The Need for a Control Orchestration Protocol in Research Projects on Optical Networking

R. Vilalta*, V. López†, A. Mayoral*, N. Yoshikane‡, M. Ruffini ‖, D. Siracusa **, R. Martínez*, T. Szyrkowiec§, A. Autenrieth§, S. Peng¶, R. Casellas*, R. Nejabati¶, D. Simeonidou¶, X. Cao‡, T. Tsuritani‡, I. Morita‡, J. P. Fernández-Palacios†, and R. Muñoz*

*CTTC, Castelldefels, Spain
†Telefónica I+D, Madrid, Spain
‡KDDI R&D Laboratories Inc. Fujimino, Japan
§ADVA Optical Networking, Munich, Germany
¶University of Bristol, Bristol, U.K.
‖Trinity College Dublin, Dublin, Ireland
**CREATE-NET, Trento, Italy

***Abstract*—The Control Orchestration Procotol (COP) abstracts a common set of control plane functions used by an various SDN controllers, allowing the interworking of heterogenous control plane paradigms (i.e., OpenFlow, GMPLS/PCE). COP has been defined using YANG model language and can be transported using RESTconf, which is being incorporated by industry.**

COP has been defined in the scope of STRAUSS due to the need for an overarching control plane protocol for network orchestration. In this paper, several research projects describe how the COP could fit in their architecture and propose a use case for COP usage. The proposed COP use cases cover the following research projects: STRAUSS, IDEALIST, DISCUS, COMBO, INSPACE.

***Index Terms*—Control Plane, Network Orchestration, YANG, RESTCONF, use cases, research projects.**

I. Introduction

Software Defined Networking (SDN) is defined as a control framework that supports the programmability of network functions and protocols by decoupling the data plane and the control plane, which are currently integrated vertically in most network equipment. The SDN and virtualization technologies allow network operators to manipulate the logical map of the network and create multiple co-existing network slices (virtual networks) independent of the underlying transport networks. Furthermore, the separation of control plane and data plane makes SDN a suitable candidate for end-to-end network service orchestration across multiple domains with heterogeneous and incompatible control plane and transport technologies. Introducing SDN to the optical layer can potentially facilitate application specific network slicing at the optical layer, as well as coordination and orchestration of higher network layers. It can also provide a unified control plane platform for integration of packet and circuit switched networks for access, metro and core network segments.

The need of offering end-to-end Ethernet service provisioning and orchestration across multiple domains with heterogeneous transport and control plane technologies is clear. Central to this capability is the SDN based service and network orchestration layer as depicted in Fig. 1. The proposed Control Orchestration Procotol (COP) abstracts a set of control plane functions used by an SDN controller, allowing the interworking of heterogenous control plane paradigms (i.e., OpenFlow, GMPLS/PCE).

The COP a) is aware of the existing background in network programmability and b) applies new Software Defined Networking principles to enable cost reduction, innovation and reduced time to market of new services, while covering multi-domain and multi-technology path/packet networks.

This COP provides two main functionalities. The first functionality is a network-wide, centralized orchestration. This high level, logically centralized entity exists on top of and across the different network domains and is able to drive the provisioning (and recovery) of connectivity across heterogeneous networks, dynamically and in real time. The second COP functionality is the introduction of a new interface and protocol that abstracts the particular control plane technology of a given domain. In this sense, the proposed architecture applies the same abstraction and generalization principles that OpenFlow/SDN have applied to data networks.

The SDN orchestrator works under the assumption that each domain is composed of a data plane controlled by an instance of a given control plane technology, but transport and/or control plane technologies for each domain can be different. The main functionalities of the SDN orchestrator are abstract and not technology related. This control plane abstraction must enable the provisioning of data services using the underlying configuration technology.

COP has been defined in the scope of STRAUSS due to the need for an overarching control plane protocol for network orchestration. In this paper, we first present the requirements for a COP, we propose an specification and finally, several EU funded research projects describe how the COP could fit in their architecture and propose a use case for COP usage. The proposed COP use cases cover the following research projects: STRAUSS, IDEALIST, DISCUS, COMBO, INSPACE.

