

Article

A Service-Oriented Approach for Dynamic Chaining of Virtual Network Functions over Multi-Provider Software-Defined Networks

Barbara Martini ^{1,*} and Federica Paganelli ^{2,*}

¹ National Interuniversity Consortium for Telecommunications (CNIT),
National Laboratory of Photonic Networks, via G. Moruzzi 1, 56124 Pisa, Italy

² National Interuniversity Consortium for Telecommunications (CNIT),
Research Unit at the University of Firenze, via S. Marta 3, 50139 Firenze, Italy

* Correspondence: barbara.martini@cnit.it (B.M.); federica.paganelli@cnit.it (F.P.); Tel.: +39-050-5492245 (B.M.); +39-055-2758524 (F.P.)

Academic Editor: Dino Giuli

Received: 23 February 2016; Accepted: 18 May 2016; Published: 1 June 2016

Abstract: Emerging technologies such as Software-Defined Networks (SDN) and Network Function Virtualization (NFV) promise to address cost reduction and flexibility in network operation while enabling innovative network service delivery models. However, operational network service delivery solutions still need to be developed that actually exploit these technologies, especially at the multi-provider level. Indeed, the implementation of network functions as software running over a virtualized infrastructure and provisioned on a service basis let one envisage an ecosystem of network services that are dynamically and flexibly assembled by orchestrating Virtual Network Functions even across different provider domains, thereby coping with changeable user and service requirements and context conditions. In this paper we propose an approach that adopts Service-Oriented Architecture (SOA) technology-agnostic architectural guidelines in the design of a solution for orchestrating and dynamically chaining Virtual Network Functions. We discuss how SOA, NFV, and SDN may complement each other in realizing dynamic network function chaining through service composition specification, service selection, service delivery, and placement tasks. Then, we describe the architecture of a SOA-inspired NFV orchestrator, which leverages SDN-based network control capabilities to address an effective delivery of elastic chains of Virtual Network Functions. Preliminary results of prototype implementation and testing activities are also presented. The benefits for Network Service Providers are also described that derive from the adaptive network service provisioning in a multi-provider environment through the orchestration of computing and networking services to provide end users with an enhanced service experience.

Keywords: network function virtualization; software defined networking; service-oriented architecture; network service; service chaining; virtual network function

1. Introduction

Today's users continuously demand interactive, content-rich, and immersive networked experiences while imposing increasingly stringent requirements on the delivery capability of the network infrastructure. With a number of mobile digital services, *i.e.*, apps, which are racing toward the two million mark, the over-the-top (OTT) service providers have a consistent margin for innovating their infrastructure to optimize service provisioning and to enrich their service offerings to users at a rapid pace. On the other hand, network service providers (NSPs) have many difficulties to roll out innovative network-based services and to benefit from increasing revenues of the new digital economy [1]. A relevant obstacle is the inflexibility of the network infrastructure, which is static

and costly to change. In fact, the deployment of network services and functions (e.g., routers, middleboxes) traditionally require the acquisition and operation of specialized hardware devices and their interconnections. This results in static chains of network services that cannot flexibly cope with dynamic user and service requirements (e.g., delay constraints) [2]. Moreover, the inflexibility in the service delivery is even more prominent across different NSPs and geographical domains where the level of resource accessibility and exploitation in the service delivery chain is still limited. The lack of a multi-provider infrastructure service coordination and deployments prevents NSPs to benefit from agile service customization, service enhancement, and reaching new markets [3].

Recent research efforts on promising network technologies, *i.e.*, software defined networking (SDN) [4] and network function virtualization (NFV) [5], go in the direction of conceiving and developing novel network virtualization models and programmable network service delivery functions to address the aforementioned flexibility and scalability requirements. Indeed, in accordance with recent service virtualization paradigms adopted in the computing domain (*i.e.*, cloud computing), they are expected to promote a service-centric vision where the data delivery function is decoupled from the underlying network physical infrastructure and is conceived as a chain of ready-to-use functional capabilities, dynamically deployed where more appropriate in the physical network as virtual resources and provisioned as a service in potentially multi-provider environments [6,7]. Challenges related to dynamic service composition and provisioning in a multi-provider environment have been faced in the last two decades in the IT domain, especially with the efforts related to service-oriented architecture (SOA) [8]. We, thus, deem it important to take into account principles and best practices of SOA in order to foster the deployment of SDN and NFV solutions for the provision of network services that can also be dynamically discovered, negotiated, and composed from different providers to meet specific user and service requirements.

Below, we introduce the main concepts of NFV and SDN in order to provide the background information for our work. We also introduce main concepts of SOA, since it provides a reference framework for service modeling and provisioning.

1.1. Background

Network Function Virtualization [5] is one of the most innovative manifestations of virtualization in networking enabling network functions and capabilities to be implemented as software components and executed in virtual machines or containers provisioned in general-purpose hardware systems, *i.e.*, VNFs. Thanks to a more agile lifecycle management of virtual machines, VNFs can be instantiated, updated, deleted when and where needed, as well as dynamically combined to provide more complex capabilities on demand. This allows for a strong reduction of capacity over-provisioning and for an opportunistic deployment and/or re-arrangement of VNFs to be performed, thus optimizing the usage of the underlying resource infrastructure toward different management targets (e.g., consolidated or balanced usage of physical servers, proximity to users).

Software-Defined Networking [4] is a new approach to the design, building, and management of networks. Basically, SDN decouples the software-based control plane (e.g., routing decision functions) from the hardware-based data plane (*i.e.*, packet forwarding engine) while abstracting the underlying network infrastructure and moving the network intelligence to a centralized software-based controller where network services, such as traffic engineering and path provisioning, are deployed. Such separation allows for a more agile and cost-effective network operation thanks to full programmability of forwarding capabilities and enhanced decision-making capabilities based on a global view of the network status. As a result, SDN opens the way toward a more effective interaction between applications and networks for the establishment of data delivery paths while addressing resource usage optimization and reducing the complexity of the network operation [4].

Service Oriented Architecture [8] is an architectural paradigm for the interoperability of heterogeneous systems from different administrative domains. Basically, service orientation principles define how resources and capabilities can be handled as independent services that can be flexibly and

dynamically composed to provide more complex functionalities and address dynamic requirements. A service can be defined as a component that performs a simple, granular, and self-contained function that can be invoked by external clients through well-defined interfaces.

Erl [8] proposed the following main principles for service oriented design: service contract, loose coupling, service abstraction, service reusability, and service composability. The service contract expresses the capabilities offered by services and their technical interface details. The loose coupling principle emphasizes the need for reducing dependencies among the service implementation, its published contract, and service consumers. Service abstraction is a cross-cutting aspect of service-orientation that consists in hiding as much as possible the low-level details of a service interface and implementation. Service Reusability implies that services should be designed for serving more than one consumer. The Service Composability principle expresses the need for services that can be used as building blocks of more complex services (*i.e.*, composite services).

A SOA is typically characterized by three main roles: (i) the Service Provider makes the service available and publishes its description profile (contract) in a service registry handled by a Service Broker; (ii) the Service Consumer uses the service; and (iii) the Service Broker mediates the interactions between Service Providers and Consumers by offering discovery, matchmaking, and composition capabilities.

1.2. Contribution

SOA principles have been adopted in the past for the rapid and flexible development and management of value-added telecom services. However, a literature review on SOA models for Next Generation Networks [9] shows that, while many works focused on the integration of distributed service components at the service stratum, the benefits of applying the SOA paradigm to the transport functionalities have been scarcely investigated. Furthermore, recent works on network and service virtualization [10] show that the opportunity to adopt the SOA approach in telecom cloud environments remains quite unexplored. We argue that the adoption of SOA principles in the NFV and SDN technological landscape can ease the provision of network services that can be dynamically discovered, negotiated, and composed also from different providers to meet specific user and service requirements [7,11].

In this work, we propose a service-oriented approach for the orchestration of network services deployed as a cloud of Virtual Network Functions (VNF) on top of a NFV infrastructure that also takes full advantage of granular traffic steering capabilities provided by SDN. To this purpose, we refer to the SOA technology-agnostic architectural guidelines for organizing, (re)using, and integrating distributed networking capabilities provided by different systems [8]. First, we provide a survey of reference service scenarios for virtualized networks and, then, we elaborate how SOA, NFV, and SDN provide complementary features toward adaptive composite network service delivery, and organize these features in a set of tasks required for realizing dynamic network service chaining: service composition specification, service selection, service delivery, and placement. Then, we describe the architectural design of a SOA-inspired NFV orchestrator with SDN network control capabilities, also providing references to related standardization initiatives and illustrating its main workflows. We also present preliminary results of our ongoing activities for the implementation and testing of a prototype to validate the proposed approach. Finally, we conclude the paper with a discussion of benefits for NSPs along with a description of research challenges posed by the proposed approach.

2. Related Work

In this section we first provide an overview of main standardization activities in the NFV and SDN domains. Then, we discuss research works on dynamic network service chaining, highlighting contributions in terms of architectural and methodological guidelines and, finally, we motivate the contribution of our work.

