Peer SDN Orchestration: End-to-End Connectivity Service Provisioning Through Multiple Administrative Domains

R. Vilalta(1), A. Mayoral(1), V. López(2), V. Uceda(2), R. Casellas(1), R. Martínez(1), R. Muñoz(1), A. Aguado(3), J. Marhuenda(3), R. Nejabati(3), D. Simeonidou(3), N. Yoshikane(4), T. Tsuritani(4), I. Morita(4), T. Szyrkowiec(5), A. Autenrieth(5)

(1) CTTC (rvilalta@cttc.es), (2) Telefónica I+D, (3) University of Bristol, (4) KDDI R&D Labs., (5) ADVA Optical Networking

Abstract *This papers proposes the usage of Control Orchestration Protocol (COP) as an East-West interface in order to interconnect different SDN controllers (peer model) through multiple administrative domains. An experimental validation of network connectivity provisioning is presented in an international testbed.*

Introduction

Within the transport SDN community, it is commonly accepted that deploying a single, integrated controller for a large or complex network may present scalability issues, or may not be doable in practice. Two main reason are: a) Network size, in terms of controllable elements, which has a direct impact on the controller requirements (e.g. active and persistent TCP connections on top of which control sessions are established, memory requirements to store in memory e.g. a data structure representing the network graph that abstracts the network and CPU requirements for processing message exchange); and b) Network complexity, in terms of having a network that combines multiple technology layers[1].

To address such shortcomings, it is important to consider the deployment of multiple controllers, arranged in a specific setting, along with inter-controller protocols. Such network architectures apply both to heterogeneous and homogeneous control (different or same control plane and data plane technologies within the domain of responsibility of a given controller). Two approaches to controller interconnection are identified, which depend on the directivity of the interconnection model: hierarchical and peer.

In a recursive hierarchical interconnection model, controllers are ranged in a topology which is, typically a tree, with a given root being the top-most controller. For a given hierarchy level, a parent SDN controller handles the automation and it has a certain number of high-level functions, while low level controllers (usually referred to as children) cover low-level, detail functions and operations. A recurring example is a 2-level hierarchy in which a parent SDN controller is responsible for connectivity provisioning at a higher, abstracted level, covering inter-domain aspects, while specific per-domain (child) controllers map the abstracted control plane functions into the underlying control plane technology. The Control Orchestration Protocol (COP)[2] has been proposed as a Transport API to enable this.

Peer interconnection model corresponds to a set of controllers, interconnected in an arbitrary mesh, which cooperate to provision end-to-end services. In this setting, we can often assume that the mesh is implicit by the actual (sub)domains connectivity; the controllers hide the internal control technology and synchronize state using East/West interfaces. The SDN controllers manage detailed information of their own, local topology and connection databases, as well as abstracted views of the external domains and the East/West interfaces should support functions such as network topology abstraction, path computation and connectivity provisioning.

In a realistic scenario[3], which typically includes multiple administrative domains, services span across multiple domains and they all must find a way to interconnect and deliver end-to-end services. In this multi-carrier environment there's no hierarchy, no cross-domain control, no cross-domain visibility. It is reasonable that a peer interconnection model is needed. Within a single domain, a parent SDN controller, might handle different underlying technology-specific child SDN controllers. It is in this context, where a hybrid scenario is needed, using hierarchical interconnection within the domain, and peer interconnection through different administrative domains.

This paper proposes extends the work on hierarchical SDN orchestration[4] within a single administrative domain, to incorporate peer SDN orchestration between domains. An international control plane testbed between CTTC, Telefónica, KDDI R&D Labs, University of Bristol and ADVA optical networking has been set-up in order to demonstrate the feasibility of this approach.

