.g., current link utilisation). That is, in a stateless PCE each request is processed independently, without considering the resources allocated by previous requests, which results in a much simpler path computation.

In addition, BGP-LS is not only a protocol that network devices can use to inform about the topological and link state information. It can also be used in Hierarchical-PCE (H-PCE) to exchange TED information between PCEs [168]. In multi-domain path computation, child-PCEs are in charge of the computation of the paths in each domain, while the parent-PCE is in charge of what in the multi-domain terminology is known as the inter-domain path computation. The multi-domain path computation is in fact a two step process in which first the parent-PCE computes the domain sequence and then the child-PCEs compute the paths inside each domain. Now that the different PCE types have been

TABLE IV
SUMMARY OF TE-RELATED RESEARCH WITH BGP-LS/PCEP. *M-PCE* STANDS FOR "MULTIPLE-PATH COMPUTATION ELEMENT" AND *H-PCE* FOR "HIERARCHICAL-PATH COMPUTATION ELEMENT". THE SYMBOL ✓ INDICATES THAT THE CAPABILITY IS PRESENT IN THE SOLUTION

| Authors | Summary | Performance objective<br>Packet loss minimisation | Scope | PCE Type | Multi-domain |
|---|---|---|---|---|---|
| Martínez *et al.* [165] | Resilience and LSPs re-optimisation through a coordinated PCE where BGP-LS is used to synchronise the TED. | ✓ | GMPLS/Flexigrid | Stateful M-PCE | |
| Giorgetti *et al.* [166] | Proactive update of parent TED in H-PCE using BGP-LS to reduce the blocking probability. | ✓ | GMPLS/EON | Stateful H-PCE | ✓ |
| Casellas *et al.* [167] | Multi-domain EON lightpath provisioning involving multiple domains taking into account the availability of network resources. | | GMPLS/EON | Stateful H-PCE | ✓ |
| Cuaresma *et al.* [168] | Compares two algorithms that use BGP-LS to exchange a different amount of TE information to build the TED. | | GMPLS/EON | H-PCE | ✓ |
| Casellas *et al.* [169] | Multi-domain E2E service provisioning across heterogeneous domains taking into account frequency availability. | | OpenFlow & GMPLS/Flexigrid | Stateful H-PCE | ✓ |

described, the following subsections present the TE solutions based on BGP-LS/PCEP.

*1) Solutions Dealing With Packet Loss:* In flexigrid optical networks, PCEs are in charge of executing a Routing and Spectrum allocation algorithm to compute the physical route and the frequency slot that each LSP will use. In order to re-optimise LSPs, which is a time and resource consuming task, Martínez *et al.* [163] propose to use a front-PCE and a back-PCE, the first one in charge of the algorithm execution for new and restored LSPs and the second one in charge of LSPs re-optimisation. Having this architecture requires both PCEs to be coordinated, which is achieved using BGP-LS for TED synchronisation and the PCEP REPORT message to exchange information about LSPs. In case of a failure affecting an established LSP, the front-PCE computes another path and once it is setup, it communicates with the back-PCE to inform about the new TED and to request a re-optimisation of the affected LSPs.

In such a scenario, Giorgetti *et al.* [164] has proposed a proactive scheme to update the parent-PCE TED to provide multi-domain connectivity services in Elastic Optical Networks (EON). The updates are triggered upon path computation requests, resulting in a lower control traffic exchange between the child-PCEs and the parent-PCE. In addition, this solution reduces the blocking probability, that is, the same network is able to accept more service demands. The parent-PCE computes the paths taking into consideration per-link spectrum availability information, information that it uses to optimise the utilisation of such resource. In addition, the solution also provides restoration capabilities, thus, it aims to minimise the packet loss.

*2) Solutions That Do Not Deal With Packet Loss:* De Dios *et al.* [165] extend BGP-LS, OSPF-TE, PCEP and RSVP-TE to provide multi-domain service provisioning in flexigrid optical networks. In this particular case, BGP-LS is used by the optical devices to inform the child-PCEs about the frequencies they use at each link. The same approach is followed in [166] to provide multi-domain path computation in EONs. In this solution, each domain contains a TED built using IGP information that the child-PCEs use to compute the paths inside their domain. Each domain also has a BGP Route Reflector that uses BGP-LS to send the topological information of the domain to the parent-PCE. The authors compared two algorithms to determine which strategy is more efficient for the E2E path computation. In the first one, only the TE information necessary to perform the inter-domain path computation is sent to the parent-PCE. Whereas in the second one, all the TE information is sent, which allows the parent-PCE to compute more optimal paths, enabling a better utilisation of the network resources.

BGP-LS can be useful in numerous ways for multi-domain TE. A typical challenge in multi-domain solutions is how to deal with technology diversity, that is, how to operate different transport technologies and their control planes in a unified manner. Casellas *et al.* [167] integrate OpenFlow and flexigrid networks with a H-PCE to solve the problem of heterogeneous control plane interworking. In this case, BGP-LS has been extended to support the encoding of OpenFlow datapath identifiers and to inform about the status of the nominal central frequencies that characterise flexigrid links.

## VI. TE SOLUTIONS BASED ON A-CPI PROTOCOLS

The ALTO protocol is the only protocol analysed in this survey that fits into the A-CPI category. It is worth noting its ability to provide information about the state of the network, information that can be later used to improve network performance. This section provides a literature review until 2015 of TE-related solutions in which the ALTO protocol has been used. Since all solutions aim to optimise the resource utilisation, they have been categorised depending on their scope: *Inter-Data Center networks*, *Wide area networks*, *P2P Overlays* or *Mobile networks*. As a quick reference, Table V summarises the TE-related research with ALTO, including information about the scope in which the solution is applied, if the solution is used together with other SDN protocols and about their performance objectives.

### *A. Inter-Data Center Networks*

As mentioned before, the retrieval of accurate and up-to-date network state information has a direct impact on the success of the TE mechanisms and strategies applied. An efficient abstraction of the network resources and topology plays a key role in TE, since the optimisation algorithms would have simpler information to process. In SDN, network state information is retrieved by the controller plane and later abstracted

TABLE V

Summary of TE-Related Research With ALTO. *Inter-DC* Stands for "Inter-Data Center network", *P2P Overlay* for "Peer-to-Peer Overlay Network", *WAN* for "Wide Area Network" and *Mobile* for "Mobile Network". *RUO* Stands for "Resource Utilisation Optimisation", *QoE* for "Quality of Experience" and *Peer sel.* for "Optimise Peer/Endpoint Selection". The Symbol ✓ Indicates That Other SDN Protocols Are Involved in the Solution

| **Authors** | **Summary** | **Scope** | **Other SDN** | **Performance Objective** | | |
|---|---|---|---|---|---|---|
| | | | | **RUO** | **QoE** | **Peer sel.** |
| Scharf *et al.* [169] | Expose and orchestrate information to control VPN and bandwidth reservation between the VPN sites on-demand | Inter-DC | | ✓ | | |
| Gurbani *et al.* [170] | ALTO to provide network information to an SDN application that selects among multiple replicas of a resource to provide the best service. | Inter-DC & DC | ✓ | ✓ | | ✓ |
| Li *et al.* [171] | ALTO client colocated with an SDN controller that setups paths to interconnect CDN networks. | Inter-DC | ✓ | ✓ | | ✓ |
| Zhang *et al.* [172] | Multipath transport framework for the application layer to improve the network resource utilisation. | WAN | | ✓ | ✓ | |
| Scharf *et al.* [173] | Abstract the topology information retrieved from multiple sources using different levels of detail to ease path computation. | WAN | | ✓ | | |
| Gurbani *et al.* [174] | Build bandwidth and latency cost maps using public records to improve P2P applications performance. | P2P overlay | | ✓ | | ✓ |
| Wang *et al.* [175] | Study the interaction between ALTO-assisted P2P overlay and the ISP's application agnostic TE strategies. | P2P overlay | | ✓ | | ✓ |
| Faigl *et al.* [112] | ALTO client that selects the preferred endpoints is embedded in an SDN controller that programs the network devices to setup a path between them. | Mobile | ✓ | ✓ | | ✓ |

to the application plane. As a result, SDN applications running optimised routing algorithms can benefit from the information provided by interfaces like ALTO.

ALTO can be helpful to orchestrate and expose information in distributed clouds, allowing the provisioning of high quality services among DCs [169]. In this solution, the network topology information provided by ALTO is used to grow Virtual Private Networks (VPN) and modify the reserved bandwidth between the different sites of the VPN on-demand service. The authors of this work state that this ALTO-based solution presents some benefits. For instance, it allows applications to communicate through the VPN using the most appropriate path without the need for them to discover the topology or the available resources on the network. With this solution, users do not need to perform measurements and are abstracted from the underlying routing protocols, while the NSP does not need to advertise all the details of its network.

Since the SDN paradigm emerged, some authors have stated that ALTO fits perfectly into the SDN environment, where network state abstractions are envisaged [170]. They argue that ALTO can be used to provide network information to SDN applications able to use different replicas of the same resource to provide the service, which they call rendezvous applications. For instance, ALTO is able to provide the necessary information to enable the utilisation of optimised routing algorithms inside the DC and between different DCs, and to enhance the network resource management.

Li *et al.* [171] have proposed an architecture for CDNi (Content Delivery Interconnection) that relies on ALTO and an unspecified D-CPI protocol. An ALTO server provides an ALTO Map Service containing information about the network state, including topology, costs and additional security information. This information is transmitted to the ALTO Client, which resides in a Control Center with an SDN controller in charge of establishing paths between the most appropriate edge servers as selected by the CDNi controller. There are multiple applications in which the peer or endpoint selection plays a key role, since it can influence other performance objectives such as congestion, delay or the QoE [176]. In such a context, the ALTO protocol can be highly beneficial, since it provides network state information that can facilitate the optimal peer or endpoint selection.

### B. Wide Area Networks

Zhang *et al.* [172] propose a multipath transport framework called MPTS-AR that operates at the application layer to improve the network resource utilisation, to increase the reliability and throughput and to enhance the users' experience. In order to select the best combination of multiple paths they use ALTO, to take into account not only routing costs but also the actual load in the paths involved. One of the main benefits provided by this multipath framework is that thanks to ALTO it is possible to find a superior relay path. In this context, a relay path refers to a path whose relative performance is under certain threshold and balances the overall traffic inside the network in the most efficient way.

As mentioned in Section IV-B, ALTO provides the means to aggregate the information obtained through various protocols to provide simplified and complete network state information. This technique is used in [173], where ATLAS, the Accurate Topology Level-of-Detail Abstraction System, extracts topological information directly from the network management system and protocols like IS-IS or BGP and abstracts it using a contraction algorithm. This system relies on ALTO to expose the abstracted topological information with different levels of details, taking into account the different policies of the

TABLE VI
SUMMARY OF TE-RELATED RESEARCH WITH MI PROTOCOLS. *NA* STANDS FOR "NOT APPLICABLE"

| Authors | Summary | Protocol | Objective | Network type | Controller |
|---|---|---|---|---|---|
| Palma *et al.* [180] | Architecture with support for queue configuration messages | OVSDB | Basic queue setup | OpenFlow | FloodLight |
| Sharma *et al.* [181] | QoS framework to guarantee high priority traffic delivery even in case of failure | OVSDB | Dynamic queue reconfiguration | OpenFlow | FloodLight |
| Caba *et al.* [182] | QoS architecture to provide BoD | OVSDB | Dynamic queue reconfiguration | OpenFlow | FloodLight |
| Pereira *et al.* [183] | TE-related policy management in DiffServ-aware networks | NETCONF | Policy provisioning | MPLS | NA |
| Martini *et al.* [184] | Resource Admission Control Function to provide QoS in NGN. | NETCONF | Policy provisioning | MPLS | NA |
| Oliveira *et al.* [185] | Adoption of SDN paradigm for reconfigurable optical testbed | NETCONF | LSP instantiation and configuration | GMPLS / Flexigrid | NA |
| Aoki *et al.* [186] | L2 circuit on-demand provisioning and fine grained traffic analysis | NETCONF | LSP instantiation and configuration | MPLS | NA |
| Loureiro *et al.* [187] | Agent for link state monitoring | NETCONF | Failure detection | MPLS | NA |
| Malishevskiy *et al.* [188] | Efficient and secure network resource manager | OF-COFIG | Queue management | OpenFlow | Unknown |
| Wedong *et al.* [189] | Automatic QoS management framework for SDN | OF-COFIG | Queue management | OpenFlow | NOX |

Network Service Providers (NSP) and the nature of the application that will use that topology information. Using this mechanism they are able to produce an order of magnitude smaller network and two orders of magnitude smaller costs maps. Since the size of these maps affects the processing time of the algorithms, smaller maps imply lower processing times to select the most appropriate routes, while the traffic optimisation inside the applications is not considerably affected.