2.1. Standardization Activities

The NFV ISG (Industry Specification Group) of the European Telecommunications Standard Institute (ETSI) is the most noteworthy standardization initiative so far regarding the NFV domain aiming at specifying general, open, and scalable architectural and operation solutions to meet challenges placed by NFV. More specifically, current standardization efforts in ETSI [5] are paving the way toward the realization of a multi-provider NFV ecosystem through the definition of a comprehensive architectural framework composed of: (i) the NFV Infrastructure (NFVI), which includes the hardware and software resources that support a cloud of virtualized network, computational and storage resources; (ii) the VNFs, which refer to the software implementations of network functions capable of running over the NFVI; and (iii) NFV Management and Orchestration (MANO), which covers the lifecycle management of both physical and software resources as well as management and orchestration of VNF instances [12]. An important step for NFV MANO has been to include software-defined networking solutions into the NFV architectural framework. Such solutions consider SDN controllers acting as Network Controllers for the network part of the NFVI (*i.e.*, network infrastructure domains) to deliver connectivity services involving physical or virtual resources (*i.e.*, OpenFlow switches) while providing an abstract view of such resources for orchestration purposes [13]. Moreover, in [14] ETSI identified the most common design patterns for using SDN in an NFV architectural framework along with recommendations to be fulfilled by the entities that perform the integration.

Current standardization efforts in the Internet Engineering Task Force (IETF) around SDN and NFV include, firstly, the Service Function Chaining (SFC) Working Group [15] efforts aiming at addressing the dynamic specification and instantiation of an ordered list of instances of service functions (such as firewalls, load balancers, *etc.*) while subsequently steering of data traffic flows through those service functions, accordingly. Secondly, another related initiative carried out in IRTF is the Network Function Virtualization Research Group (NFVRG) [16] aiming at developing new architectures, systems, and software, and to explore trade-offs and possibilities for leveraging virtualized infrastructure to provide support for network functions. Thirdly, efforts in IETF are also devoted to the design of a comprehensive SDN controller, namely Application-Based Network Operations (ABNO) architecture [17], including instrumental guidelines and operational workflow specifications to coordinate network control functions to compute paths, enforce policies and manage topology while providing a full network automation and programmability for the benefit of the applications that use the network. To the best of our knowledge, these efforts are still poorly integrated.

IEEE with the Next Generation Service Overlay Network (NGSON) standard [18] specifies a framework for the control and delivery of composite services over diverse IP-based networks (*e.g.*, Internet, P2P overlay, IMS, PSTN, Mobile) with context-aware, dynamically-adaptive, and self-organizing capabilities. Dynamic context of users, devices, services, and networks are considered to adapt composite service delivery while optimizing network and computing resources consumption. In particular, network awareness allows for adapting service provisioning to the current status of the network (*e.g.*, avoiding hot spots or congestions) while satisfying the ever-changing requirements of users. In this regard, recent emerging paradigms such as SDN and initiatives such as NFV can significantly contribute to achieve greater elasticity in network service deployments and accelerate a prominent innovation in NGSON service and data delivery functions. In this regard, a liaison has been established between the IEEE SDN Initiative and the NGSON WG to identify potential new standards in SDN/NFV areas to be developed in IEEE [19]. "The goals are to accelerate the proliferation of SDN services and applications and to offer a more efficient way of providing them through a service-architecture ecosystem of one-stop shopping for service-specific challenges" [20]. In [11] the authors report on their contributions to IEEE NGSON architecture to include SDN and NFV technologies and to provide a more powerful service composition and orchestration functions through generalized service chains including both application and network services while dynamically establishing data delivery paths across services.

The Open Networking Foundation (ONF) is a user-driven organization dedicated to the promotion and adoption of SDN through the development of the OpenFlow specifications as an open standard for the communication between the controller and the data forwarding network elements. ONF is also devoted to the promotion of open source developments as a way of consolidating impacts of SDN in the Industry. Moreover, ONF is very active in the definition of an SDN architecture framework with focus on the controller capabilities and on the interfaces with the other elements in the architecture both at north-bound and south-bound with applications and network elements, respectively [21,22]. Within the services area, an initiative started about providing an end-to-end orchestration, abstraction and resource optimization across data center SDN controllers and Wide Area Network (WAN) SDN controllers so that user-applications can be created and managed seamlessly [23]. In the NFV arena, the ONF has proposed a flexible NFV networking solution [24] for an NFV deployment of an OpenFlow-enabled SDN approach to deal with the dynamic provisioning of networking services. However, both initiatives do not cover the dynamic composition and orchestration aspects as this work does.

Other related standardization efforts are in progress in the Broadband Forum (BBF) [25], which is working on how cloud-based technologies including NFV can be used in the implementation of the Multi Service Broadband Network. The TMForum (TMF) rolled out the Zero-touch Orchestration, Operations and Management (ZOOM) program to develop best practices and standards to deliver true business agility and new digital services and revenue opportunities in the area of virtualization, NFV, and SDN [26].

2.2. State of the Art of Research Works

Dynamic service chaining is considered a relevant challenge for the evolution of network deployments toward flexible, cost-effective, and on-demand service delivery models. Recently, several works have been focused on this topic and related issues. Based on the analysis of existing works, we distinguish related literature in two main areas: architectural approaches for NFV/SDN integrated management; algorithms and techniques for specific issues in dynamic service chaining.

Within the first area, several works include an architectural model integrating NFV management with SDN control capabilities as a main goal of the work or as a reference framework within which more focused contributions can be positioned.

Some authors limited their discussion to high-level guidelines and architectural views. Reference [27] is a position paper that analyses the case of multi-domain distributed deployment of NFV. Some reference use cases are analyzed and challenges and research directions are discussed. Lopez *et al.* [28] argue the need of conceiving a Network Operating System (NOS) to cope with the lack of network-wide abstractions, thus favoring the foundation for true network programmability. However, this is a short paper that not handles in detail the integration of network control capabilities with NFV management functions. Naudts *et al.* [29] present a three-layer architectural view for both network and cloud domains (infrastructure, control and application layers). The integration of these two architectures is discussed by introducing an orchestration functional box, which is used for services that require a combination of these resources. However, its interaction with cloud and network control functions is only briefly described. Reference [30] analyzes benefits and challenges related to the adoption of the NFV paradigm toward the 5G cellular framework. The work moves from the high-level architecture conceived within the T-NOVA European Project to support the dynamic provision of VNFs on-demand and as-a-service. However, the authors provide only an introductory and conceptual description of the architecture.

Other related works provide deeper insights into architectural design and related implementation issues [31–34].

Reference [31] presents a functional architecture supporting automated, dynamic service creation leveraging NFV, SDN, and cloud virtualization techniques, which has been conceived in the framework of UNIFY, a three year European Project that has just concluded. The proposed architecture comprises

a service layer, an orchestration layer and an infrastructure layer. They also briefly discuss how some functions/layers of the architecture can be mapped to ETSI NFV MANO and ONF SDN architectural elements. UNIFY do not explicitly refer to SOA principles, although some similar concepts exist, such as the description of services at different levels of abstractions (*i.e.*, service graphs and network function forwarding graph). Nevertheless, UNIFY does not aim at supporting a multi-provider marketplace of network services as this work proposes using SOA practices, which were, in fact, designed to ease the operation of multi-provider environments where services can be dynamically-advertised and discovered. Giotis *et al.* [32] propose a modular VNF architecture providing policy-based management of VNF and service chains. Their main contribution consists in a basic ontology-based information model to describe network resources, network control functions, and VNFs capabilities with a uniform language. However, the role of network control functions (*i.e.*, SDN controller) in the architecture and its interaction with the NFV orchestrator are not explained. Munoz *et al.* [33] address the problem of SDN-based virtual connectivity over multi-technological domains. Therefore they propose an integrated SDN/NFV management and orchestration architecture which is specifically conceived for the dynamic deployment of Virtual Tenant Networks and the related SDN Controllers (implemented as VNF in DCs). However, they do not address the service chaining problem, as this work does.

Soares *et al.* [34] propose the CloudNFV Platform for enabling a telecom operator to deploy and manage service functions in a distributed cloud infrastructure. Although they focus on service chaining, their work differs from our contribution in that the proposed architecture specifically reflects the software design of their prototype (*i.e.*, instead of providing general functional specifications, some functional blocks are defined in terms of software implementation OpenStack and OpenDayLight). Moreover, they do not explicitly refer to service oriented principles and architectural guidelines.

On the other side, Garay *et al.* [35] stress the importance of a description model for network service. Although they do not explicitly refer to SOA guidelines, this approach is in principle compliant with those guidelines. However, their contribution consists in a straw man model for service descriptions, not on architectural frameworks.

As regards the second area of research we identified several authors focusing on the problem of VNF placement, *i.e.*, the (sub)optimal placing of a set of virtual network functions on a network of physical nodes to serve a set of service chain requests while optimizing a certain utility function, for instance to minimize the number of utilized servers [36,37] or to apply load balancing policies [38]. In [39] the authors provide a formalization of the VNF scheduling problem, *i.e.*, finding the corresponding time slots for functions to be executed over a given set of machines. Other authors [40] focused on the routing problem, *i.e.*, assigning paths to the incoming traffic flow requests in order to connect nodes running virtual functions. Ultimately, such works focused on specific issues of service chaining, proposing focused algorithms and techniques, but, to the best of our knowledge none of them discuss how the proposed solution fits into a comprehensive architectural framework.

2.3. Motivation of Our Work

The analysis of the literature confirms the need of reference architectural frameworks supporting the integration of VNF orchestration and network control functions toward the dynamic provisioning of network services in multi-provider environments.

