future internet

MDPI

Article

Effectiveness of Segment Routing Technology in Reducing the Bandwidth and Cloud Resources Provisioning Times in Network Function Virtualization Architectures †

Vincenzo Eramo [1,*], Francesco G. Lavacca [2], Tiziana Catena [1], Marco Polverini [1] and Antonio Cianfrani [1]

[1] DIET, ”Sapienza” University of Rome-Via Eudossiana 18, 00184 Rome, Italy; tiziana.catena@uniroma1.it (T.C.); marco.polverini@uniroma1.it (M.P.); antonio.cianfrani@uniroma1.it (A.C.)
[2] Fondazione Ugo Bordoni, Viale del Policlinico 147, 00161 Roma, Italy; francescogiacinto.lavacca@uniroma1.it
* Correspondence: Vincenzo.Eramo@uniroma1.it; Tel.: +39-06-44585372

† This paper is an extended version of paper “Proposal and Investigation of a Scalable NFV Orchestrator Based on Segment Routing Data/Control Plane” published in Proceedings of the 14th International Conference on Network and Service Management (CNSM), Rome, Italy, 5–9 November 2018.

Received: 31 December 2018; Accepted: 5 March 2019; Published: 12 March 2019

Abstract: Network Function Virtualization is a new technology allowing for a elastic cloud and bandwidth resource allocation. The technology requires an orchestrator whose role is the service and resource orchestration. It receives service requests, each one characterized by a Service Function Chain, which is a set of service functions to be executed according to a given order. It implements an algorithm for deciding where both to allocate the cloud and bandwidth resources and to route the SFCs. In a traditional orchestration algorithm, the orchestrator has a detailed knowledge of the cloud and network infrastructures and that can lead to high computational complexity of the SFC Routing and Cloud and Bandwidth resource Allocation (SRCBA) algorithm. In this paper, we propose and evaluate the effectiveness of a scalable orchestration architecture inherited by the one proposed within the European Telecommunications Standards Institute (ETSI) and based on the functional separation of an NFV orchestrator in Resource Orchestrator (RO) and Network Service Orchestrator (NSO). Each cloud domain is equipped with an RO whose task is to provide a simple and abstract representation of the cloud infrastructure. These representations are notified of the NSO that can apply a simplified and less complex SRCBA algorithm. In addition, we show how the segment routing technology can help to simplify the SFC routing by means of an effective addressing of the service functions. The scalable orchestration solution has been investigated and compared to the one of a traditional orchestrator in some network scenarios and varying the number of cloud domains. We have verified that the execution time of the SRCBA algorithm can be drastically reduced without degrading the performance in terms of cloud and bandwidth resource costs.

Keywords: network function virtualization; scalable orchestrator; segment routing; software defined network

1. Introduction

The Network Function Virtualization (NFV) concept is one of the most appealing research outcomes for network operators. The research activity on NFV has been focused on the solution of the SFC Routing and Cloud and Bandwidth resource Allocation (SRCBA) problem [1]; it is composed of two subproblems: (i) the Service Function Chaining (SFC) routing and (ii) the allocation of network and computing resources for the execution of Virtual Network Function Instances (VNFI). The performance indexes used to characterize an SRCBA algorithm are the number of SFCs accepted, the server power consumption [2], the bandwidth used to interconnect the VNFIs [3], the cloud resource amount needed to activate the VNFIs and so on [4–6].

All of these solutions assume that the NFV orchestrator, responsible for implementing the SRCBA algorithm, knows in detail the network topology and the available resources in the datacenter in which the VNFIs have to be instantiated. This leads to a high computational complexity of the SRCBA algorithm increasing versus the number of servers and switches located in the datacenters [7].

Furthermore, this traditional solution may not be applicable in multi-provider NFV environments that implement business models with the following two players [6,8–10]: the Cloud Infrastructure Provider (CInP) and the Internet Service Provider (ISP). The function of an CInP is to deploy and manage the physical resources on which the virtual resources (i.e., Virtual Machine) may be provisioned. The function of an ISP is to rent the physical resources from one or more than one CInP to execute the VNFIs. It provides interconnecting VNFIs through bandwidth resources to create services to the end users. It also manages the NFV Orchestrator that acquires from the CInP through the Virtual Infrastructure Manager (VIM), information about the network topology and the available resources (bandwidth, processing, disk memory) of the Cloud Infrastructure. When CInP and ISP are two different players, the CInP could need to hire from the ISP some of this information and only a partial description could be sent to the NFVO. For this reason, the application of classical SRCBA algorithms is not possible in a Multi-Provider NFV environment.

In this paper, we propose and investigate a scalable NFV orchestration architecture inherited by ETSI [11]. The functions of the NFVO are split into two functional entities: (i) the Resource Orchestrator (RO) whose role is to orchestrate the resources; and (ii) the Network Service Orchestrator (NSO) whose role is to orchestrate the network service. In the proposed solution, one RO is placed in each Cloud domain and its role is to simplify and abstract the Cloud Infrastructure of which the the VIM notifies him. The RO only notifies the NSO—information about the amount of available cloud resources without providing any detail on the network topology of the datacenter and the available number of servers. This abstraction process implemented by the ROs allows for a simplification of the input data of the SRCBA algorithm and a drastic decrease in its computational complexity.

A further contribution of the paper is to propose the use of Segment Routing (SR) [12–14] to implement the data plane of our architecture. SR is a novel technology allowing the implementation of advanced TE features with a reduced complexity. In this work, we exploit the features of SRv6, i.e., SR over IPv6, to simplify the management of the proposed NFV orchestration solution. In more detail, we make use of the Binding Segment Identifier (SID) tool to hide the complexity of datacenter routing without requiring the use of different technologies, such as NSH or SDN.

The main innovative contributions of the paper are the following:

- the proposal of a scalable and ETSI compliant orchestration architecture in Multi-Provider NFV environments;
- the definition of a SRv6-based data plane for the SFC routing applicable for the proposed scalable orchestration;
- the proposal of an ad hoc algorithm for the proposed orchestration architecture and the investigation of its computational complexity;
- the comparison in terms of computational complexity of the proposed solution with respect to the one of a traditional orchestration.

This paper is an extension of [15] in which only some preliminary results are presented. Conversely, in this paper, we add the following contributions: (i) a description of ETSI compliant NFV orchestration architecture is illustrated with the introduction of a architectural solution in which the Network Function Virtualization Orchestrator (NFVO) is split in RO and NSO; Segment Routing is extended to the Network Function Virtualization Infrastructure- Point-of-Presence (NFVI-PoP) domain with the introduction of the Binding Segment IDentifier (BSID) to hide the routing management to the RO, without requiring the use of Software Defined Networks (SDN); (ii) the SRCBA algorithms for the traditional and proposed orchestration solutions are highlighted and their computational complexity is evaluated; (iii) the comparison results between traditional and proposed orchestration solutions are also reported for a long distance network such as NSFNET.

The paper is organized as follows: the Related Work is reported in Section 2. We illustrate the scalable orchestration solution in Section 3. A data plane based on Segment Routing technology and a SRCBA algorithm for the proposed NFV orchestration architecture are shown in Sections 4 and 5, respectively. The advantages in terms of reduction in computational complexity are evaluated in Section 6. Finally, conclusions and future research items are described in Section 7.

2. Related Work

The NFV resource allocation and routing problem has been studied in [16] by formulating it as a multi-commodity-chain network design problem on a cloud-augmented graph. The paper investigates a static traffic scenario and adopts a simple model in which the sharing of VNFI among service functions and their maximum resource availability are not considered.