### C. P2P Overlays

A recent study has demonstrated that it is possible to create ALTO networks and topology maps using DNS, active performance measurements and video simulation data available at public source. Additionally during this process the subscribers' anonymity is guaranteed [174]. The information is used to construct bandwidth and latency cost maps that allow to select the best peers to increase P2P applications' performance without requesting private or sensitive information to the NSPs.

However, the utilisation of ALTO imposes some challenges, especially when it is applied in networks where the technology is not TE-ready. For instance, Wang *et al.* [175] studied the interaction between an ALTO-assisted P2P overlay and the ISP's application agnostic TE strategies. They demonstrated that the lack of cooperation between the two of them affects the overall network performance. For instance, in the cases where the ALTO-assisted P2P overlay does not take into account the overall network traffic performance to select the peers, the non-P2P traffic can be negatively impacted. They concluded that in the cases where multiple entities affect the routing decisions, the overall system stability and performance has to be taken into account to avoid a negative impact on non-TE traffic.

### D. Mobile Networks

Faigl *et al.* [112] present an ALTO-SDN architecture where the ALTO client is implemented as an SDN application and assumes the selection of the preferred endpoint. In theory, this decision-making should be implemented at the ALTO server, but they have decided to follow this approach because it is better to implement communication intensive applications as modules in the SDN controller. The ALTO server acts as a network and cost map information service that the ALTO clients query whenever a distributed service requires the ALTO guidance for its establishment. In this proposal, the ALTO server can dynamically request network information to the SDN controller, which provides an up-to-date network view including load information retrieved from the switch port statistics. Once the ALTO client has selected the best destination using the information provided by ALTO, the SDN controller enforces the connectivity between the required endpoints by installing the flow entries.

## VII. TE SOLUTIONS BASED ON MI PROTOCOLS

This section surveys a comprehensive list of TE solutions where MI protocols are used. The solutions are classified depending on the MI protocol that they rely on: OVSDB Management Protocol, NETCONF or OF-CONFIG. Since these are MI protocols, their contribution to TE is associated to their capability to enforce the selected QoS or TE strategy in the data plane. In addition, the solutions have been further categorised taking into account the objective of the solution. Table VI summarises a comprehensive list of TE-related solutions that rely on MI protocols, including information about the MI protocol on which relies, their objective, the network type in which the solution is applied and the controller that is used.

### A. OVSDB Management Protocol

As mentioned in Section IV-C1, OVSs expose two different interfaces: OpenFlow for control purposes and OVSDB for the management of OVSs' configuration databases. This means that by means of OVSDB it is possible to create, delete and modify datapaths, ports, tunnels, queues and their configuration. OpenFlow by itself does not provide the means to enforce

QoS at the data plane, making the use of an MI protocol absolutely necessary in order to provide powerful and automated TE solutions. Many of the controllers and network operating systems available at the time of writing this paper have started to include OVSDB plugins in their architectures, such as ODP and ONOS. The solutions presented in this section have been categorised depending on whether they deal with basic queue setup or with dynamic configuration of queues.

*1) Basic Queue Setup:* First SDN controllers did not include OVSDB plugins in their architectures and it was not until 2013 that the first projects trying to solve this limitation appeared. For instance, back in 2014 Palma *et al.* [177] proposed an architecture with support for queue configuration messages through OVSDB, called QueuePusher. It has been implemented as an extension to the FloodLight controller, and it has been designed to be easily integrated with other controllers and third parties' software through a REST API. This module provides the means to create, update, delete and modify queues in OVSs, although it does not specify how it configures these queues once they are created.

*2) Dynamic Queue Configuration:* Sharma *et al.* [178] implement a QoS framework with failure recovery mechanisms that guarantees, even in case of failure, that high priority traffic is handled before best effort traffic. They add OVSDB support to the FloodLight controller to support remote and dynamic queue configuration.

A more recent work, proposed by Caba and Soler [179], presents a data plane QoS architecture to provide BoD service, where the QoS is enforced at the data plane by using queues and rate limiters. The proposed solution uses OVSDB to configure the different priority queues associated to an output port, which are later used by the different QoS classes to share the available bandwidth of an output port. Each priority queue is configured through four parameters: minimum serving rate, maximum serving rate, queue size and priority.

### *B. NETCONF Protocol*

As mentioned before, NETCONF is an unified management interface able to extend the basic operation of the network elements. Presently, there are a few published proposals related to TE with NETCONF, although they represent a very significant and interesting collection. The solutions presented in this section have been categorised depending on their scope, distinguishing solutions for policy provisioning, LSP instantiation and configuration and failure detection.

*1) Policy Provisioning:* First, Pereira and Granville [180], evaluate the performance of NETCONF for the management of DiffServ-aware MPLS networks. They conclude that NETCONF can replace other protocols such as COPS-PR to transfer policies to DiffServ-aware devices, which can have a direct impact on TE, as stated in [187].

Later, Martini *et al.* [181] propose NETCONF for the QoS provisioning in Next Generation Networks (NGN) [188]. Their solution is based on the configuration of the edge-nodes through the NETCONF protocol in order to exploit the DiffServ-aware TE capabilities present in MPLS. In Next Generation Networks, the Resource Admission Control Function (RACF) is responsible for the admission control of network service requests. It takes into account the available resources in the network, and in the cases where the service is admitted, allocates the necessary resources to support the service with the required QoS. In this proposal, the QoS is provided through DiffServ-aware TE, where the NETCONF protocol is used to configure the edge nodes. The RACF is consists of two elements: the Policy Decision Functional Entity (PD-FE) and the Transport Resource Control Functional Entity (TRC-FE). Upon a new service request, the PD-FE obtains the addresses of the relevant edge routers, the TRC-FEs connected to those routers and all the information necessary to translate the service parameters into network resource requirements, i.e., required bandwidth and traffic category. Then, the PD-FE communicates with the TRC-FE to inform about the service requirements, and the TRC-FE uses NETCONF to communicate with the involved edge routers to check whether there is an LSP with enough resources available or not and to modify some parameters of the LSPs, like bandwidth.

*2) LSP Instantiation and Configuration:* NETCONF-based TE solutions can also be applied to flexigrid optical networks. Oliveira *et al.* [182] present a reconfigurable optical testbed composed of Reconfigurable Optical Add-Drop Multiplexers (ROADM) with a controller daemon programmable through Command Line Interface (CLI) and NETCONF interfaces. Furthermore, they use the local NETCONF database to store topological information obtained through LLDP which is then used to instantiate TE-links, which is the term they use for TE-enabled paths.

For TE, traffic measurements and analysis is fundamental. In the Japanese SINE4 REN, technologies such as NetFlow and NETCONF are used to provide fine grained traffic analysis [183]. Furthermore, SINE4 provides L2 circuits on demand using MPLS and NETCONF, where the latter protocol is used to configure the routers (i.e., to specify the flow sampling rate). In addition, they also use NETCONF to obtain the configuration information of the routers and extract topology information.

*3) Failure Detection:* Finally, NETCONF can also be used for fast failure detection, which is very important to provide resilience. Loureiro *et al.* [184] have developed a NETCONF agent for link state monitoring. The agent uses event notifications to inform the manager about link failures. The manager creates an event subscription in the agent that results in the generation of two independent threads: one in the agent to monitor and detect link failures and another one in the manager to listen for notifications. Once a link failure event is detected, the agent asynchronously notifies the manager, using SOAP to transport the NETCONF messages.

### *C. OF-CONFIG*

TE solutions involving OF-CONFIG have started to appear, confirming that OF-CONFIG complements OpenFlow to provide complete QoS and TE solutions, since it is a protocol that allows to enforce the required QoS at the data plane.

NETMAN [185] is a new network manager that allows to manage the network resources efficiently and securely. It relies on OF-CONFIG to manage and configure OpenFlow switches and the authors conclude that OF-CONFIG and OVSDB support the same set of functionalities regarding ports, tunnel or QoS configuration, which makes OF-CONFIG the perfect candidate to substitute OVSDB in non OVS-based network devices.

Wedong *et al.* [186] propose AQSDN, an autonomic QoS management framework for SDN. They have included a QoS scheme decision module in their SDN application, which determines which are the best queue management and scheduling schemes for the newly connected switches. This module uses OF-CONFIG to configure the QoS scheme and the parameters of the associated queues (i.e., minimum and maximum transmission rates) once a switch is connected to the controller or when the operator of the network decides to do it. Then, the application uses these queues to direct the traffic to them according to the QoS policies. The authors demonstrate the feasibility of their solution and measure the performance of their management framework for video delivery. They conclude that the performance obtained is higher compared to what is achieved with DiffServ-aware TE.

## VIII. Qualitative Evaluation of SDN-Based TE Solutions

This section provides a qualitative evaluation of the impact and contributions to TE of a comprehensive list of SDN protocols. The evaluation and further discussion is arranged taking into consideration the taxonomy described in Section III-B, in order to determine how the three interface types described in the ONF's SDN architecture impact TE. Given that the PCE-based architecture is the latest TE architecture, the SDN protocols analysed in this paper have been compared to it taking into consideration the metrics described in the following subsection.

### A. Evaluation Metrics

The SDN architecture supports the utilisation of dedicated elements for the computation of the paths and the implementation of TE strategies. Since this approach is also followed by the PCE-based architecture, the metrics used for the evaluation of PCE solutions can also be applied to SDN-based TE solutions. The RFC 4655 [17] proposes the following set of metrics to evaluate the performance, efficiency and applicability of the different PCE solutions:

- *Optimality:* the ability to maximise network utilisation and minimise cost, considering QoS objectives, multiple regions and multiple layers.
- *Scalability:* the implications of routing, TE LSP signalling, and PCE communication overhead, such as the number of messages and the size of the messages. At the time of evaluating SDN protocols, it will depend on the type of the signalling and control mechanisms used. With out-of-band mechanisms, these parameters will not introduce any overhead in the data traffic, while in the case of in-band mechanisms, they will.
- *Load sharing:* the ability to allow multiple PCEs to spread the path computation load by allowing multiple PCEs to take responsibility for a subset of the total path computation requests. It should not be confused with load balancing the traffic among multiple paths. In the case of SDN protocols, it refers to the ability to have multiple controllers implementing the TE solution.
- *Multi-path computation:* the ability to compute multiple and potentially diverse paths to satisfy load-sharing of traffic and protection/restoration needs including E2E diversity and protection within individual domains.
- *Re-optimisation:* the ability to perform TE LSP path re-optimisation. In the case of SDN protocols, it refers to the ability to relocate flows onto alternative paths.
- *Network stability:* the ability to minimise any perturbation on existing TE state resulting from the computation and establishment of new TE paths.
- *Accurate TED synchronisation:* the ability to maintain accurate synchronisation between TED and network topology and resource states.
- *TED synchronisation speed:* the speed which TED synchronisation is achieved with.
- *Impact on data flows:* the impact of the synchronisation process on the data flows in the network.

In addition, the authors of this paper also propose two additional evaluation metrics to better characterise the technologies analysed in this manuscript:

- *Granularity:* refers to the number of possible classifiers that can be used at the network devices to forward the packets. That is, the number of header fields and wildcarding options that can be taken into account to classify the packets at the networking devices (e.g., an IPv4 source address and its mask). The higher the number of available packet classifiers the higher the granularity is, resulting in finer-grain flows. It has clear implications on the multi-path capabilities of the solutions, since a higher granularity allows to split the traffic more conveniently. Similarly, it also impacts the optimality and the re-optimisation capabilities of the solutions.
- *Equipment configurability:* capacity to configure the network equipment to enforce the establishment of paths.

Given these metrics, Table VII provides a quick overview of the impact on TE of the D-CPI, A-CPI and MI protocols analysed in this paper.

### B. Contributions of D-CPI Protocols to TE

In a nutshell, all the D-CPI technologies analysed in this paper are suitable to improve TE in current networks. The present section provides a detailed evaluation of the TE capabilities of the four technologies included in this category, ForCES, OpenFlow, I2RS and BGP-LS/PCEP. Table VIII provides a detailed evaluation of the current proposals where D-CPI protocols are used in traffic engineered networks. The evaluation and further discussion has been arranged taking into account the metrics described in the previous section.

