NFV Research Group
Internet-Draft
Intended status: Informational
Expires: October 12, 2017

G. Bernini
G. Landi
Nextworks
D. Lopez
Telefonica
P. Aranda Gutierrez
UC3M

April 10, 2017

VNF Pool Orchestration For Automated Resiliency in Service Chains
draft-bernini-nfvrg-vnf-orchestration-04

Abstract

Network Function Virtualisation (NFV) aims at evolving the way network operators design, deploy and provision their networks by leveraging on standard IT virtualisation technologies to move and consolidate a wide range of network functions and services onto industry standard high volume servers, switches and storage. The primary target for operators, stimulated by the recent updates on NFV and SDN, is the network edge. In fact, operators are considering their future datacentres and Points of Presence (PoPs) as increasingly dynamic infrastructures where Virtualised Network Functions (VNFs) and on-demand chained services with high elasticity will be deployed.

This document presents an orchestration framework for automated deployment of highly available VNF chains. Resiliency of VNFs and chained services is a key requirement for operators to improve, ease, automate and speed up services lifecycle management. The proposed VNFs orchestration framework is also positioned with respect to current NFV and Service Function Chaining (SFC) architectures and solutions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 12, 2017.

Table of Contents

1. Introduction

Current Telco infrastructures are facing the rapid development of the cloud market, which includes a broad range of emerging virtualised services and distributed applications. Network Function

Virtualisation (NFV) is gaining wide interest across operators as a means to evolve the way networks are operated and provisioned, with network functions and services traditionally integrated in hardware devices executed in virtualised environments.

A Virtualised Network Function (VNF) provides the same function as its non virtualised equivalent (e.g. firewall, load balancer) but is deployed as a software instance running on general purpose servers using virtualisation technologies. The main idea, therefore, is to run network functions in datacentres or commodity network nodes that are, in some cases, close to the end user premises. With NFV, network functions are moved from specialised hardware devices to self-contained virtual machines running in general purpose servers. These virtualised functions can be deployed in multiple instances or moved to various locations in the network, adapting themselves to traffic dynamicity and customer demands without the overhead cost and management of installing new equipment.

Operator networks are populated with a large and increasing variety of proprietary software and hardware tools and appliances. The deployment of new network services in operational environments is often a complex and costly procedure, where additional physical space and power are required to accommodate new boxes. Additionally, current hardware-based appliances rapidly reach end of life. This requires that much of the design integration and deployment cycle be repeated with little revenue benefit. In this context, the transition of network functions and appliances from hardware to software solutions by means of NFV promises to address and overcome these hindrances for network operators.

The considerations above are valid for stand-alone VNFs running independently. However, additional challenges and requirements raise for network operators when services offered to customers are built by the composition of multiple VNFs. In this case, the deployment and provisioning of each (virtual) service component for the customer needs to be coordinated with the other VNFs, applying control functions to steer the traffic through them following a predefined order (i.e. according to the specific service function path). An orchestration framework capable of coordinating the automated deployment, configuration, provisioning and chaining of multiple VNFs would ease the management of the whole lifecycle of services offered to customers. Additionally, when dealing with virtualised functions, resiliency and high availability of chained services pose additional requirements for a VNF orchestration framework, in terms of detection of software failures at various levels (including hypervisors and virtual machines, hardware failure), and dynamic and intellingent reaction (virtual appliance migration, deployment of new VNFs, re-adapt the VNF chain).

This document presents an orchestration framework for automated deployment of high available VNF chains, and introduces its architecture and building blocks. Resiliency for both stand-alone VNFs and chained services is considered in this document as a key control function based on VNF pool concepts. The proposed VNF pool orchestration framework is also positioned with respect to approaches and architectures currently defined for Network Function Virtualisation and Service Function Chaining (SFC).

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following acronyms are used in this document:

NFV: Network Function Virtualisation.

SDN: Software Defined Networking.

VNF: Virtualised Network Function.

SFC: Service Function Chaining.

CPE: Customer Premise Equipment.

VPN: Virtual Private Network.

EMS: Element Management System.

PoP: Point of Presence.

VM: Virtual Machine.

