Experimental Assessment of Inter-datacenter Multicast Connectivity for Ethernet services in Flexgrid Networks

Ll. Gifre[1*], F. Paolucci[2], J. Marhuenda[3], A. Aguado[3], L. Velasco[1], F. Cugini[2], P. Castoldi[2], O. Gonzalez de Dios[3], L.M. Contreras[3] and V. Lopez[3]
[1] Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Spain. lgifre@ac.upc.edu
[2] Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Italy.
[3] Telefónica Investigación y Desarrollo (TID), Spain.

Abstract *We demonstrate an orchestrated inter-datacenter multicast connectivity for Ethernet services. An ABNO-driven workflow is experimentally validated to provision p2mp connectivity over a multilayer Ethernet-over-Flexgrid network. Experimental validation was carried out on a distributed infrastructure connecting Telefonica, CNIT, and UPC premises.*

Introduction

The distributed nature of cloud computing entails that applications can run on servers belonging to datacenters (DC) in geographically distant locations. Huge data transfer is thus needed to synchronize databases and to distribute content among DCs. To provide connectivity between such DCs, not only point-to-point, but also multicast support to emulate group communication is required.

The Flexgrid optical network technology is being extensively investigated because of its inherent spectrum efficiency and connection flexibility (slot of almost any arbitrary width can be created). The focus of this paper is to demonstrate dynamic connectivity among a set of DCs belonging to a federation, assuming a Flexgrid-based interconnection network.

In the service plane, the federation uses a cloud manager in charge of managing intra-DC resources. Typically, one or more core Ethernet switches inside each DC are connected to a Flexgrid optical core network that interconnects the DCs of the federation. To control the network workflows the Application-Based Network Operations (ABNO) architecture [1] is used. To orchestrate cloud and network systems we use an Application Service Orchestrator (ASO), which is deployed in between the OpenStack cloud manager and the ABNO controller. ASO is responsible of translating the connectivity requests from the DCs to the ABNO. In this work we focus on demonstrating multicast connectivity service provisioning.

Multicast Connectivity Services

To implement multicast connectivity services in a multi-layer network, a virtual topology needs to be created connecting every source switch to every other leaf switch. Several alternative approaches can be considered to support that virtual topology: *i*) create a set of point-to-point (p2p) optical connections between each pair of source-leaf switches or *ii*) create a set of point-to-multipoint (p2mp) optical connections, one for each source router connecting all the leaves.

The first approach is partially illustrated in Fig. 1a, where three p2p optical connections have been set-up connecting switch in DC-A to the rest of DCs (B, C, and D). To create a multicast service, we need to set-up all the connections where routers in DCs B, C, and D are source, so this approach requires 12 p2p connections.

An example of the p2mp approach is depicted in Fig. 1b, where one of the p2mp optical connections for the multicast service is set-up connecting the router in DC-A to the routers in DCs B, C, and D. The feasibility of creating p2mp connections on a flexgrid optical network was demonstrated in [1]. Note that only 4 p2mp connections are needed to implement the multicast service. Not only the number of optical connections is reduced, but also the amount of spectral resources needed, as clearly observed.

Fig. 1c and Fig. 1d present the virtual topology resulting from the two approaches. Whereas in the p2p approach multicast capabilities are needed in each router, the p2mp approach relaxes source router from such task thus reducing its forwarding requirements.

All the rationale above thus provides motivation to use optical p2mp connections for multicast service provisioning.

It is clear that, similarly as for p2p connections, specific p2mp routing and spectrum allocation (RSA) algorithms similar to the one [2] are needed to compute the route from the source to every leaf in the connection and allocate a contiguous and continuous spectrum slot.

Datacenter management

The delivery of distributed cloud services implies the configuration of the networks involved in the service (both in the distributed DCs and in the domains connecting them).

In the case of federated DCs, each DC is operated and managed separately. Service provisioning in such distributed scenario requires a tight coordination to ensure the consistency of the service delivery. This problem space is covered by the Software Defined Networking (SDN) framework, which is seen as the facilitator of this capability.