*1) Optimality:* All the D-CPI protocols analysed in this survey contribute to TE by enhancing the optimality of the solutions. In the case of OpenFlow and I2RS, the logically

TABLE VII
Summary of the Impact of SDN Protocols on TE. The Symbols Indicate That the SDN Protocol Capabilities Are (✓):*better*, (✗):*worse* or (=):*equal* Compared to the Capabilities of the PCE-Based Architecture

| | **Optimality** | **Scalability** | **Load sharing** | **Multi-Path comp.** | **Re- opt.** | **Network Stability** | **TED accu.** | **TED Sync speed** | **Impact on data flows** | **Granularity** | **Equipment config.** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ForCES | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | = | ✓ | ✓ |
| OpenFlow | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| I2RS | ✓ | ✗ | ✓ | ✓ | ✓ | = | ✓ | ✗ | = | = | |
| BGP-LS/PCEP | ✓ | * | ✓ | ✓ | ✓ | = | ✓ | ✗ | = | = | |
| OVSDB | | | | | | | | | | | ✓ |
| NETCONF | | | | | | | ✓ | | | | ✓ |
| OF-CONFIG | ✗ | | | | | | ✗ | | | | ✓ |
| ALTO | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | |

centralised control plane contributes to the implementation of more ambitious TE strategies using up-to-date information about the network state [38]. The case of OpenFlow is of particular interest because it does not impose a specific routing algorithm or any legacy routing protocol. In OpenFlow, the controller plane can be fully programmed from scratch and implement, for instance, alternative routing protocols, even non-IP ones [144].

Nevertheless, there are some constraints imposed by OpenFlow to the optimality of the TE solution as the result of different optional features supported at network devices. Furthermore, the way these features are implemented in the hardware devices can affect the performance of the overall solution. For instance, Durner *et al.* [190] evaluate the performance of different OpenFlow-enabled switches regarding dynamic QoS enforcement of the network traffic. More specifically, they study how the different queueing techniques impact TCP traffic. The measurement results show that in some devices, both priority queueing and bandwidth guaranteed queueing lead to packet duplication, thus, affecting the effective bandwidth. They also demonstrate that priority queueing mechanisms can cause flow depletion and TCP connection interruption. In addition, each vendor implements QoS mechanisms in a different way, slowing down the pace to implement QoS mechanisms in OpenFlow networks effectively. However, some proposals are also focused on solving the switch diversity problem. Tango [191] is a framework that allows to deal with switch diversity based on a proactive probing engine able to measure key properties that can affect the switches' performance.

Having in mind that the OpenFlow protocol can be considered a subset of ForCES protocol [72], it can be deducted that ForCES presents the same advantages in this regard. However, even if this is what the ForCES-related RFC documents specify, the reality is that ForCES-based solutions still rely on classic routing protocols [133]. In such solutions, ForCES maintains the distributed control plane. What varies is the way the control plane communicates with the forwarding plane. As a result, OpenFlow has better re-optimisation capabilities than ForCES. As mentioned before, the granularity has also an impact on the optimality. Therefore, the high granularity of both ForCES and OpenFlow also enhances the optimality achievable with these two protocols.

In the case of BGP-LS/PCEP, the utilisation of BGP-LS enhances the optimality of the PCE-based architecture. As outlined in Section II-D3, one of the current limitations in the PCE-based architecture is that it lacks the mechanisms to guarantee up-to-date TE information in the TED. Since BGP-LS is used to provide information about the topology and the links' state, the TED can better reflect the network state in real-time. As a result, the optimality is enhanced, since the algorithms in charge of the path computation can take into account more reliable information. In addition, the optimality also takes into account the capability of the solutions to optimise the network resources across multiple regions. In that regard, BGP-LS can be used to exchange accurate topology information between M-PCEs [163] or in H-PCE [164]. In this latter case, the utilisation of BGP-LS can improve TE solutions involving multiple domains [165].

Finally, I2RS provides an alternative way to control the configuration and the diagnose of the operation of MPLS links [99]. I2RS Clients are able to control the MPLS-TE network by analysing its operational state and TE LSP data, so as to manipulate the configuration of these TE LSPs [162].

*2) Scalability:* In this survey, the scalability refers to the overhead introduced by the analysed technologies and the implications of routing. In general, there is a trade-off between the lower number of protocols and the additional elements required by each solution. On the one hand, a higher number of elements can reduce scalability, while a lower number of protocols can enhance it. Furthermore, if the solution presents a higher scalability or not does not only depend on the number of protocols being replaced, but also on the overhead introduced by them, and a thorough study should be performed in each case.

For instance, in the case of ForCES and OpenFlow, both protocols have the potential to enhance the scalability of the TE solution since many protocols may become expendable. Notwithstanding, the impact of the traffic exchange between the network devices and the controller should also be taken into account. This is of special relevance for in-band control, where the control traffic of multiple devices is exchanged using the same channel. In this regard, it is a well-known issue that the OpenFlow controller can be a bottleneck in large deployments, which can result in the controller being unable to process all the requests of incoming packets [194].

TABLE VIII
DETAILED EVALUATION OF THE D-CPI PROPOSALS. THE SYMBOLS INDICATE THAT THE SDN PROTOCOL CAPABILITIES ARE (✓):*better* OR (✗):*worse* COMPARED TO THE PCE-BASED ARCHITECTURE

| Protocol | Ref. | Optimality | Scalability | Load sharing | Multi-Path comp. | Re- opt. | Network Stability | TED accu. | TED Sync speed | Impact on data flows | Granularity | Equipment config. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ForCES | Bin *et al.* [131] | ✓ | | | | | | | | | | ✓ |
| | Wang *et al.* [132] | ✓ | ✗ | | ✓ | | | | | | | |
| | Jiang *et al.* [133] | ✓ | | ✓ | | | | | | | | |
| | Chen *et al.* [135] | | ✓ | | | | | | | | | |
| | Li *et al.* [135] | | ✓ | | | | | | | | | |
| | Luo *et al.* [136] | | ✓ | | | | | | | | | |
| | Tarnaras *et al.* [137] | | ✓ | | | | | ✓ | ✓ | | | |
| | Zhong *et al.* [138] | | | | | ✓ | | | | | | |
| | Jeong *et al.* [139] | | | | | ✓ | | | | | | |
| | Yoon *et al.* [140] | | | | | ✓ | | ✓ | | | | |
| OpenFlow | Jain *et al.* [38] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | |
| | Hong *et al.* [142] | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | ✓ | |
| | Van deer Pol *et al.* [143] | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | Mendiola *et al.* [144] | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | ✓ | |
| | ESNet [16] | ✓ | ✓ | | ✓ | ✓ | | ✓ | | | ✓ | |
| | Bredel *et al.* [145] | ✓ | ✓ | | ✓ | | | ✓ | | | ✓ | |
| | Das *et al.* [147] | ✓ | ✓ | | ✓ | ✓ | | ✓ | | | ✓ | |
| | Das *et al.* [146] | ✓ | ✓ | | | | | ✓ | | | | |
| | Agarwal *et al.* [148] | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | Koerner *et al.* [149] | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | Gharbaoui *et al.* [150] | ✓ | | | | | | ✓ | | | | |
| | Wang *et al.* [151] | ✓ | | | | | | ✓ | | | | |
| | Huang *et al.* [152] | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | Braun *et al.* [153] | ✓ | ✓ | | ✓ | ✓ | | ✓ | | | ✓ | |
| | Li *et al.* [154] | ✓ | ✓ | | | | | ✓ | | | | |
| | Tso *et al.* [155] | ✓ | | | | | | ✓ | | | | |
| | Trestian *et al.* [156] | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | Kassler *et al.* [37] | ✓ | ✓ | | | ✓ | | ✓ | | | | |
| | Phemius *et al.* [157] | | ✓ | | ✓ | ✓ | | ✓ | | | ✓ | |
| | Nguyen *et al.* [158] | | ✓ | | ✓ | ✓ | | ✓ | | | ✓ | |
| | Pisa *et al.* [159] | | ✓ | | | ✓ | | ✓ | | | | |
| | Iyer *et al.* [189] | | ✓ | | | | | | | | | |
| | Durner *et al.* [190] | | | | | | | | | ✗ | | |
| | Lazaris *et al.* [191] | | | | | | | | | ✓ | | |
| | Kuzniar *et al.* [192] | | | | | | ✗ | | | ✗ | | |
| | Rotsos *et al.* [193] | | | | | | | | | ✗ | | |
| I2RS | Huang *et al.* [99] | ✓ | | | | ✓ | | | | | | |
| | Sgambelluri *et al.* [162] | ✓ | ✗ | | | | | | | | | |
| BGP-LS / PCEP | Martínez *et al.* [163] | ✓ | ✗ | ✓ | | ✓ | | ✓ | ✗ | | | |
| | Giorgetti *et al.* [164] | ✓ | ✓ | ✓ | | | | ✓ | ✗ | | | |
| | Casellas *et al.* [165] | ✓ | ✗ | ✓ | | | | ✓ | ✗ | | | |
| | Cuaresma *et al.* [166] | ✓ | ✗ | ✓ | | | | ✓ | ✗ | | | |
| | Casellas *et al.* [167] | ✓ | ✗ | ✓ | | | | ✓ | ✗ | | | |

However, all in all, many of the OpenFlow-based solutions analysed in this document improve the scalability as a consequence of keeping the number of protocols to the minimum [145].

In the case of ForCES, the scalability improvement also depends on whether the legacy protocols are kept or not. It has to be taken into account that under the same circumstances, for instance, a solution where OSPF-TE is used, the adoption of the ForCES framework will impose additional elements and an extra protocol [132].

This is precisely what happens with I2RS, where the additional elements reduce the general scalability of the TE solutions based on this protocol; apart from the element in charge of the path computation, at least an I2RS Agent and an I2RS Client are involved, which can be physically separated [96].

Finally, in BGP-LS, the TE and link state information is embedded in the BGP protocol [103]. Furthermore, the BGP protocol can also be used to distribute the MPLS labels [195]. As a consequence, although additional header space is needed to distribute the MPLS labels, the scalability of the solution can be considered to be improved, since no additional protocols, infrastructure or software agents are needed to deploy BGP-LS in the TE solution. However, the amount of data transferred with BGP-LS can be huge in big networks or complex scenarios and this could negatively impact the scalability [166].

*3) Load Sharing:* Regarding the load sharing, this is one of the metrics in which the four D-CPI technologies behave similarly. In the four cases it is possible to have multiple control elements working simultaneously. There are different strategies that could be followed. For instance, as it happens in ForCES, each CE could be in charge of a subset of the control functionalities [133] or, as it happens in the BGP-LS/PCEP using the H-PCE architecture [168], the path computation process could be shared among multiple PCEs. It is also possible to apply load sharing techniques among OpenFlow controllers. In fact, some network operating systems like ONOS [91] or ODP [92] support clustering, where each cluster node is in charge of a subset of the network devices.

*4) Multi-Path Computation:* The multi-path computation capabilities of both OpenFlow and ForCES are higher than the multi-path capabilities of I2RS and BGP-LS/PCEP. There are two parameters analysed here that can influence the multi-path capabilities of the TE solutions, the TED accuracy and the granularity, both explained later in this section. In the four technologies the TED accuracy is enhanced thanks to the centralised control plane and the high possibilities to obtain up-to-date TE-related information, which positively impacts the multi-path computation capabilities. However, what posits the difference between the D-CPI technologies in this regard is the granularity, which in the case of OpenFlow and ForCES is higher due to the possibility that these technologies provide to re-define the forwarding plane. As a consequence, the granularity at the forwarding plane of these technologies is higher, meaning that the traffic can be divided into more fine-grained sub-flows. Anyway, as it can be deducted from the literature review, the traffic splitting and the computation of possible backup paths are common techniques in the TE solutions that rely on D-CPI interfaces.

*5) Re-Optimisation:* Another parameter that gets improved by the D-CPI technologies is the re-optimisation, mainly thanks to the utilisation of dedicated elements in charge of the path computation, which is possible in all the D-CPI technologies reviewed in this paper, and the centralisation of the control plane. In the case of ForCES, the framework does not specify how the control plane must be implemented. Therefore, the re-optimisation capabilities of the solution depend on the utilisation of legacy protocols or not. On the one hand, if the solution still uses RSVP-TE for signalling and OSPF or BGP for routing, everything will remain unchanged in this regard. On the other hand, if the CE implements a custom path computation algorithm with re-optimisation support, the performance of the solution would be improved.