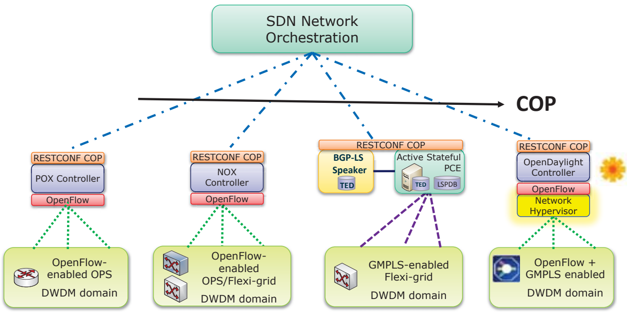

Fig. 1: STRAUSS use case

II. The need for a Control Orchestration Protocol

The design of COP between the orchestration and control layers allows the simplification and optimization, in terms of scalability and compatibility between the different modules which compose the SDN architecture. COP unifies all the orchestration functionalities into a single protocol paradigm.

The main reason for the need of this protocol is the heterogeneousity of NorthBound Interfaces (NBI) of SDN controllers. Most of the current SDN controllers (e.g., OpenDayLight, ONOS) provide their own northbound Application Programming Interface (API), which allows applications to directly program the underlying network resources, which are exposed by the SDN controller. The proposed COP provides a common NBI API so that SDN controllers can be orchestrated using a single common protocol.

From a Transport SDN perspective, it is needless to mention that there is a lack of specifications on how user applications should interact with the underlying network resources and how services should be requested. The proposed COP provides the necessary commands to bring the full benefits of programable SDN transport networks to applications. The latest OIF/ONF Transport SDN API is in line with COP objectives. COP provides a research-oriented multi-layer approach using YANG/RESTconf, while OIF/ONF Transport SDN API is focused on standardization efforts for orchestration of REST NBI for SDN controllers.

A draft COP definition is open for discussion and can be downloaded and contributed to at:

https://github.com/ict-strauss/COP

In the following subsections the base definition for COP is presented. The usage of YANG models and RESTCONF protocol is also discussed.

A. COP definition

1) Call: The first common service identified as a COP requirement is the design of a common provisioning model which defines an end-to-end connectivity provisioning service. In the scope of COP the service Call is defined as the provisioning interface.

A Call object must describe the type of service that is requested or served by it (e.g., DWDM, Ethernet, MPLS). It also contains the endpoints between whom the service is provided.

The Call object also includes the list effective connections made into the data plane, to support the service call. A Connection object is used for a single network domain scope. It should include the path or route across the network topology the data traverses, which may be fully described or abstract depending on the orchestration/control schemes used. Each connection must be associated with a single control plane entity (e.g. a SDN controller) responsible for the configuration of the data path.

Finally, the Call also introduces the necessary TE parameters (e.g., bandwidth) that the service requests.

2) Topology: The COP definition also covers the topological information about the network, which must include a common and homogeneous definition of the network topologies included in the TE Databases (TED) of the different control instances.

A Topology object may consist of a set of nodes and edges, which form a tree structure. A Node must contain a list of ports or endpoints and their associated switching capabilities. An Edge object is defined as the connection link between two Endpoints. Due to the need of conforming to a common model among different transport network technologies, the definition of the three main objects described (Node, Edge, Endpoint) must be extensible, able to include TE extensions to describe different switching capabilities (i.e., time-slots, packets, wavelengths, frequency slots).

3) Path Computation: The Path Computation service should provide an interface to request and return Path objects which contain the information about the route between two Endpoints.

Path computation is highly related to the previous group of resources. In the service Call, the Connection object has been designed to contain information about the traversed Path. The Path model should be the same in both, the service Call and at the Path Computation. Furthermore each component in the Path object is represented as an Endpoint with TE information associated to it.

B. YANG data model of COP

After identifying the different COP models that a common orchestration interface must fulfill in order to provide useful orchestration mechanisms in a wide range of possible SDN scenarios, it is necessary to select the data modeling language to describe the COP data models.