Some solutions have been proposing bandwidth and cloud resource allocation in dynamic traffic scenarios [17–19]. Ghaznavi et al. [20,21] design online algorithms to optimize placement of VNF instances in response to on demand workload, considering the trade-off between bandwidth and cloud resource consumption. Eramo et al. [1] introduce VM migration policies to save the server power consumption and by limiting the QoS degradation or the energy consumption occurring during the migrations. Wang et al. [22] propose a procedure to evaluate the number of VNFIs over time with the objective to minimize the cloud resources used; their study does not consider the cost of the bandwidth resources. Liu et al. [23] study the problem of reallocation of VNFIs in a dynamic traffic scenario by constraining the number of reallocations so as to limit the deployment costs; they determine the number and the location of the VNFIs that are executed in Virtual Machines (VM) equipped with a pre-determined number of cores dependent on the type of VNF supported by the VNFI. Wang et al. [10] study cloud and bandwidth resource allocation problems in geographically-distributed data centers interconnected by elastic optical networks; the scenario considered suits the business model of the NFV ecosystem very well.

Pei et al. [24] and Cappanera et al. [25] achieve good performance for the SFC routing and VNFI resource allocation problems by modeling the network as a multi-layer graph.

In the case of ETSI NFV architecture [26], the algorithms for the cloud and bandwidth resource allocation are implemented in the Network Function Virtualization Orchestrator (NFVO). The NFVO receives from the Virtual Infrastructure Managers (VIM) of the Network Function Virtualization Infrastructure-Point of Presence (NFVI-PoPs) information about the available cloud and bandwidth resources and the NFVI-PoP network topologies. Most of the solutions proposed in literature assume that the VIMs expose a complete description of the NFVI-PoP network topologies with the consequence of drastically increasing the computational complexity of the SRCBA algorithm. Only a few papers propose more scalable solutions [23] in which the NFVO has a limited and simplified visibility of the NFVI-PoP network topology. However, none of these works propose an NFV orchestration architectural solution able to support the scalable algorithms proposed. In this paper, we employ one of the architectural options proposed within the ETSI [11] to support a solution in which: (i) each NFVI-PoP domain is equipped with a Resource Orchestration (RO) whose role is to abstract the NFVI-PoP network topology exposed by the VIM; and (ii) a Network Service Orchestrator (NSO) has the role of deciding where to allocate the resources and to route the SFCs. In addition, we also illustrate how the application of a Segment Routing technology allows for a reduction in the addressing overhead of the VNFIs.

Several research works proposed the use of Segment Routing in the context of the Service Function Chaining, but none of them considered SR as an enabler for the implementation of a scalable orchestrator architecture. Ref. [27] exploits the network programming feature of SR to implement a load balancer able to distribute the computation load over a set of servers, by defining a new SR function (named service hunting) to be performed at the gateway. SRV6Pipes [28] is an extension of the IPv6 implementation of Segment Routing in the Linux kernel [29] that enables chaining and operation of in-network functions operating on streams, such as transparent proxies, transparent compression or encryption, transcoding, etc (all the functions that cannot be performed on per-packet based). The authors of [30] investigate the problem of applying Virtual Network Functions on SR encapsulated packets. First, a definition of SR aware and unaware applications is provided and then a new element, called SR Proxy, is introduced in order to allow the use of SR unaware applications in the realization of SFCs with SR. The SERA firewall, presented in [31], is the first example of an SR aware virtual network function. Its main feature is the possibility to consider specific SRH fields to perform the filtering actions typical of a firewall application. SR network programming feature is used in [32] to realize a zero loss VM migration tool. It is based on the definition of new SR functions that modify the forwarding behaviour of the routers according to local conditions (e.g., the VM is locally present).

3. Proposal of a Scalable and ETSI Compliant NFV Orchestration Architecture

The network scenario considered is the one of an Internet Service Provider (ISP), composed of a core network interconnected to multiple datacenters. The datacenters, hosting computing resources for the execution of VNFIs can be managed by the ISP or by different CInPs. The orchestration architecture is based on the ETSI standard [26] described in Figure 1. It details the key functional blocks in NFV systems: Virtualized Network Function (VNF), NFV Infrastructure (NFVI) and NFV Management and Orchestration (MANO) framework [26]. The MANO is comprised of a Virtualized Infrastructure Manager (VIM), VNF Manager (VNFM) and NFV Orchestrator (NFVO). The communication between various functional blocks in NFV architectural framework is enabled through a set of well-defined reference points.

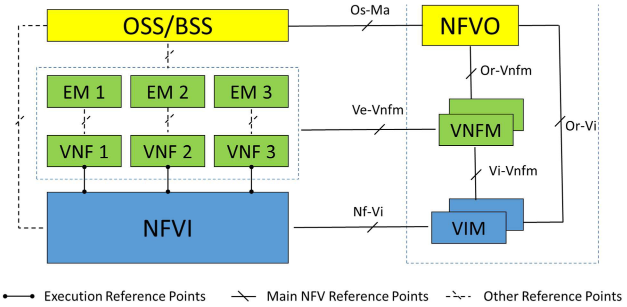


Figure 1. ETSI NFV standard.

The VNF Instance is the software implementation of a network function. It runs on NFV Infrastructure (NFVI) that encompasses the necessary software and hardware components. NFVI can span geographically distributed locations, called NFVI Point of Presences (NFVI-PoPs). NFVI resources (e.g., compute, storage and network) can be managed and controlled by one or more VIMs. The VIM is in charge of instantiating and managing the cloud resources (processor cores, RAM memory, disk memory) used by the VNF [33]. The VNFM is responsible for the lifecycle management (e.g., instantiation, scaling, healing and monitoring [34,35]) of one or more VNF instances. The NFVO is in charge of the orchestration of NFVI resources and lifecycle management of network services. The NFVO and VNFM work jointly to ensure that the network services and their corresponding VNFs meet the service quality requirements.

In the traditional operation mode, the NFVO acquired from the VIMs, information about the NFVI-PoP network topology and available resources as illustrated in Figure 2a, where the NFVI-PoP-1 sends to the NFVO the information about a network topology composed of four servers and three switches connected according to a tree topology; two processing cores per each server also notified the NFVO.

There are the following problems in the implementation of this solution:

- the NFVO needs to acquire a lot of information and the graph representing the entire network topology and the available cloud and bandwidth resources can become very complex;
- the SFC Routing and Cloud and Bandwidth resource Allocation (SRCBA) algorithm does not scale with the increase in both the number of servers and switches of the NFVI-PoP networks;
- the solution is not applicable in the case of multi-provider NFV environments in which the providers of each NFVI-PoP may need to hire the organization of their own NFVI-PoP network and not make it visible to the owner of the NFVO.