Both OpenFlow and I2RS benefit from having a logically centralised control plane. In the case of I2RS, one of the most interesting capabilities regarding TE is that an I2RS Client is able to trigger global concurrent re-optimisation at a specific time on multiple nodes by communicating with the I2RS Agent of each node [162]. Furthermore, the I2RS Client is able to manually re-optimise the MPLS-TE network and send the new constraints including the calculated path to each node via the I2RS Agents [99]. Though, the establishment of the TE LSPs still relies on some legacy protocols like RSVP-TE.

In the case of OpenFlow, being aware of the global state of the network clearly enhances the re-optimisation capabilities, since the controller is able to select which paths to use depending on the network load or other factors [142]. In addition, OpenFlow also includes some mechanisms at the OLS to support flow re-optimisation and re-allocation. The fast failover groups allow to program a list of possible action buckets that are applied in order [196]. With this mechanism, a secondary path can be programmed in the network devices, which is only used in case the primary path fails.

Regarding BGP-LS/PCEP, being BGP-LS a suitable protocol to exchange up-to-date topological information between the elements of a composite PCE, it enhances considerably the re-optimisation capabilities of BGP-LS/PCEP compared to simple PCE. TE solutions based on these technologies support having backup PCEs in charge of path re-optimisations working in background [166].

*6) Network Stability:* Regarding the network stability, both OpenFlow and ForCES provide some mechanisms to minimise perturbation on already existing flows. For instance, in both cases it can be specified in which order the nodes are programmed, which can reduce considerably possible service disruptions. Furthermore, the ability to prioritise some flow entries available in OpenFlow can also be beneficial to increase the network stability [196], since lower priority flow entries can handle the traffic forwarding while the higher priority flows are relocated to alternative paths. I2RS can also provide a higher network stability, since it has been designed to augment the capabilities of the existing mechanisms in MPLS-TE to configure, interrogate and analyse the LSPs. More precisely, I2RS is able to coordinate the configuration of the LSPs to avoid some network devices to be configured out of order [99] . Finally, BGP-LS/PCEP still relies on RSVP-TE for signalling, as it happens in the PCE-based architecture. Therefore, since it inherits the stability of RSVP-TE, the stability is neither improved nor worsened, it remains unchanged.

*7) Accurate TED Synchronisation:* Regarding the TED Accuracy, the logically centralised control plane of I2RS and OpenFlow can be beneficial for TE, since the controller is aware of the entire state of the network, including topological information and already installed services, etc. BGP-LS/PCEP can also improve the TED accuracy, since BGP-LS carries a lot of information related to TE directly obtained from the link state databases of the devices. Since in ForCES the centralisation of the control plane is not a must, but it is a possibility, this parameter can also be improved. However, in a complex architecture such as the one proposed in [197], with two or more of the protocols analysed in this paper coexisting, or in complex network operating systems like ODP or ONOS, special care must be taken when constructing the TED, and possible inconsistencies in the information provided by the different technologies must be considered. All in all, all the D-CPI technologies have the power to enhance the TED accuracy and, therefore, provide more quality TE. This becomes clear in the case of BGP-LS used in conjunction with PCE. The PCE-based architecture lacks the mechanisms to maintain a real-time synchronised TED (at least in stateless PCEs) and BGP-LS can be used to compensate that. Even more, the

PCE-based architecture also lacks the mechanisms to obtain real-time network state information, which BGP-LS solves.

*8) TED Synchronisation Speed:* The architecture of the four D-CPI technologies entails a penalty in the TED synchronisation speed. In the cases of I2RS, ForCES and OpenFlow the delay in the communication between the network devices and the controllers can affect this parameter, an effect that is further worsened if the control channel is congested. Furthermore, the TED synchronisation speed in the case of OpenFlow depends on whether the control is done in-band or out-of-band. On the one hand, with out-of-band control the TED synchronization is faster, because the controller plane can directly obtain the information from the devices. On the contrary, with in-band control the controller is not directly connected to all the networking devices, which introduces additional delay and increases the chances to congest the control channel. In BGP-LS/PCEP the TED synchronisation speed can also be negatively impacted due to the separation between the control and forwarding elements, which can increase the latency that the BGP-LS messages experiment.

*9) Impact on Data Flows:* Impact on data flows can severely degrade the overall performance of a TE solution. For instance, regarding OpenFlow, the performance of current hardware devices has to be taken into account. Due to the flow entries re-ordering that can take place at the OpenFlow switches' Ternary Content-Addressable Memory (TCAM) [193], already installed data flows can be impacted. During the programming time, the previously installed flow entries are not available, resulting in an unavailability time that also worsens the network stability. In general, TCAMs are not immediately programmed, changes are queued and then committed in well specified time slots. However, the modify-state messages used to program the OpenFlow switches are sent at any time, without the controller being aware of the state of the switch regarding that matter. Furthermore, OpenFlow lacks the mechanisms to guarantee that the TCAMs have been correctly programmed, and can assume an incorrect network state. Regarding ForCES, since there are no available products it is hard to state the real impact on data flows. However, as it is closer to the hardware, the programming of the TCAM could be more efficient. Finally, regarding I2RS and BGP-LS/PCEP, legacy equipment and legacy protocols are still used, therefore, there is no additional impact on the data flows compared to the one that already exists as a consequence of using the legacy protocols. The impact on data flows of legacy protocols is analysed in [198], where the route modifications in legacy network devices results in packet loss and communication degradation.

*10) Granularity:* As mentioned before, a high level of granularity can impact the multi-path computation capabilities of a technology, so as the optimality and the re-optimisation capabilities. In a nutshell, a higher granularity implies that the multi-path forwarding or flow disaggregation to optimise the overall link utilisation can be done using a higher range of classifiers. Furthermore, the fine-granularity of the technology also enhances the optimality, as the network utilisation can be optimised by allocating the traffic into different paths easily. On the one hand, ForCES has a great potential granularity because it operates at the data plane and the framework includes a very flexible information model to define the forwarding plane of the network devices. The model is flexible enough to support a wide range of classifiers, i.e., L2 fields, IPv4 addresses, MPLS tags, the combination of these parameters, etc., being considered a very fine-granular technology. Since OpenFlow can be considered a subset of ForCES, it is straightforward to consider that the same header fields used in OpenFlow could be used in ForCES.

In fact, one of the most remarkable features of OpenFlow for TE is its high granularity, as the protocol is able to operate even at bit level. It provides the possibility to use more than 40 matching fields and provides the means to include new matching fields with OXM. In addition to the enhancement of the multi-path computation capabilities, due to the high granularity of the technology, this metric is further enhanced since the multi-path computation can leverage the possibility to direct the packets to groups of a selected type, where an action bucket is applied according to a switch-computed selection algorithm [196]. Still, the same features that make OpenFlow a very suitable technology to improve TE, also impose some challenges. The fine-granularity of OpenFlow results in a higher number of flow entries to keep in the switches, which increases their RAM requirements. In order to solve such a problem, Iyer *et al.* [189] propose to control the number of flow entries installed at the switches and limit the number of OpenFlow counters that are used.

On the other hand and as previously mentioned, I2RS does not operate at the forwarding layer. As a consequence, the granularity of this technology is similar to the one achieved in classic MPLS/IP networks, which results in similar multi-path computation capabilities and optimality. Particularly, the RIB contains routes formed by match conditions and associated actions. The match condition specifies the kind of route and the set of fields to match, which can be the IPv4/6 destination IP address, the outermost MPLS label, the destination MAC address, the incoming interface or IP prefixes.

The same happens with BGP-LS/PCEP. Even if BGP speakers naturally support multiple routes to a destination, the granularity is limited to L3 fields [199]. The BGP protocol is not useful to exchange L2 information or wildcarded information at this layer, which is necessary in services based on the provisioning of L2 circuits like BoD. Hence, the granularity of the flows advertised between the ASs is limited. In addition, routing in BGP is performed taking into consideration just the destination addresses, and not the source addresses. There are multiple use cases that can benefit from using the source information to route the packets. For example, traffic can be balanced to different servers based on the source address. Therefore, when comparing the granularity of BGP-LS/PCEP with the PCE-based architecture no improvement are envisaged in this regard.

*11) Equipment Configurability:* Finally, it is worth mentioning that the equipment configurability is also improved in the cases of OpenFlow and ForCES. Of particular interest is the case of OpenFlow, which allows to perform a limited set of QoS related functions such as rate limiting at the ingress port or sending packets to specific queues thanks to the messages to add, modify or delete meters. However, queues must be

TABLE IX
DETAILED EVALUATION OF THE TE-RELATED PROPOSALS WITH ALTO. THE SYMBOLS INDICATE THAT THE SDN PROTOCOL CAPABILITIES ARE (✓):*better* OR (✗):*worse* COMPARED TO THE PCE-BASED ARCHITECTURE

| | **Optimality** | **Scalability** | **Load sharing** | **Multi-Path comp.** | **Re- opt.** | **Network Stability** | **TED accu.** | **TED Sync speed** | **Impact on data flows** | **Granularity** | **Equipment config.** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Scharf *et al.* [169] | ✓ | ✗ | | | ✓ | | ✓ | ✗ | | | |
| Gurbani *et al.* [174] | | ✗ | | | | | ✓ | ✗ | | | |
| Li *et al.* [171] | ✓ | ✗ | | | | | ✓ | ✗ | | | |
| Zhang *et al.* [172] | ✓ | ✗ | | ✓ | | | ✓ | ✗ | | | |
| Scharf *et al.* [173] | ✓ | ✗ | | | | | ✓ | ✗ | | | |
| Gurbani *et al.* [170] | | ✗ | ✓ | | | | ✓ | ✗ | | | |
| Wang *et al.* [175] | ✗ | ✗ | | | | ✗ | ✓ | ✗ | ✗ | | |
| Faigl *et al.* [112] | ✓ | ✗ | | | | | ✓ | ✗ | | | |

configured as usual, and OpenFlow does not provide the means to do it. On the contrary, neither I2RS nor BGP-LS/PCEP include any additional feature for equipment configuration.

### *C. Contributions of A-CPI Protocols to TE*

As it happens in the case of the D-CPI protocols, the A-CPI protocol analysed in this survey, namely ALTO, is highly beneficial for TE. Table IX summarises the evaluation of the proposals reviewed in this paper in which ALTO is used for advanced TE.

ALTO complements the TE capabilities of the D-CPI by providing additional TE-related information. On the one hand, the TED accuracy gets considerably increased, as ALTO provides information from different sources processed in order to avoid inconsistencies. Though, being the TED held in a separate element, the TED synchronisation time is slightly worse, as it also happens in the other technologies analysed in this document that also use external servers or controllers to store this information. Furthermore, having extra elements and a protocol involved in the network architecture, it also impacts the scalability, and those factors should be taken into account at the time of designing a solution.

Regarding the optimality, the reviewed literature shows that ALTO can be used to maximise network utilisation and to improve other TE-related parameters in a wide range of scenarios, including DC networks [170], WAN [172] or mobile networks [112]. However, the coarse granularity of the technology, limited to IPv4 or IPv6 addresses and prefixes, impacts directly on the multi-path computation capabilities and the re-optimisation [111]. In fact, it limits the level of multi-path computation and re-optimisation that the SDN technology or the PCE would achieve without the constraints imposed by the ALTO limited granularity.

In summary, this technology provides useful TE information that enhances the TED accuracy and the optimality. The only metrics that are worsened by ALTO are the granularity, the scalability and the TED synchronisation speed. The rest of the metrics remain practically unchanged due to the dependence with the control plane technology used in the network.

### *D. Contributions of MI Protocols to TE*

This section discusses how the three management plane technologies analysed in this survey contribute to TE: OVSDB, NETCONF and OF-CONFIG. These technologies are focused on the equipment configurability metric described before.

When talking about TE in SDN, the role of the management plane goes sometimes unnoticed, and it is, nonetheless, of uttermost importance. It has to be taken into account that in order to support and enforce the behaviour stated by the controller plane at the data plane, the devices must be properly configured and managed. For instance, the OpenFlow protocol is able to associate a flow to a certain queue to guarantee some QoS, but it does not provide the means to create queues or configure them, and relies on other MI protocols to perform those tasks [177]. As mentioned in Section III-B, the main difference between the D-CPI and MI protocols is the time-scale at which they operate.