YANG [1] is a data modeling language designed to describe the configuration, interactions and state data managed by the NETwork CONFiguration Protocol (NETCONF) [2]. NETCONF protocol provides the mechanisms to install, manipulate, and delete the configuration of network devices. YANG and NETCONF principles can be applied for modeling COP using the following mechanisms:

- **Configuration Data** is organized in YANG as a hierarchical tree data structure where each node contains a name identifier and a set of child nodes. Each node can be created, retrieved, updated or deleted (CRUD operations).

- **Remote Procedure Calls (RPC)** can be defined in YANG as independent operations, to translate into more complex operations on the remote SDN controllers.
- **Notifications** can be received asynchronously by the SDN orchestrator to update the state of any configuration parameter in case of changes in the network. This feature is a key-requirement for efficient resilient mechanisms.

C. *RESTCONF/YANG for COP*

Typical NBI API for SDN controllers are implemented using RESTful (or REST) HTTP-based technology. REST encodes data into a uniform media type such as JSON or XML, that is specified into the message header and every resource exchanged is uniformly described using an Uniform Resource Identifiers (URIs).

The REST paradigm is convenient for the COP implementation due to the need of stateless communication among SDN controllers and the SDN orchestrator. It is also convenient because of the flexibility, scalability and commodity for practical implementation. REST practices and architecture have been adopted by NETCONF in RESTconf [3] and considering the benefits of both communication schemes (REST and NETCONF), RESTconf has been considered to be a suitable transport protocol on which the different COP yang models are transported.

YANG/RESTconf provides the suitable combination for COP in order to provide the necessary flexibility and usability.

III. COP USE CASES

In this section we present application areas for COP in an considerable amount of research projects on optical networks.

It can be observed that the two main usages for COP, which are described in the following use cases are basically the use of COP as a NBI for SDN controllers, and the use of COP as a NBI for SDN orchestrators.

A. *STRAUSS use case*

The STRAUSS project proposes a future software defined optical Ethernet transport network architecture [4]. The STRAUSS architecture addresses the provisioning of end-to-end network services across multiple network domains with heterogeneous transport and control planes technologies.

Fig. 1 shows the overview of the STRAUSS architecture and the use case of COP in the project. In the considered use case, the network consists of four network domains including one OpenFlow-enabled OPS DWDM network domain, one OpenFlow-enabled hybrid OPS/Flexi-grid OCS DWDM network domain, one GMPLS/PCE-enabled Flexi-grid OCS DWDM network domain, and one OpenFlow-enabled OCS DWDM network domain. These network domains with heterogeneous control planes technologies (i.e. OpenFlow, GMPLS/PCE) are coordinated by an SDN orchestrator in order to enable the automatic provisioning of end-to-end Ethernet transport services spanning a multi-layer and multi-domain network.

Since the current SDN controller implementations have their own specific north-bound interface due to the lack of a standard, it is necessary to define common models for the control plane functions available in all domains (e.g. provisioning, topology and path computation) and a common transport orchestration protocol. Therefore, the COP is introduced between the network orchestration layer and the network control layer in order to unify all the orchestration functionalities into a single protocol paradigm and support the orchestration of multi-layer, multi-technology network domains.

B. *IDEALIST use case*

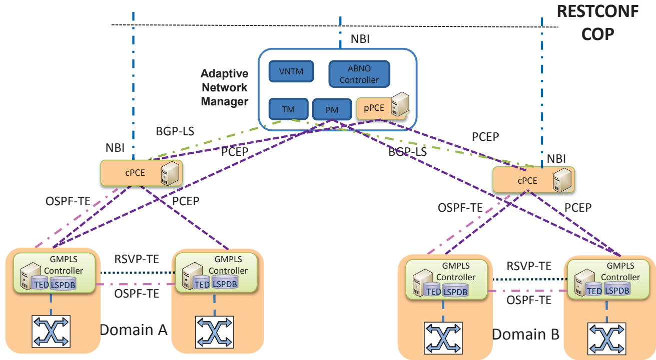

Fig. 2: IDEALIST use case

The IDEALIST project has designed and implemented a Generalized MultiProtocol Label Switching (GMPLS) control plane for flexigrid optical Dense Wavelength Division Multiplexing (DWDM) networks [5].