The main limitation of related work is the level of abstraction adopted for the specification of the reference architectural framework: some works provide high-level and conceptual guidelines [27–30], while others present architectural design which are coupled with implementation choices [33,34].

Our objective is, thus, to propose a functional architecture which is implementation-independent and whose specifications clearly identify the role of the components and shape their mutual interactions.

Since the “as-a-service” abstraction is considered a key element in the prospected evolution of networking technology, our aim is to take advantage of principles and practices that have emerged in the domain of service lifecycle management in the area of SOA design and implementation.

Moreover, since SOA is an architectural solution which can be mapped into different implementation solutions [41], we deem that the adoption and re-visitation of SOA principles and main architectural guidelines to target our problem domain have the benefit of achieving the desired level of specification abstraction (as mentioned above), while also taking into account architectural patterns and practices developed in more than one decade.

Therefore, our contribution consists first in providing a survey of reference scenarios for virtualized networks and elaborating how SOA, NFV, and SDN provide complementary features toward adaptive composite network service delivery in multi-provider environments. Then, we propose a SOA-inspired architectural model integrating NFV orchestration and SDN network control capabilities, also providing references to related standardization initiatives and illustrating its main workflows. We also describe our ongoing prototyping and testing activities, which aim at validating the proposed approach.

3. Reference Service Scenarios

In this section, we survey two main reference service scenarios for virtualized networks made possible by dynamic VNF composition and orchestration (see Figure 1).

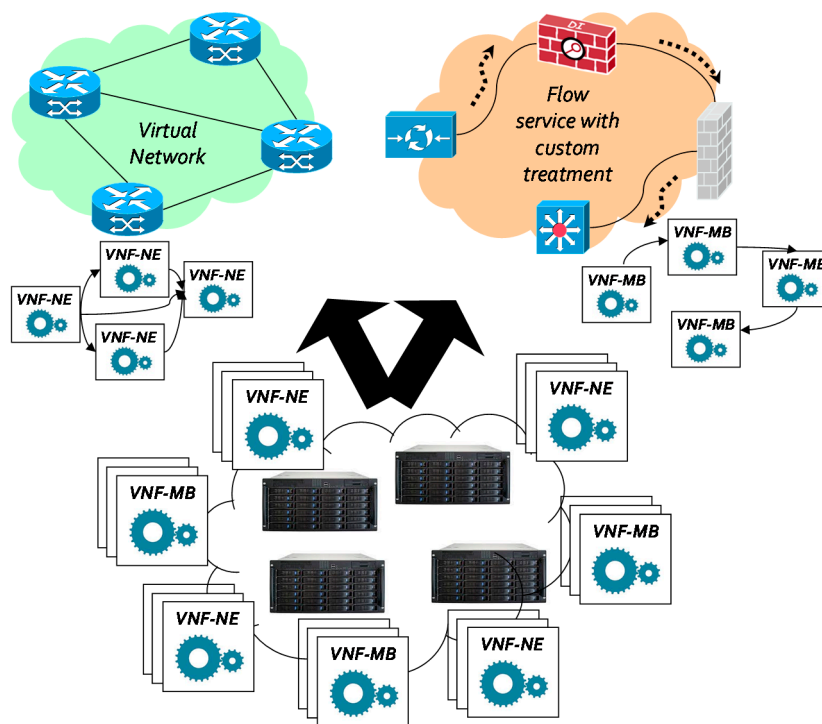


Figure 1. Scenarios for dynamic composition of Virtual Network Functions.

The first scenario is enabled by the deployment of VNFs that provide network element (NE) functions, *i.e.*, VNF-NEs. Examples of NEs might be either switching elements (e.g., routers, broadband remote access server), mobile network nodes (HLR/HSS, SGSN, GGSN/PDN-GW, base stations), or signaling control systems (e.g., session border controllers) [12]. Different functions of the same NE can be also partitioned and deployed as different VNFs operated in different network locations instead as one single comprehensive VNF. For instance, a session border controller can be split in the following functions deployed as VNFs, *i.e.*, session terminations executed at the edge of the network, admission control executed in the core network, and statistics and billing data collection executed at a data center [2].

In general terms, the sequence of deployed VNFs formally represents a virtual network (VN) established to deliver different kinds of network infrastructure services (e.g., IMS, 4G, IP/MPLS) and loosely coupled with the exact physical location of constituent VNFs. On the one hand, the deployment of VNF-NE chains allows for NSPs to set-up/upgrade the network infrastructure while taking advantage of the deployment of network functions as virtual machines. On the other hand, the NSPs can enrich their service offerings through the delivery of virtualized network infrastructure services to third-parties, *i.e.*, “VN service”. In a scenario where network functions can be dynamically discovered, negotiated and elastically composed as services, application service providers may lease VNF-NE chains with given communication capabilities from different NSPs and compose them to operate an end-to-end virtual service infrastructure to offer value-added application services to users (e.g., delay-optimized infrastructure for high-definition video applications [6]). Accordingly, a request for “VN service” set-up can be internally issued by a Network Management System (NMS) or an Operations Support System (OSS) on behalf of a network planning function of a NSP to establish/update the network infrastructure. Furthermore, a request for a virtual service infrastructure deployment, such as a next generation service overlay network (NGSON), can be issued by an application service provider.

The second scenario derives from the deployment of VNFs providing middlebox (MB) functions, *i.e.*, VNF-MBs. Example of MBs are: firewall, network address translation, WAN optimization controller (WOC), deep packet inspection (DPI), intrusion detection systems, load balancers, multimedia transcoders, virus scanning. Elastic deployments of VNF-MBs allow for differentiated data processing since service data flows can traverse only the needed VNFs, while skipping the unnecessary ones. For instance, in the case of a long-lived multimedia data flow it could be beneficial to avoid a DPI appliance in order to confine delays and save processing resources, while it could be beneficial including a WOC function so that traffic would be routed over links with proper level of delay and jitter guarantees. Accordingly, a service delivery path need to be established to serve an application data flow which includes specified network processing functions to be traversed for addressing given application requirements.

The capability to dynamically establish chains of VNF-MBs allows NSPs enriching their Quality of Service (QoS) offerings through the delivery of paths with extended QoS guarantees that do not only address delay or throughput requirements but also availability, reliability and security requirements demanded by the application data flows, *i.e.*, “flow service with custom data treatment”. In fact, in a dynamic landscape of service delivery, VNF-MB chains can be dynamically provisioned that extend the network forwarding capabilities with customizable data processing features. Accordingly, the sequence of deployed VNFs formally represents a request for a “flow service with custom data treatment”, which can be issued by a service delivery platform, e.g., NGSON, on behalf of an application service provider during a negotiation phase for adequate transport QoS guarantees [11].

In the rest of the paper, such service scenarios will be generally referred to as network service chaining, irrespective of the fact that VNFs in the chain deploy NE or MB functions.

4. Service-Oriented Approach for Dynamic Network Services

The adaptive composition and delivery of network services within the NFV framework requires the conception of architectural models and techniques for dynamic definition, orchestration and management of network functions. We argue that SOA provides an effective solution for coordinating and elastically composing virtualized network functions, *i.e.*, VNFs, across heterogeneous systems while leveraging SDN capabilities to programmatically set up data delivery paths through the dynamically established sequence of VNFs. Thus, as depicted in Figure 2, we envision a synergistic connection among SOA, NFV and SDN for addressing more effective service delivery models that allow NSPs to stay competitive and keep the pace with the service offer and infrastructure innovation of OTT providers.

Although virtualization does not necessarily imply the adoption of service-oriented principles, it favors the usage of service-oriented abstractions to model and handle network functions executed on top of virtual resources. Indeed, thanks to the definition of open and well-defined interfaces between network functions and their management entities, VNFs can be represented as “black boxes” in the form of SOA-compliant network services (arrow 1 in Figure 2). By leveraging service-oriented principles, these network services can be dynamically consumed and composed to provide more complex and adaptive services based on specified requirements, even in a multi-provider environment (arrow 2). To complete the picture, the dynamic provision of VNF-enabled service chains leverages the capability offered by SDN of enforcing adequate traffic forwarding capabilities through the constituent VNFs (arrow 3).