3. VNF Pool Orchestration for Resilient Virtual Appliances

The telco market is rapidly moving towards an "Everything as a Service" model, where the virtualisation of traditionally in-the-box network functions can benefit from Software Defined Networking (SDN) tools and technologies. As said, the recent updates and proposed solutions on NFV and SDN is practically bringing, from an operator perspective, a deep evolution on how the network edge is architected, operated and provisioned, since that is the place where VNFs and virtual services can be deployed and provisioned close to the customers. Operators target to evolve their datacentres and PoPs into increasingly dynamic infrastructures where VNFs and chained services can be deployed with high availability and high elasticity

to scale up and down while optimizing performances and resources utilization.

This section introduces the VNF pool orchestration framework for the deployment, provisioning and chaining of resilient virtual appliances and services within operator data-centres proposed in this document.

3.1. Problem Statement

The orchestration framework proposed in this document aims at solving some of the challenges that operators face when, trying to apply the base NFV concepts, they replace hardware devices implementing well-known network functions with software-based virtual appliances. In particular, this VNF orchestration framework targets an automated, flexible and elastic provisioning of service chains within operators' datacentres.

When operators need to compose and chain multiple VNFs to provision a given service to the customer, they need to operate network and computing resources in a coordinated way, and above all to implement control mechanisms and procedures to steer the traffic through the different VNFs and the customer sites. As an example, the virtualisation of the Customer Premises Equipment (CPE) is emerging as one of the first applications of the Network Functions Virtualisation (NFV) architecture that is currently being commercialized by several software (and hardware) vendors. It has the potential to generate a significant impact on the operators businesses. The term virtual CPE (vCPE) refers to the execution in a virtual environment of the network functions that traditionally integrated in hardware gears at customer premises, like BGP speakers, firewall, NAT, etc.

Different scenarios and use cases exist for the vCPE. Currently, the typical scenario is the vCPE in the PoP, that actually provides softwarization and shift to the first PoP of the operator for those network functions normally deployed at customer premises (e.g. NAT, Firewall, etc.). The goal is to manage the whole LAN environment of the customer, while preserving QoE of its services, providing added value services to users not willing to get involved with technology issues, and reducing maintenance trouble tickets and the need for in-house problem solving. In addition, the vCPE can be used in the operator's datacenter to implement in software those chained network functions to provision automated VPN services for customers [VCPE-T2], in order to dynamically and automatically extend existing L3 VPNs (e.g. connecting remote customer sites) to incorporate new virtual assets (like virtual machines) into a private cloud.

As additional requirements for the proposed orchestration framework, the use of VNFs opens new challenges concerning the reliability of provided virtual services. When network functions are deployed on monolithic hardware platforms, the lifecycle of individual services is strictly bound to the availability of the physical device, and management tools may detect outages and migrate affected services to new instances deployed on backup hardware. When introducing VNFs, individual network functions may still fail, but with more risk factors such as software failure at various levels, including hypervisors and virtual machines, hardware failure, and virtual appliance migration. Moreover, when considering chains of VNFs, the management and control tools used by the operators have to consider and apply reliability mechanisms at the service level, including transparent migration to backup VNFs and synchronization of state information. In this context, VNF pooling mechanisms and concepts are valid and applicable, thus considering VNF instances grouped as pools to provide the same function in a reliable way.

3.2. Orchestration Framework

The VNF pool orchestration framework proposed in this document aims to provide automated functions for the deployment, provisioning and composition of resilient VNFs within operators' datacentres. Figure Figure 1 presents the high level architecture, including building blocks and functional components.

This VNF pool orchestration framework is built around two key components: the orchestrator and the SDN controller. The orchestrator includes all the functions related to the management, coordination, and control of VNFs instantiation, configuration and composition. It is the component at the highest level of the architecture and represents the access point to the VNF pool orchestration framework for the operator. On the other hand, the SDN controller provides dynamic traffic steering and flexible network provisioning within the datacenter as needed by the VNF chains. The basic controller functions are augmented by a set of enhanced network applications deployed on top, that might be themselves control and management VNFs for operator use (i.e. not related to customers and users functions).

Therefore, the architecture depicted in Figure Figure 1 is a practical demonstration of how SDN and NFV technologies and concepts can be integrated to provide substantial benefits to network operators in terms of robustness, ease of management, control and provisioning of their network infrastructures and services. SDN and NFV are clearly complementary solutions for enabling virtualisation of network infrastructures, services and functions while supporting dynamic and flexible network traffic engineering.