The control plane (CP) architecture is based on the concept of domain topology abstraction and on the use of a stateful hierarchical PCE (H-PCE). On top of the CP an Adaptive Network Manager (ANM) is introduced. The ANM triggers, by means of a provisioning interface towards the parent PCE (pPCE), the activation of network connectivity services, which maps to the actual establishment and release, via the CP, of elastic connections. The actual provisioning of the connection is coordinated by the pPCE and ultimately delegated to the underlying GMPLS control plane at each domain.

In this use case, COP is presented as a possible NBI for the ANM. COP might provide a single point of enty interface to ANM application for requesting dedicated services (i.e., using the service Call) or to request an abstract network topology (i.e., using the service Topology). The introduction of this flexible NBI will result in more network-aware applications, which are able to request the necessary connectivity services.

C. *DISCUS use case*

The DISCUS architecture [6]- [7] builds on the concept of long-reach PON (LR-PON) in the access connected to a flat optical core, which makes up a transparent optical island.

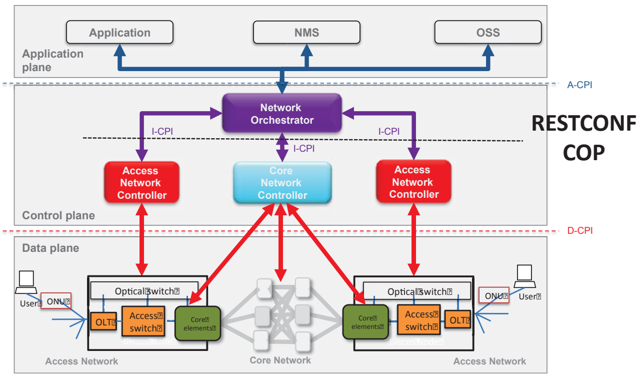

Fig. 3: DISCUS use case

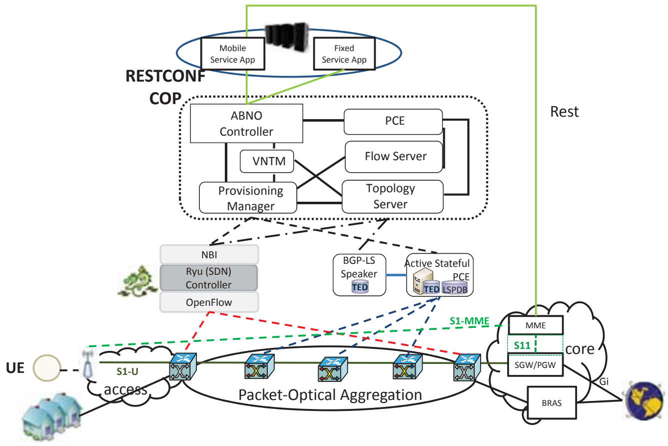

Fig. 4: COMBO use case

One of DISCUS' aims is to remove all packet processing in the metro-access section, which is achieved by terminating all PONs directly into metro-core (MC) nodes. MC nodes are the only electronic packet processing interfaces between PONs and the core transmission network. The core is a flat optical network interconnecting DISCUS MC nodes through a full mesh of wavelength channels. These wavelengths traverse intermediate core nodes without entering the electronic sub-layers.

The DISCUS network is operated by the CP architecture shown in Fig. 3, which identifies three main elements: the access network controller, in charge of controlling the access network elements; the core network controller, in charge of controlling the elements carrying out core transmission; the network orchestrator, in charge of taking requests from applications (e.g., SP requests, management systems,..) and translating them into high-level commands for the access and core network controllers. Following the Open Networking Foundation (ONF) SDN architectural document [8], the DISCUS control plane identifies three types of interfaces among its components: Application-Controller Plane Interfaces (A-CPI), determining the interaction between external applications and orchestrator; Intermediate-Controller Plane Interfaces (I-CPI) describing the interaction between orchestrator and access and core controllers; Device-Controller Plane Interfaces (D-CPI), defining the interfaces between the controllers and the devices (e.g., using OF or GMPLS protocols). Where service provisioning requires interaction among two or more MC nodes, the coordination is provided by the orchestrator, which communicates via I-CPI to both access controllers and core controller.