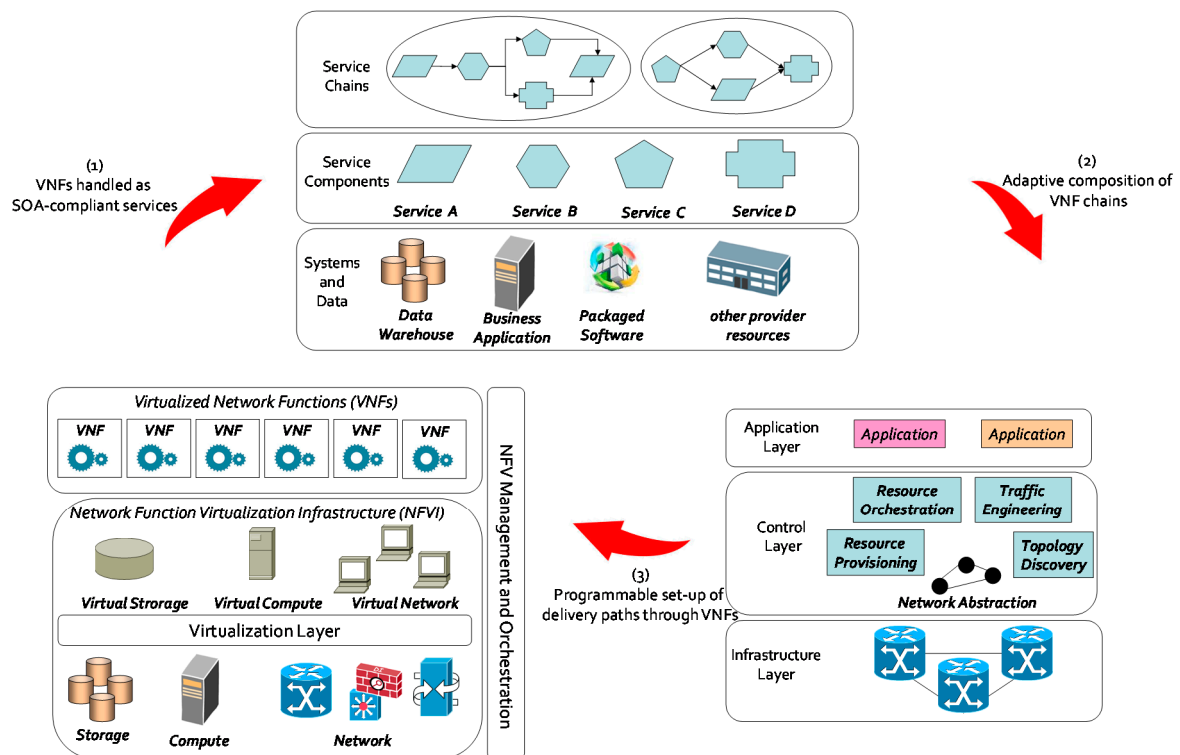


Figure 2. Service Oriented Architecture, Network Function Virtualization, and Software Defined Networking synergy for adaptive network service composition and delivery models.

In this perspective, the adoption of service-oriented models can help in defining the proper level of abstraction of network capabilities provided by the NFV infrastructure, thus enabling a loose-coupled, flexible, and effective collaboration among providers of infrastructural resources, network and application services. Indeed, by offering a virtualized access to the physical infrastructure, a NSP can cooperate with other providers to offer advanced network services and capabilities to different consumers (e.g., OTT providers) on top of a virtual resource infrastructure, *i.e.*, NFVI.

Dynamic Service Chaining of Virtual Network Functions

The deployment of network functions as VNFs paves the way for network resource and capabilities to be handled as independent and self-contained services that can be flexibly composed to provide a dynamically-established sequence of functionalities leveraging SOA principles, *i.e.*, dynamic service chaining.

In dynamic service chaining, the invocation flow of VNFs is specified at run time for addressing flexible and adaptive service deployments. We define as a “Composite Service” a generalized network service to be provisioned as a result of proper associations of VNF instances according to specified

requirements. As depicted in Figure 3, the provisioning of a “Composite Service” through the orchestration of VNFs is enabled through the following functions: “service composition specification and retrieval”, “service selection”, “service delivery”, and “service placement”. The first two functions are typical of service-oriented architectures and are here slightly revisited to align with the terminology adopted in the VNF standard specifications. The latter two functions, instead, strongly depend on the peculiarities of reference service scenarios for network service chaining introduced in Section 3.

Service composition specification consists in defining a chaining logic as a workflow of functional network capabilities, *i.e.*, “Abstract Service Chain”. More specifically, the “Abstract Service Chain” corresponds to the VNF-graph defined in ETSI [12] and specifies the types of VNFs that should be connected to provide a service (or part of a service), their order and the connectivity requirements among them (e.g., dependencies between VNFs, maximum latency and/or minimum bandwidth of inter-VNF connection, *etc.*), while not specifying yet the location and implementation details of the instances that should actually provide those functions. When a service request has to be handled, the related service composition specification is retrieved. The “Abstract Service Chain” is thus a template for service chain instances. The level of abstraction thus introduced is essential to assure loose coupling between the exposed network service and the specific implementation and management issues of the VNF chain and its constituent elements. This promotes the flexibility to choose among available VNF instances at runtime (*i.e.*, service selection), especially in a multi-provider environment. Furthermore, this level of abstraction eases the delivery of dynamically adaptable services to accommodate policy changes as well as requirements of clients accessing the service. Indeed, context-aware adaptation rules can be applied to modify at run time the “Abstract Service Chain” according to the current situation, e.g., user location and network load [11].

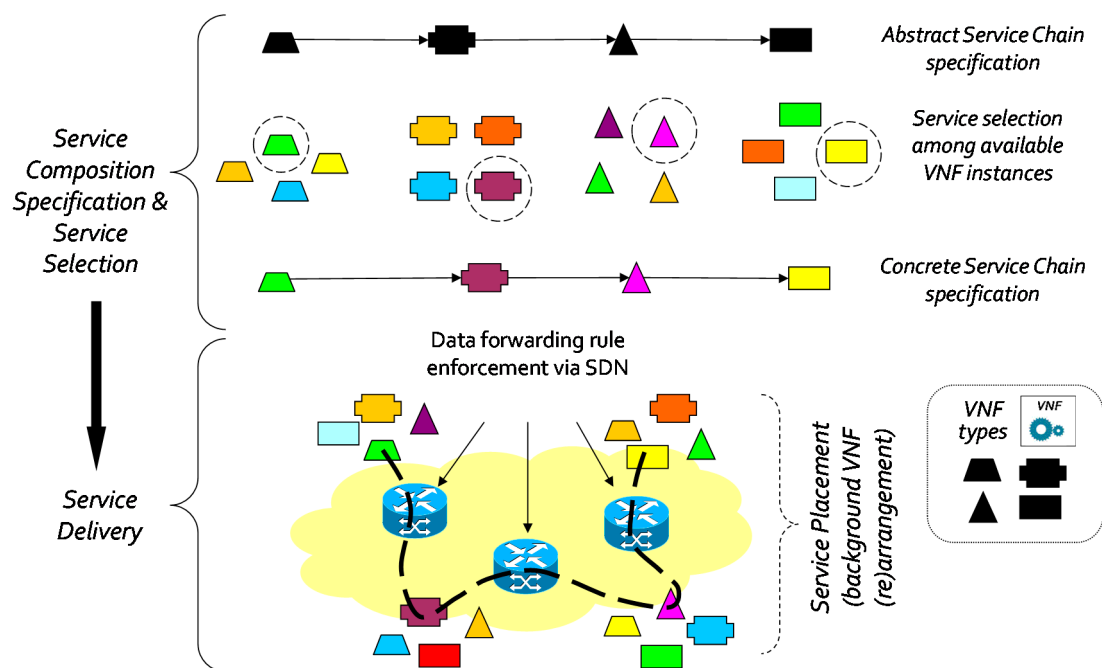


Figure 3. Service-oriented mechanisms for dynamic service chaining of Virtual Network Functions (VNFs).

Service selection consists in the proper identification of the sequence of VNF instances among available candidates thereby mapping the “Abstract Service Chain” into a “Concrete Service Chain”. Different algorithms can be used to this purpose that optimize a QoS-based utility function (e.g., minimizing the latency per-application traffic flows) for a given composition plan. Such algorithms can consider the computation capabilities and load status of resources executing the

VNF instances, either deduced through estimations from usage historical data or collected through real-time monitoring data (*i.e.*, context-aware selection). For this reason a “Concrete Service Chain” should include references to dynamic information on the status of the service instance and its constituent elements, *i.e.*, monitoring information related to individual VNF instances and links connecting them as well as derived monitoring information at the chain level (e.g., end-to-end delay). At runtime, if one or more VNF instances are no more available or QoS degrades below a given threshold, the service selection task can be rerun to perform service substitution.

Service delivery consists in provisioning delivery paths throughout the selected chain of VNFs thereby accomplishing the required VNF associations. It concerns the use of forwarding functions provided by the underlying network infrastructure to guarantee the proper connectivity among VNF instances specified in the “Concrete Service Chain”. The service delivery task is realized through the enforcement of data flow forwarding rules able to address the connectivity requirements among the selected VNF instances. Here SDN plays a key role since it offers the capability to programmatically enforce traffic forwarding rules across network nodes on per-flow or per-tenant basis. Such capability can be exploited to selectively deliver service data through the dynamically established sequence of VNF instances thereby accomplishing the deployment of a network service chain [42,43].

Service Placement consists in the allocation of virtual resources executing VNFs and/or re-arrangement of running VNFs (e.g., migration) while optimizing a specified cost function (e.g., energy consumption, network congestions). Indeed, the VNF instances that take part to a service chain may be deployed in different locations and be provided by different parties. This task runs in the background with respect to the above-mentioned tasks. Indeed, the orchestration assumes that the VNF instances are available at the moment of the service deployment as well as throughout the delivery phase. In order to cope with dynamic application requirements (e.g., user mobility, QoS, or security requirements) or unexpected events (e.g., load congestions, failures), NSPs implement proper resource re-arrangement strategies to deploy VNFs according to the new context while guaranteeing the continuity of the service chain.

5. Architectural Design

In this section we present the functional architecture of a NFV orchestrator with SDN network control capabilities and describe the main functional entities (FEs) involved in the proposed VNF orchestration approach.