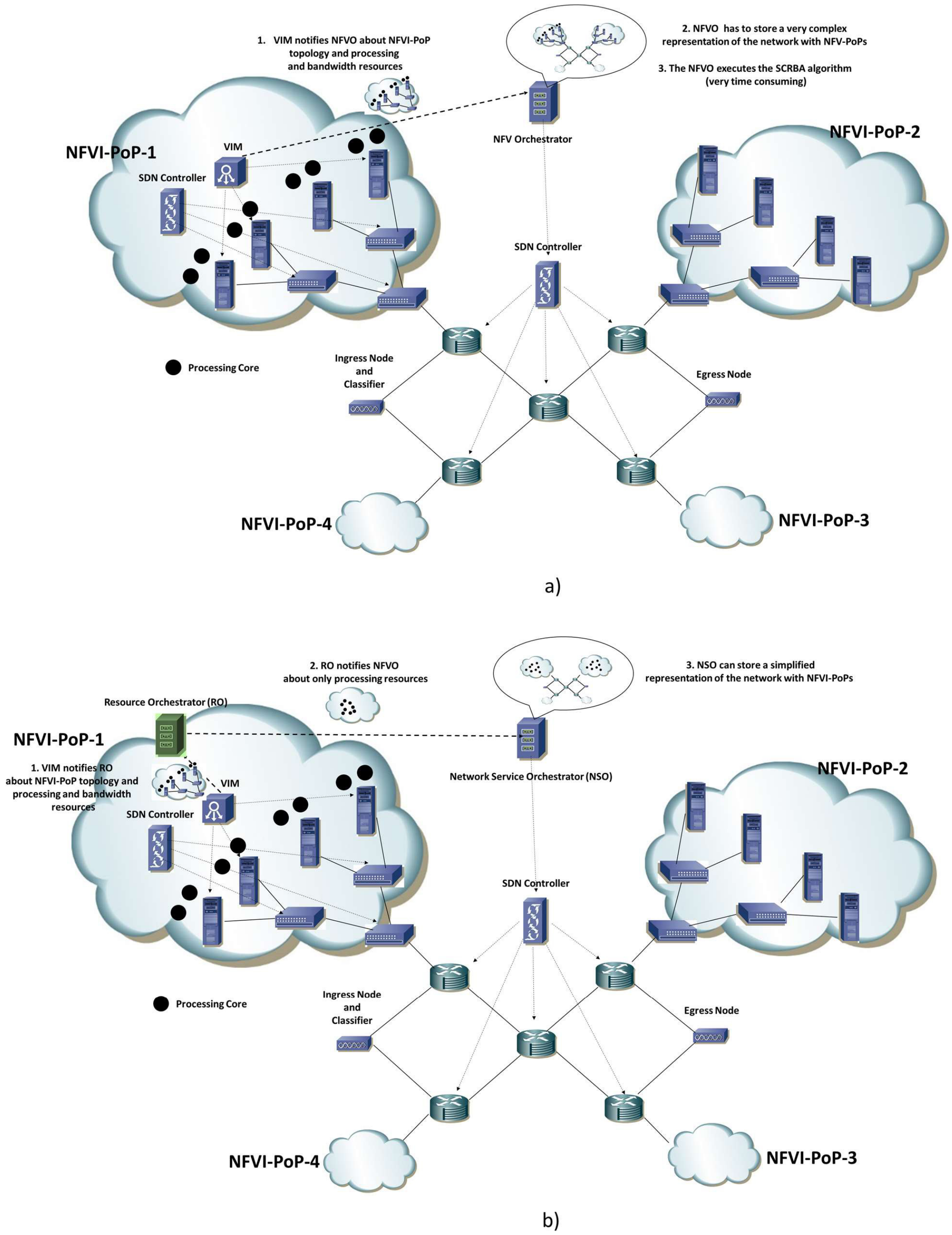

1. VIM notifies NFVO about NFVI-PoP topology and processing and bandwidth resources
2. NFVO has to store a very complex representation of the network with NFV-PoPs
3. The NFVO executes the SCRBA algorithm (very time consuming)
NFV Orchestrator
NFVI-PoP-1
VIM
SDN Controller
NFVI-PoP-2
SDN Controller
Processing Core
Ingress Node and Classifier
Egress Node
NFVI-PoP-4
NFVI-PoP-3
a)
2. RO notifies NFVO about only processing resources
3. NSO can store a simplified representation of the network with NFVI-PoPs
Resource Orchestrator (RO)
NFVI-PoP-1
1. VIM notifies RO about NFVI-PoP topology and processing and bandwidth resources
VIM
SDN Controller
Network Service Orchestrator (NSO)
NFVI-PoP-2
SDN Controller
Processing Core
Ingress Node and Classifier
Egress Node
NFVI-PoP-4
NFVI-PoP-3
b)

Figure 2. Traditional (**a**) and scalable (**b**) NFV orchestration.

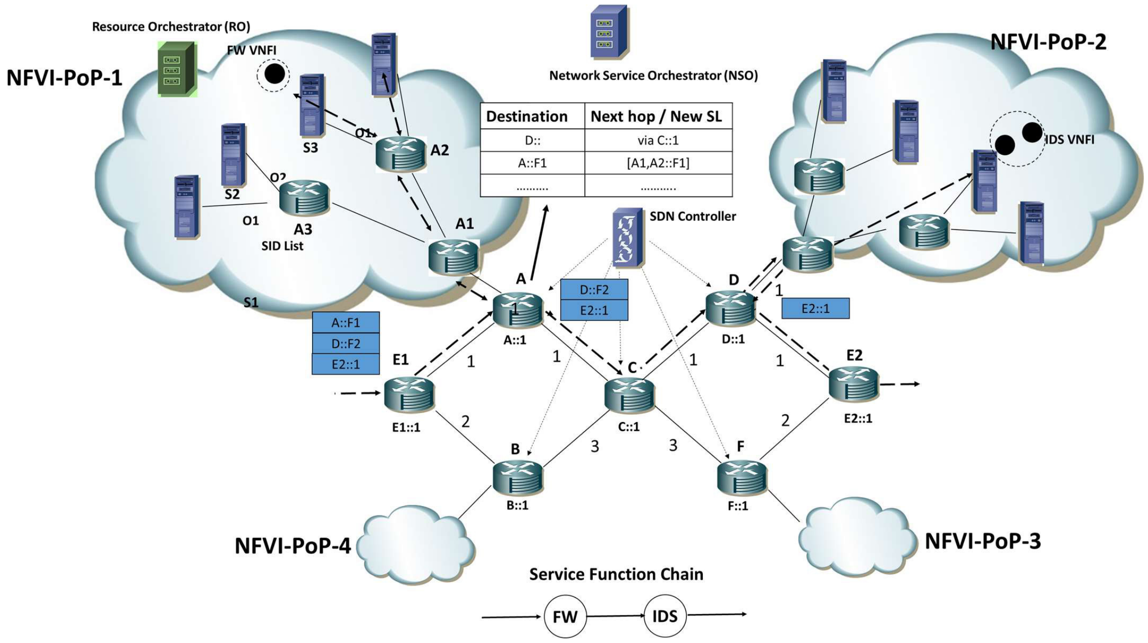
Our idea exploits the ETSI architectural option [11] in which the NFVO is split into two functional entities. The first one, referred to as a Resource Orchestrator (RO), is placed in each NFVI-PoP and has the role of a resource orchestrator. The second one, referred to as a Network Service Orchestrator (NSO), has the role of orchestrating the service. According to this functional split, we propose the scalable orchestration solution depicted in Figure 2b, where the RO abstracts the topology received by the VIM and only notifies the NSO of about the available cloud resources; the servers' status and the NFVI-PoP network topology is not reported to the NSO; for instance, the RO only notifies the NSO that eight processing cores are available and can be allocated to the VNFIs executing the Service Functions (SF) of any SFC. This solution drastically decreases the complexity of the SRCBA algorithm whose output will be the NFVI-PoPs hosting each VNFI and the amount of cloud resources to be assigned, without specifying the routing paths followed by the SFC inside the NFVI-PoP networks. The RO will be responsible for internal NFVI-PoP path computation, by trying to balance as much as possible the load of the VNFIs executing a given Service Function according to the number of processing cores assigned to the VNFIs.

4. Segment Routing Based SFC Routing

For the data plane of the architecture, we propose to use SRv6, i.e., Segment Routing over IPv6. SRv6 represents a novel technology for the packet forwarding and path encoding in a network domain (referred to as SR domain), able to provide advanced features, such as Traffic Engineering (TE) support, failure protection and Service Function Chaining support, with no need of complex flow state protocols such as Multi-Protocol Label Switching (MPLS). The SRv6 paradigm is based on explicit and source routing features: the ingress node of a SR domain is able to code the network path to be followed by a packet inserting into the SR header a Segment List, i.e., a list of Segment Identifiers (SIDs). The SID of a node (i.e., a router) is an IPv6 address used to globally identify the node in the SR domain. The distribution of the SIDs among network nodes can be performed making use of IGP routing protocol with specific extensions [36]. In this way, the Segment List is a list of nodes to be crossed by the packet; the final path will be composed of the IPv6 paths interconnecting the nodes reported in the Segment List.

