Fichera, S., Martínez, R., Casellas, R., Martini, B., Vilalta, R., Muñoz, R., Castoldi, P. y Manzalini, A. (2018). Experimental Evaluation of Orchestrating Inter-DC Quality-enabled VNFFG Services in Packet/Flexi-Grid Optical Networks. In: *2018 European Conference on Optical Communication (ECOC)*.

DOI: https://doi.org/10.1109/ECOC.2018.8535231

Experimental Evaluation of Orchestrating Inter-DC Quality-enabled VNFFG Services in Packet/Flexi-Grid Optical Networks

S. Fichera[(1)], R. Martínez[(2)], R. Casellas[(2)], B. Martini[(3)], R. Vilalta[(2)], R. Muñoz[(2)], P. Castoldi[(1)] and A. Manzalini[(4)]

[(1)] Scuola Superiore Sant'Anna, Pisa (Italy). s.fichera@santannapisa.it
[(2)] Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona (Spain)
[(3)] CNIT, Pisa (Italy); [(4)] Telcom Italia, Turin (Italy)

Abstract. *An implemented Cloud/Network Orchestrator to dynamically serve VNFFGs in remote DCs through a Multi-Layer Network (packet/flexi-grid optical) is evaluated. Two network information and path computation approaches are adopted by the Orchestrator being experimentally benchmarked with a number of performance metrics.*

Introduction

Upcoming 5G network services will be very heterogeneous and will demand different network functions with various levels of QoS (bandwidth, latency, etc.). To this end, virtualization techniques such as Network Function Virtualization (NFV) are being adopted. NFV virtualizes network functions in general-purpose servers (VNFs) deployed in datacenters (DCs) inter-connected by, e.g. a metro/core infrastructure. In this context, a network service is determined by VNF Forwarding Graphs (VNNFG)[1], which specifies an ordered set of VNFs potentially placed in remote DCs.

Considering this scenario, in this paper we investigate the dynamic VNFFG allocation over multiple DCs within a transport multi-layer network (MLN). The MLN combines both packet and flexi-grid optical technologies leveraging the effective packet traffic grooming and the optical spectral efficiently and flexibility. VNFFGs are handled by a Cloud/Network Orchestrator composed of two main entities: the *Allocator* for processing VNFFG requests and DC resource allocation; and a Transport SDN controller (based on a PCE Central Controller, PCECC[2,3]) to compute/configure MLN connections.

Two different orchestration approaches (*no network info,* NNI and *abstracted network info, ANI*) are proposed. They differ from the amount of network resources information (i.e., topology and resources) disseminated by the T-SDN controller to the Allocator. In the NNI, no network information is handled by the Allocator only performing DC resource allocation. In the ANI, in addition to DC resource allocation, the Allocator operates with partial network information at the packet level which allows computing inter-DC connections. Each approach relies on a different MLN path computation mechanism for computing packet inter-DC paths. The devised Cloud/Network Orchestrator and both approaches are experimentally evaluated within the CTTC ADRENALINE testbed in terms of blocking, setup delay and IT/network resource consumption.

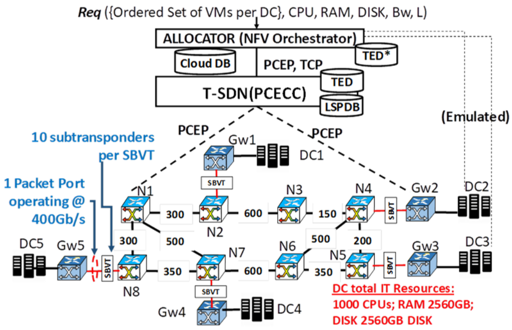

Fig. 1: Cloud/Network Orchestrator Architecture

Cloud/Network Orchestrator Architecture

Fig.1 shows the two building blocks forming the Cloud/Network Orchestrator: the *Allocator* and the T-SDN controller. The Allocator receives, computes and dynamically accommodates VNFFG requests (*req*). Each received *req* describes an ordered set of DCs specifying for each DC the required IT resources: number of Virtual Machines (VMs) and IT capabilities such as CPU, RAM and DISK. Additionally, a *req* also specifies the required packet inter-DC connectivity in terms of bandwidth (*Bw*, b/s) and maximum tolerated latency (*L*, ms). The Allocator keeps track of all DC resources via the Cloud database (DB). Hence, the Allocator checks for each *req* whether it can be accommodated with regard to the IT resources. The Allocator interfaces the T-SDN controller (depending on the adopted approach) to request inter-DC path computations, to retrieve network information and to instantiate inter-DC connections.