As shown in Figure 4, we distinguish two main functional network control layers, *i.e.*, the “Network Service Provisioning and Control” and the “Infrastructure Control and Management” layer. The former interacts with an “Application Layer”, while the latter interacts with the VNF instances and the underlying resource infrastructure (*i.e.*, the ETSI NFVI).

The “Application Layer” includes instantiation and lifecycle management functions of applications, including network operation support applications. Typically, these functions provide service control and delivery capabilities and are operated at service delivery platforms, e.g., NGSON, or at network service operation systems, e.g., NMS/OSS.

The “Network Service Provisioning and Control” layer supports the dynamic establishment of composite network services on behalf of applications while addressing adequate connectivity requirements. From a SOA perspective, it covers the role of service broker by interfacing with service consumers residing at the “Application Layer”, while handling the interactions with VNF instances by leveraging the control and management features provided by the underlying layer.

The “Service Control” Functional Entity (FE) is in charge of coordinating the provision of network services implemented as a chain of VNFs, according to pre-defined “Abstract Service Chain” specifications. It offers a north-bound interface to applications (e.g., NGSON and NMS/OSS) to request the setup of “Composite Services” (e.g., flow with custom data treatment and VN service, respectively). It leverages the “Service Orchestrator” FE capabilities for instantiating a service chain according to the specifications of the selected “Abstract Service Chain”.

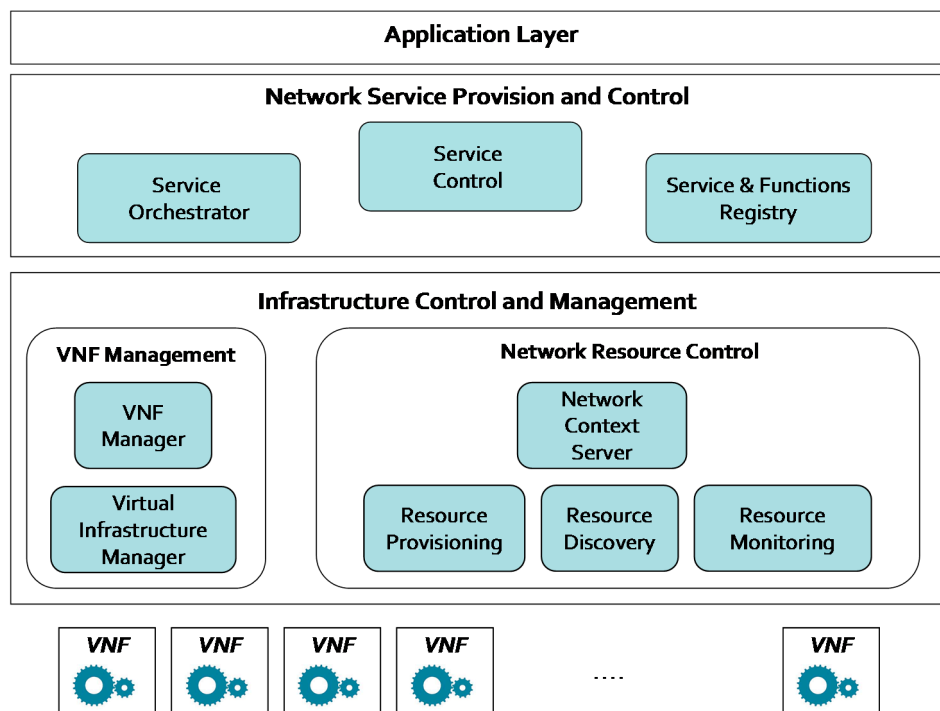


Figure 4. Functional architecture for adaptive service composition and delivery.

The “Service Orchestrator” FE maps the “Abstract Service Chain” into a “Concrete Service Chain”, thereby selecting the proper VNF instances. Moreover, it coordinates the instantiation, operation and connection of VNFs to satisfy the requirements of instantiated services and to manage adaptation mechanisms for coping with service requirement changes or service degradations. It also instructs the “Network Resource Control” FE for enforcing the proper forwarding rules to establish the delivery path through the specified chain of VNF instances. This process runs transparently to the “Application Layer”.

The “Service and Functions Registry” FE maintains the information base required by the “Service Orchestrator” for performing its decision and orchestration tasks. Thus, it handles descriptive and operational information on individual VNFs and on links connecting them (Table 1). Such information is collected through the “VNF Management and Network Context Server” functions of the “Infrastructure Control and Management” layer, described below. If some performance degradation at either VNFs or link is detected, the “Service Orchestrator” is notified. The “Service Orchestrator” performs the appropriate actions for modifying the network service implementation (e.g. substituting VNF instances or the path interconnecting them), otherwise it informs the “Service Control” about the need of cooperative renegotiation with the “Application Layer” functions to reconsider service requirements.

The “Infrastructure Control and Management” functions are responsible for the proper transfer of data across network nodes and VNF management.

The “VNF Management” FE is in charge of managing the lifecycle of individual VNFs and underlying virtual resources. Indeed, it can be decomposed into two main functional components, also according to ETSI MANO specifications: (i) VNF Manager(s), which in accordance with the instructions received by the “Service Orchestrator”, initiates or terminate VNF instances or modifies the existing configuration and (ii) virtual infrastructure manager(s), which manages the lifecycle of virtual resources, such as virtual machines and containers. Both components collect monitoring information on the VNFs and resources operational status and performance metrics (Table 1) and notify the “Service and Function Registry” when events of interest occur (e.g., unavailability of a VNF instance).

Table 1. Network Function Virtualization data model.

Entity	Information Category	Description	Data
VNF	descriptive	It contains the information required to automate the instantiation and use of the VNF capability into a service.	VNF id, VNF type, network location and interface capability, requirements for the VNF instantiation and operation (e.g., Virtual Machine specification, required storage and computation resources), management information (e.g., configuration information, initiation and termination scripts), performance indicators, VNF dependencies, horizontal scale thresholds (e.g., maximum load in terms of percentage of nominal computational and traffic capabilities).
	operational	It contains the monitoring information of VNFs in a running service instance.	Operational status (e.g., start, stop, pause), usage of CPU, memory and storage resources, traffic load at network interfaces.
VNF link	descriptive	It contains the information related to the connectivity (established or to be established) between two or more VNFs, possibly including NFs.	Link type (e.g., point-to-point <i>vs.</i> point-to-multipoint, logical link among VNFs <i>vs.</i> physical link between a VNF and a Network Function), performance indicators (e.g., QoS parameters).
	operational	It contains the monitoring information of established links between VNFs in a running service instance.	Delay, throughput.

The “Network Resource Control” FE is in charge of setting up and managing the network links required to connect the VNF instances. It handles the requests originated from the “Service Orchestrator” FE for connecting a set of (newly) instantiated VNFs for new (or modified) service contracts by properly enforcing the forwarding rules in the network devices along the delivery path. More specifically, the “Resource Discovery” FE handles the discovery of resource capabilities and network topology thus enabling routing, path computation and traffic engineering decisions. The “Resource Provisioning” FE handles low-level configuration directives to address the programming of the network nodes based on information provided by the “Resource Discovery” FE and under the coordination of the “Service Control” FE to consistently steer data flows through the desired VNF chain. The “Resource Monitoring” FE handles low-level directives for network monitoring (*i.e.*, collection of traffic statistics) to feed the “Network Context Server” FE, which is in charge of collecting the network topological and dynamic information of VNF links and making it available to the “Services and Functions Registry” FE with the proper level of abstraction and granularity. More specifically, it handles the descriptive and operational information of links connecting VNFs (Table 1). Efforts for defining VNF and VNF link data models are on-going both in ETSI [12] and in IETF [44] standardization bodies.

The “Network Resource Control” FE and the “Network Context Server” FE can be identified within the IETF ABNO architecture. Specifically, the “Network Context Server” FE refers to the IETF Application-Layer Traffic Optimization (ALTO) server [45] while the “Network Resource Control” FE mainly refers to the ABNO controller. The ALTO server is envisioned to provide abstract network topology information on an end-to-end basis (e.g., network location structure, topological distance between locations and cost between them) for allowing applications to take decisions about service deployments. Relevant network costs information is, for instance, the maximum bandwidth, minimum cross-domain traffic, and lower cost to the user.

The ABNO controller is the main component of the ABNO architecture [13] and is responsible for orchestrating the workflows among ABNO components (e.g., path computation element, topology managers policy agents, and provisioning manager) while addressing application requests of end-to-end network services, e.g., point-to-point path between specified endpoints. Moreover, the ABNO controller is envisioned to provide dynamic network performance information to applications, such as bandwidth usage, available capacity and end-to-end delay.