In this work, we show that the NFV orchestration architecture proposed can be implemented using SRv6 and, more importantly, that, exploiting the flexibility of SRv6, it is possible to hide the routing management of the NFVI-PoP networks to the NSO. To do that, SRv6 must be supported in the NFVI-PoP networks and the Binding SID (BSID) concept must be used. A BSID is an SID associated with a specific SR policy, i.e., a new Segment List to be applied to the traffic; in our work, we introduce the Service Function-BSID (SF-BSID), an SID able to globally identify the SF running in a specific NFVI-PoP, independently of the specific instance (NFVI) where the SF is running. The SF-BSID will be managed by the NFVI-PoP hosting the NF: the new Segment List (or Segment Lists) associated with it represents the internal routing and depends on the SF allocation performed at NFVI-PoP level. In other words, SRv6 allows for abstracting the datacenter resources at the data plane level.

To better explain this concept, let us consider the example reported in Figure 3 where a seven-node core network with two NFVI-PoPs is providing SFC composed of two functions, a Firewall (FW) and an Intrusion Detection System (IDS). Each core node have its own SID: for the easiness of representation, we associate with each node an SID having as first bits the node name and as last bits 1, i.e., An SID is A::1. The FW and IDS SFs running at NFVI-PoP-1 are identified by two SIDs of router A, i.e., A::F1 and A::F2, respectively; in a similar way, D::F1 and D::F2 are the SID of FW and IDS SFs running at NFVI-PoP-2. When router A receives a packet having A::F1 as an active segment, it checks its SR forwarding table: as reported in Figure 3, A::F1 is a BSID, i.e., the new Segment List {A1::, A2::F1} is attached to the packet. In this way, the packet is forwarded into an NFVI-PoP-1 domain, where SR forwarding will allow for reaching the proper VNFI. The NFVI-PoP-1 controller is responsible for configuring and managing the SR forwarding table of A, as a result of the resource allocation strategy. In a similar way, node D will use the D::F2 BSID to associate a new Segment List to packets directed to the IDS SF. It is important to highlight that load balancing among several VNFIs executing the same SF is possible: an SF-BSID can be associated with several SLs, and a weight to each SL can be set (so that to support non-uniform balancing).



Destination	Next hop / New SL
D::	via C::1
A::F1	[A1,A2::F1]
....	....

Figure 3. A segment routing based data/control plane for the scalable NFV orchestration architecture.

In Figure 3, we show the overall set of operations performed for a packet crossing the ISP network. In the example, we consider router E1 as the Ingress router, i.e., the one responsible for packet classification and SR-based SFC routing (FW + IDS), and router E2 as Egress node. The Segment List used to encode the SFC is {A::F1, D::F2, E2::1}, where A::F1 identifies the FW SF running at NFVI-PoP-1, D::F2 identifies the IDS running at NFV-PoP-2, and E2::1 identifies the egress node E. The FW and IDS SFs are executed in VNFIs instantiated in the NFVI-PoP-1 and NFVI-PoP-2, respectively.

5. Cloud and Bandwidth Resource Allocation Problem

The cloud and bandwidth resource allocation problem in NFV architectures is Nondeterministic Polynomial Hard (NP-Hard) [1]. For this reason, we will compare the computational complexity of two heuristics for the resource allocation in NFV architectures when a Traditional Orchestration (TO) solution and the proposed Scalable Orchestration (SO) solution are considered, respectively. The objective of both the heuristics is to minimize the total cost of the allocated cloud and bandwidth resources.

For the TO case, the NFV orchestrator has the visibility of the internal structure of every NFVI-PoP (its network topology, the number of servers, ...). For this reason, a modified version of the algorithm proposed in [7] can be applied for solving the SFC routing and cloud and bandwidth resource allocation problem in the TO case; the algorithm has been modified so that the total cost of allocated cloud and bandwidth resources is only minimized. It is shown in [7] that the computational complexity of the algorithm is $C_{TO} = O(NM_{max}N_s^2 log(M_{max}N_s))$, where N is the number of offered SFCs, M_{max} is the number of SFs in the offered longest SFC and N_s denotes the number of servers in each NFVI-PoP.

Next, we illustrate an algorithm, inherited by [7,37,38], for the SFC routing and resource allocation in NFV architectures in the Scalable Orchestration case proposed in Section 3. We will introduce the main notations in Section 5.1 while the main steps of the algorithm are illustrated in Section 5.2.

5.1. NFVI-PoP, Network and Traffic Model

We represent the NFVI-PoP and network infrastructures with the graph $\bar{G} = (\bar{V}, \bar{L})$, where $\bar{V} = \bar{V}_A \bigcup \bar{V}_S \bigcup \bar{V}_{NFVI}$ is a set of nodes that represents the access nodes in which the Service Function Chains are generated/terminated ($\bar{V}_A$), the network switches ($\bar{V}_S$) and the NFVI-PoPs ($\bar{V}_{NFVI}$). $\bar{L}$ is the set of edges interconnecting the nodes of the set $\bar{V}$ and models the network links interconnecting the

NFVI-PoPs, access nodes and switches. We denote with $C_{\bar{e}}$ the capacity of the link $\bar{e} \in \bar{L}$ while c_b is the cost unit (Dollars/KmGb) of carrying one traffic Gb on any link of length 1 Km.

The NFVI-PoPs host the cloud resources needed to implement the network service. In this paper, we consider only processing resources, though our study can be easily extended to the case in which memory and disk resources are also considered. We assume that every NFVI-PoP is equipped with a total number N_{core} of processing cores. Virtual Machines (VM) are activated in the NFVI-PoP to instantiate VNFIs supporting service functions as Network Address Translation (NAT), firewall (FW), Load Balancer (LB), etc. If we denote with F the number of service function (SF) types, we assume that the VNFI hosting the i-th ($i = 1, \cdots, F$) SF is implemented with a software module providing the processing capacity $C_i^{pr,max}$ and requiring the allocation of n_i^c cores [7]. Next, we denote with c^{core} the cost unit (Dollars/hour) of renting for one hour one processing core.

We assume that the ISP has to route and to allocate resources for N SFCs—each one characterized by a 5-tuple $< \bar{v}_i, \bar{w}_i, M_i, \vec{s}_i, b_i >$ ($i = 1, \cdots, N$) where:

- $\bar{v}_i \in \bar{V}_A$ and $\bar{w}_i \in \bar{V}_A$ denote the originating and terminating access nodes the SFC;
- M_i is the number of service functions to be executed for the SFC according to a given order; we assume that the SFs belong to a set of F types;
- $\vec{s}_i$ is a 1's or 0's matrix of size $M_i \times F$; the component $s_i(j,k)$ assumes the value 1 if the j-th SF to be executed is a k-th type SF;
- b_i is the bandwidth offered by the i-th SFC.

5.2. SFC Routing and Cloud and Bandwidth Resource Allocation (SRCBA) Algorithm

The SFC Routing and Cloud and Bandwidth resource Allocation (SRCBA) algorithm is based on the following two procedures. In the first one, the N SFCs are sorted in decreasing bandwidth order. The second procedure allows for: (i) the determination of the NFVI-PoP in which the VNFIs are located; (ii) the determination of the VNFIs in which the SFs are executed; and (iii) the determination of the network paths used to interconnect the VNFIs.

