First experimental demonstration of a distributed cloud and heterogeneous network orchestration with a common Transport API for E2E services with QoS

A. Mayoral[1], R. Vilalta[1], R. Muñoz[1], R. Casellas[1], R. Martínez[1], M. Svaluto Moreolo[1], J. M. Fàbrega[1], S. Yan[2], A. Aguado[2], E. Hugues-Salas[2], S. Peng[2], F. Meng[2], Y. Shu[2], G. Zervas[2], R. Nejabati[2], D. Simeonidou[2], J. M. Gran[3], V. López[3], O. González de Dios[3], J.P. Fernández-Palacios[3], P. Kaczmarek[4], R. Szwedowski[4], T. Szyrkowiec[4], A. Autenrieth[4], N. Yoshikane[5], X. Cao[5], T. Tsuritani[5], I. Morita[5], M. Shiraiwa[6], N. Wada[6], M. Nishihara[7], T. Tanaka[7], T. Takahara[7], J. C. Rasmussen[7], Y. Yoshida[8], K. Kitayama[8]

[1]CTTC, [2]University of Bristol, [3]Telefónica I+D, [4]ADVA Optical Networking, [5]KDDI R&D Laboratories , [6] NICT, [7]Fujitsu LTD., [8]Osaka University, arturo.mayoral@cttc.es

Abstract: A common Transport API is proposed for joint cloud/network orchestration, allowing interworking of heterogeneous control planes to provide provisioning/recovery of QoS-aware E2E services. We present its first demonstration in a multi-partner testbed including data-plane monitoring.

OCIS codes: (060.4256) Networks, network optimization; (060.4253) Networks, circuit-switched.

1. Introduction

To offer End-to-End (E2E) transport service provisioning and orchestration across multiple domains with heterogeneous transport and control plane technologies, multi-domain control plane interworking is required. The OIF and ONF presented the results of their Global Transport SDN Prototype Demonstration [1] last year, where several SDN controllers and hierarchy levels were analyzed. Currently, a Transport API is standardized at ONF.

The Control Orchestration Protocol [2] (COP) is a well-positioned candidate for a Transport API as it abstracts a set of control plane functions commonly used by SDN controllers, allowing the interworking of heterogeneous control plane paradigms (e.g., OpenFlow (OF) or GMPLS/PCE). The proposed COP provides a common NorthBound / SouthBound Interface (NBI/SBI) for SDN controllers. Moreover, COP enables the joint orchestration of cloud (i.e., computing, storage) and network resources by allowing the provisioning of E2E transport services between Virtual Machines (VM) across multiple network domains.

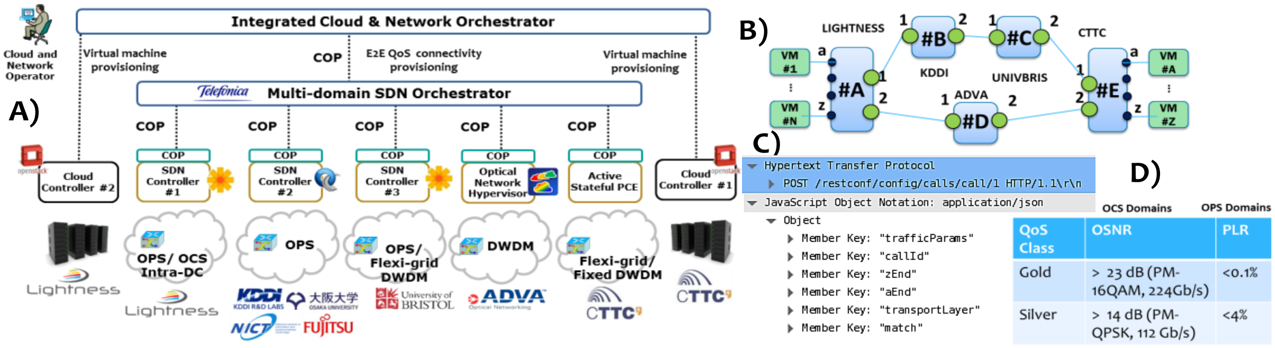
The COP definition covers the topological information about the network, a call service for establishing E2E connections and a path computation service. The COP information model is described in YANG, with RESTconf as transport protocol. Notifications of the monitored optical parameters are also included in the COP definition models.