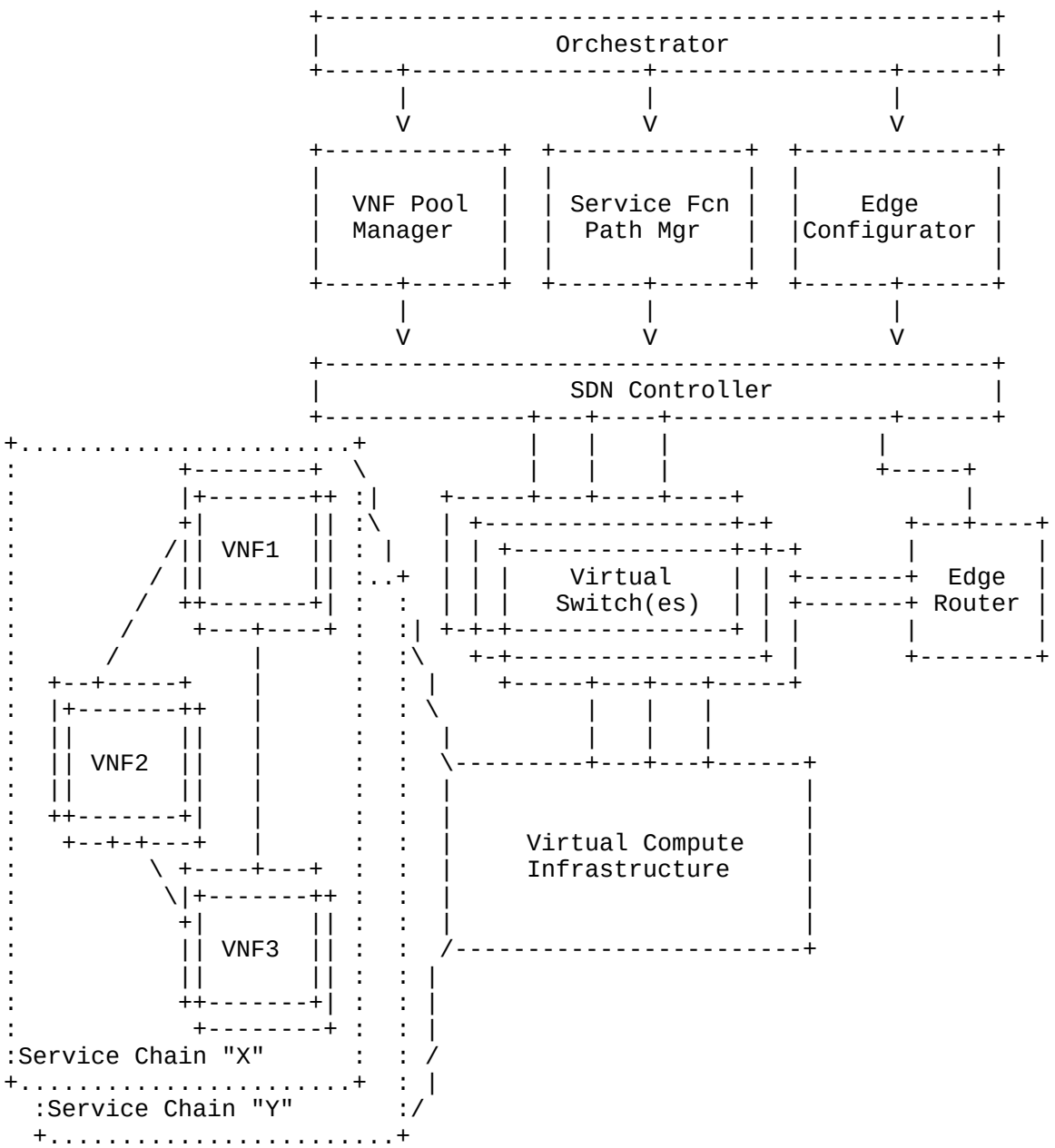

Figure 1: VNF Pool Orchestration Framework Architecture.

SDN focuses on network programmability, traffic steering and multi-tenancy by means of a common, open, and dynamic abstraction of network resources. NFV targets a progressive migration of network elements, network appliances and fixed function boxes into VMs that can be ran on commodity hardware, enabling the benefits of cloud and datacentres to be applied to network functions.

