Demonstration of Latency-Aware and Self-Adaptive Service Chaining in 5G/SDN/NFV infrastructures

M. Gharbaoui*, C. Contoli‡, G. Davoli‡, G. Cuffaro*, B. Martini*, F. Paganelli*, W. Cerroni‡, P. Cappanera †, P. Castoldi§
*CNIT, Italy ; †University of Florence, Italy ; ‡DEI - University of Bologna, Italy ; §Scuola Superiore Sant'Anna, Pisa, Italy

***Abstract*—This paper will demonstrate an orchestration system for 5G infrastructure supporting latency-minimized and self-adaptive service chaining over geographically distributed edge clouds interconnected through SDN. The demo will show an orchestration system that comprises dynamic virtual function selection and intent-based traffic steering control functionalities to provide optimized service chains in terms of end-to-end latency and adjustable with respect to the context (e.g., service dynamics, network status).**

***Index Terms*—SDN; NFV; orchestration; service chaining.**

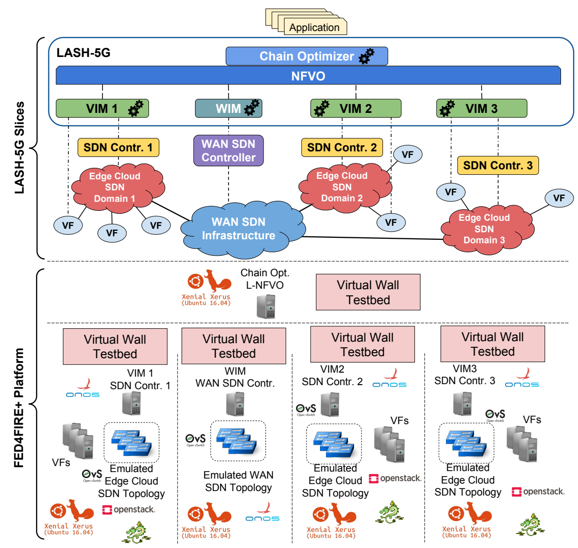

Fig. 1. Reference architecture and LASH-5G deployment on Fed4FIRE+.

I. Introduction

With the advent of SDN, NFV and the comprehensive 5G context, new applications can be conceived (e.g., cloud robotics, smart cities) by dynamically composing (i.e., chaining) computational and network services (e.g., robot balancers, traffic accelerators) deployed as virtual functions (VFs) in distributed micro-clouds located at the network Edge. However, the heterogeneity of the infrastructures, the high dynamicity of services and the geographical distribution of VFs pose new challenges in terms of multi-technology resource control/management capabilities, adaptive usage of resources, and fulfillment of end-to-end latency requirements considering both processing and network delays [1].

This demo will show a latency-aware and self-adaptive service chaining orchestration system for 5G environments (i.e., LASH-5G) able to exploit heterogeneous resource control/management capabilities offered by cloud and network domains to dynamically provide service chains while (i) optimally selecting VFs over the path that minimizes the offered end-to-end latency across the clouds and the network resource domains; and (ii) promptly adapting established service chains to the current context offered by multi-technology resource domains (e.g., network load causing different network paths to be used to connect VFs, service dynamics causing different VFs to be included in the chain). To optimally select VFs and to promptly trigger adaptations, LASH-5G orchestration system exploits monitoring data that are collected from the underlying cloud and network domains so that decisions for VF selection and path adaptation are taken by considering the current status of the infrastructure resources. The LASH-5G system has been developed to be compliant with ETSI MANO framework [2] and put into operation on top of the Fed4FIRE experimentation infrastructure provided within the Fed4FIRE+ Horizon 2020 Project [3].