The SRCBA specifies the number and the VNFI type to be allocated and the networks paths involved between NFVI-PoPs. Conversely, the cloud resource allocation in the servers and the bandwidth allocation in the NFVI-PoPs in order to interconnect servers and NFVI-PoP access nodes are delegated to another algorithm out of the scope of the paper. We can only say that the computational complexity of such an algorithm is negligible with respect to the one of the SRCBA algorithm because the application of the SRCBA algorithm involves the bulk of the computation by determining the number of VNFIs, VNFI type, VNFI locations and network paths in a domain composed of a set of NFVI-PoPs.

The objective of SRCBA is both to share the VNFIs among SFCs and to re-use them as much as possible. A new VNFI is activated only when either cloud resources are not available among the ones already active or the re-use leads to high bandwidth costs.

Next, we give further clarification about the second procedure of SRCBA. It allows for the choice of NFV-PoPs and network paths so as to minimize the sum of the cloud and bandwidth resource cost; it is applied for each SFC, it is devoted to build a multi-stage graph and to evaluate on it a least cost path that identifies the NFVI-PoPs on which the SFs of the SFCs have to be executed either by using a VNFI already activated or by activating one new VNFI. The construction of the multi-stage graph G_i^{MS} for the i-th SFC is accomplished as follows. If M_i is the number of SFs in the i-th SFC, G_i^{MS} is composed of $M_i + 2$ stages, numbered from 0 to $M_i + 1$. The 0-th and the ($M_i + 1$)-th stages are composed of one node for each one; a least cost path will be evaluated between these two nodes once the graph is built. The j-th stage contains n_j nodes, whose generic one is named $v_{j,k}$ $(j = 1, \cdots, M_i, k = 1, \cdots, n_j)$, where n_j is the number of nodes in each stage and it equals the number of NFVI-PoPs in which processing resources are available to execute the j-th SF of the i-th SFC. In particular, notice how the execution of the SF may occur on a VNFI already activated or involve the instantiation of a new VNFI. The node $v_{j,k}$ of the multi-stage graph is characterized by a cost referred to as $c_{v_{j,k}}$ and characterizing the cost of executing the j-th SF of the i-th SFC in the NFVI-PoP associated with the node $v_{j,k}$. The cost depends on whether a new VNFI needs to be activated and, in this case, it depends on the core cost c^{core}. We assume that the cost is equal to zero in the case in which a VNFI can be re-used; otherwise, the cost equals $c^{core}n_h^c$ if the SF is of the h-th type. An edge of the multi-stage graph characterizes the possibility of interconnecting the two VNFIs located in the NFVI-PoPs associated with the two nodes of the edge. The edge is labeled with the average bandwidth cost involved in carrying the bandwidth b_i on the K-Shortest Paths connecting the NFVI-PoPs associated with the two nodes of the edge.

Once the multi-stage graph has been built, a shortest path is evaluated between the 0-th and the $(M_i + 1)$-th stages. The computed path determines: (i) whether new VNFI have to be allocated; (ii) in which NFVI-PoPs the cloud resources have to allocated and the SFs of the SFC have to be executed; and (iii) between which NFVI-PoPs the bandwidth resource has to be allocated. In the allocation phase, the bandwidth allocation occurs on one out of the K-Shortest Paths.

The computational complexity of the SRCBA heuristic depends on the evaluation procedure of the shortest path in the multi-stage graph. It is easy to prove that it is given by $C_{SO} = O(NM_{max}N_{NFVI}^2 log(M_{max}N_{NFVI}))$, where N_{NFVI} denotes the number of NFVI-PoPs.

Next, we provide an example of application of the SRCBA algorithm in the case of a scenario reported in Figure 4. We consider the NFVI-PoP-1, NFVI-PoP-2 and NFVI-PoP-3, composed of two, two and one servers, respectively; the SRCBA algorithm needs to allocate cloud and bandwidth resources for an SFC composed of a Firewall (FW) and one Intrusion Detection System (IDS); we assume that one FW VNFI and one IDS VNFI have already been allocated in NFVI-PoP-1 and NFVI-PoP-3, respectively. We also report the average bandwidth costs of the network paths interconnecting the NFVI-PoPs; the interconnection costs equal 10$ and 15$ for the tuples (NFVI-PoP-1,NFVI-PoP-2) and (NFVI-PoP-2,NFVI-PoP-3) respectively; conversely, we denote with $x$$ the interconnection cost for the tuple (NFVI-PoP-1,NFVI-PoP-3). The core cost c^{core} is assumed to be equal to 0.25$; because we also assume that four and eight cores are needed to activate one FW VNFI and one IDS VNFI [21], their activation cost equals 2$ and 4$, respectively.

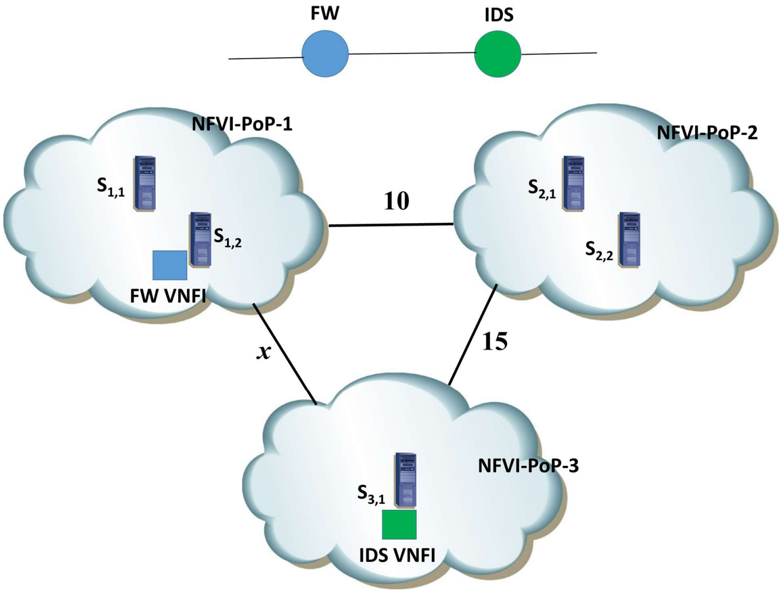
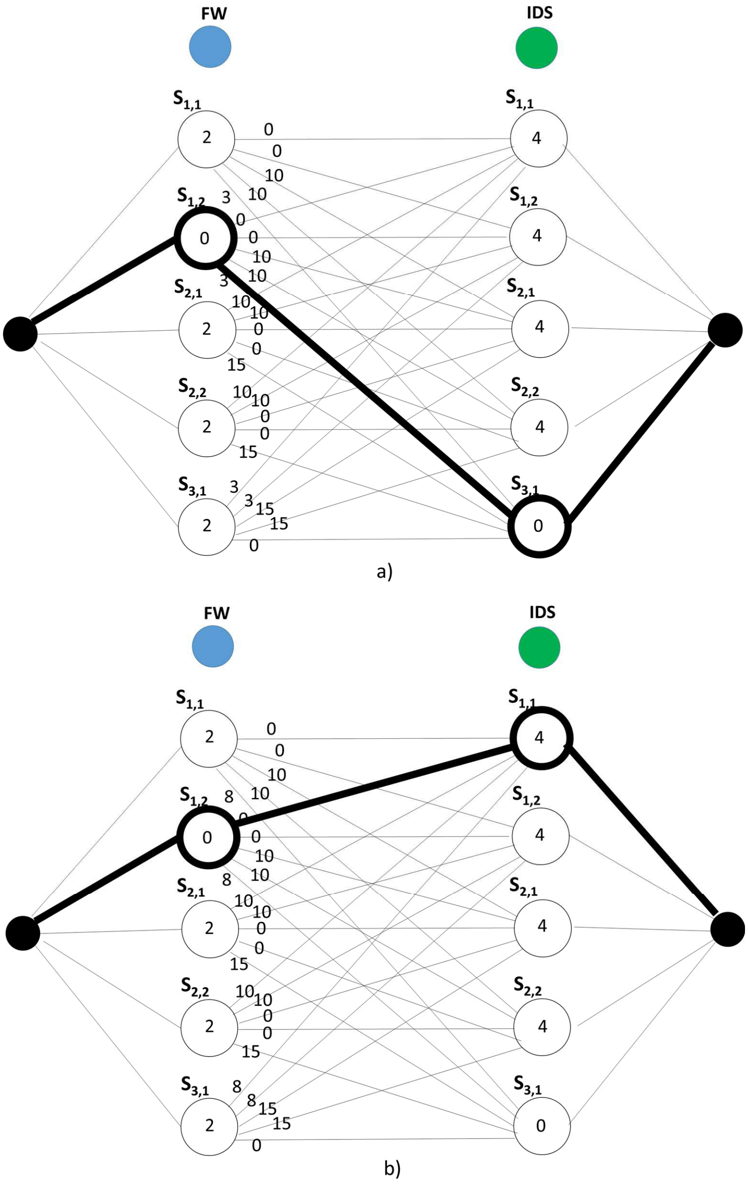