In Figure 5 the sequence diagram of orchestration workflows is reported for the considered reference service scenarios. The dotted boxes depict the workflows related to the background collection of monitoring information (top) and to the service recovery in case of degradation (bottom), while the main workflow shows the main interactions among FEs for processing a VNF chain set-up request as a result of the orchestration process. More specifically, the main workflow is triggered by a service request issued by an application client to the “Service Control” FE (arrow 1). This FE first retrieves the abstract service description that matches the received request and asks the “Service Orchestrator” FE to return a concrete service chain for that abstract service (arrow 2). The service orchestrator selects the VNF instances that implement the concrete service chaining by properly taking into account current context information, such as VNF and network monitoring data (arrows 3 and 4). Then, it asks the “Network Resource Control” FE for setting up the data path connecting the VNF instances in the service chain (arrows 5–6) and, finally, returns a response to the “Service Control” FE (arrow 7). The workflow in the upper dotted box shows how the Service and Functions registry receives updates on the status of VNF and infrastructure resources from the “VNF Management” FE (arrow 1.a) and on the status of the network (arrows 2.a and 3.a). The workflow in the lower dotted box shows an example of service chain dynamic adaptation to possible network service performance metrics degradation. In the depicted example, we suppose that the service degradation is due to the network path status (e.g., congestion at switches) and, therefore, the “Service Orchestrator” FE, upon the notification issued by the “Service and Functions Registry” (arrow 1.b) requests the setup of a new path connecting the VNF instances (arrows 2.b and 3.b). Of course, more complex workflow could be executed for the service chain dynamic adaptation (e.g., for substituting a subset of VNF instances).

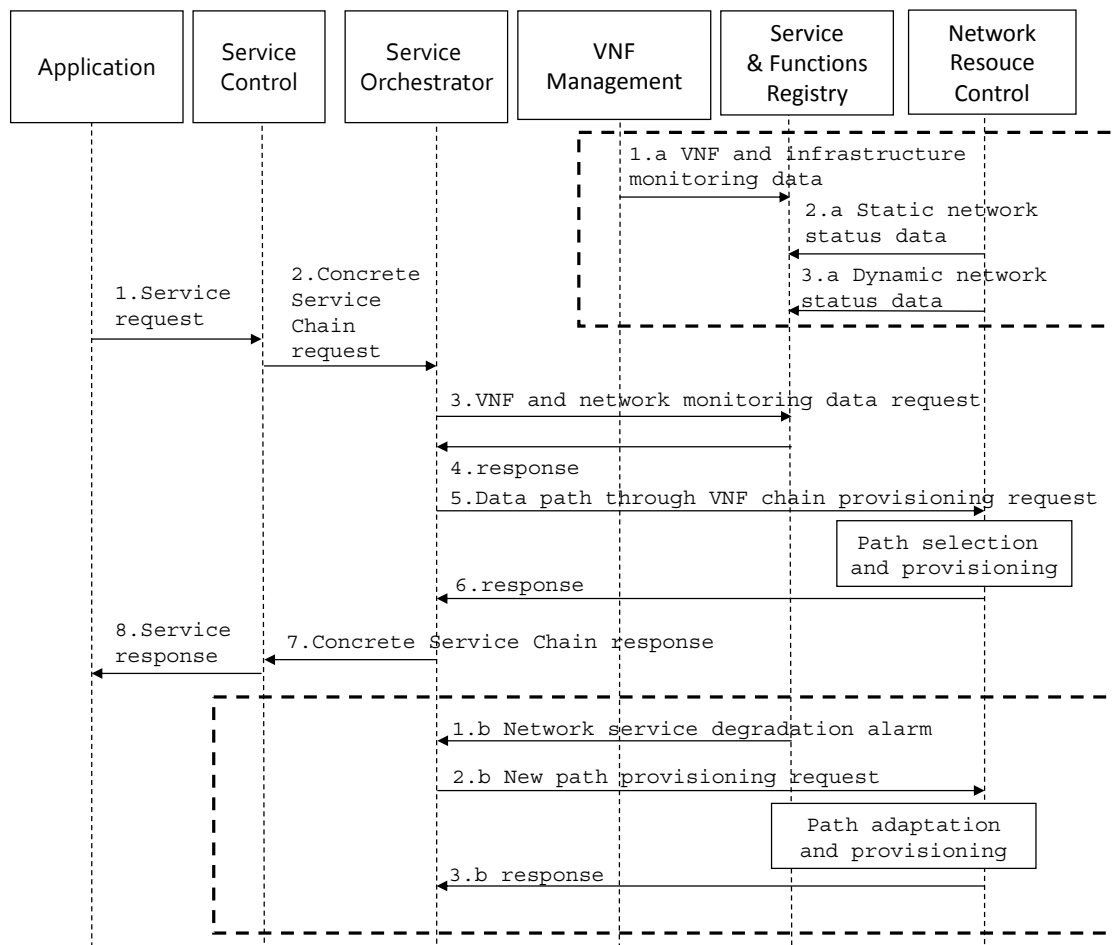


Figure 5. Orchestration workflows for adaptive service delivery.

6. Architecture Preliminary Validation via Prototype

In order to provide a preliminary validation of the proposed approach we implemented a prototype that provides a subset of the functionalities of the service-oriented architecture presented above. The prototype consists of an SDN controller enhanced with orchestration capabilities implemented on top of a Floodlight controller [46] to establish dynamic chains of network functions (e.g., middlebox). Basically, the prototype plays the role of an orchestrator for dynamic service chaining operating over SDN networks, *i.e.*, SDN orchestrator. In particular, the SDN orchestrator addresses the second service scenario envisioned in the Section 3, *i.e.*, “flow service with custom data treatment”, and performs functions both at the “Network Service Provision and Control” layer and at the “Infrastructure Control and Management” layer.

6.1. SDN Orchestrator

The software design of the SDN orchestrator is shown in Figure 6b. More specifically, a subset of the “Service Orchestrator” functions is implemented in the request manager, the service data delivery control and coordination and the adaptation module. In fact these blocks handle requests for the establishment of data delivery paths throughout a specified set of middlebox services. These blocks leverage the basic network server (BNServer) and the Floodlight components that implement the “Network Resource Control” functions. As a result of interactions between the above blocks triggered by path set-up requests, a consistent set of forwarding rules can be enforced across network nodes in order to steer service data flows through the desired VNF chain. In addition to the ordinary workflow for the setup of a new request, the data delivery path set-up requests can be also generated by other SDN orchestrators for the establishment of paths to be concatenated to other paths in a multi-provider scenario or by other internal blocks as a result of an orchestration process to adapt established paths to modified network status. The forwarding rules are enforced in the network nodes via the south-bound interface (SB-I) through the OpenFlow protocol [47]. Moreover, these blocks put in place adaptation mechanisms to cope with degradation events. In fact, degradation events are likely to occur as a result of concurrent usage of network node capabilities among multiple services and the prototype allows for reactively adapting loads across network nodes while addressing required data delivery performance. To this purpose, the prototype also includes “Network Context Server” functions implemented by the statistics collector block, which collects, elaborates, and aggregates OpenFlow statistics about network nodes and links (e.g., switch throughput computed from per-port byte counters), while making them available with the proper level of granularity in internal databases acting as “Service and Function Registry”. More specifically, such databases contain descriptive information (e.g., IP addresses of end-points, identifier and port number of intermediate network nodes) as well as operational data about the data delivery (segment) paths established throughout the network realizing (part of) running service chain instances, e.g., delay on a per-chain basis. Such data are elaborated by the adaptation module that is in charge of triggering the service data delivery control and coordination block for the adaptation of (part of) service paths throughout different set of switches (*i.e.*, redirection) as soon as a degradation event is detected (e.g., switch load exceeding a certain threshold).

Without lack of generality, the only orchestration and adaptability at the level of the data service delivery path is addressed by the SDN orchestrator prototype. The adaptability also at the level of VNFs and the respective lifecycle management via a cloud management platform is considered as future work. Further details on the prototype design, functional blocks and implementation can be found in [48].

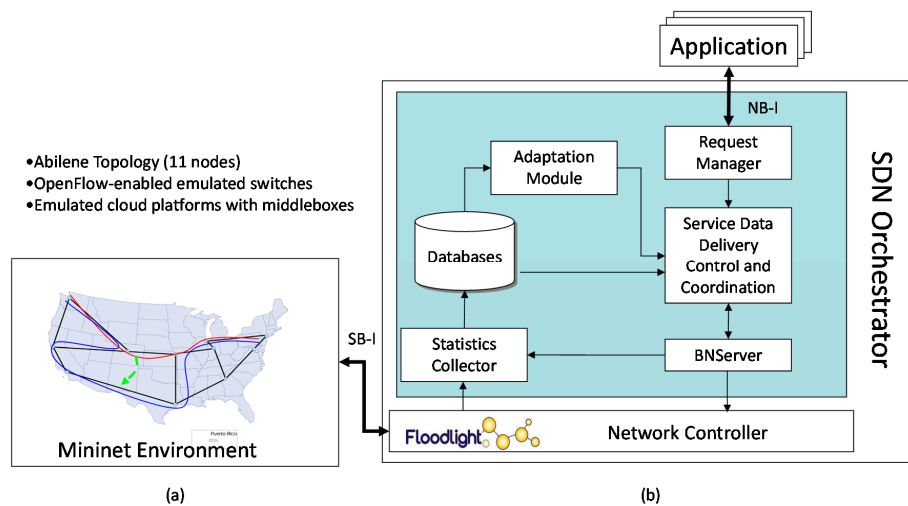


Figure 6. Prototype design and emulated testbed environment. (a) Emulation environment adopting the Mininet tool and the Abilene network topology; (b) architecture of the SDN Orchestrator prototype.