3.2.1. Orchestrator

The VNF orchestrator implements a set of functions to seamlessly control and manage in a coordinated way the instantiation and deployment of VNFs on one hand, and their composition and chaining to steer the traffic through them on the other. It is fully controlled and operated by the network operator, and basically it is the highest control and orchestration layer that sits above all the softwarized and virtualised components in the proposed architecture.

Therefore the VNF orchestrator provides a consistent way to access the system and provision chains of VNFs to the operator. It exposes a set of primitives to instantiate, configure VNFs and compose them according to the specific service chain requirements. Practically, it aims at enabling an efficient and dynamic management of operator’s infrastructure resources with great flexibility by means of a consistent set of APIs.

To enable this, the VNF orchestrator can be seen as a composition of several internal functionalities, each providing a given coordination function needed to orchestrate the lower layer control and management functions depicted in Figure Figure 1 (i.e. VNF chain configuration, VNF pool provisioning, etc.). In practice, the VNF orchestrator needs to include at least an internal component to manage the instantiation and configuration of stand-alone VNFs (e.g. implemented by a self-contained VM) that might be directly interfaced with the physical servers in the datacenter. And also a dedicated component for programmatic coordination and provisioning of VNF chains is needed to properly orchestrate the traffic steering through VNFs belonging to the same service chain. This should also provide multi-tenant functionalities and maintain isolation across VNF chains deployed for different customers. It is then clear that the VNF orchestrator is the overall coordinator of the proposed framework, and it drives all the lower layer components that implement the actual control logic.

3.2.2. SDN Controller

The SDN controller provides the logic for network control, provisioning and monitoring. It is the component where the SDN abstraction happens. This means it exposes a set of primitives to

configure the datacenter network according to the requirements of the VNF chains to be provisioned, while hiding the specific technology constraints and capabilities of the software switches and edge routers underneath. The deployment of an SDN controller allows to implement a software driven VNF orchestration, with flexible and programmable network functions for service chaining and resilient virtual appliances.

At its southbound interface, the SDN controller interfaces with software switches running in servers, physical switches interconnecting them and edge routers connecting the datacenter with external networks. Multiple control protocols can be used at this southbound interface to actually provision the datacenter network and enable traffic steering through VNFs, including OpenFlow, OVSDB, NETCONF and others.

Therefore the SDN controller provides the basic network provisioning functions needed by upper layer coordination functions to perform service chain and VNF pool-wide actions. Indeed, the logic and the state at the service level is only maintained and coordinated by network applications on top of the SDN controller.

3.2.3. Service Function Path Manager

The service function path manager is deployed as a bridging component between the orchestrator and the SDN controller, and it is mostly dedicated to the implementation of VNF chaining and composition logic. It computes a suitable path to interconnect the involved VNFs (already instantiated and identified by the orchestrator) and forwards the network configuration request to the SDN controller for each new VNF chain requested by the orchestrator.

Following the datacenter service chains and related traffic types defined in [I-D.ietf-sfc-dc-use-cases], the service function path manager should implement its coordination logic to support both north-south and east-west chains. The former refer to network traffic staying within the datacenter but coming from a remote datacenter or a user through the edge router connecting to an external network. In this case, the service function path manager should also coordinate with the edge configurator to properly provision the datacenter edge router. Moreover, this north-south case may also refer to VNF chains spanning multiple datacentres, thus requiring a further inter-datacenter coordination between service function path managers and orchestrators. These coordination functions are out of the scope of this document. On the other hand, the east-west chains refer to VNFs treating network traffic that do not exit the datacenter. For both cases, the service function path manager (in combination with the SDN controller) should implement and

support proper service chains encapsulation solutions [I-D.ietf-sfc-nsh] to isolate and segregate traffic related to VNF chains belonging to different tenants.

Different deployment models may exist for the service function path manager: a dedicated configurator for each chain, or a single configurator for all the VNF chains. In the first approach, the orchestrator needs to implement some coordination logic related to the dynamic instantiation of configurators when new VNF chains are provisioned.