II. System deployment

The top part of Fig. 1 shows the SDN/NFV setup we reproduced to run LASH-5G. It comprises three Edge Cloud SDN Domains where VFs are deployed by means of a cloud-computing platform (e.g., OpenStack) and where SDN is used as network control technology. An SDN Wide Area Network (WAN) infrastructure domain is assumed to provide inter-DC connectivity. Each SDN domain adopts its own controller, which is assumed to provide a fully-featured northbound interface (e.g., via REST API) that allows to dynamically program traffic flow steering rules across network nodes. The LASH-5G orchestration system lies on top of the SDN controllers and includes three main components: (i) ***Chain Optimizer*** that optimally selects service chain components among a set of VF instances available from different locations and over the path that minimizes the end-to-end latency considering both processing delays and network latency information [4]; (ii) ***WAN Infrastructure Manager (WIM)*** enhanced with orchestration capabilities able to provide the programmable provision of service chain paths by means of an intent-based northbound REST API and to adapt them based on current network context (e.g., switch load) to prevent service data delivery degradations [5]; (iii) ***Virtual Infrastructure Manager (VIM)*** enhanced with orchestration capabilities able to expose an intent-based northbound REST API that allows to specify a service chain by means of a high-level descriptive syntax, agnostic to the specific SDN technology adopted. This API allows service chains to be established and dynamically modified to adapt them to the current context of users or services (e.g., current location of users) or to varying needs of the service provider (e.g., different resource management policy) [6].

The bottom part of Fig. 1 shows the deployment of the above reference scenario on top of the Fed4FIRE+ platform. In particular, we deployed 5 *experiment containers*, i.e., slices of resources, from the Virtual Wall testbed facility [7] to run LASH-5G components as well as SDN and cloud domains. More specifically, we dedicated (i) 1 slice to put in operation the Chain Optimizer and a lightweight version of the NFVO (L-NFVO) we developed to perform specific tasks needed to run LASH-5G, i.e., translation of forwarding graph descriptors generated by the Chain Optimizer into suitable intents for VIMs/WIM (encoded as JSON files), and provision of processing and network latency information obtained by WIM and VIMs; (ii) 1 slice to deploy the WAN domain composed of 5 OvS switches, the ONOS SDN controller and WIM; (iii) three slices to deploy SDN-based cloud domains each hosting an Openstack cluster composed of (a) compute nodes where VF instances are deployed and respective cloud and network controllers, (b) an instance of OvS representing the emulated SDN topology among compute nodes and controlled by a ONOS controller, (c) a VIM instance, (d) an egress router connected to the SDN WAN slice through VXLAN tunnels.

III. Demo Workflows

In this demo, we show the capability of LASH-5G in (i) deploying service chains while optimally selecting VFs over the path that minimizes the overall end-to-end latency (i.e., *service chains deployment*), and (ii) promptly adapting established service chains to the current context offered by the cloud and the network domains (e.g., network load, user demands, operator needs), i.e., *service chains adjustment*. The demo workflows are described in Fig.2 and will be shown during the demo through a set of GUIs provided with the orchestration system.

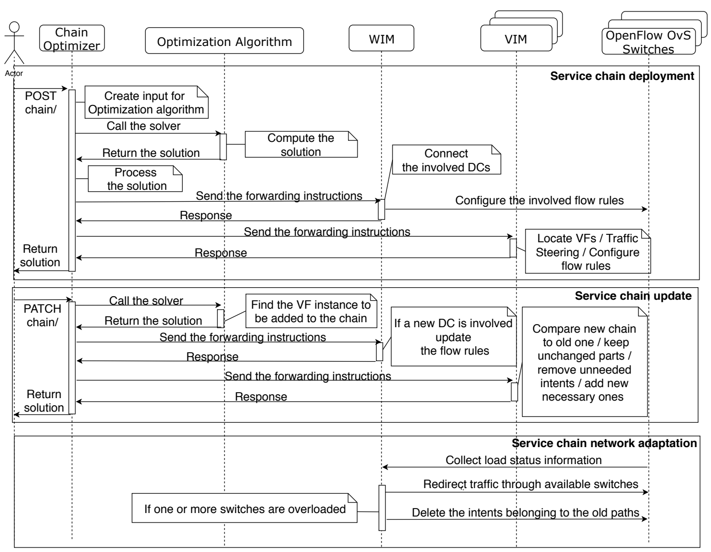

Fig. 2. Demo Workflow.