6.2. Experimental Results

In this subsection we describe the tests that we carried out to verify, first, the operation of the prototype and, then, the advantages and disadvantages of the orchestration approach, as implemented in the prototype. Figure 6a shows the adopted emulation environment, which uses Mininet, a network emulator for deploying large networks [49]. In line with other current works in literature [43], we have used the Abilene topology which is a high-performance backbone network created by the Internet2 community [50] as a reference network. Thus, 11 switches have been deployed and collocated at as many nodes as the Abilene topology. A subset of the switches are connected to emulated cloud platforms that contain the middleboxes, *i.e.*, VNF cloud, whether the remaining switches are simply transit switches. The emulated cloud platforms are supposed to contain the same types of middlebox services. For each middlebox service instance (*i.e.*, VNF) to select, we exploit the Dijkstra’s shortest-path algorithm provided by the Floodlight controller to find the network path connected to the closest switch associated with a cloud platform. Each instance of middlebox, *i.e.*, VNF, specified in the request is selected from different cloud platforms. Moreover, all the nodes in the topology are randomly chosen to behave as a source/destination of the service delivery path requests.

In the first round of experiments we used 5 bursts of 100 requests each to collect results on the general performance of the SDN orchestrator. We fixed the number of switches connected to a cloud platform to five (*i.e.*, five out of 11 switches connected to a VNF cloud), we vary the number of middlebox services in the request (*i.e.*, VNFs in the chain) and we measured the average number of flow entries in the switches and the network set-up time (*i.e.*, time required to setup the flow entries in all the switches across the chosen path). From Table 2, we notice that both the number of flow entries installed in these switches (truncated value) and the network set-up time increases with the number of required services in a chain due to the higher number of traversed switches (from two to five) and, thus, to the higher number of flow entries to be setup to forward the traffic along the data delivery path. Then, we fixed the number of requested virtual functions in a chain to three and we vary the number of VNF clouds in the network. From Table 3, we notice that increasing the number of available cloud platforms and spreading them on different switches decrease the number of paths traversing the same switch. This alleviates the load (*i.e.*, in terms of number of flow entries) on the switches and prevents from possible failures or packets loss. Additionally, the network setup time decreases as the more VNF clouds are available in the network while showing to be less dependent on respect the number of VNF clouds. In fact, in this case the total path length differs just due to the decreasing need to traverse additional transit switches because the number of VNFs in the chain is the same.

Table 2. Orchestrator performance with different number of VNFs in the chain and with five VNF clouds.

Number of VNFs in the Chain	Average Number of Flow Entries	Network Set-up Time
3	153	36.32 ms
5	253	50.14 ms

Table 3. Orchestrator performance with different number of VNF clouds and with three VNFs in the chain.

Number of VNF Clouds in the Network	Average Number of Flow Entries	Network Set-up Time
3	249	43.64 ms
6	132	36.05 ms

In the second round of experiments we evaluated the effectiveness of the orchestration feature in terms of level of usage of switches and in terms of overhead due to data delivery path redirections. To this purpose, we carried out five tests using sequences of 100 requests. Each request is characterized by a random source, a random destination and 4 VNFs. The number of switches connected to a VNF cloud is equal to five. The request follows a Poisson distribution characterized by an average inter-arrival time of 20 s and an average flow duration of 20 s.

Table 4 shows the average number of transmitted bytes along with the standard deviation related to the switches co-located with the VNF clouds. We can observe that the redirection of path has beneficial effects on the overall load of switches since the number of transmitted bytes is more fairly-distributed among all the available switches. This is in line with the IETF guidelines for the control functionalities governing the service function chaining [51]. It is worth highlighting that while addressing data delivery performance, our approach preserves the perceived quality of the services since no packets are lost during the tear-down of the data delivery paths that are rapidly re-established through unloaded switches. However, this enhanced performance is obtained at the cost of the increased number of exchanged messages and the presence of redirection time, as shown in Table 5 in comparison with a case without redirection considered in the literature [43].

Table 4. Switch load with and without path redirection.

Redirection Option	Average Number of Transmitted Bytes	Standard Deviation
With redirection feature	23 MBytes	5.56
Without redirection feature	43 MBytes	34.43

The redirection feature increases the number of messages exchanged internally since every time a redirection is performed, a new path is calculated and new flow entries are setup which necessitates further communication messages (e.g., path delivery set-up, path teardown, etc.). Moreover, these operations require a certain time, evaluated to 0.3 s in these tests, which is acceptable with respect to the overall flow duration.

Table 5. Redirection overhead.

Redirection Option	Number of Exchanged Messages	Redirection Time
With redirection feature	11,195	30 ms
Without redirection feature	3449	0

7. Conclusions

In this paper we discussed how service-oriented principles can be applied to effectively orchestrate virtualized network resources, *i.e.*, VNFs, to provide dynamically-established VNF chains while taking

full advantage of programmable data forwarding capabilities provided by SDN to adaptively deliver data throughout VNFs. The level of abstraction introduced by SOA principles provides a generalized mechanism for composing heterogeneous resources (in the computing, as well as in the networking domain) across different providers to provide users with an enhanced service experience. Moreover, the exposure of VNFs as independent services advertised through a service contract description promotes the delivery of complex network services implemented as a composition of dynamically-selected and bound VNFs. In addition, composition adaptation and service selection and substitution mechanisms can be put in place to assure the provision of QoS-aware dynamic adaptive services. We also present current results of our ongoing activities in the implementation of a prototype to validate the proposed approach. In the near future we plan to extend the prototype with VNF and virtual infrastructure management capabilities.

This approach can give several benefits to NSPs: a faster innovation speed of their network infrastructure without the need of radically changing hardware systems, reduction in the time and the cost for rolling out new services, adaptive network service provisioning and, finally, new business opportunities fostered by a dynamic environment of multi-provider cooperation.

The proposed approach poses a number of challenges for NFV and SDN research areas. Firstly, efficient mechanisms for the automated orchestration of VNF instances and their placement are needed for adaptively addressing service requirements. Indeed, optimization algorithms would be required to provide the best set of VNF instances (in case of orchestration) and the set of locations where to put VNFs (in case of placement) while minimizing a cost function, e.g., overall latency along the VNF path. Secondly, in the SDN area the main envisioned challenge is how to provide dynamic and granular traffic steering capabilities while scaling at the tenant or application flow level. In both areas, high-scale monitoring functions are required for tracing the actual service availability and properly trigger service recovery operations. Moreover, such data need to be exposed to the orchestration functions with the proper level of abstraction and granularity to keep the problem scalable [2].

Acknowledgments: We acknowledge Molka Gharbaoui and Ahmed Mohammed Ali for their technical assistance.

Author Contributions: The authors equally contributed to this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABNO	Application-Based Network Operations
DPI	Deep Packet Inspection
ETSI	European Telecommunication Standard Institute
FE	Functional Entity
GGSN	Gateway GPRS (General Packet Radio Service) Service Node
HLR	Home Location Register
HSS	Home Subscriber Server
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP/MPLS	Internet Protocol/Multi-Protocol Label Switching
MB	middlebox
NE	Network Element
NFV	Network Function Virtualization
NGSON	Next Generation Service Overlay Network
NSP	Network Service Providers
PDN-GW	Packet Data Network Gateway
SDN	Software-Defined Network
SGSN	Serving GPRS (General Packet Radio Service) Service Node (aka Serving GPRS Support Node)
SOA	Service Oriented Architecture
OTT	Over-The-Top
VN	Virtual Network
VNF	Virtual Network Function
WOC	WAN optimization controller
4G	4 Generation