Figure 4. Application example of the SFC Routing and Cloud and Bandwidth resource Allocation (SRCBA) algorithm to the case of a cloud infrastructure composed by three Network Function Virtualization Infrastructure-Point of Presence (NFVI-PoP), that is NFVI-PoP-1, NFVI-PoP-2 and NFVI-PoP-3 equipped with two, two and one servers, respectively; one FW VNFI and one IDS VNFI are already activated in the NFVI-PoP-1 and NFVI-PoP-3, respectively. The SRCBA algorithm has to accommodate a Service Function Chain (SFC) composed of one Firewall (FW) and one Intrusion Detection System (IDS).

The application of the SRCBA algorithm in the TO solution case is illustrated in Figure 5a,b when the bandwidth interconnection cost x between NFVI-PoP-1 and NFVI-PoP-3 equals 3 and 8, respectively. As you can observe that the multi-stage graph is composed of four stages; because we assume that all of the servers have sufficient resources to support the Service Functions of the SFC, then all of the five servers are reported in the second and third stage; the nodes are labeled with the costs that equal zero for those servers in which one VNFI is already instantiated; otherwise, the activation cost of the VNFI is reported. The nodes are interconnected by edges labeled with the bandwidth costs. We report with thick lines the shortest path evaluated in the multi-stage graph and identify where the resources have to be allocated. You can observe that, when the bandwidth interconnection cost between NFVI-PoP-1 and NFVI-PoP-3 is low and equals 3, no activation of VNFI is decided and the VNFIs already instantiated are used as illustrated in Figure 6a; conversely, when the bandwidth interconnection cost is high and equals 8, in order to limit the bandwidth cost, the SFC is embedded in the NFVI-PoP-1 by activating one new IDS VNFI as highlighted in Figure 6b.



a)

b)

Figure 5. Construction of the multi-stage graph for the Scalable Orchestration solution of the SRCBA algorithm. The cases $x = 3\$$ (**a**) and $x = 8\$$ (**b**) for the bandwidth interconnection cost between NFVI-PoP-1 and NFVI-PoP-3 of Figure 4 are reported.

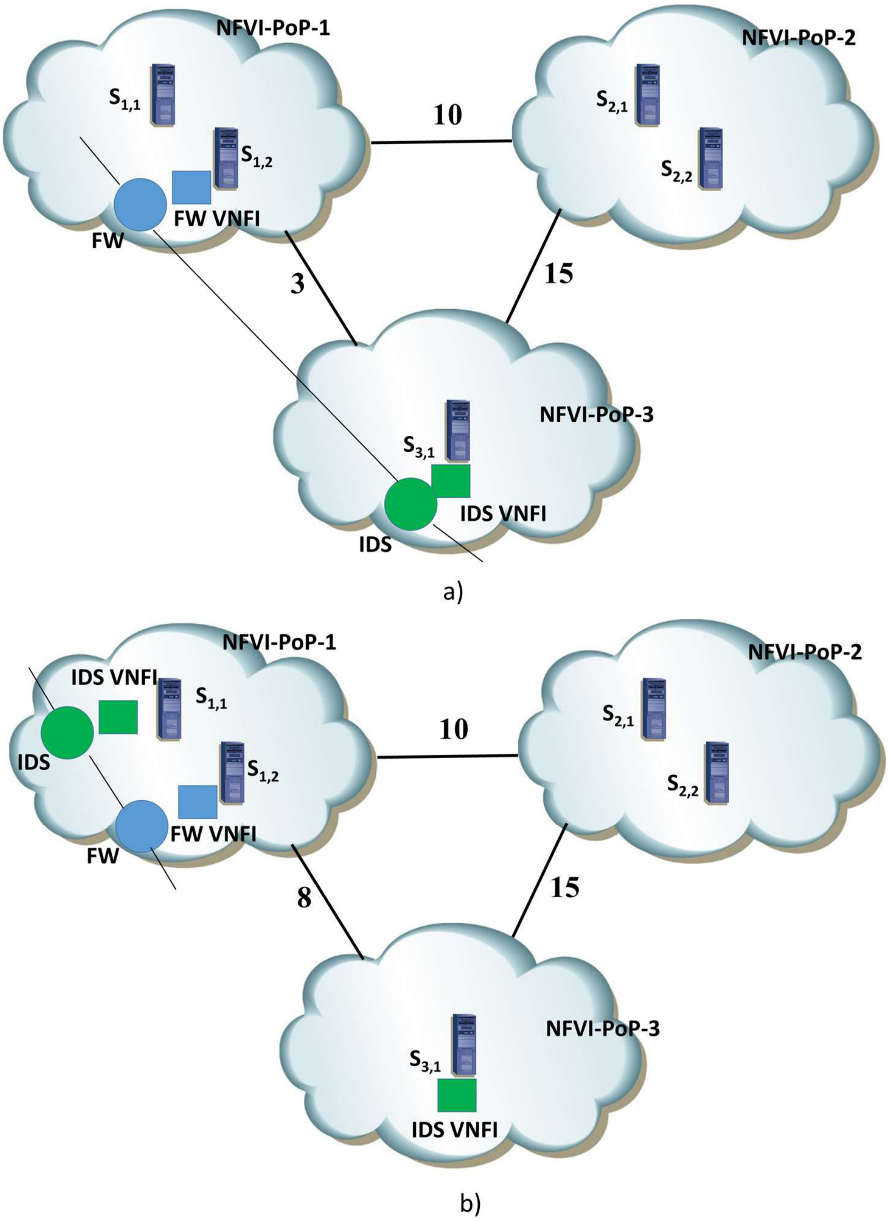


Figure 6. Resource allocation for the SFC of Figure 4. The cases x = 3$ (**a**) and x = 8$ (**b**) for the bandwidth interconnection cost between NFVI-PoP-1 and NFVI-PoP-3 of Figure 4 are reported.

The application of the SRCBA algorithm in the scalable solution case is illustrated in Figure 7a,b when the interconnection bandwidth cost x between NFVI-PoP-1 and NFVI-PoP-3 equals 3 and 8, respectively. We achieve the same solution of the TO case but at lower computational complexity. The nodes of the multi-stage graph are no longer associated with the servers but to the NFVI-PoPs, and this leads to reducing the size of the graph and the computation time of the shortest path.