COP fits well into the DISCUS CP architecture, as it could operate at the I-CPI level, thus creating a standardized communication protocol allowing seamless orchestration of access and core controller technologies. Where multiple domains are concerned, it is envisaged COP will facilitate interaction among multiple DISCUS orchestrator entities, each controlling a different optical island.

D. COMBO use case

The COMBO project aims at proposing and investigating a unified access and aggregation network architecture allowing fixed and mobile networks to converge (Fixed Mobile Convergence, FMC) [9]. This convergence of fixed and mobile networks will be driven by an improved network infrastructure ensuring reduced cost (both OpEx and CapEx). One of the COMBO proposed solutions is the centralized functional convergence depicted in Fig. 4, with the goal to deploy a common and unified orchestration system based on the SDN principles to seamlessly handle the automatic provisioning and recovery of both fixed and mobile data flows. The example considers a multi-layer (MPLS-TP/WSON) aggregation infrastructure. Thereby, both fixed and mobile services are grouped at the packet layer and transported over the same optical tunnels. For example, mobile bearers are encapsulated at the MPLS layer and then transported transparently towards the mobile core, i.e. the 3GPPP communication between mobile base stations (ENodeBs) and the centralized Evolve Packet Core (EPC).

The control of the packet and optical network elements is provided by two independent CP which are coordinated by a higher-level SDN orchestrator. For the packet switching, a SDN controller is used which commands the forwarding configuration via an extended OpenFlow protocol. On the other hand, the optical infrastructure (WSON) is governed by an Active Stateful PCE which is responsible for computing and triggering the optical LSP establishment via the GMPLS signaling. The role of the SDN orchestrator (based on ABNO [10]) is twofold: first, it allows the coordination of the heterogeneous control plane instances for governing the multi-layer aggregation network; second, it attends the service requests for both fixed and mobile connections arriving from the respective service applications located at the cloud. Observe that such service applications running on top of the SDN orchestrator are responsible for requesting, modifying and releasing the connections to serve both fixed and mobile services.

COP fits well into the COMBO approach since it provides a common communication protocol (over the northbound interface) between the SDN orchestrator and the fixed and mobile services applications. Consequently, this enables to seamlessly handle the operations of both service types relying on a converged functional SDN orchestrator.

E. INSPACE use case

The INSPACE project works in the Space-Division Multiplexing (SDM) domain, which is a nascent technology for scalable and high capacity transmission networks. The INSPACE CP for SDM networks differs from fixed-grid or flexi-grid WDM solutions because it has to capture the nuances needed to support a more complex network model, where parallel transmission takes place over multiple media (e.g. bundles of fibers, multi-core fibers, multi-mode fibers). Furthermore, new resource allocation schemes that exploit the presence of the additional space dimensions and take the related physical limitations into account must be developed.

The INSPACE control plane (shown in Figure 5) adopts a centralized SDN architecture, which can be described using an operating system analogy. At the highest level sits the System API layer, which exposes functionalities to the client third-party applications. These functionalities are implemented by the Core Applications layer modules; these are the Connection Manager, responsible for handling setup and teardown requests, the Topology Service, which exposes an abstracted vision of the network, the PCE, responsible for path computations, and the Virtualization Engine, in charge of creating virtual topologies. These modules rely on the Kernel layer, which is responsible for managing devices (including memory, i.e., the TED), and which offers a unified interface irrespective of the peculiarities of the underlying hardware, by translating network state between its own representation (i.e., the one used by the Core Apps) and that used by potentially multiple south-bound protocols.