Regarding the OVSDB Management Protocol, it is worth mentioning that it stores information about the physical interfaces of the device, the flow table configuration and monitoring protocols such as NetFlow [200] or sFlow [201]. These monitoring protocols can be really helpful for TE as they collect information about the status of the network. However, the OVSDB holds information about the monitoring configuration and not about the statistics obtained through the monitoring tools. What the OVSDB does hold is information about the queues configured at the OVSs and the configuration of the QoS. As a consequence, it is possible to create, delete or modify queues and QoS configuration through the OVSDB Management Protocol. As previously stated, the D-CPI protocols are not in charge of these kinds of management and configuration aspects, that is why the OVSDB Management Protocol is a very powerful tool that can complement other SDN protocols regarding this matter.

Notwithstanding, OVSDB can only be used with OVSs, which are oriented to virtual environments. In recent years, a huge number of DCs have migrated their systems to OVS enabled hypervisors, which makes the OVSDB Management Protocol a relevant SDN technology. Moreover, this technology could also be applied for the management and configuration of physical devices using the control stack of OVS. Briefly, although OVSDB is a very powerful tool that greatly increases equipment configurability, it cannot be applied to non OVS-based devices.

Meanwhile, NETCONF presents the advantage that being based on the YANG modelling language, it can be applied to all kinds of network devices. This means that unlike OVSDB, it does not depend on a specific type of networking device. However, even that NETCONF provides the means to simplify the equipment configurability, it does not guarantee that the equipment supports QoS or TE. All in all, it can be concluded, NETCONF is a very powerful MI protocol with the power of making a substantial impact on TE. It does not directly operate over the forwarding plane, but it is able to indirectly change the RIB information by altering the information used by the routing protocols for the computation of the routes, as stated in RFC 6241 [127]. NETCONF is able to create or delete interfaces into the running configuration, so as to modify the configuration parameters of the interfaces. In this regard, TE solutions can leverage its ability to create and modify queues attached to the interfaces. Nonetheless, as the reviewed literature points out, NETCONF is broadly used in MPLS networks to configure LSPs and enforce QoS for TE purposes. Furthermore, NETCONF has been proven useful to obtain link state and other resource-related information, which can improve the TED accuracy.

As in the case of NETCONF and OVSDB, the OF-CONFIG protocol is useful for TE because it makes possible the creation and configuration of queues, which are then used to guarantee some QoS constraints. In addition to this, OF-CONFIG has been specifically designed for the management of OpenFlow devices. This means that it makes possible to use the high granularity of OpenFlow for its application to TE. In summary, OF-CONFIG is a technology that greatly increases the equipment configurability. Nevertheless, there is no mechanism available to communicate the OpenFlow controllers with the OCPs. As a result, the data obtained through these two entities may not be synchronised, resulting in a low level of TED Accuracy (which is built from the data collected from both elements). Additionally, the information obtained through the OF-CONFIG protocol may not be enough to optimise network performance. For instance, OF-CONFIG does not inform about the CPU usage of the network device, which can influence its forwarding capability. In order to solve these problems, the EU Seventh Framework Programme (FP7) SPARC project proposed a mechanism to share information between the OpenFlow controllers and the OCPs [202], [203].

From our analysis it can be deducted that it is clear that in order to support TE, some equipment configurability capabilities are required. Even so, control plane technologies mostly lack these capabilities. They are able to minimally configure some features but it is not their purpose. In such a scenario, the management plane technologies are perfect to complement the aforementioned control plane technologies. Among the different technologies analysed, OF-CONFIG appears as the most promising one. On the one hand, OVSDB is a very powerful technology, but it only works with OVS and equipment that follows the OVS schema. On the other hand, OF-CONFIG does not depend so much on the network device, it only requires NETCONF support, and provides all the configurability that D-CPI protocols lack. Regarding this matter, it is clear that OF-CONFIG is the technology that should be used with OpenFlow.

## IX. Future Work and Ongoing Challenges

After reviewing the current proposals dealing with TE solutions in SDN, we outline five research areas of special relevance for TE in SDN.

### A. Design of Integral TE Solutions

As shown in the previous section, the SDN environment includes very different protocols with different purposes and strengths, where some protocols can complement others. There are already some network operating systems that include support for a wide range of SDN protocols operating at different levels, such as OpenFlow, BGP-LS/PCEP or OVSDB. However, most of the solutions dealing with TE in SDN do not leverage this SDN protocol diversity and usually lack the mechanisms necessary to manage the network devices and enforce the required QoS.

Consequently, being aware that the three interface types proposed by the ONF in their SDN architecture proposal, we encourage researchers working on TE in SDN to design integral TE solutions. To provide good TE solutions it is necessary to rely on a flexible and granular D-CPI protocol together with an MI protocol and a powerful A-CPI protocol. The former to enforce the associated QoS at the networking devices and the latter to provide information easy to process by the optimisation algorithms that reside in the application layer.

### B. Impact of Non-TE Applications

According to the ONF, multiple applications should be able to operate over the same network infrastructure, which raises some policy enforcement related concerns. At the moment, most SDN application are not ready to operate in parallel with other SDN applications. For example, lets assume a TE application with a stateful PCE that takes into account previous service requests and the available resources in the network to compute the paths. This TE application should be aware of the resources consumed by other applications, even if those applications do not maintain a detailed inventory of the consumed resources such as a *simple forwarding* application. Such a situation could lead to the TE application considering some resources as available when in reality, they are not. In summary, TE applications should be aware of non-TE applications, and the impact of these applications on a shared network infrastructure.

### C. State Consistency

Another problem related to the high protocol diversity available at the network operating system is the population of the TED by multiple D-CPIs. Researchers working on TE solutions where the TED is built from information provided by multiple D-CPIs should be aware of possible inconsistencies in the TED.

### D. Scalability

One of the main benefits of SDN is the high granularity available at the data plane of technologies such as OpenFlow and ForCES. These protocols provide the means to revolutionise TE, since they support novel and disruptive traffic splitting levels that bring Multi-Commodity Flow algorithms back to the front. This is a blooming research area, where there are not many real contributions yet. According to our point of view, the application of flow aggregation and disaggregation mechanisms that leverage the fine-granularity of OpenFlow are of special relevance, as they will enable the optimisation of the network resource utilisation.

However, researchers working on flow aggregation and disaggregation mechanisms to improve network performance should be aware of the scalability constraints imposed by the hardware devices available in the market. The limited number of flow entries available in many of the hardware devices impose additional requirements for these types of mechanisms, that researchers should take into account.

### *E. Switch Re-Programming*

Another future challenge, tightly coupled to OpenFlow would be to provide solutions for TE dealing with network devices continuous re-programming. As previously stated, one of the main drawbacks in OpenFlow networks is that changes in the flow tables can result in an unavailability time that impacts directly the network stability and the already installed data flows. As a result, future research should be done in techniques that try to improve TCAM programmability in OpenFlow networks.

Additionally, it would also be very interesting to provide TE solutions aware of this problem at some hardware devices, either proposing hardware-independent solutions where this sort of constraints do not affect the performance of the solution or either proposing new TE solutions in which the performance objective would be precisely to minimise the re-programming of the switches. It has to be taken into account the time-scale at which D-CPIs like OpenFlow operate. With these technologies, switch re-programming can occur very often. Therefore, it is of uttermost importance to handle this constraint imposed by current hardware devices appropriately.

### *F. Maintain Basic Control Functionalities at the Data Plane*

Keeping some control functionalities at the data plane is already mentioned in the ONF's SDN architecture [65]. In fact, Tarnaras *et al.* [137] presented a solution that keeps LLDP at the FE like an LFP in a ForCES router. However, there are not many solutions yet in this regard, especially in OpenFlow. We consider that this is a clear future research direction that would greatly benefit TE in SDN.

Currently, most OpenFlow controllers automatically discover the network devices encapsulating LLDP packets into the OpenFlow protocol, which introduces great overhead in the control channel, consumes one of the most limited resources in the OpenFlow devices, the flow entries, and imposes additional computational load in the controller. Keeping the topology discovery at the data plane, without involving the controller, would benefit TE since the congestion in the control plane would be reduced, less flow entries would be consumed and the controller would have more computational resources available to perform complex computational operations for the TE solutions.

## X. Conclusion

The SDN environment clusters very diverse and varying protocols, which are useful for TE in very different ways. The SDN protocols analysed in this manuscript have been classified depending on the interface type where they operate, as stated in the ONF's SDN architecture, where D-CPI, A-CPI and MI protocols are differentiated. Furthermore, the technologies have been evaluated using the parameters proposed in the PCE-based architecture to evaluate the performance of TE solutions.

Among the protocols analysed, the D-CPI protocols appear as the most promising and beneficial ones for TE. This is a consequence of the interface at which they operate, even if they present major differences among them. What first comes to the reader's attention is the difference between OpenFlow and ForCES and the other two, BGP-LS/PCEP and I2RS. The first two protocols propose to support new forwarding models, while the latter two do not. Having the capability of defining new forwarding models enables finer granularity at the data plane, which directly impacts the optimality and the multi-path computation capabilities of the TE solution. In the case of OpenFlow, the granularity that it provides at the forwarding plane is well known, while in ForCES it has not been defined yet.

Furthermore, current SDN frameworks such as Cisco ONE or ODP support various southbound protocols. This allows to operate the network not only using OpenFlow, but in conjunction with other technologies. The best and most complete solutions to improve TE will involve D-CPI, A-CPI and MI protocols working together. Using this approach will allow to leverage the great granularity of the D-CPI protocol, while obtaining better TE information through an A-CPI protocol and enhancing the equipment configurability by means of an MI protocol.

To conclude, the revolution in TE that started with the PCE-based architecture has its continuity guaranteed thanks to the appearance of these novel and disruptive SDN technologies. Furthermore, there is a huge room for research in SDN-based TE, especially regarding the optimisation of the network resource utilisation leveraging the granularity provided by some forwarding models.

## References

[1] K. Greene, *TR10: Software-Defined Networking*, MIT Technol. Rev., Cambridge, MA, USA, Feb. 2009. Accessed on May 25, 2016. [Online]. Available: http://www2.technologyreview.com/article/412194/tr10-software-defined-networking/

[2] *The Top 5 Emerging Technology Trends of 2014*. Accessed on Jun. 24, 2014. [Online]. Available: http://www.avaya.com/usa/perspectives/articles/the-top-5-five-emerging-technology-trends-of-2014

[3] *Cisco's SDN Solutions*. Accessed on Jun. 22, 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html

[4] *HP SDN Solutions*. Accessed on May 25, 2016. [Online]. Available: http://www8.hp.com/us/en/networking/sdn/portfolio.html

[5] *NEC SDN Solutions*. Accessed on May 25, 2016. [Online]. Available: http://www.nec.com/en/global/solutions/sdn/

[6] *Corsa*. Accessed on May 25, 2016. [Online]. Available: http://www.corsa.com/

[7] A. Nygren *et al.*, "OpenFlow switch specification 1.5.0," Open Netw. Found., Tech. Rep. ONF TS-020, Dec. 2014

[8] N. Damouny and J. Harcourt, *OpenFlow-Enabled Hybrid Cloud Services Connect Enterprise and Service Provider Data Centers*, ONF Solution Brief, Nov. 2012. Accessed on May 25, 2016. [Online]. Available: https://www.opennetworking.org/sdn-resources/technical-library#pub

[9] S. Katukam *et al.*, "SDN in the campus environment," ONF Solution Brief, Sep. 2013. Accessed on May 25, 2016. [Online]. Available: https://www.opennetworking.org/sdn-resources/technical-library#pub

[10] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[11] D. Allan *et al.*, "OpenFlow-enabled SDN and network functions virtualization," ONF Solution Brief, Feb. 2014. Accessed on May 25, 2016. [Online]. Available: https://www.opennetworking.org/sdn-resources/technical-library#pub

[12] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of Internet traffic engineering," Internet Requests for Comments, RFC 3272, May 2002.

[13] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," Internet Requests for Comments, RFC 2702, Sep. 1999.