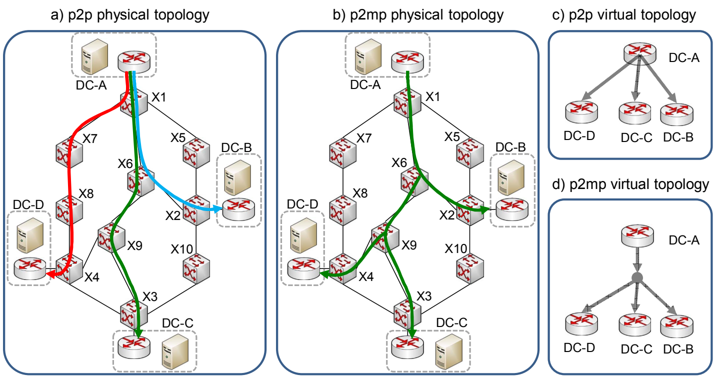


Fig. 1. Physical and Virtual Topologies

For this work, four main software components are developed to govern and control the network connectivity service. The *Cloud computing manager* is OpenStack Grizzly [3], which handles the computing resources in each data center. Neutron plugin is in charge of providing the local overlay networks. As Neutron is a technology dependent plugin, a customized Neutron plugin has been developed to manage the inter-data center connectivity using an SDN approach; the plugin interacts with the *Application Service Orchestrator* (ASO). The Local SDN controller configures the resources within the DC, so it is out-of-the-scope of this paper. The ASO maintains the network information from the DCs requests and interacts with the network orchestrator. The *network orchestrator* is based on Telefonica's ABNO implementation [4], which configures the Ethernet Switches (HP 5406zl) and the flexgrid network.

ASO and each of the OpenStack instances exchange information via the Neutron plugin, so the ASO is aware of which VMs in each data belong to the same network. When multiple networks in each federated DCs are merged, the ASO asks to the ABNO to create unicast and multicast services between these DCs (Fig. 2) Unicast services were demonstrated for multi-layer scenarios in previous work [4]. Here, Ethernet services over flexgrid are used instead of IP/MPLS, but the orchestration process for p2p services is similar.

Workflow: Datacenter multicast connectivity

The ABNO architecture includes *i*) a controller, responsible for implementing workflows orchestrating operations among ABNO modules; *ii*) a Layer 0 PCE (L0 PCE), responsible for path computation on the optical topology; *iii*) a Virtual Topology Network Manager (VNTM), responsible for maintaining a virtual topology between the DCs using resources in the optical topology; *iv*) a Layer 2 PCE (L2 PCE), which computes paths on the virtual topology; *v*) a Provisioning Manager (PM) dealing with the configuration of the network elements (switches or optical nodes) and vi) a Topology Module (TM) in charge of obtaining the TED. Besides, in this paper the L0 PCE is split into two dedicated modules: a front-end PCE (L0 fPCE), responsible of performing impairment-aware computation and initiating the optical connections: and a back-end PCE (L0 bPCE), capable of performing computationally intensive tasks, such as solving the p2mp routing and spectrum allocation algorithm. As described in previous work [4], most of interfaces among ABNO modules are PCEP. To configure the Ethernet switches, the PM uses OpenFlow via a Floodligth [5] controller, but PCEP is used to instantiate connections in the flexgrid nodes.

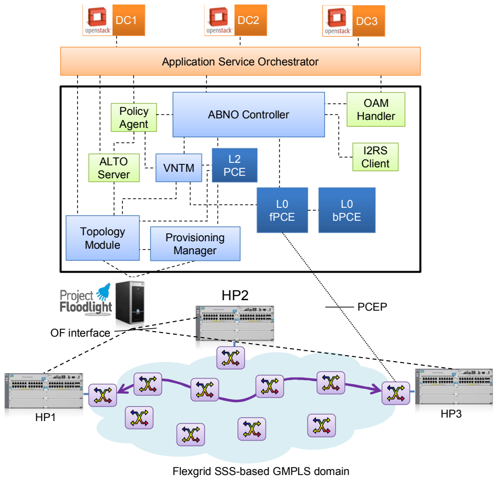


Fig. 2. Experimental set-up scenario

When multiple DC networks are merged, the ASO module asks the ABNO controller for unicast and multicast services between the Ethernet switches. Fig. 3 shows the workflow for one single p2mp connection request. The same workflow is repeatedly executed for the rest of the requests belonging to the same multicast service request.