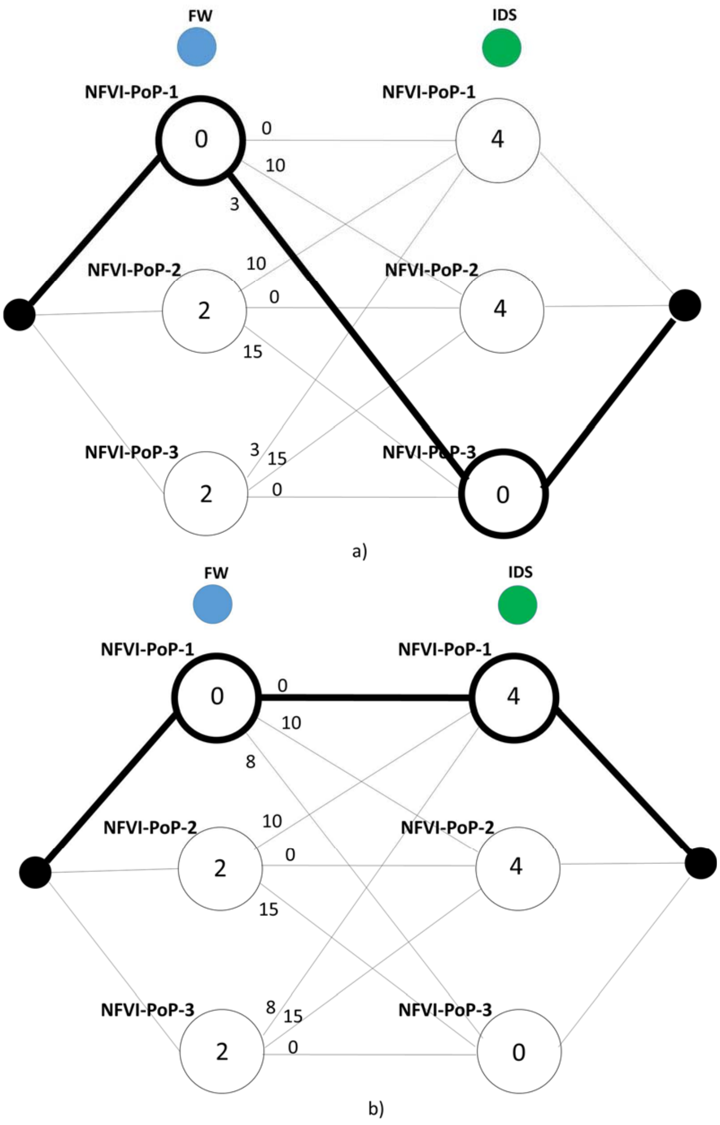


Figure 7. Construction of the multi-stage graph for the SO solution of the SRCBA algorithm. The cases x = 3$ (**a**) and x = 8$ (**b**) for the bandwidth interconnection cost between NFVI-PoP-1 and NFVI-PoP-3 of Figure 4 are reported.

6. Numerical Results

We provide some results to compare the computational complexity of the Traditional Orchestration (TO) and proposed Scalable Orchestration (SO) solutions. In particular, we evaluate the execution times of SRCBA algorithms proposed in [7] and in Section 5.1 for TO and SO solutions, respectively. The times are measured on a Personal Computer characterized by a 3.40 GHz Intel i7-3770 processor and by an 8 GB RAM memory (Intel Core i7-3770).

The comparison is carried out when two realistic network scenarios are considered. The first one is reported in Figure 8 and considers a medium distance network such as Deutsche Telekom (Germany) in which N_{NFVI} = 4 NFVI-PoPs are considered as illustrated in Figure 8. The second one is reported in Figure 9 and considers a long distance network as NSFNET with 16 switches and 25 links; the four cases with a number N_{NFVI} of NFVI-PoPs equal to 4 (a), 7 (b), 10 (c) and 14 (d) are considered. For these four case studies, the NFVI-PoPs are placed as illustrated in Figure 9a–d, respectively.

The NFVI-PoPs are equipped with $N_{core} = 96$ processing cores. The link bandwidth of the network interconnecting the NFVI-PoPs is fixed equal to 40 Gb/s. The NFVI-PoP network topology is assumed to be a fat tree. We assume a core cost c^{core} equal to 1 \$/h while the bandwidth cost c_b equals 0.1 cent/KmGb.

We consider four Service Functions: Firewall (FW), Intrusion Detection System (IDS), Network Address Translator (NAT) and Proxy. VNF instances can be instantiated in the NFVI-PoPs to support these SFs. They are supported by software modules characterized by the maximum processing capacity and by the number of cores reported in Table 1. The possible compositions of the SFs in SFC are reported in Figure 10, where the nodes v_s and v_e are the access nodes originating and terminating the SFC.

Table 1. Maximum processing capacity and allocated number of cores for the software modules implementing FW, IDS, NAT and Proxy.

	Maximum Processing Capacity ($C^{pr,max}$)	**Number of Cores Allocated (n^c)**
FW	0.9 Gbps	4
IDS	0.6 Gbps	8
NAT	0.9 Gbps	2
Proxy	0.9 Gbps	4

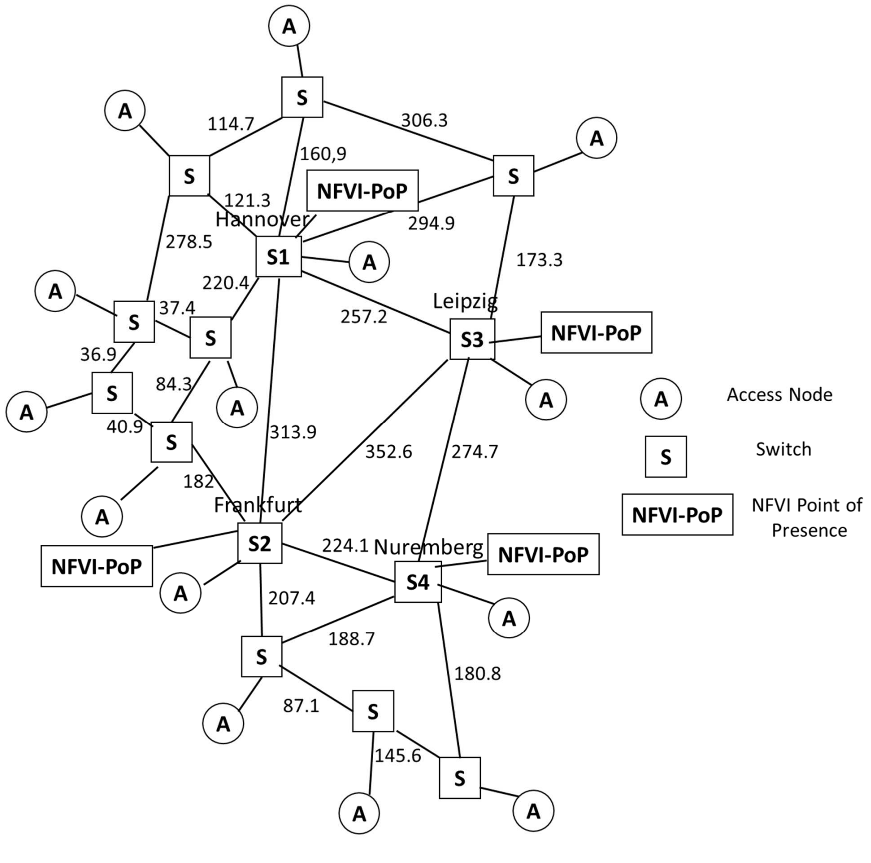


Figure 8. Deutsche Telekom network with four NFVI-PoPs.