3.2.4. Edge Configurator

The edge configurator is a network control application deployed on top of the SDN controller. Its main role is to coordinate the provisioning and configuration of the edge router for those north-south VNF chains exiting the datacenter. In particular, it keeps the binding between the traffic steered through the VNFs and the related network service outside the datacenter terminated at the edge router (e.g. a L3 VPN, VLAN, VXLAN, VRF etc), possibly considering the service chain encapsulation implemented within the VNF chain. The mediation of the SDN controller allows to support a variety of control and management protocols for the actual configuration of the datacenter edge router.

3.3. Resiliency Control Functions for Chained VNFs

In the proposed orchestration architecture, the resiliency control functions that have been identified as a key feature for a flexible and dynamic provisioning of chained VNF services are implemented by the VNF pool manager depicted in Figure Figure 1. It is the entity that manages and coordinates VNFs reliability providing high availability and resiliency features at both stand-alone and chained VNFs level.

The deployment of VNF based services requires moving the resiliency capabilities and mechanisms from physical network devices (which are typically highly available and often specialized) to entities (like self-contained VMs) running VNFs in the context of pools of virtualised resources. When moving towards a resilient approach for VNF deployment and operation, in line with ETSI NFV Resiliency Requirements (NFV-REL001), the generic high availability requirements to be matched are translated into:

Service continuity: when a hardware failure or capacity limits (memory and CPU) occur on platforms hosting VMs (and therefore VNFs), it is necessary to migrate VNFs to other VMs and/or

hardware platforms to guarantee service continuity with minimum impact on the users

Topological transparency: the hand-over between live and backup VNFs must be implemented in a transparent way for the user and also for the service chain itself. The backup VNF instances need to replicate the necessary information (configuration, addressing, etc.) so that the network function is taken over without any topological disruption (i.e. at the VNF chain level)

Load balancing or scaling: migration of VNF instances may also happen for load-balancing purposes (e.g. for CPU, memory overload in virtualised platforms) or scaling of network services (with VNFs moved to new hardware platforms). In both cases the working network function is moved to a new VNF instance and the service continuity must be maintained.

Auto scale of VNFs instances: when a VNF requires increased resource allocation to improve overall service performance, the network function could be distributed across multiple VMs, and to guarantee the performance improvement dedicated pooling mechanisms for scaling up or down resources to each VNF in a consistent way are needed.

Multiple VNF resiliency classes: each type of end-to-end service (e.g. web, financial backend, video streaming, etc.) has its own specific resiliency requirements for the related VNFs. While for operators it is not easy to achieve service resiliency SLAs without building to peak, a basic set of VNF resiliency classes can be defined to identify some metrics, such as: if a VNF needs status synchronization; fault detection and restoration time objective (e.g. real-time); service availability metrics; service quality metrics; service latency metrics for VNF chain components.

The aim of the VNF pool orchestration presented in this document is to address the above requirements by introducing the VNF pool manager that follows the principles of the IETF VNFPOOL architecture [I-D.zong-vnfpool-arch], where a pool manager coordinates the reliability of stand-alone VNFs, by selecting the active instance and interacting with the Service Control Entity for consistent end-to-end service chain reliability and provisioning. In the VNF pool orchestration architecture illustrated in Figure Figure 1, the Service Control Entity is implemented by the combination of the orchestrator (for overall coordination of service chains) and the VNF chain configuration (for actual provisioning and coordination of individual service chains).

Different deployment models may exist for the VNF pool manager: a dedicated manager for each VNF chain, or a single one for all the chains.

In terms of offered resiliency functionalities, the VNF pool manager provides some post-configuration functions to instantiate VNFs (as self-contained VMs) with the desired degree of reliability and redundancy. This translates into further actions to create and configure additional VMs as backups, therefore building a pool for each VNF in the chain.

The VNF pool manager is conceived to offer several types and degrees of reliability functions. First, it provides specific functions for the persistence of VNFs configuration, including making periodic snapshots of the VMs running the VNF. Moreover, at runtime (i.e. with the service chain in place), it monitors the operational status and performances of the master VNFs VMs, and collects notifications about VMs status, e.g. by registering as an observer to dedicated services offered by the virtualisation platform used within the virtual compute infrastructure. Moreover, VNF pool manager reacts to any failure condition by autonomously replacing the master VNF with one of its backup on the pool, basically implementing a swap of VMs for service chain recovery purposes. Thus, the VNF pool manager also takes care in coordination with the service function path manager of implementing those resiliency mechanisms at the chain level. Two options have been identified so far: cold recovery and hot recovery. In the former, backup VNFs, properly configured with the same master configuration, are kept ready (but switched off) to be started when the master dies. In this case the recovery time depends on the specific VNF and its type of function, e.g. it may depend on convergence time for a virtual BGP router. In the hot recovery, active backup VNFs are kept synchronized with the master ones, and the recovery of the service chain (mostly performed at the service function path manager) in case of failure is faster than cold recovery.