The COP service call is defined as the E2E provisioning interface. A *Call* object describes the type of service that is requested or served (e.g., DWDM link, Ethernet E2E transport, MPLS). It contains the endpoints between whom the service is provided. The *Call* object also includes the list of effective *Connections* created in the data plane, to implement the E2E service. A *Connection* object is used for a single network domain scope. The *Connection* includes the path across the network topology that the data traverses, which may be fully described or abstracted depending on the orchestration/control schemes used. Finally, the Call also introduces the necessary TE parameters (e.g., bandwidth, QoS class, latency) that the service may request.

This paper is the first architectural proposal and experimental validation of an integrated orchestration of IT and heterogeneous network resources using a common Transport API for E2E provisioning and recovery services with QoS. The E2E recovery mechanism without QoS and without COP has been previously presented in [3]. In this paper we present the provisioning and QoS recovery results in an experimental multi-partner testbed with distributed Data Centers (DCs) from the LIGHTNESS project and at CTTC premises and multi-domain Optical Circuit/Packet Switching (OCS/OPS) networks [3], covering Spain, U.K., Germany and Japan. Data plane quality of transmission (QoT) monitoring is integrated into the control plane for the first time to monitor cross-domain connections, which enables both per-domain and E2E QoS recovery.

2. Dynamic VM and E2E transport service deployment with QoS provisioning

Fig.1A shows the integrated experimental scenario, where LIGHTNESS and CTTC DCs can be jointly orchestrated with a heterogeneous set of transport networks, including OPS/OCS domains. Each network domain is controlled either by an SDN controller, an Optical Network Hypervisor (ONH) or an Active Stateful PCE. Each domain provides its abstracted topology (node abstraction) to the multi-domain SDN orchestrator, which is based on ABNO [4] (Fig.1B shows the topology observed by the SDN orchestrator). Finally, an integrated cloud and network orchestrator is introduced to provide joint orchestration of IT and network resources.



```
▼ Hypertext Transfer Protocol
  ▶ POST /restconf/config/calls/call/1 HTTP/1.1\r\n
▼ JavaScript Object Notation: application/json
  ▼ Object
    ▶ Member Key: "trafficParams"
    ▶ Member Key: "callId"
    ▶ Member Key: "zEnd"
    ▶ Member Key: "aEnd"
    ▶ Member Key: "transportLayer"
    ▶ Member Key: "match"
```

	OCS Domains	OPS Domains
QoS Class	OSNR	PLR
Gold	> 23 dB (PM-16QAM, 224Gb/s)	<0.1%
Silver	> 14 dB (PM-QPSK, 112 Gb/s)	<4%

Fig. 1: A) Proposed LIGHTNESS-STRAUSS scenario; B) Abstracted network/cloud scenario; C) Call example, D) QoS classes.