In our implementation, the L0 fPCE is active whereas L2 PCE is not. Therefore, the ABNO controller requests a L2 p2mp path computation to the L2 PCE (message 1 in Fig. 3). Let us assume that no enough resources are available at this time, so the L2 PCE returns a NO-PATH message (2). The controller delegates to the VNTM module updating the virtual topology, possibly adding more resources to serve the L2 p2mp request. To that end, a request is sent containing the end points of the requested p2mp connection (3). Upon reception, the VNTM sends a request to L0 fPCE to create optical connectivity among the specified end points (4); in this particular use case, a p2mp optical connection needs to be created.

Because L0 p2mp path computation might take long time, L0 fPCE delegates it to the specialized L0 bPCE (5). When the computation ends, the L0 bPCE sends back the solution to the fPCE (6). When the L0 fPCE receives the solution, it delegates its setting-up to the PM, which sends the appropriate commands to the

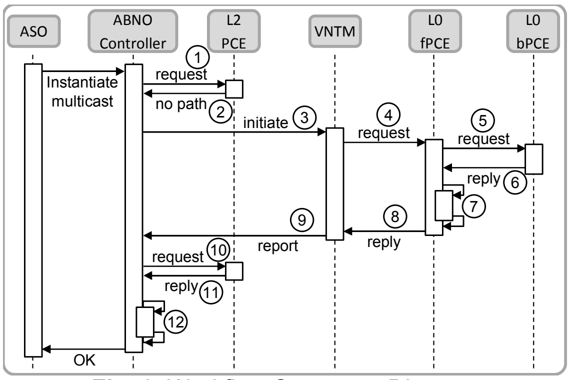
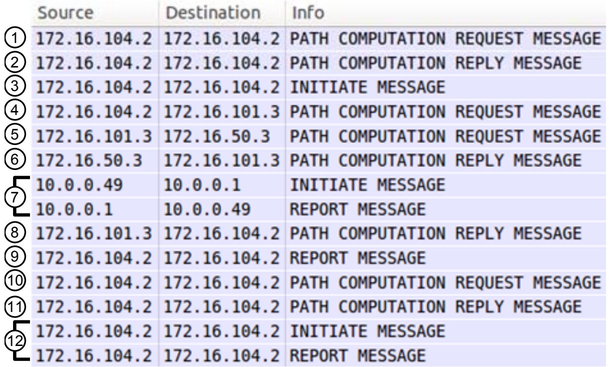
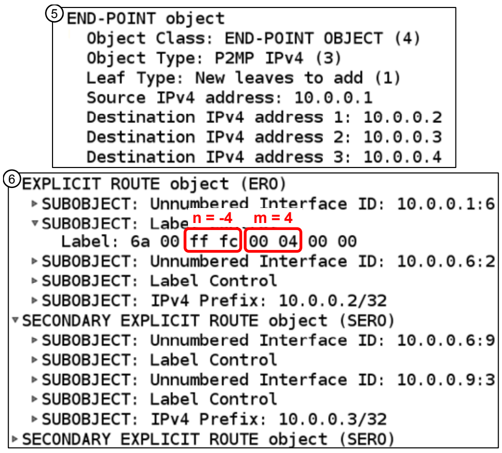


Fig. 3. Workflow Sequence Diagram



	Source	Destination	Info
①	172.16.104.2	172.16.104.2	PATH COMPUTATION REQUEST MESSAGE
②	172.16.104.2	172.16.104.2	PATH COMPUTATION REPLY MESSAGE
③	172.16.104.2	172.16.104.2	INITIATE MESSAGE
④	172.16.104.2	172.16.101.3	PATH COMPUTATION REQUEST MESSAGE
⑤	172.16.101.3	172.16.50.3	PATH COMPUTATION REQUEST MESSAGE
⑥	172.16.50.3	172.16.101.3	PATH COMPUTATION REPLY MESSAGE
⑦	10.0.0.49	10.0.0.1	INITIATE MESSAGE
	10.0.0.1	10.0.0.49	REPORT MESSAGE
⑧	172.16.101.3	172.16.104.2	PATH COMPUTATION REPLY MESSAGE
⑨	172.16.104.2	172.16.104.2	REPORT MESSAGE
⑩	172.16.104.2	172.16.104.2	PATH COMPUTATION REQUEST MESSAGE
⑪	172.16.104.2	172.16.104.2	PATH COMPUTATION REPLY MESSAGE
⑫	172.16.104.2	172.16.104.2	INITIATE MESSAGE
	172.16.104.2	172.16.104.2	REPORT MESSAGE