The T-SDN Controller supports the following functions: processing packet connection requests from the Allocator (via a NBI API that uses the PCEP protocol), performing path computations, and configuring the computed packet and optical network elements (i.e., switches and SBVTs).

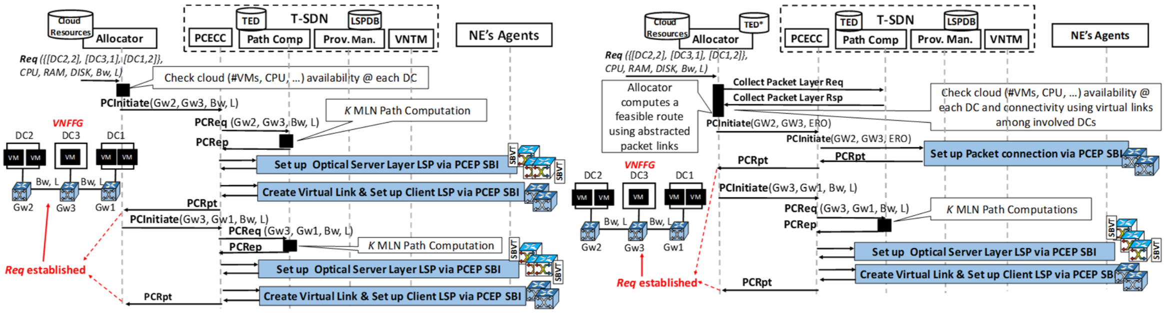

Fig. 2: Workflows for VNFFG establishment: (left) *NNI* approach; (right) *ANI* approaches

As shown in Fig. 1, each DC is connected to a packet node (Gateway, Gw) which via Sliceable Bandwidth Variable Transponders (SBVT) connects to a flexi-grid network for provisioning the required packet inter-DC connections.

On-line Orchestration Approaches

In the NNI approach, as mentioned, the Allocator only performs DC resource configuration delegating the inter-DC connectivity to the T-SDN controller. On the other hand, in the ANI approach, the Allocator is granted, via the T-SDN controller, with *abstracted* view of the network topology exclusively related to the packet virtual links created upon setting up flexi-grid optical path segments[3]. Created virtual link information is passed from the T-SDN controller to the Allocator via a TCP protocol and stored in the Allocator's TED*. This includes the unreserved bandwidth, aggregated cost and accumulated (propagation) delay. Hence, the Allocator can compute packet inter-DC paths for the received VNFFG *req*. If the Allocator cannot find a feasible path (e.g., no DC connectivity or insufficient available bandwidth), it queries such a path computation to the T-SDN controller.

Fig. 2 details the workflows (i.e., functions and exchanged control messages) to set up a received VNFFG *req*. Specifically, Fig. 2(left) deals with the NNI approach. If the *req* demands an amount of IT resources that cannot be satisfied (e.g., in a particular DC), the Allocator blocks the *req*. Otherwise, the Allocator decomposes the *req* in a set of packet inter-DC connections (*Pkt_LSPis*) to be set up. For a *req* involving *N* DCs, *N-1* unidirectional *Pkt_LSPis* need to be established. To do this, the Allocator sends respective PCEP PCInitiate messages to the T-SDN controller. Each message carries the endpoints DCs' Gw nodes, the required bandwidth (*Bw*) and maximum latency (*L*). The T-SDN controller computes a MLN path using the TED information (updated via BGP-LS protocol running at every node's agent). If a *Pkt_LSPi* computation succeeds, this is established using extended PCEP control connection capabilities[3]. In the considered MLN, a computed path may combine new flexi-grid optical path segments and existing virtual links with sufficient spare capacity. Once a *Pkt_LSPi* path is established, this is reported to the Allocator via a PCEP PCRpt message. If not, a PCError is sent. Notice that a *req* is successfully completed when all *Pkt_LSPis* are established. Otherwise, the *req* is blocked releasing all allocated IT and network resources.