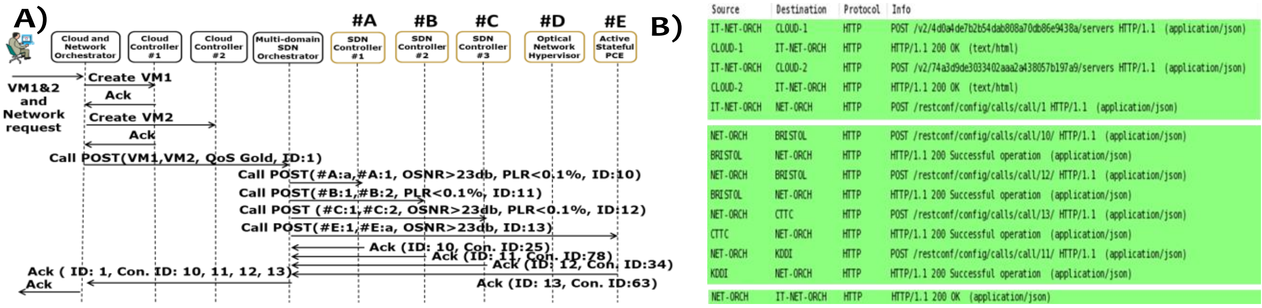
With the objective of providing E2E transport services between the two DC with QoS, we have introduced two QoS classes (Fig.1D) in the COP *Call* definition (Fig.1C, *trafficParams*). Each QoS class defines a certain packet loss rate (PLR) for OPS domains, and a certain OSNR for OCS domains, for a given bandwidth request. The SDN orchestrator will translate the high level QoS classes into the corresponding parameters in the call requests sent to the different SDN controllers. Fig.2A shows the message exchange between the involved computing and network elements in order to jointly provide interconnected VMs with QoS. The provisioning of the VMs is requested to each responsible cloud controller, while the VM interconnection is requested to the SDN orchestrator with an E2E call (ID: 1) including a QoS class. The SDN orchestrator computes the E2E path and requests the necessary calls (IDs: 10, 11, 12, 13) to the different SDN Controllers. Fig.2B shows the Wireshark captures at the integrated cloud and network orchestrator and at the SDN orchestrator.

3. Per-domain / E2E service recovery with QoS

Fig.3A shows three conducted experiments for QoS recovery: in an OPS domain (scenario A), in an OCS domain (scenarios B, C) and finally E2E QoS recovery (scenario D).

Per-domain QoS recovery through adaptive route control in the OPS network: Fig. 3B shows the experimental setup of the OPS domain at the NICT premises in Japan. The OPS nodes used are optical packet and circuit integrated nodes [5], including one SOA-based 4 × 4 OPS. In the control plane, an OF-based SDN controller is used to control the OPS nodes. Four OPS nodes with optical packet counters are used, including OF agents and OPS transmitters and receivers. The OF agent periodically reads and provides to the SDN controller the optical packet count information that is measured. In this use case (scenario A), two E2E connections are setup involving the OPS domain, flow1 with a packet occupancy rate of 10% and flow2 with a packet occupancy rate of 2%. In this case, the PLR for flow1 measured by a tester is around 4%. When we increase the packet occupancy rate of flow2 from 2% to 6%, the optical packet counter of OPS node 4 reaches the pre-defined threshold indicating packet congestion. Fig. 3C shows the measured packet counts of Node 4. With the increase of the packet occupancy rate, the packet count at OPS node 4 is finally smaller than 17000, the OF agent attached to OPS node 4 detects packet congestion and sends an alarm to the SDN controller. The SDN controller receives the alarm and then issues the route adaption for the switching table of node 2, in order to improve the PLR. After the route control, the obtained PLR for flow1 measured by the traffic tester is reduced from around 4% to 0.1%. The route adaptation is announced to SDN orchestrator via the COP notification mechanism which employs WebSockets as transport technology.