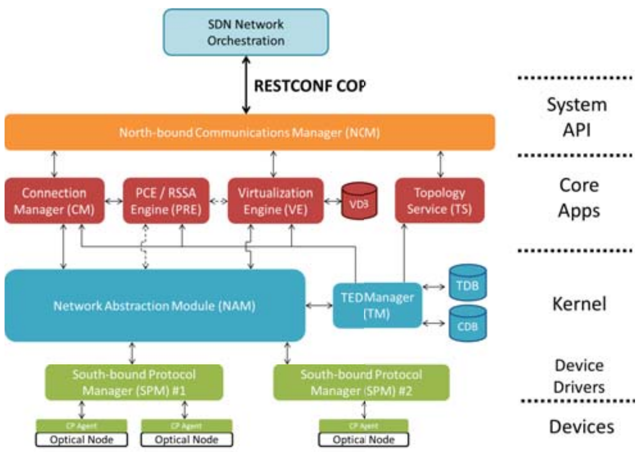

Fig. 5: INSPACE use case.

By means of the aforementioned Topology Service module, the INSPACE CP can expose to a network orchestrator (via the System API and by using COP) an abstracted (e.g. simple graph) representation of the underlying network that hides the SDM-related details. The orchestrator can then see the SDM network as just another transport network, which simply provides connectivity services (via the Connection Manager module). More in detail, the orchestrator can issue requests for connection setups and teardowns, even including detailed resource allocation descriptions (e.g. by using a third-party PCE). Although, the current COP definition does not support virtualization, in the future, network partitions should be requested to the Virtualization Engine through COP.

IV. Conclusions

We have presented COP as a common protocol for the interworking of heterogenous control plane paradigms. COP abstracts a set of control plane functions used by an SDN Controller, allowing the SDN orchestrator to uniformly interact with several domains, each controlled by a single SDN controller. COP has been defined using the YANG modeling language and can be transported using RESTconf.

COP has been defined in the scope of STRAUSS due to the need for an overarching control plane protocol for network orchestration. We have presented several research projects which describe how COP could fit into their architecture. The proposed COP use cases cover the following research projects: STRAUSS, IDEALIST, DISCUS, COMBO, INSPACE.

V. Acknowledgements

This paper was supported by the European Community's Seventh Framework Programme FP7/2007-2013 through the STRAUSS project (608528), IDEALIST project (317999), DISCUS project (318137), COMBO project (317762), INSPACE project(619732), and PACE CSA (619712).

References

[1] M. Bjorklund, "YANG - A Data Modeling Language for he Network Configuration Protocol (NETCONF)," RFC 6020 (Proposed Standard), Internet Engineering Task Force, Oct. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc6020.txt

[2] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network Configuration Protocol (NETCONF)," RFC 6241, Internet Engineering Task Force, Jun. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6241

[3] A. Bierman, M. Bjorklund, and K. Watsen, "RESTCONF Protocol," Internet Engineering Task Force, Jan. 2015. [Online]. Available: https://tools.ietf.org/html/draft-ietf-netconf-restconf-04

[4] R. Muñoz *et al.*, "Network Virtualization, Control Plane and Service Orchestration of the ICT STRAUSS Project," in *in Proc. of European Conference on Networks and Communications (EuCNC)*, 2014.

[5] R. Casellas *et al.*, "IDEALIST Control Plane Architecture for Multi-domain Flexi-Grid Optical Networks," in *in Proc. of European Conference on Networks and Communications (EuCNC)*, 2014.

[6] M. Ruffini *et al.*, "DISCUS: End-to-end network design for ubiquitous high speed broadband services (Invited)," in *in Proc. of ICTON*, 2013.

[7] ——, "DISCUS: An end-to-end solution for ubiquitous broadband optical access," *IEEE Communications Magazine*, vol. 52, no. 2, February 2014.

[8] ONF, "SDN Architecture 1.0." June 2014. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf

[9] S. Gosselin *et al.*, "Fixed and Mobile Convergence: Needs and Solutions," in *in Proc. of European Wireless (EW)*, 2014.

[10] D. King and A. Farrel, "A PCE-based Architecture for Application-based Network Operations," draft-farrkingel-pce-abno-architecture (work in progress), Internet Engineering Task Force, 2015.