Fig. 2(right) addresses the workflow for serving a new VNNFG *req* for the ANI approach. After checking that the required IT resources can be allocated, the Allocator triggers the computation for each *Pkt_LSPi*. To this end, it first retrieves (via a TCP protocol) an updated view of the existing packet virtual links which are then stored in the Allocator's TED* repository. Next, for each *Pkt_LSPi* a Constrained Shortest Path First (CSPF) algorithm based on Dijkstra is triggered aiming at fulfilling the demanded *Bw* and *L* requirements. If a feasible path is found, the *Pkt_LSPi* is instantiated sending a PCInitiate message to the T-SDN controller with the computed path (Explicit Route Object, ERO). Otherwise, the *Pkt_LSPi* path computation and its establishment is asked to the T-SDN controller likewise the NNI approach. As above, if all the *Pkt_LSPs* are set up, the *req* is successfully completed.

The path computation at the T-SDN controller relies on a modified Yen algorithm providing *K-CSPF* calculations. Specifically, for a k^{th} iteration, an on-line MLN path computation proposed in[3] is used. This path computation provides the shortest path cost satisfying the *Bw* requirement where both electrical and optical grooming opportunities are fostered. To do that, both existing virtual packet links and optical flexi-grid links are considered to output a feasible path. For the latter, the MLN path computation adopts a distance-adaptive mechanism where different modulation formats (*MFs*) supported by the SBVTs are explored. The MLN path computation prioritizes selecting the most advanced *MF* as long as the maximum permitted distance (in km) is not exceeded[3]. The path computation also tackles additional technological restrictions such as the packet port capacity at the Gw nodes (400Gb/s) and the spectrum continuity and contiguity constraints in the optical domain.

The K-CSPF algorithm sorts the computed k^{th} MLN paths with respect to their total cost. Paths with the same cost are sorted by the lowest latency. The first resulting computed MLN path satisfying the latency restriction is chosen.

Experimental Performance Evaluation

The experimental performance evaluation of the proposed approaches is conducted within the CTTC ADRENALINE testbed rolling out a cloud/network infrastructure as shown in Fig. 1. This infrastructure is formed by 5 (emulated) DCs linked to 5 Gw nodes. Each Gw node has a single port (operating at 400Gb/s) and is connected to the optical flexi-grid network via an SBVT with 10 subtransponders. Each subtransponder can use 3 different MFs (i.e., DP-16QAM, DP-8QAM and DP-QPSK) enabling 3 respective bit rates (200, 150 and 100 Gb/s) for different maximum distances (650, 1000 and 3000 km)[3]. Optical links support 128 Nominal Central Frequencies spaced 6.25GHz. Optical fiber distances, indicated in Fig. 1, are used for exploring MF during the K-CSPF computation and determines the accumulated propagation path delay needed for checking the latency restriction.

Fig.3.a. depicts the captured PCEP control messages exchanged among the Allocator, the T-SDN controller and the network nodes' agents. It shows: the Allocator's PCEP message sent to the T-SDN controller (step 1) requesting a *Pkt_LSPi*; the T-SDN controller PCEP messages to the network nodes' agents for setting up a flexi-grid optical connection (via PCLabelUpd, msg type 30 in step 2). This optical path eventually derives on a packet virtual link used to accommodate a packet inter-DC LSP (step 3). Step 4 deals with the response at the Allocator.

The assumptions for dynamically generating the VNFFGs *reqs* are gathered in Fig.3.b. Finally, Fig. 3.c. shows the obtained results for both approaches in terms of the successful (in %) established *reqs*, average setup time, average consumed IT resources (CPU, RAM and DISK) in all DCs, total amount of created virtual links and average number of used subtransponders.

In general, both approaches behave similarly in terms of the successful *req* establishment. As expected, as we increase the *req* duration (HT), the successful establishment is worsened. The main reason for that is due to the Gw's packet port capacity. Indeed, it represents the main limitation and, as HT increases, it becomes exhausted thus leading to not accommodating *Pkt_LSPs* of a *req*. That said, it is worth pointing out that adopting the ANI model slightly improves the successful served *req* compared to the NNI. In the ANI, the Allocator better exploits the packet grooming. This in turn does improve the network resource usage in terms of the packet port capacity and SBVT subtransponders. Consequently, since more VNFFG *reqs* are served, larger amount of IT resources is consumed at the DCs. Such an improvement, however, is attained at the expenses of increasing the average setup time (more than 50% wrt NNI). The rationale behind this is that the path computation at the Allocator for the ANI requires multiple iterations with the T-SDN to collect the existing virtual links for computing each *Pkt_LSPi*.