QoS recovery in an OCS domain: For same BER performance, the required OSNR value will relax when a signal with a lower order modulation format is used [6]. Fig.3E shows the tested OSNR vs. BER curve for our 28Gbaud PM-QPSK and PM-16QAM transmitters (Txs). QPSK requires an OSNR value about 9dB less than 16QAM at HD-FEC threshold (3.8E-3). Moreover, OSNR monitoring of a circuit flow can detect the OSNR degradation for optical links. The receiver-side error-vector-magnitude (EVM) based monitor provides in-band OSNR monitoring without deploying new hardware [7]. With these monitoring information, the COP can orchestrate multi-domain E2E service efficiently and reconfigure the network according to the traffic and link conditions to maintain QoS. Fig.3F shows



```
Source       Destination  Protocol  Info
IT-NET-ORCH  CLOUD-1      HTTP      POST /v2/4d0a4de7b2b54dab808a70db86e9438a/servers HTTP/1.1  (application/json)
CLOUD-1      IT-NET-ORCH  HTTP      HTTP/1.1 200 OK  (text/html)
IT-NET-ORCH  CLOUD-2      HTTP      POST /v2/74a3d9de3033402aaa2a438057b197a9/servers HTTP/1.1  (application/json)
CLOUD-2      IT-NET-ORCH  HTTP      HTTP/1.1 200 OK  (text/html)
IT-NET-ORCH  NET-ORCH     HTTP      POST /restconf/config/calls/call/1 HTTP/1.1  (application/json)

NET-ORCH     BRISTOL      HTTP      POST /restconf/config/calls/call/10/ HTTP/1.1  (application/json)
BRISTOL      NET-ORCH     HTTP      HTTP/1.1 200 Successful operation  (application/json)
NET-ORCH     BRISTOL      HTTP      POST /restconf/config/calls/call/12/ HTTP/1.1  (application/json)
BRISTOL      NET-ORCH     HTTP      HTTP/1.1 200 Successful operation  (application/json)
NET-ORCH     CTTC         HTTP      POST /restconf/config/calls/call/13/ HTTP/1.1  (application/json)
CTTC         NET-ORCH     HTTP      HTTP/1.1 200 Successful operation  (application/json)
NET-ORCH     KDDI         HTTP      POST /restconf/config/calls/call/11/ HTTP/1.1  (application/json)
KDDI         NET-ORCH     HTTP      HTTP/1.1 200 Successful operation  (application/json)

NET-ORCH     IT-NET-ORCH  HTTP      HTTP/1.1 200 OK  (application/json)
```

Fig. 2: A) VM connectivity provisioning workflow; B) Wireshark capture.

the performance of OSNR monitor for both QPSK and 16QAM Txs. The monitor notifies the SDN controller, when the OSNR is degraded up to a threshold specified by the multi-domain orchestrator during the provisioning. Then SDN controller reconfigures the link either to use another path (scenario B) or to adopt a lower order modulation-format signal with a multi-format transceiver (scenario C).

Fig.3D shows the experimental setup for the OCS domain. At node A, 10 carriers with QPSK or 16QAM are launched into the optical network. The 28Gbaud QPSK/16QAM transmitter with central wavelength at 1548.9nm is connected to the OPS domain with an integrated OPS/OCS interface. The signals are transmitted 175km from node A, through node B toward D, and demultiplexed at node D for coherent detection. The coherent receiver deploys an EVM-based OSNR monitor. ASE noise is added between node A and B to emulate the OSNR degradation.

By adding more ASE noise to the link, the monitored OSNR will decrease to a threshold of 23dB. Then the monitor instructs the SDN controller to reconfigure the link. In scenario B, the SDN controller reconfigures the optical path from node A to node D to the direct route with a reduced distance link. In this scenario, the total link bitrate won't change, but the SDN orchestrator is notified about the new connection. In scenario C, the QPSK/16QAM Tx is instructed by the SDN controller to reconfigure its modulation format from 16QAM to QPSK, to provide a bitrate of only half of the initial value. Recovered constellation diagrams are shown as insets in Fig.3D for both 16QAM and QPSK. The SDN orchestrator is notified of the new bitrate.

E2E QoS recovery: When a transport domain is unable to recover itself from a failure, or QoS cannot be ensured, it sends a request to the SDN orchestrator to perform an end-to-end QoS recovery (Fig.3A – scenario D), involving all the available domains in order to find an alternative route that satisfies the required QoS. The SDN orchestrator requests the necessary call deletion, modification and establishment from the involved SDN controllers (IDs: 11, 12, 10, 14, 13). Finally, it informs the cloud and network orchestrator about the E2E call modification (ID :1).

4. Conclusions

This paper first introduced and validated an architecture for jointly orchestrate cloud and network resources with the open-source Control Orchestration Protocol in a multi-partner international control and data plane testbed, including per-domain and end-to-end QoS recovery based on data-plane QoT monitoring.

Acknowledgements

EU FP7 STRAUSS (608528) and LIGHTNESS (318606) and FARO (TEC2012-38119) projects.