A. Service chains deployment

The demo starts by generating 5 different service chain requests with different requirements in terms of bandwidth and maximum latency and sending them to the Chain Optimizer through a CO-Client GUI. The Chain Optimizer handles each request and it computes a solution of an optimization problem by leveraging a CPLEX solver. If this procedure is successful, a response is returned to the CO-Client GUI with the solution (set of cloud domains hosting the selected VFs to be connected while minimizing end-to-end latency) along with the computed end-to-end latency and computation time. The Chain Optimizer then sends the forwarding instructions (ordered sequence of VFs to connect) to each involved VIM. The VIMs are responsible of discovering in which compute nodes the VFs instances with minimum processing latency are located in order to properly enforce traffic steering by consistently configuring the flow rules. The Chain Optimizer also sends the forwarding instructions (set of cloud domains to be connected across the WAN) to the WIM. The WIM computes the network path(s) and, accordingly, sets-up the forwarding rules in the involved switches.

B. Service chains adjustment

The second part of the demo consists in showing the adaptive capability of LASH-5G in dynamically adjusting established service (i.e., VF) chains. Firstly, the updating of two service chains out of the established ones is shown (i.e., *service chain update*). The requested update consists in adding one new VF in the chain, while keeping the rest of the chain unchanged. In order to trigger such a chain update, a CO client sends an update request by specifying the id of the chain, the new VF type and its ordered position in the chain. The Chain Optimizer handles the request by invoking the optimization algorithm to find the VF instance to be added to the chain, taking into account how the pre-existing chain has been deployed. The Chain Optimizer then processes the algorithm output to provide appropriate instructions to VIMs and WIM for updating the chain accordingly. Specifically, WIM is updated whenever the updated chain needs to traverse an Edge cloud data center that was not involved in the original chain deployment, while the VIMs process update requests by comparing the new chain against the old one: parts of the chain that remain unchanged are kept as they are, parts of the chain that are no longer needed are removed, while new parts of the chain are added. Secondly, the adaptation feature offered by LASH-5G is shown with respect to the network status in the SDN WAN (i.e., *service chain network adaptation*). In this case, the WIM comes into play by adapting the network paths connecting Edge cloud domains and underpinning the VF chain path segments with respect to the load status information of switches/links derived from a selective set of monitoring data (e.g., data throughput). During the demo, we show an example in which after the deployment of a service chain request, a subset of the switches in the WAN SDN domain becomes overloaded which triggers the dynamic adaptation capability, thus redirecting the traffic through other available switches.

IV. Conclusions

In this demo we will show a service chaining orchestration system comprising dynamic VF selection and intent-based traffic steering control capabilities to enhance MANO framework components with latency-minimized and self-adaptive service chaining features over geographically distributed SDN-based cloud DCs interconnected through SDN WAN. We will also demonstrate that the times necessary for the setup and update of the service chains do not exceed few seconds.

References

[1] S. Zhang *et al.*, “5g: Towards energy-efficient, low-latency and high-reliable communications networks,” in *2014 IEEE International Conference on Communication Systems*, Nov 2014, pp. 197–201.

[2] G. ETSI, “Network functions virtualisation (nfv): Architectural framework,” *ETSI GS NFV*, vol. 2, no. 2, p. V1, 2013.

[3] https://www.fed4fire.eu/.

[4] B. Martini *et al.*, “Latency-aware composition of virtual functions in 5g,” in *Proceedings of IEEE NetSoft 2015*, April 2015, pp. 1–6.

[5] A. Mohammed *et al.*, “Sdn controller for network-aware adaptive orchestration in dynamic service chaining,” in *Proceedings of IEEE Netsoft 2016*. IEEE, 2016, pp. 126–130.

[6] F. Callegati *et al.*, “Performance of intent-based virtualized network infrastructure management,” in *Proceedings of IEEE ICC*, May 2017.

[7] Virtual wall web site. https://www.fed4fire.eu/virtual-wall/.