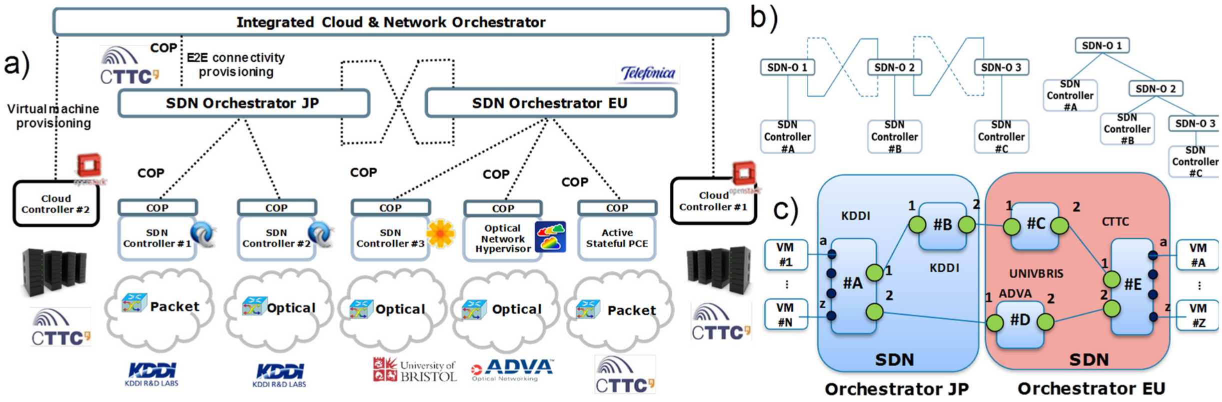

Fig. 1: a) Proposed integrated cloud and network architecture; b) Neighbor recursion pattern; c) Topological views from SDN-O-EU and SDN-O-JP

Peer SDN Orchestration

Fig.1.a shows the proposed architecture for integrated cloud and network orchestration. Two Data Centers (DC) are interconnected though different provider networks. In the proposed architecture, each provider network is controlled though an SDN orchestrator (parent SDN controller, pSDN), which handles several child SDN controllers (cSDN). Each cSDN is responsible for a single network segment. A recursive hierarchy could be based on technological, vendor, SDN controller type, geographical domains or network segment basis. COP[4] was demonstrated as a viable protocol for recursive hierarchical inter-connection between pSDN and cSDN. Finally, we propose to extend the COP to allow a pSDN from Provider A (SDN-O-JP) being able to interact with its peer from Provider B (SDN-O-EU).

Neighbour recursion has been proposed[1] as the pattern in which SDN controllers peer to deliver services across multiple SDN control domains. All participants would be expected to expose comparable levels of abstraction and services. Using neighbour recursion, any SDN controller can act as either client or server to its neighbours, depending on the requested service endpoints. It can be noted, that requested peer services will be understood in a call model, including service creation, service usage, and service release.

Peer SDN orchestrators might use neighbour recursion to provision End-to-End (E2E) services, such as DC interconnection. In all cases, the service endpoints must be coordinated and therefore recursively visible, while the internal details of the network are typically abstracted, and left for the immediate controller.

Fig.1.b provides an example for neighbour recursion. Fig.1.b (left) shows three SDN orchestrators (pSDN), each responsible for a cSDN. For an E2E connectivity request starting in a service endpoint handled by SDN-O 1 and ending at service endpoint handled by SDN-O 3, Fig.1.b (right) shows the neighbour recursion pattern, which results in a balanced hierarchy of SDN-Os. The proposed neighbour recursion pattern does not detail how inter-domain topology is obtained, as a mechanism to avoid topological loops shall be implemented. It is assumed that service end-point reachability is known.

Fig.1.c provides the different overall topological views from the proposed scenario in Fig.1.a. The provided topological view corresponds with the proposed experimental validation, where SDN-O-JP is responsible for SDN controllers A and B; and SDN-O-EU is responsible for SDN controllers C, D, and E. Two inter-domain links are provided between #A:2-#D:1 and #B:2-#C:1. Moreover, COP has been extended in order to offer the context for a client, which includes the abstracted topology and the available service endpoints. In the proposed scenario, each SDN-O announces as service endpoints the different network endpoints on which virtual machines might be interconnected (SDN-O-JP service endpoints: a-z; SDN-O-EU service endpoints: A-Z).

Fig.2.a shows the proposed workflow between a cloud and network orchestrator which requests two virtual machines (VM) (step 1) and requests their interconnection through a dedicated E2E connectivity service (step 2). The E2E connectivity request is sent to the SDN orchestrator responsible for the source service endpoint (SDN-O-JP). The intra-domain connectivity is provisioned through SDN controller #A (step 3) and #B (step 4). Through neighbour recursion, the necessary connectivity is requested to SDN-O-EU (step 5). SDN-O-EU