5. References

[1] OIF-ONF white paper, Global Transport SDN Prototype Demonstration, 2014.
[2 R. Vilalta et al., The Need for a COP in Research Projects on Optical Networking, EuCnC, 2015.
[3] A. Aguado et al., Dynamic Virtual Net. Reconfiguration over SDN Orchestrated Multi-Technology Optical Transport Domains, ECOC 2015
[4] Y. Yoshida et al., SDN-based Network Orch. of Variable-capacity OPS Network over Programmable Flexi-grid Elastic Optical Path Network, JLT, vol. 33, n. 3, 2015.
[5] H. Harai, et al., "Optical Packet and Circuit Integrated Networks and SDN Extension," JLT, vol.32, no.16, 2014.
[6] G. Bosco et al., "Analytical results on cc in uncompensated optical links with coherent detection," Opt. Exp., vol. 19, no. 26, 2011.
[7] R. Schmogrow et al., "Error Vector Magnitude as a Performance Measure for Advanced Mod. Formats," IEEE PTL, 24, 1. 2012.

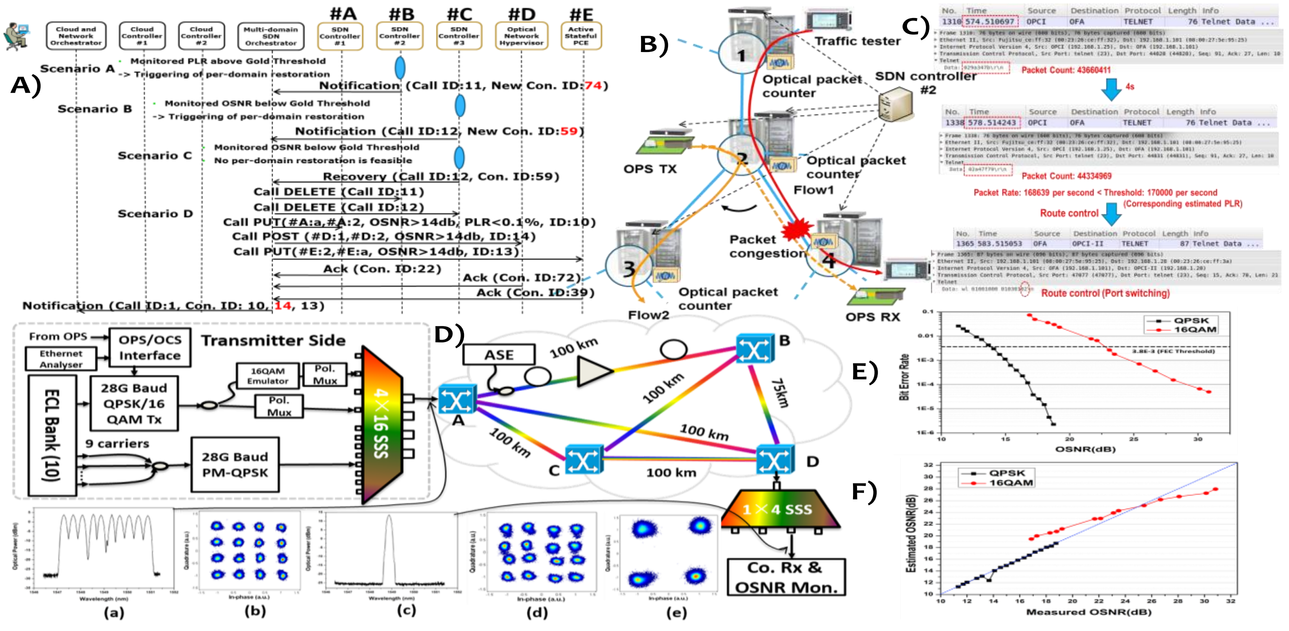


Fig. 3: A) Service restoration workflow, B) OPS domain, C) Adaptative Route Control, D) OCS domain QoS recovery, E)OSNR vs. BER curve for PM-QPSK and PM-16QAM Tx, F) EVM-based in-band OSNR monitored for PM-QPSK and PM-16QAM