References

1. Han, B.; Gopalakrishnan, V.; Ji, L.; Lee, S. Network function virtualization: Challenges and opportunities for innovations. *IEEE Commun. Mag.* **2015**, *53*, 90–97. [[CrossRef](#)]
2. GEx Multi-Domain Service Creation—From 90 Days to 90 Minutes. White Paper. March 2016. Available online: <http://www.5gex.eu/wp/wp-content/uploads/2016/03/5GEx-White-Paper-v1.pdf> (accessed on 20 April 2016).
3. John, W.; Pentikousis, K.; Agapiou, G.; Jacob, E.; Kind, M.; Manzalini, A.; Meirosu, C. Research directions in Network Service Chaining. In Proceedings of the 2013 IEEE SDN for Future Networks and Services (SDN4FNS), Trento, Italy, 11–13 November 2013; pp. 1–7.
4. Nunes, B.; Mendonca, M.; Nguyen, X.; Obraczka, K.; Turletti, T. A survey of Software-Defined Networking: Past, present, and future of Programmable Networks. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1617–1634. [[CrossRef](#)]
5. European Telecommunications Standard Institute (ETSI). *Network Functions Virtualisation (NFV); Architectural Framework*; GS NFV 002 V1.2.1; ETSI: Sophia Antipolis Cedex, France, 2014.
6. Latre, S.; Famaey, J.; de Turck, F.; Demeester, P. The fluid internet: Service-centric management of a virtualized future internet. *IEEE Commun. Mag.* **2014**, *52*, 140–148. [[CrossRef](#)]
7. Duan, Q.; Yan, Y.; Vasilakos, A.V. A Survey on Service-Oriented Network Virtualization toward convergence of networking and cloud computing. *IEEE Trans. Netw. Serv. Manag.* **2012**, *9*, 373–392. [[CrossRef](#)]
8. Erl, T. *SOA, Principles of Service Design*; Prentice Hall: Upper Saddle River, NJ, USA, 2008.
9. Branca, G.; Atzori, L. A survey of SOA technologies in NGN Network Architectures. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 644–661. [[CrossRef](#)]
10. Pentikousis, K.; Meirosu, C.; Lopez, D.R.; Denazis, S.; Shiimoto, K.; Westphal, F.J. Guest editorial: Network and service virtualization. *IEEE Commun. Mag.* **2015**, *53*, 88–89. [[CrossRef](#)]
11. Paganelli, F.; Ulema, M.; Martini, B. Context-aware service composition and delivery in NGSONs over SDN. *IEEE Commun. Mag.* **2014**, *52*, 97–105. [[CrossRef](#)]
12. ETSI. *Network Functions Virtualisation (NFV); Management and Orchestration*; GS NFV-MAN 001 V1.1.1; ETSI: Sophia Antipolis Cedex, France, 2014.
13. ETSI. *Network Functions Virtualisation (NFV); Infrastructure; Network Domain*; GS NFV-INF 005 V1.1.1; ETSI: Sophia Antipolis Cedex, France, 2014.
14. ETSI. *Network Functions Virtualisation (NFV); Ecosystem*; Report on SDN Usage in NFV Architectural Framework; GS NFV-EVE 005 V1.1.1; ETSI: Sophia Antipolis Cedex, France, December 2015.
15. Quinn, P.; Nadeau, T. Problem Statement for Service Function Chaining. RFC 7498. April 2015. Available online: <https://tools.ietf.org/html/rfc7498> (accessed on 20 April 2016).
16. Network Function Virtualization Research Group. Available online: <https://irtf.org/nfvrg> (accessed on 6 May 2016).
17. King, D.; Farrel, A. A PCE-Based Architecture for Application-Based Network Operations. Available online: <http://tools.ietf.org/html/rfc7491> (accessed on 31 May 2016).
18. IEEE Std. 1903–2011, Standard for the Functional Architecture of Next Generation Service Overlay Networks. Available online: <https://standards.ieee.org/findstds/standard/1903-2011.html> (accessed on 6 May 2016).
19. IEEE Software Defined Networks (SDN). Available online: <http://sdn.ieee.org/> (accessed on 6 May 2016).
20. IEEE. Working Toward the Next Generation of Networks. Available online: <http://theinstitute.ieee.org/benefits/standards/working-toward-the-next-generation-of-networks> (accessed on 6 May 2016).
21. Open Networking Foundation. SDN Architecture—Issue 1 TR-502, June 2014. Available online: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf (accessed on 20 April 2016).
22. Open Networking Foundation. SDN Architecture—Issue 1.1 TR-521, January 2016. Available online: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-521_SDN_Architecture_issue_1.1.pdf (accessed on 20 April 2016).
23. Open Networking Foundation. Available online: <https://www.opennetworking.org/technical-communities/areas/services/2860-cross-stratum-orchestration-cso> (accessed on 6 May 2016).

24. Open Networking Foundation. OpenFlow-Enabled SDN and Network Functions Virtualization. Available online: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-sdn-nvf-solution.pdf> (accessed on 20 April 2016).
25. Broadband Forum. Available online: <https://www.broadband-forum.org/> (accessed on 6 May 2016).
26. Framework 15.0 Foundational Studies. Available online: <https://www.tmforum.org/zoom/framework-15-0-foundational-studies> (accessed on 6 May 2016).
27. Rosa, R.V.; Silva Santos, M.A.; Rothenberg, C.E. MD2-NFV: The case for multi-domain distributed network functions virtualization. In Proceedings of the IEEE 2015 International Conference and Workshops on Networked Systems (NetSys), Cottbus, Germany, 9–12 March 2015; pp. 1–5.
28. Lopez, V.; de Dios, O.G.; Fuentes, B.; Yannuzzi, M.; Fernández-Palacios, J.P.; Lopez, D. Towards a network operating system. In Proceedings of the 2014 Optical Fiber Communication Conference, San Francisco, CA, USA, 9–13 March 2014; pp. 1–3.
29. Naudts, B.; Tavernier, W.; Verbrugge, S.; Colle, D.; Pickavet, M. Deploying SDN and NFV at the speed of innovation: Toward a new bond between standards development organizations, industry fora, and open-source software projects. *IEEE Commun. Mag.* **2016**, *54*, 46–53. [[CrossRef](#)]
30. Giannoulakis, I.; Kafetzakis, E.; Xylouris, G.; Gardikis, G.; Kourtis, A. On the applications of efficient NFV management towards 5G networking. In Proceedings of the 1st International Conference on 5G for Ubiquitous Connectivity (5GU), Akaslompolo, Finland, 26–28 November 2014; pp. 1–5.
31. Sonkoly, B.; Szabo, R.; Jocha, D.; Czentye, J.; Kind, M.; Westphal, F.J. UNIFYing cloud and carrier network resources: An architectural view. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–7.
32. Giotis, K.; Kryftis, Y.; Maglaris, V. Policy-based orchestration of NFV services in Software-Defined Networks. In Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), London, UK, 13–17 April 2015; pp. 1–5.
33. Muñoz, R.; Vilalta, R.; Casellas, R.; Martinez, R.; Szyrkowicz, T.; Autenrieth, A.; López, D. Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks [invited]. *IEEE/OSA J. Opt. Commun. Netw.* **2015**, *7*, B62–B70. [[CrossRef](#)]
34. Soares, J.; Goncalves, C.; Parreira, B.; Tavares, P.; Carapinha, J.; Barraca, J.P.; Sargento, S. Toward a TELCO cloud environment for service functions. *IEEE Commun. Mag.* **2015**, *53*, 98–106. [[CrossRef](#)]
35. Garay, J.; Matias, J.; Unzilla, J.; Jacob, E. Service description in the NFV revolution: Trends challenges and a way forward. *IEEE Commun. Mag.* **2016**, *54*, 68–74. [[CrossRef](#)]
36. Mehraghdam, S.; Keller, M.; Karl, H. Specifying and placing chains of virtual network functions. In Proceedings of the 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, 8–10 October 2014; pp. 7–13.
37. Moens, H.; de Turck, F. VNF-P: A model for efficient placement of virtualized network functions. In Proceedings of the 2014 10th International Conference on Network and Service Management (CNSM), Rio de Janeiro, Brazil, 17–21 November 2014; pp. 418–423.
38. Abujoda, A.; Papadimitriou, P. MIDAS: Middlebox discovery and selection for on-path flow processing. In Proceedings of the 7th IEEE International Conference on Communication Systems and Networks (COMSNETS 2015), Bangalore, India, 6–10 January 2015.
39. Ferrer Riera, J.; Hesselbach, X.; Escalona, E.; Garcia-Espin, J.A.; Grasa, E. On the complex scheduling formulation of virtual network functions over optical networks. In Proceedings of the 2014 16th International Conference on Transparent Optical Networks (ICTON), Graz, Austria, 6–10 July 2014; pp. 1–5.
40. Lombardo, A.; Manzalini, A.; Riccobene, V.; Schembra, G. An analytical tool for performance evaluation of software defined networking services. In Proceedings of the 2014 IEEE Network Operations and Management Symposium (NOMS), Krakow, Poland, 5–9 May 2014; pp. 1–7.
41. Stal, M. Using architectural patterns and blueprints for service-oriented architecture. *IEEE Softw.* **2006**, *23*, 54–61. [[CrossRef](#)]
42. Matias, J.; Garay, J.; Toledo, N.; Unzilla, J.; Jacob, E. Toward an SDN-enabled NFV architecture. *IEEE Commun. Mag.* **2015**, *53*, 187–193. [[CrossRef](#)]

43. Zhang, Y.; Beheshti, N.; Beliveau, L.; Lefebvre, G.; Manghirmalani, R.; Mishra, R.; Patney, R.; Shirazipour, M.; Subrahmaniam, R.; Truchan, C.; *et al.* StEERING: A software-defined networking for inline service chaining. In Proceedings of the 2013 21st IEEE International Conference on Network Protocols (ICNP), Goettingen, Germany, 7–10 October 2013.
44. Xu, W.; Jiang, Y.; Zhou, C. Data Models for Network Functions Virtualization. Available online: <https://datatracker.ietf.org/doc/draft-xjz-nfv-model-datamodel/> (accessed on 31 May 2016).
45. Seedorf, J.; Burger, E. Application-Layer Traffic Optimization (ALTO) Problem Statement. Available online: <https://tools.ietf.org/html/rfc5693> (accessed on 31 May 2016).
46. Floodlight Openflow Controller. Available online: <http://www.projectfloodlight.org/floodlight/> (accessed on 6 May 2016).
47. Openflow Specifications. Available online: <http://www.openflow.org> (accessed on 6 May 2016).
48. Mohammed, A.A.; Gharbaoui, M.; Martini, B.; Paganelli, F.; Castoldi, P. SDN controller for Network-aware Adaptive Orchestration in Dynamic Service Chaining. In Proceedings of the NetSoft 2016, Seoul, Korea, 6–10 June 2016. (to appear).
49. Mininet. Available online: <http://www.mininet.org> (accessed on 6 May 2016).
50. Abilene Network. Available online: <https://en.wikipedia.org/wiki/AbileneNetwork> (accessed on 6 May 2016).
51. Boucadair, M., Ed.; Service Function Chaining (SFC) Control Plane Components & Requirements. Available online: <https://tools.ietf.org/html/draft-ww-sfc-control-plane-04> (accessed on 6 May 2016).



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).