4. Positioning in Existing NFV and SFC Frameworks

4.1. Mapping into NFV Architecture

For the presented solution to be integrated in the ETSI NFV reference architecture, some modifications need to be applied to it, with main focus on the Management and Orchestration (MANO) functions. The VNF pools replace the VNFs in the architecture. They are then controlled by the Element Management System (EMS) on the northbound. So the EMS has to be made VNFPOOL aware. Additional elements that need to support the mechanisms proposed by VNFPOOL are the VNF managers, which need to implement the resiliency and VNF scaling

(up-/downscale) functions. This has also implications on the NFV Orchestrator, which has to be aware of the augmented functionality offered by the VNF Manager. In fact, the NFV Orchestrator also provides primitives for VNFs chaining, matching the Service Control Entity in the VNFPool architecture. Therefore, even if ETSI NFV MANO does not include explicitely SDN, the Edge Configurator, and part of the Service Function Path Manager features might be also covered by an augmented NFV Orchestrator.

4.2. Mapping into SFC Architecture

[I-D.ietf-sfc-architecture] describes the Service Function Chaining (SFC) architecture. It describes the concept of a service function (SF) and how to chain SFs and provides only little detail of the SFC control plane, which is responsible with the coordination of the SFs and their stitching into SFCs. The combination of orchestrator, service function path manager and VNF pool manager functionalities described in this document cover most of the functions expected from the SFC control plane.

The interaction with the SFC Classifier is left for further study. We expect the VNFPOOL architecture to leverage on it to make sure that all VNFPOOL instances will be served traffic on during scale-up and that no traffic will be lost during scale-down.

5. IANA Considerations

This draft does not have any IANA consideration.

6. Security Considerations

Security issues related to VNF pool orchestration and resiliency of service chains are left for further study.

7. Acknowledgements

This work has been partially supported by the European Commission through the H2020 5G Crosshaul (The integrated fronthaul/backhaul, grant agreement no:H2020-671598) and Selfnet (Framework for Self-organized network management in virtualized and software defined networks, grant agreement no:H2020-671672) projects. The views expressed here are those of the authors only. The European Commission is not liable for any use that may be made of the information in this document.

Authors would also like to thank V. Maffione and G. Carrozzo from Nextworks for valuable discussions and contributions to the topics addressed in this document.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

8.2. Informative References

[I-D.ietf-sfc-dc-use-cases] Surendra, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", draft-ietf-sfc-dc-use-cases-06 (work in progress), February 2017.

[I-D.ietf-sfc-architecture] Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-11 (work in progress), July 2015.

[I-D.ietf-sfc-nsh] Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-12 (work in progress), February 2017.

[I-D.zong-vnfpool-problem-statement] Zong, N., Dunbar, L., Shore, M., Lopez, D., and G. Karagiannis, "Virtualized Network Function (VNF) Pool Problem Statement", draft-zong-vnfpool-problem-statement-06 (work in progress), July 2014.

[VCPE-T2] G. Bernini, G. Carrozzo, P. A. Gutierrez, D. R. Lopez, , "Virtualising the Network Edge: Virtual CPE for the datacenter and the PoP", European Conference on Networks and Communications , June 2014.

Authors' Addresses

Giacomo Bernini
Nextworks
Via Livornese 1027
San Piero a Grado, Pisa 56122
Italy

Phone: +39 050 3871600
Email: g.bernini@nextworks.it

Giada Landi
Nextworks
Via Livornese 1027
San Piero a Grado, Pisa 56122
Italy

Phone: +39 050 3871600
Email: g.landi@nextworks.it

Diego R. Lopez
Telefonica
C. Zurbaran, 12
Madrid 28010
Spain

Email: diego.r.lopez@telefonica.com

Pedro A. Aranda Gutierrez
Universidad Carlos III Madrid
Leganes 28911
Spain

Email: paranda@it.uc3m.es