Fig. 4. Relevant p2mp PCEP Messages



```
⑤END-POINT object
    Object Class: END-POINT OBJECT (4)
    Object Type: P2MP IPv4 (3)
    Leaf Type: New leaves to add (1)
    Source IPv4 address: 10.0.0.1
    Destination IPv4 address 1: 10.0.0.2
    Destination IPv4 address 2: 10.0.0.3
    Destination IPv4 address 3: 10.0.0.4
⑥EXPLICIT ROUTE object (ERO)
  ▸SUBOBJECT: Unnumbered Interface ID: 10.0.0.1:6
  ▾SUBOBJECT: Label            n = -4    m = 4
     Label: 6a 00 ff fc 00 04 00 00
  ▸SUBOBJECT: Unnumbered Interface ID: 10.0.0.6:2
  ▸SUBOBJECT: Label Control
  ▸SUBOBJECT: IPv4 Prefix: 10.0.0.2/32
▾SECONDARY EXPLICIT ROUTE object (SERO)
  ▸SUBOBJECT: Unnumbered Interface ID: 10.0.0.6:9
  ▸SUBOBJECT: Label Control
  ▸SUBOBJECT: Unnumbered Interface ID: 10.0.0.9:3
  ▸SUBOBJECT: Label Control
  ▸SUBOBJECT: IPv4 Prefix: 10.0.0.3/32
▸SECONDARY EXPLICIT ROUTE object (SERO)
```

Fig. 5. Detailed p2mp PCEP Messages

underlying data plane (7).

When the PM module receives the confirmation from the L0 fPCE (8), it is forwarded to the ABNO controller (9). Then the ABNO controller requests a L2 p2mp path computation to the L2 PCE (10), which now finds a solution and returns it to the ABNO controller (11). Since L2 PCE is not active, the ABNO controller module delegates connection set-up to the PM (12), which configures the appropriate rules in each switch using the OpenFlow protocol.

Experimental Assessment

The experimental validation was carried out on a distributed field trial set-up connecting Telefonica, CNIT, and UPC premises. The network topology depicted in Fig. 1 was used for the experiments. CNIT's flexgrid data plane includes programmable spectrum selective switches (SSS) with 1GHz frequency slice granularity. Each SSS is handled by a co-located controller. GbE interfaces are used to connect each controller to the L0 fPCE.

Fig. 4 shows the relevant PCEP messages for a single workflow execution. All the modules in ABNO except L0 fPCE and bPCE are configured with the same IP address: 172.16.104.2. L0 fPCE used IP 172.16.101.3 and bPCE used IP 172.16.50.3. Each message is identified with the same sequence used to describe the workflow.

The details of messages (5) and (6) are given in Fig. 5. PCReq message (5) requests a single L0 p2mp LSP. The end points are specified using an ENDPOINTS class 3 (p2mp IPv4) object [7], which includes the IP addresses of source (X1) and leaves (X2, X3, and X4). PCRep message (6) contains the computed route and spectrum allocation for the LSP, which is conveyed in the ERO/SERO objects. According to [7], aiming at describing p2mp routes in an efficient way, we use one single ERO object to define the route and spectrum allocation from the source node to one of the leaves, and additional SERO objects for each of the rest of leaves; the starting node in each SERO object can be whatever node already in the ERO or SERO objects. To illustrate that, the route of the ERO in Fig. 5 is for leaf X2 and includes X6 as intermediate node. The first SERO object is for leaf X3 and starts in X6.

Conclusions

A use case of multicast connectivity service has been presented and experimentally demonstrated in a realistic DC scenario. OpenStack Neutron plugin is modified to allow the interoperability with the ASO layer, and consequently with the ABNO architecture, in charge of the interconnection network.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2012-2015 under grant agreements no. 317999 IDEALIST project.

References

1 D. King, A. Farrel, IETF draft, 2014
2 N. Sambo et al., "Demonstration of Data and Control Plane for Opt. Multicast at 100 and 200 Gb/s w/ and wo/ Freq. Conversion," JOCN 2013.
3 L. Velasco et al., "Solving Routing and Spectrum Allocation Related Optimization Problems: from Off-Line to In-Operation Flexgrid Network Planning," JLT, 2014.
4 OpenStack. http://www.openstack.org/
5 A. Aguado et al., "ABNO: a feasible SDN approach for multi-vendor IP and optical networks," OFC 2014.
6 Floodligth http://www.projectfloodlight.org/
7 Q. Zhao, et al., IETF RFC 6006, 2010