Conclusions

The implementation of a Cloud/Network Orchestrator (Allocator and T-SDN controller) for dynamically serving quality-enabled VNFFG requests in a MLN is experimentally evaluated. The networking capabilities handled by each of these Orchestrator elements lead to propose two approaches which are benchmarked with respect to a number of performance metrics. The results show that having an abstracted network view at the Allocator allows attaining a better use of the overall resources, and may enable adopting more effective VNF placement strategies.

Acknowledgements

This work is partially funded by the Spanish MINECO DESTELLO project (TEC2015-69256-R) and the EU H2020 5G TRANSFORMER project (761536).

References

[1] M. Zeng, et. al., "Orchestrating Tree-Type VNF Forwarding Graphs in inter-DC Elastic Optical Networks", J. Lightwave Technol., Vol. **34**, no. 14, p. 3330 (2016).

[2] A. Farrel, Q. Zhao, Ed. "An Architecture for Use of PCE and the PCE Communication Protocol (PCEP) in a network with Central Control", IETF RFC8283, Dec. 2017.

[3] R. Martínez et al "Experimental Evaluation of a PCE Transport SDN Controller for Dynamic Grooming in Packet over Flexi-Grid Optical Networks" ECOC2017.

Source	Destination	Protocol	Info	
ALLOCATOR	T-SDN	PCEP	Path Computation LSP Initiate (PCInitiate)	1
T-SDN	OPTICAL N8	PCEP	Path Computation LSP Initiate (PCInitiate)	2
T-SDN	OPTICAL N8	PCEP	Unknown Message (30).	
T-SDN	OPTICAL N7	PCEP	Unknown Message (30).	
T-SDN	OPTICAL N6	PCEP	Unknown Message (30).	
T-SDN	OPTICAL N4	PCEP	Unknown Message (30).	
OPTICAL N8	T-SDN	PCEP	Path Computation LSP State Report (PCRpt)	
T-SDN	PACKET GW5	PCEP	Path Computation LSP Initiate (PCInitiate)	3
T-SDN	PACKET GW5	PCEP	Unknown Message (30).	
T-SDN	PACKET GW2	PCEP	Unknown Message (30).	
PACKET GW5	T-SDN	PCEP	Path Computation LSP State Report (PCRpt)	
T-SDN	ALLOCATOR	PCEP	Path Computation LSP State Report (PCRpt)	4
ALLOCATOR	T-SDN	PCEP	Close	

a.

VNFFG *req* Dynamic Generation	
Description	Ordered set of DCs allocating different VMs with IT (CPU, RAM, DISK) and Network (Bandwidth, Latency) demands
Num. *reqs*	1000; ***Poisson*** (mean inter-arrival time: 25s); ***Exponential*** duration (Holding Time, HT) varied from 25, 50 and 100s
Num. DCs per *req*	Uniformly distributed [2, 5]
Num. VMs per DC*i* in a *req*	Uniformly distributed [1, 5]
IT resources per VM	Uniformly distributed: CPU [1, 50]; RAM & DISK [4, 128]
Network Resources per *Pkt_LSPi*	***Bandwidth*** demand uniformly distributed: 10, 40 and 100 Gb/s; ***Latency*** uniformly distributed [6, 12] ms

b.

	no_network_info			***abstracted_network_info***		
HT (s)	**50**	**75**	**100**	**50**	**75**	**100**
%Successful	90.3	87.4	83.9	91.0	87.3	84
Setup time (ms)	4.0	3.1	2.4	8.26	7.4	7.0
Avg. used CPU	77.79	121.1	158.4	78.8	117.19	152.8
Avg. used RAM	192.06	306.6	398.4	202.2	303.6	381.7
Avg. used DISK	191.16	304.4	397.4	199.2	294.8	383.3
Num. Vlinks	1869	1412	1089	1999	1433	1137
Av. Used Subtransponder	0.88	1.21	1.47	0.91	1.18	1.39

c.

Fig. 3: a. VNNFG *req* generation details; b. PCEP messages setting up a new packet inter-DC LSP; c. Performance results