[14] "AT&T research areas," AT&T, Dallas, TX, USA, Tech. Rep., May 2016. Accessed on May 25, 2016. [Online]. Available: http://www.research.att.com/evergreen/what_we_do/research.html

[15] *ESnet*. Accessed on May 25, 2016. [Online]. Available: http://www.es.net/

[16] *ESnet's OSCARS With FloodLight*. Accessed on May 25, 2016. [Online]. Available: https://github.com/hsr/oscars-gui

[17] A. Farrel, J.-P. Vasseur, and J. Ash, "A path computation element (PCE)-based architecture," Internet Requests for Comments, RFC 4655, Aug. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4655.txt

[18] *AutoBAHN*. Accessed on May 25, 2016. [Online]. Available: http://geant3.archive.geant.net/service/autobahn/pages/home.aspx

[19] D. Zhang and D. Ionescu, "QoS performance analysis in deployment of DiffServ-aware MPLS traffic engineering," in *Proc. 8th ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel Distrib. Comput. (SNPD)*, vol. 3. Qingdao, China, 2007, pp. 963–967.

[20] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.

[21] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1955–1980, 4th Quart., 2014.

[22] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[23] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2317–2346, 4th Quart., 2015.

[24] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 623–654, 1st Quart., 2016.

[25] F. A. Lopes, M. Santos, R. Fidalgo, and S. Fernandes, "A software engineering perspective on SDN programmability," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1255–1272, 2nd Quart., 2016.

[26] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 655–685, 1st Quart., 2016.

[27] J. Xie, D. Guo, Z. Hu, T. Qu, and P. Lv, "Control plane of software defined networks: A survey," *Comput. Commun.*, vol. 67, pp. 1–10, Aug. 2015.

[28] P. Bhaumik *et al.*, "Software-defined optical networks (SDONs): A survey," *Photon. Netw. Commun.*, vol. 28, no. 1, pp. 4–18, 2014.

[29] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.

[30] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using OpenFlow: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 493–512, 1st Quart., 2014.

[31] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," *Comput. Netw.*, vol. 71, pp. 1–30, Oct. 2014.

[32] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and OpenFlow: From concept to implementation," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2181–2206, 4th Quart., 2014.

[33] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2015.

[34] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[35] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, 1st ed. Amsterdam, The Netherlands: Elsevier, 2004.

[36] M. Suchara, D. Xu, R. Doverspike, D. Johnson, and J. Rexford, "Network architecture for joint failure recovery and traffic engineering," in *Proc. ACM SIGMETRICS Joint Int. Conf. Meas. Model. Comput. Syst. (SIGMETRICS)*, San Jose, CA, USA, 2011, pp. 97–108.

[37] A. Kassler, L. Skorin-Kapov, O. Dobrijevic, M. Matijasevic, and P. Dely, "Towards QoE-driven multimedia service negotiation and path optimization with software defined networking," in *Proc. 20th Int. Conf. Softw. Telecommun. Comput. Netw. (SoftCOM)*, Split, Croatia, Sep. 2012, pp. 1–5.

[38] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, Hong Kong, Aug. 2013, pp. 3–14.

[39] S. Sen, D. Shue, S. Ihm, and M. J. Freedman, "Scalable, optimal flow routing in datacenters via local link balancing," in *Proc. 9th ACM Conf. Emerg. Netw. Experiments Technol. (CoNEXT)*, Santa Barbara, CA, USA, 2013, pp. 151–162.

[40] Z. Jia and P. Varaiya, "Heuristic methods for delay constrained least cost routing using $\kappa$-shortest-paths," *IEEE Trans. Autom. Control*, vol. 51, no. 4, pp. 707–712, Apr. 2006.

[41] S. Rai, B. Mukherjee, and O. Deshpande, "IP resilience within an autonomous system: Current approaches, challenges, and future directions," *IEEE Commun. Mag.*, vol. 43, no. 10, pp. 142–149, Oct. 2005.

[42] V. Foteinos, K. Tsagkaris, P. Peloso, L. Ciavaglia, and P. Demestichas, "Operator-friendly traffic engineering in IP/MPLS core networks," *IEEE Trans. Netw. Service Manag.*, vol. 11, no. 3, pp. 333–349, Sep. 2014.

[43] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation" in *Proc. NSDI*, vol. 8. San Francisco, CA, USA, 2008, pp. 323–336.

[44] "Quality of experience (QoE) requirements for real-time communication services," ETSI, Sophia Antipolis, France, Tech. Rep. TR 102 643 V1, Nov. 2009.

[45] I. Foster *et al.*, "A distributed resource management architecture that supports advance reservations and co-allocation," in *Proc. 7th Int. Workshop Qual. Service (IWQoS)*, London, U.K., 1999, pp. 27–36.

[46] D. O. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 42–47, Dec. 1999.

[47] J. McQuillan, I. Richer, and E. Rosen, "The new routing algorithm for the ARPANET," *IEEE Trans. Commun.*, vol. 28, no. 5, pp. 711–719, May 1980.

[48] R. Cole, D. Shur, and C. Villamizar, "IP over ATM: A framework document," Internet Requests for Comments, RFC 1932, Apr. 1996.

[49] I. Castineyra, N. Chiappa, and M. Steenstrup, "The Nimrod routing architecture," Internet Requests for Comments, RFC 1992, Aug. 1996.

[50] K. Nichols, S. Blake, F. Baker, and D. L. Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers," Internet Requests for Comments, RFC 2474, Dec. 1998.

[51] C. Hopps, "Analysis of an equal-cost multi-path algorithm," Internet Requests for Comments, RFC Editor, RFC 2992, Nov. 2000.

[52] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," Internet Requests for Comments, RFC 3031, Jan. 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3031.txt

[53] T. Tsuritani, M. Miyazawa, S. Kashihara, and T. Otani, "Optical path computation element interworking with network management system for transparent mesh networks," in *Proc. Nat. Fiber Optic Eng. Conf. (NFOEC)*, San Diego, CA, USA, 2008, pp. 1–10.

[54] K.-C. Leung, V. O. K. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (TCP): Problems, solutions, and challenges," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 522–535, Apr. 2007.

[55] M. Alizadeh *et al.*, "CONGA: Distributed congestion-aware load balancing for datacenters," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 503–514, 2014.

[56] A. Pathak, M. Zhang, Y. C. Hu, R. Mahajan, and D. Maltz, "Latency inflation with MPLS-based traffic engineering," in *Proc. ACM SIGCOMM Conf. Internet Meas. (IMC)*, Berlin, Germany, 2011, pp. 463–472.

[57] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, Apr. 2014.

[58] A. T. Campbell, I. Katzela, K. Miki, and J. Vicente, "Open signaling for ATM, Internet and mobile networks (OPENSIG'98)," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 1, pp. 97–108, 1999.

[59] D. L. Tennenhouse and D. J. Wetherall, "Towards an active network architecture," in *Proc. DARPA Active Netw. Conf. Expo. (DANCE)*, 2002, pp. 2–15.

[60] J. E. van der Merwe and I. M. Leslie, "Switchlets and dynamic virtual ATM networks," in *Proc. 5th IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, San Diego, CA, USA, 1997, pp. 355–368.

[61] M. Caesar *et al.*, "Design and implementation of a routing control platform," in *Proc. 2nd Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, MA, USA, 2005, pp. 15–28.

[62] J. Rexford *et al.*, "Network-wide decision making: Toward a wafer-thin control plane," in *Proc. 3rd Workshop Hot Topics Netw. (HotNets)*, College Park, MD, USA, 2004, pp. 59–64.

[63] *Open Networking Foundation*. Accessed on May 25, 2016. [Online]. Available: https://www.opennetworking.org/

[64] O. M. E. Committee *et al.*, "Software-defined networking: The new norm for networks," ONF White Paper, Open Netw. Found., Palo Alto, CA, USA, 2012.

[65] O. M. E. Committee *et al.*, "Software-defined networking: Arquitecture overview," White Paper, Open Netw. Found., Palo Alto, CA, USA, 2013.

[66] *ForCES Working Group*. Accessed on May 25, 2016. [Online]. Available: http://datatracker.ietf.org/wg/forces/charter/

[67] E. Haleplidis, S. Denazis, O. Koufopavlou, J. Halpern, and J. H. Salim, "Software-defined networking: Experimenting with the control to forwarding plane interface," in *Proc. Eur. Workshop Softw. Defined Netw. (EWSDN)*, Darmstadt, Germany, 2012, pp. 91–96.

[68] L. Yang, R. Dantu, T. Anderson, and R. Gopal, "Forwarding and control element separation (ForCES) framework," Internet Requests for Comments, RFC 3746, Apr. 2004.

[69] A. Crouch, H. Khosravi, A. Doria, X. Wang, and K. Ogawa, "Forwarding and control element separation (ForCES) applicability statement," Internet Requests for Comments, RFC 6041, Oct. 2010.

[70] E. Haleplidis, K. Ogawa, W. Wang, and J. H. Salim, "Implementation report for forwarding and control element separation (ForCES)," Internet Requests for Comments, RFC 6053, Nov. 2010.

[71] J. Halpern and J. H. Salim, "Forwarding and control element separation (ForCES) forwarding element model," Internet Requests for Comments, RFC 5812, Mar. 2010.

[72] E. Haleplidis *et al.*, "Network programmability with ForCES," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1423–1440, 3rd Quart., 2015.

[73] *NOX*. Accessed on May 25, 2016. [Online]. Available: http://www.noxrepo.org/nox/about-nox/

[74] *POX*. Accessed on May 25, 2016. [Online]. Available: http://www.noxrepo.org/pox/about-pox/

[75] *Trema*. Accessed on May 25, 2016. [Online]. Available: http://trema.github.io/trema/

[76] *Ryu*. Accessed on May 25, 2016. [Online]. Available: http://osrg.github.io/ryu/

[77] *FloodLight*. Accessed on May 25, 2016. [Online]. Available: http://www.projectfloodlight.org/floodlight/

[78] *Beacon*. Accessed on May 25, 2016. [Online]. Available: https://openflow.stanford.edu/display/Beacon/Home

[79] Z. Cai, A. L. Cox, and T. E. Ng, "Maestro: A system for scalable OpenFlow control," Dept. Comput. Sci., Rice Univ., St. Houston, TX, USA, Tech. Rep. TR10-08, 2010.

[80] A. Voellmy, B. Ford, P. Hudak, and Y. R. Yang, "Scaling software-defined network controllers on multicore servers," Dept. Comput. Sci., Yale Univ., New Haven, CT, USA, Tech. Rep. YALEU/DCS/TR-1468, 2012.

[81] *Jaxon*. Accessed on May 25, 2016. [Online]. Available: http://jaxon.onuos.org/

[82] *SNAC*. Accessed on May 25, 2016. [Online]. Available: http://www.openflowhub.org/display/Snac/SNAC+Home

[83] T. Koponen *et al.*, "Onix: A distributed control platform for large-scale production networks," in *Proc. 9th USENIX Conf. Oper. Syst. Design Implement. (OSDI)*, Vancouver, BC, Canada, 2010, pp. 351–364.

[84] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proc. Internet Netw. Manag. Workshop/Workshop Res. Enterprise Netw. (INM/WREN)*, San Jose, CA, USA, 2010, p. 3.

[85] *Helios*. Accessed on May 25, 2016. [Online]. Available: http://www.nec.com/

[86] R. Sherwood *et al.* (2009). *FlowVisor: A Network Virtualization Layer*. Accessed on May 25, 2016. [Online]. Available: http://OpenFlowSwitch.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf

[87] *OpenVirteX*. Accessed on Dec. 7, 2016. [Online]. Available: http://ovx.onlab.us/

[88] Z. Bozakov and P. Papadimitriou, "Autoslice: Automated and scalable slicing for software-defined networks," in *Proc. ACM Conf. CoNEXT Student Workshop*, Nice, France, 2012, pp. 3–4.

[89] M. F. Bari *et al.*, "Data center network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 909–928, 2nd Quart., 2013.

[90] P. Rygielski and S. Kounev, "Network virtualization for QoS-aware resource management in cloud data centers: A survey," *PIK-Praxis der Informationsverarbeitung und Kommunikation*, vol. 36, no. 1, pp. 55–64, 2013.

[91] *Open Network Operating System*. Accessed on May 25, 2016. [Online]. Available: http://www.onosproject.org/

[92] *Open Daylight Project*. Accessed on May 25, 2016. [Online]. Available: http://www.opendaylight.org/

[93] *Cisco ONE*. Accessed on May 25, 2016. [Online]. Available: http://www.cisco.com/web/solutions/trends/open_network_environment/indepth.html

[94] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky, "Advanced study of SDN/OpenFlow controllers," in *Proc. 9th Central Eastern Eur. Softw. Eng. Conf. Russia (CEE-SECR)*, 2013, Art. no. 1.