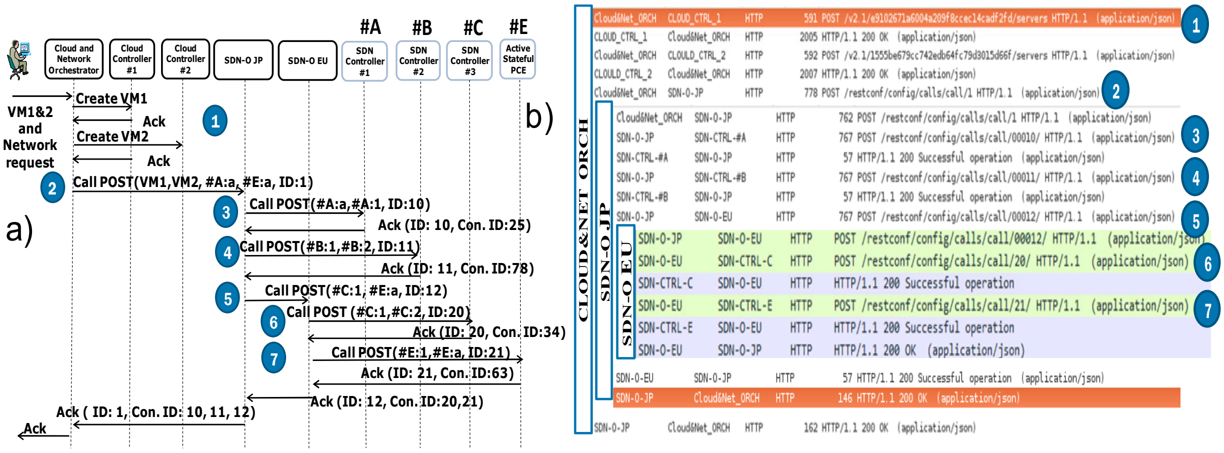

Fig. 2: a) Message workflow for VM and Connectivity Service creation; b) Wireshark captures from three viewpoints: Cloud and network orchestrator, SDN-O-JP, and SDN-O-EU

is responsible for the provisioning of the remaining intra-domain connectivity though SDN controllers #C (step 6) and #E (step 7). Once the necessary connections have been established the Cloud and Network Orchestrator is notified.

Experimental validation

The experimental setup is shown in Fig.1.a, where Bristol, CTTC, and ADVA domains are controlled by an SDN orchestrator (SDN-O-EU), which is run by Telefónica, based on ABNO[5]. The controllers hide the internal setup of each domain. CTTC SDN orchestrator (SDN-O-JP) is responsible for handling multiple technology SDN controllers from KDDI. Each SDN controller provides through COP the abstracted topology as a node. The multiple SDN orchestrators and controllers are interconnected through an OpenVPN over the public internet offering a control plane testbed.

Fig.2.b shows the messages exchanged from three different perspectives: a) the captured from the Cloud and Network Orchestrator; b) from the SDN-O-JP; and c) from the SDN-O-EU.

The objective of the experimental validation is to create two VMs in different DC, and provide a connectivity service between them. In step 1, two VMs are requested to each respective cloud controller. Later, in step 2 a connectivity service is requested to SDN-O-JP. This results in a HTTP POST command to create a Call object, which includes the necessary connection endpoints, as well as the requested QoS, including bandwidth details. The SDN-O-JP processes the request, and triggers the connectivity service through SDN controllers A and B (steps 3 and 4).

In order to establish the requested service, SDN-O-JP requests the connectivity service to the peer SDN-O-EU (step 5). This service request includes as a source endpoint the inter-domain endpoint #C:1 and as destination endpoint the destination endpoint included in the E2E call. Once this request is processed in SDN-O-EU, a connectivity service is requested through the SDN controllers C and E (steps 6 and 7).

Finally, the requested E2E connectivity service has been provisioned in our setup with an average time of 1.6s; while the VM creation on each cloud controller is in the order of 40s.

Conclusions

This paper has proposed the peer SDN Orchestration in order to handle SDN network orchestration in multiple administrative domains. Neighbour recursion and extensions to Control Orchestration Protocol have been proposed and experimentally validated in a control plane international testbed.

Acknowledgements

EU FP7 project STRAUSS (FP7-ICT-2013-EU-Japan 608528), and Spanish MINECO project DESTELLO (TEC2015-69256-R).

References

[1] SDN Architecture 1.1, ONF, 2016.

[2] A. Mayoral et al., First experimental demonstration of a distributed cloud and heterogeneous network orchestration with a common Transport API for E2E services with QoS, OFC 2016.

[3] Shahar Steiff, The Real Story Behind Multi-Operator Automation, Open Networking Submit, 2016.

[4] R. Vilalta et al., Hierarchical SDN Orchestration for Multi-technology Multi-domain Networks with Hierarchical ABNO, ECOC 2015.

[5] D. King and A. Farrel, A PCE-Based Architecture for Application-Based Network Operations, IETF, RFC 7491, 2015.