[95] *I2RS Working Group*. Accessed on May 25, 2016. [Online]. Available: http://datatracker.ietf.org/wg/i2rs/charter/

[96] S. Hares and A. Dass, "I2RS protocol strawman," Internet-Draft draft-hares-i2rs-protocol-strawman-02, Internet Eng. Task Force, Fremont, CA, USA, May 2016.

[97] A. Atlas, J. Halpern, S. Hares, D. Ward, and T. Nadeau, "An architecture for the interface to the routing system," Internet Requests for Comments, RFC 7921, Jun. 2016.

[98] A. Atlas, T. Nadeau, and D. Ward, "Problem statement for the interface to the routing system," Internet Requests for Comments, RFC 7920, Jun. 2016.

[99] T. Huang and Z. Li, "Use cases for an interface to mpls te," Working Draft, IETF Secretariat, Fremont, CA, USA, Internet-Draft draft-huang-i2rsmpls-te-usecases-00, Oct. 2013. [Online]. Available: http://www.ietf.org/internet-drafts/draft-huang-i2rs-mpls-te-usecases-00.txt

[100] N. Bahadur, R. Folkes, S. Kini, and J. Medved, "Routing information base info model," Working Draft, IETF Secretariat, Fremont, CA, USA, Internet-Draft draft-nitinb-i2rs-ribinfo-model-02, Aug. 2013. [Online]. Available: http://www.ietf.org/internet-drafts/ draft-nitinb-i2rs-rib-info-model-02.txt

[101] M. Bjorklund, "Yang—A data modeling language for the network configuration protocol (NETCONF)," Internet Requests for Comments, RFC 6020, Oct. 2010.

[102] *IDR Working Group*. Accessed on May 25, 2016. [Online]. Available: http://datatracker.ietf.org/wg/idr/charter/

[103] H. Gredler, J. Medved, S. Previdi, A. Farrel, and S. Ray, "North-bound distribution of link-state and traffic engineering (TE) information using BGP," Internet Requests for Comments, RFC 7752, Mar. 2016.

[104] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (BGP-4)," Internet Requests for Comments, RFC 4271, Jan. 2006.

[105] *Cisco ONE*. Accessed on May 25, 2016. [Online]. Available: http://www.cisco.com/web/solutions/trends/open_network_environment/indepth.html

[106] J. Ash and J. L. Le Roux, "Path computation element (PCE) communication protocol generic requirements," Internet Requests for Comments, RFC 4657, Sep. 2006.

[107] D. Awduche *et al.*, "RSVP-TE: Extensions to RSVP for LSP tunnels," Internet Requests for Comments, RFC 3209, Dec. 2001.

[108] *ALTO Working Group*. Accessed on May 25, 2016. [Online]. Available: http://datatracker.ietf.org/wg/alto/charter/

[109] R. Alimi *et al.*, "Application-layer traffic optimization (ALTO) protocol," Internet Requests for Comments, RFC 7285, Sep. 2014.

[110] J. Seedorf and E. Burger, "Application-layer traffic optimization (ALTO) problem statement," Internet Requests for Comments, RFC 5693, Oct. 2009.

[111] R. Alimi, R. Penno, and Y. Yang, "Alto protocol," Working Draft, IETF Secretariat, Fremont, CA, USA, Internet-Draft draft-ietfalto- protocol-27, Mar. 2014. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-alto-protocol-27.txt

[112] Z. Faigl, Z. Szabó, and R. Schulcz, "Application-layer traffic optimization in software-defined mobile networks: A proof-of-concept implementation," in *Proc. 16th Int. Telecommun. Netw. Strategy Plan. Symp. (Networks)*, Funchal, Portugal, Sep. 2014, pp. 1–6.

[113] B. Pfaff and B. Davie, "The open vSwitch database management protocol," Internet Requests for Comments, RFC 7047, Dec. 2013.

[114] P. Gorja and R. Kurapati, "Extending open vSwitch to L4-L7 service aware OpenFlow switch," in *Proc. IEEE Int. Adv. Comput. Conf. (IACC)*, Gurgaon, India, 2014, pp. 343–347.

[115] J. Corbet. *Routing Open vSwitch Into the Mainline*. Accessed on May 25, 2016. [Online]. Available: http://lwn.net/Articles/469775/

[116] J. Pettit, J. Gross, B. Pfaff, M. Casado, and S. Crosby, "Virtual switching in an era of advanced edges," in *Proc. 2nd Workshop Data Center Converged Virtual Ethernet Switching (DC-CAVES)*, Amsterdam, The Netherlands, 2010, pp. 1–7.

[117] *Pica8*. Accessed on May 25, 2016. [Online]. Available: http://www.pica8.com/

[118] B. Pfaff *et al.*, "Extending networking into the virtualization layer," in *Proc. Workshop Hot Topics Netw. (HotNets)*, New York, NY, USA, 2009, pp. 1–6.

[119] *Open vSwitch Database Schema*, Nicira Netw., Palo Alto, CA, USA, accessed on May 19, 2014. [Online]. Available: http://openvswitch.org/ovs-vswitchd.conf.db(5)

[120] *Open vSwitch*, Nicira Netw., accessed on May 25, 2016. [Online]. Available: http://openvswitch.org/

[121] *JSON-RPC Specification Version 1.0*. Accessed on May 25, 2016. [Online]. Available: http://json-rpc.org/wiki/specification

[122] R. Cafini, W. Cerroni, C. Raffaelli, and M. Savi, "Standard-based approach to programmable hybrid networks," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 148–155, May 2011.

[123] Cisco. *Cisco Networking Services Configuration Guide, Network Configuration Protocol*. Accessed on Dec. 7, 2016. [Online]. Available: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cns/ configuration/ xe-16/cns-xe-16-book/cns-netconf.html

[124] J. Networks. (2014). *NETCONF XML Management Protocol Developer Guide*. Accessed on Dec. 7, 2016. [Online]. Available: http://www.juniper.net/documentation/en US/junos15.1/information-products/pathway-pages/netconf-guide/netconf.html

[125] NEC. (2014). *Network Management Core Technologies*. Accessed on Dec. 7, 2016. [Online]. Available: http://uk.nec.com/en GB/ emea/about/neclab eu/projects/gmm.html

[126] *Network Configuration Working Group*. Accessed on May 25, 2016. [Online]. Available: http://datatracker.ietf.org/wg/netconf/charter/

[127] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network configuration protocol (NETCONF)," Internet Requests for Comments, RFC 6241, Jun. 2011.

[128] M. Scott and M. Bjorklund, "YANG module for NETCONF monitoring," Internet Requests for Comments, RFC 6022, Oct. 2010.

[129] S. Kim, "OpenFlow management and configuration protocol 1.2 (OF-Config 1.2)," Open Netw. Found., Tech. Rep. ONF TS-016, 2014.

[130] S. Kim, "OpenFlow notifications framework 1.0," Open Netw. Found., Tech. Rep. ONF TS-014, Oct. 2013.

[131] Z. Bin *et al.*, "Resource scheduling algorithm and ecnomic model in ForCES networks," *China Commun.*, vol. 11, no. 3, pp. 91–103, Mar. 2014.

[132] D.-D. Wang, L.-G. Dong, F.-R. Zhou, and W.-M. Wang, "Study of OSPF routing optimization among forwarding elements in the ForCES router," *Inf. Technol. J.*, vol. 12, no. 2, pp. 351–356, 2013.

[133] X. Jiang, M. Xu, and Q. Li, "Compact route computation: Improving parallel BGP route processing for scalable routers," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. Workshops Phd Forum (IPDPSW)*, Anchorage, AK, USA, 2011, pp. 1496–1501.

[134] J. Chen, W. Wang, and B. Zhuge, "Bandwidth allocation mechanism of ForCES transport mapping layer based on TCP/IP," in *Proc. IFIP Int. Conf. Netw. Parallel Comput. Workshops (NPC)*, Dalian, China, 2007, pp. 819–823.

[135] C. Li, S. Zhang, and W. Wang, "Scheduling model and performance analysis in transport mapping layer of control element in forwarding and control element separation system," *Int. J. Commun. Syst.*, vol. 26, no. 3, pp. 395–411, 2013.

[136] T. Luo, S. Yu, and X. Wang, "Analyzing and designing of reliable multicast based on FEC in distributed switch," in *Proc. 8th SPIE Asia Pac. Opt. Commun. (APOC)*, Hangzhou, China, 2008, Art. no. 713720.

[137] G. Tarnaras, E. Haleplidis, and S. Denazis, "SDN and ForCES based optimal network topology discovery," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., 2015, pp. 1–6.

[138] D. Zhong, W. Wang, and C. Li, "The implementation of multiple virtual FEs and their backup in the ForCES router," in *Proc. IEEE 2nd Int. Workshop Intell. Syst. Appl. (ISA)*, 2010, pp. 1–4.

[139] K.-T. Jeong, C.-H. Lee, and Y.-T. Kim, "Performance analysis of multi-layered protection switching with control-forwarding separation," in *Proc. IEEE 14th Asia–Pac. Netw. Oper. Manag. Symp. (APNOMS)*, Seoul, South Korea, 2012, pp. 1–4.

[140] S.-H. Yoon, D. Siradjev, and Y.-T. Kim, *Management of DiffServover-MPLS Transit Networks with BFD/OAM in ForCES Architecture*, Heidelberg, Germany: Springer, 2006, pp. 136–148.

[141] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, 2000.

[142] C.-Y. Hong *et al.*, "Achieving high utilization with software-driven WAN," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, vol. 43. Hong Kong, 2013, pp. 15–26.

[143] R. van der Pol *et al.*, "Multipathing with MPTCP and OpenFlow," in *Proc. High Perform. Comput. Netw. Storage Anal. (SCC)*, Salt Lake City, UT, USA, Nov. 2012, pp. 1617–1624.

[144] A. Mendiola *et al.*, "DynPaC: A path computation framework for SDN," in *Proc. IEEE 4th Eur. Workshop Softw. Defined Netw. (EWSDN)*, Bilbao, Spain, 2015, pp. 119–120.

[145] *OLiMPS's*. Accessed on Dec. 7, 2016. [Online]. Available: https://indico.cern.ch/event/212656/contributions/1508160/ attachments/ 335618/468303/presentation lhcone.pdf

[146] S. Das, A. R. Sharafat, G. Parulkar, and N. McKeown, "MPLS with a simple OPEN control plane," in *Proc. Opt. Fiber Commun. Conf. (OFC)*, Los Angeles, CA, USA, 2011, pp. 1–3.

[147] S. Das *et al.*, "Application-aware aggregation and traffic engineering in a converged packet-circuit network," in *Proc. Nat. Fiber Opt. Eng. Conf. (NFOEC)*, Los Angeles, CA, USA, 2011, pp. 1–3.

[148] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in *Proc. 32nd IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Turin, Italy, 2013, pp. 2211–2219.

[149] M. Koerner and O. Kao, "Multiple service load-balancing with OpenFlow," in *Proc. 13th IEEE Int. Conf. High Perform. Switching Routing (HPSR)*, Belgrade, Serbia, 2012, pp. 210–214.

[150] M. Gharbaoui *et al.*, "On virtualization-aware traffic engineering in OpenFlow data centers networks," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Kraków, Poland, 2014, pp. 1–8.

[151] G. Wang, T. S. E. Ng, and A. Shaikh, "Programming your network at run-time for big data applications," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, Helsinki, Finland, 2012, pp. 103–108.

[152] C. Huang, C. Nakasan, K. Ichikawa, and H. Iida, "A multipath controller for accelerating GridFTP transfer over SDN," in *Proc. 11th IEEE Int. Conf. e-Sci. (e-Science)*, Munich, Germany, 2015, pp. 439–447.

[153] W. Braun and M. Menth, "Load-dependent flow splitting for traffic engineering in resilient OpenFlow networks," in *Proc. Int. Conf. Workshops Netw. Syst. (NetSys)*, Cottbus, Germany, 2015, pp. 1–5.

[154] S. Li *et al.*, "Flexible traffic engineering: When OpenFlow meets multi-protocol IP-forwarding," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1699–1702, Oct. 2014.

[155] F. P. Tso and D. P. Pezaros, "Baatdaat: Measurement-based flow scheduling for cloud data centers," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Split, Croatia, Jul. 2013, pp. 765–770.

[156] R. Trestian, G.-M. Muntean, and K. Katrinis, "MiceTrap: Scalable traffic engineering of datacenter mice flows using OpenFlow," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, Ghent, Belgium, 2013, pp. 904–907.

[157] K. Phemius and M. Bouet, "Implementing OpenFlow-based resilient network services," in *Proc. 1st IEEE Int. Conf. Cloud Netw. (CLOUDNET)*, Paris, France, 2012, pp. 212–214.

[158] K. Nguyen, Q. T. Minh, and S. Yamada, "A software-defined networking approach for disaster-resilient WANs," in *Proc. IEEE 22nd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Nassau, The Bahamas, 2013, pp. 1–5.

[159] P. S. Pisa *et al.*, *Openflow and Xen-Based Virtual Network Migration*. Heidelberg, Germany: Springer, 2010, pp. 170–181.

[160] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, no. 2, pp. 248–264, 1972.

[161] M. Boddie *et al.*, "On extending ESnet's OSCARS with a multi-domain anycast service," in *Proc. IEEE 16th Int. Conf. Opt. Netw. Design Model. (ONDM)*, Colchester, U.K., 2012, pp. 1–6.

[162] A. Sgambellur, F. Paolucci, F. Cugini, L. Valcarenghi, and P. Castoldi, "Generalized SDN control for access/metro/core integration in the framework of the interface to the routing system (I2RS)," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Atlanta, GA, USA, 2013, pp. 1216–1220.

[163] R. Martínez *et al.*, "Experimental validation of active frontend–backend stateful PCE operations in flexgrid optical network re-optimization," in *Proc. IEEE Eur. Conf. Opt. Commun. (ECOC)*, Cannes, France, 2014, pp. 1–3.

[164] A. Giorgetti, F. Paolucci, F. Cugini, and P. Castoldi, "Proactive hierarchical PCE based on BGP-LS for elastic optical networks," in *Proc. Opt. Fiber Commun. Conf. (OFC)*, Los Angeles, CA, USA, 2015, pp. 1–3.

[165] O. G. de Dios *et al.*, "First multi-partner demonstration of BGP-LS enabled inter-domain EON control with H-PCE," in *Proc. Opt. Fiber Commun. Conf. (OFC)*, Los Angeles, CA, USA, 2015, pp. 1–3.

[166] M. Cuaresma *et al.*, "Experimental demonstration of H-PCE with BPG-LS in elastic optical networks," in *Proc. 39th Eur. Conf. Exhibit. Opt. Commun. (ECOC)*, London, U.K., 2013, pp. 1–3.

[167] R. Casellas *et al.*, "SDN orchestration of OpenFlow and GMPLS flexi-grid networks with a stateful hierarchical PCE [Invited]," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 7, no. 1, pp. A106–A117, Jan. 2015.

[168] R. Casellas *et al.*, "IDEALIST control plane architecture for multi-domain flexi-grid optical networks," in *Proc. IEEE Eur. Conf. Netw. Commun. (EuCNC)*, Bologna, Italy, 2014, pp. 1–5.

[169] M. Scharf *et al.*, "Monitoring and abstraction for networked clouds," in *Proc. IEEE 16th Int. Conf. Intell. Next Gener. Netw. (ICIN)*, Berlin, Germany, 2012, pp. 80–85.

[170] V. K. Gurbani, M. Scharf, T. V. Lakshman, V. Hilt, and E. Marocco, "Abstracting network state in software defined networks (SDN) for rendezvous services," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, 2012, pp. 6627–6632.

[171] S. Li, I. Doh, and K. Chae, "Key management mechanism in ALTO/SDN based CDNi architecture," in *Proc. IEEE Int. Conf. Inf. Netw. (ICOIN)*, Siem Reap, Cambodia, 2015, pp. 110–115.

[172] W. Zhang, W. Lei, S. Liu, Y. Guan, and G. Li, "Multipath transport based on application-level relay service and traffic optimization," in *Proc. Int. Conf. Inf. Commun. Technol. (ICT)*, May 2014, pp. 1–8.

[173] M. Scharf, T. Voith, M. Stein, and V. Hilt, "ATLAS: Accurate topology level-of-detail abstraction system," in *Proc. IEEE Netw. Oper. Manag. Symp. (NOMS)*, Kraków, Poland, 2014, pp. 1–5.

[174] V. K. Gurbani, D. Goergen, R. State, and T. Engel, "Making historical connections: Building application layer traffic optimization (ALTO) network and cost maps from public broadband data," in *Proc. IEEE 10th Int. Conf. Netw. Service Manag. (CNSM)*, Rio de Janeiro, Brazil, 2014, pp. 193–198.

[175] C. Wang, N. Wang, M. Howarth, and G. Pavlou, "An empirical study on the interactions between ALTO-assisted P2P overlays and ISP networks," in *Proc. IEEE 36th Conf. Local Comput. Netw. (LCN)*, Bonn, Germany, 2011, pp. 719–726.

[176] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang, "Cooperative content distribution and traffic engineering in an ISP network," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 1, pp. 239–250, 2009.

[177] D. Palma *et al.*, "The QueuePusher: Enabling queue management in OpenFlow," in *Proc. 3rd Eur. Workshop Softw. Defined Netw. (EWSDN)*, Budapest, Hungary, Sep. 2014, pp. 125–126.

[178] S. Sharma *et al.*, "Implementing quality of service for the software defined networking enabled future Internet," in *Proc. IEEE 3rd Eur. Workshop Softw. Defined Netw. (EWSDN)*, Budapest, Hungary, 2014, pp. 49–54.

[179] C. Caba and J. Soler, *SDN-Based QoS Aware Network Service Provisioning*. Cham, Switzerland: Springer, 2015, pp. 119–133.

[180] R. C. Pereira and L. Z. Granville, "On the performance of COPS-PR and NETCONF in an integrated management environment for DiffServ-enabled networks," in *Proc. IEEE Int. Conf. Telecommun. (ICT)*, 2008, pp. 1–6.

[181] B. Martini, F. Baroncelli, V. Martini, K. Torkman, and P. Castoldi, "ITU-T RACF implementation for application-driven QoS control in MPLS networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, 2009, pp. 422–429.

[182] J. Oliveira *et al.*, "Experimental testbed of reconfigurable flexgrid optical network with virtualized GMPLS control plane and autonomic controls towards SDN," in *Proc. IEEE SBMO/IEEE MTT-S Int. Microw. Optoelectron. Conf. (IMOC)*, Rio de Janeiro, Brazil, 2013, pp. 1–5.

[183] M. Aoki and S. Urushidani, "Flow analysis system for multi-layer service networks," in *Proc. IEEE 9th Asia–Pac. Symp. Inf. Telecommun. Technol. (APSITT)*, 2012, pp. 1–6.

[184] D. Loureiro, P. Gonçalves, and A. Nogueira, "NETCONF agent for link state monitoring," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, 2012, pp. 6565–6569.

[185] A. Malishevskiy *et al.*, "OpenFlow-based network management with visualization of managed elements," in *Proc. 3rd GENI Res. Educ. Experiments Workshop (GREE)*, Atlanta, GA, USA, 2014, pp. 73–74.

[186] W. Wendong, Q. Qinglei, G. Xiangyang, H. Yannan, and Q. Xirong, "Autonomic QoS management mechanism in software defined network," *China Commun.*, vol. 11, no. 7, pp. 13–23, Jul. 2014.

[187] P. Aukia *et al.*, "RATES: A server for MPLS traffic engineering," *IEEE Netw.*, vol. 14, no. 2, pp. 34–41, Mar./Apr. 2000.

[188] *NGN Working Definition*. Accessed on May 25, 2016. [Online]. Available: http://www.itu.int/ITU-T/studygroups/com13/ngn2004/working_definition.html

[189] A. S. Iyer, V. Mann, and N. R. Samineni, "SwitchReduce: Reducing switch state and controller involvement in OpenFlow networks," in *Proc. IFIP Netw. Conf.*, Brooklyn, NY, USA, 2013, pp. 1–9.

[190] R. Durner, A. Blenk, and W. Kellerer, "Performance study of dynamic QoS management for OpenFlow-enabled SDN switches," in *Proc. IEEE/ACM Int. Symp. Qual. Service (IWQoS)*, Portland, OR, USA, 2015, pp. 177–182.

[191] A. Lazaris *et al.*, "Tango: Simplifying SDN control with automatic switch property inference, abstraction, and optimization," in *Proc. 10th ACM Int. Conf. Emerg. Netw. Experiments Technol. (CoNEXT)*, Sydney, NSW, Australia, 2014, pp. 199–212.

[192] M. Kuźniar, P. Perešini, and D. Kostić, "What you need to know about SDN control and data planes," EPFL, Lausanne, Switzerland, Tech. Rep. EPFL-REPORT-199497, 2014.

[193] C. Rotsos, N. Sarrar, S. Uhlig, R. Sherwood, and A. W. Moore, "OFLOPS: An open framework for OpenFlow switch evaluation," in *Proc. 13th Int. Conf. Passive Act. Meas. (PAM)*, Vienna, Austria, 2012, pp. 85–95.

[194] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, Helsinki, Finland, 2012, pp. 7–12.

[195] Y. Rekhter and E. Rosen, "Carrying label information in BGP-4," Internet Requests for Comments, RFC 3107, May 2001.

[196] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[197] D. King and A. Farrel, "A PCE-based architecture for application-based network operations," Internet Requests for Comments, RFC 7491, Mar. 2015.

[198] G. Iannaccone, C.-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an IP backbone," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas. (IMW)*, Marseilles, France, 2002, pp. 237–242.

[199] D. Meyer and K. Patel, "BGP-4 protocol analysis," Internet Requests for Comments, RFC 4274, Jan. 2006.

[200] B. Claise, "Cisco systems netflow services export version 9," Internet Requests for Comments, RFC Editor, RFC 3954, Oct. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3954.txt

[201] *sFlow*. Accessed on May 25, 2016. [Online]. Available: http://www.sflow.org/

[202] A. Devlic, W. John, and P. Sköldström, "A use-case based analysis of network management functions in the ONF SDN model," in *Proc. IEEE Eur. Workshop Softw. Defined Netw. (EWSDN)*, Darmstadt, Germany, 2012, pp. 85–90.

[203] *EU FP7 Sparc Project*. Accessed on May 25, 2016. [Online]. Available: http://www.fp7-sparc.eu/

**Alaitz Mendiola** received the B.Sc. and M.Sc. degrees in telecommunication engineering, and the M.Sc. degree in information and communication systems and in wireless networks from the University of the Basque Country, in 2012 and 2014, respectively. She joined the I2T Research Laboratory in 2010 and has participated in several OpenFlow/SDN related projects. Her research interests include software-defined networking, network virtualization, and DOCSIS access networks. She has been involved in GN3✓, GN4-1, and DynPaC projects.

**Jasone Astorga** received the B.Sc. and M.Sc. degrees in telecommunication engineering and the Ph.D. degree from the University of the Basque Country (UPV/EHU), in 2004 and 2013, respectively. From 2004 to 2007, she was with Nextel S.A., a telecommunications enterprize. From 2007, she was a Lecturer with the Communications Engineering Department, UPV/EHU, and a Researcher in the I2T (Engineering and Research on Telematics, http://i2t.ehu.es) Research Laboratory. Her research interests include software-defined networking, network virtualization, IP-enabled wireless sensors, security in distributed environments, and mobility management. She has been involved in GN3, GN4-1, and DynPaC projects.

**Eduardo Jacob** was a Network Manager first and a Research and Development Project Leader later, in private business. Then he joined the University of the Basque Country, where he leads a Research Group that is participating in several national and European research and development projects. He is also Member of the Advisory Council Member of the Basque Data Protection Agency, where he occupies the ICT Expert Chair. His research interests are security in distributed systems, next generation networks, industrial applications of SDN and NFV for resilience, experimental network infrastructures, and cyber physical systems.

**Marivi Higuero** received the B.S. and M.S. degrees in electrical engineering, and the Ph.D. degree from the University of the Basque Country (UPV/EHU), in 1992 and 2005, respectively. She was with Sarenet, an Internet Service Provider, as a member of the technical department in this company. She is currently an Assistant Professor with the Communications Engineering Department, UPV/EHU, where she is also a member of the I2T Research Laboratory. Her research interests include computer networks and services, sensor environments, mobility, and security.