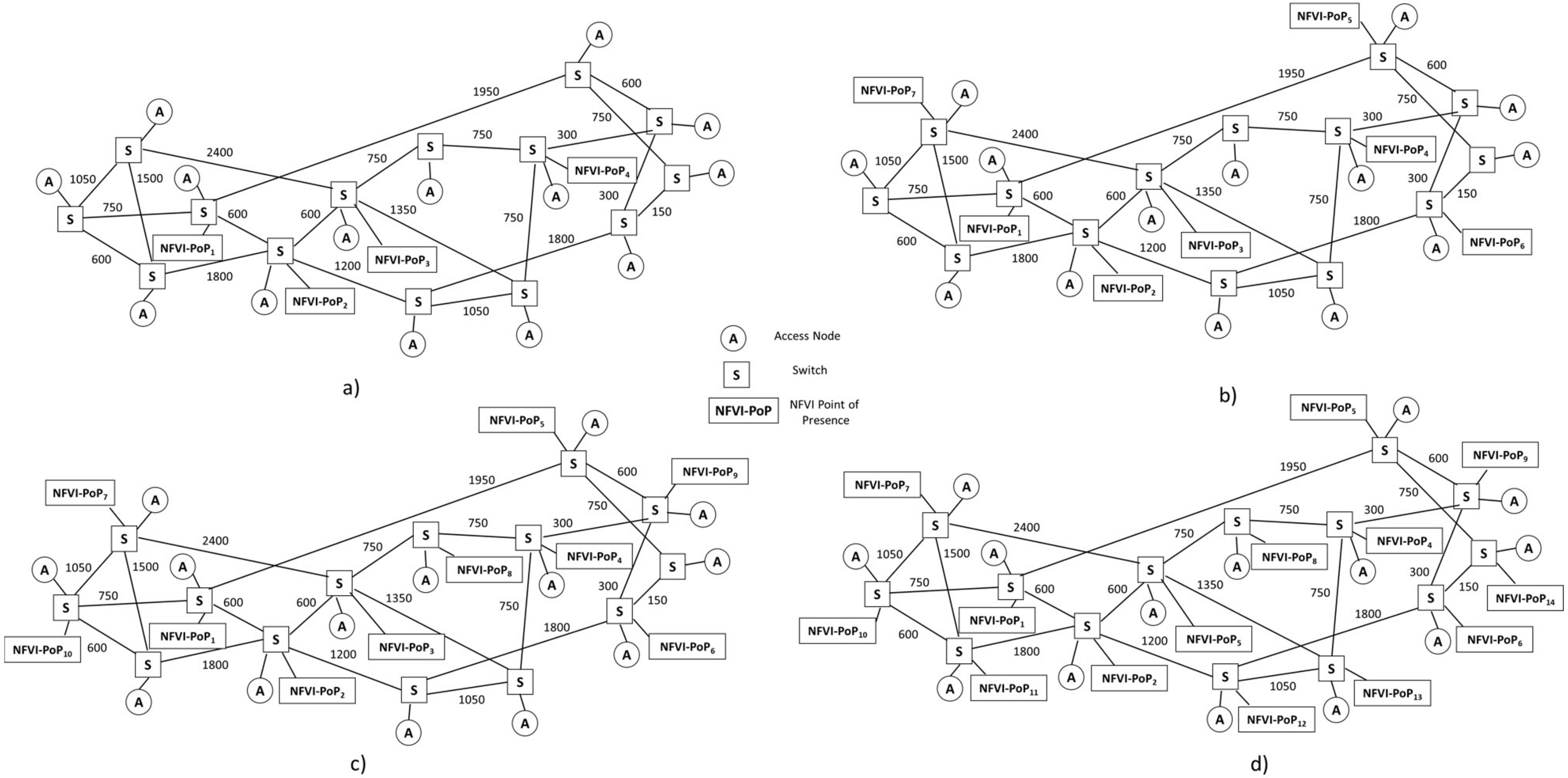


Figure 9. NSFNET (USA) network composed of 16 switches and 25 links. The four cases with a number N_{NFVI} of NFVI-PoPs equal to 4 (**a**), 7 (**b**), 10 (**c**) and 14 (**d**) are considered.

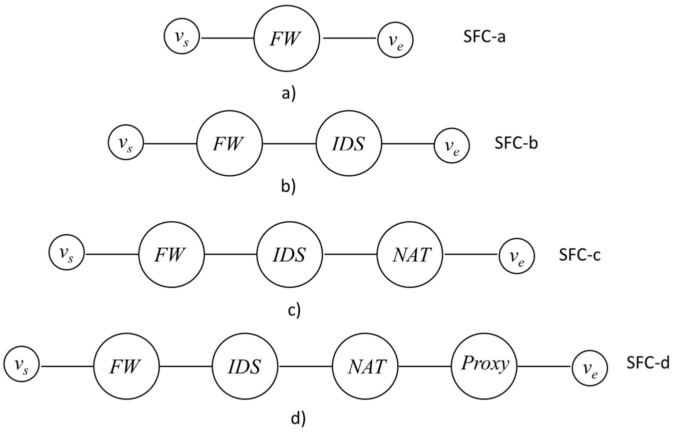


Figure 10. Composition of the Service Function Chain; four types are considered: the first one composed of a Firewall (FW) (**a**), the second one composed of an FW and an Intrusion Detection System (IDS) (**b**); the third one composed of an FW, an IDS and a Network Address Translator (NAT) (**c**); and the fourth one composed of an FW, an IDS, a NAT and a Proxy (**d**).

We assume that N SFC requests are generated. The bandwidth values of the SFCs are chosen equal to 20 Mbps. The ingress and egress nodes of each SFC are randomly chosen.

We illustrate in Figure 11 for the Deutsche Telekom network the SRCBA execution time as a function of the number of SFC requests for the Traditional Orchestration (TO) and Scalable Orchestration (SO) cases. The number N_s of servers in each NFVI-PoP equals 4. We report the results when SFCs with one SF (SFC-a), two SFs (SFC-b), three SFs (SFC-c) and four SFs (SFC-d) of Figure 10 are offered. We can observe how our proposed scalable orchestration allows for a remarkable reduction in SRCBA execution time. For example, when 2500 type-b SFCs are offered, SRCBA execution times equal 6642 ms and 536 ms for TO and SO, respectively, with an advantage in computation time by 92%.

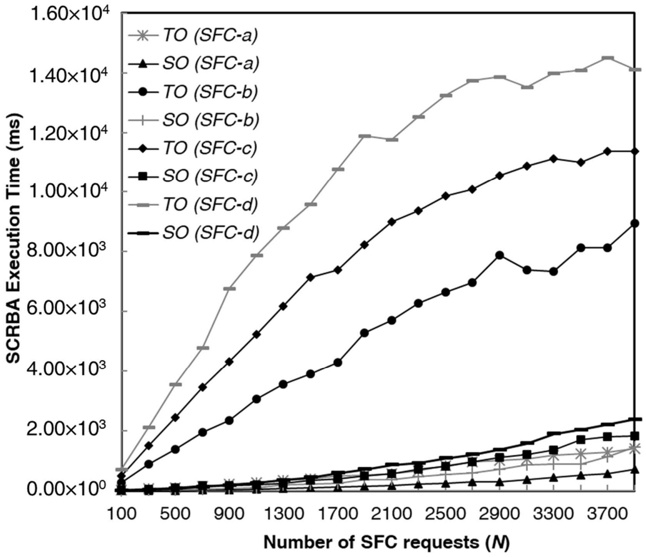


Figure 11. SRCBA execution time versus the number of SFC requests for the TO and SO solutions and when the Deutsche Telekom network is considered. The TO solution is investigated when the number N_s of servers in each NFVI-PoP equals 4.

The impact of the number N_s of servers on the SRCBA execution time is analyzed in Figure 12 for Deutsche Telekom. In this figure, we report the SRCBA execution time as a function of the number N of

SFCs offered. We show the results in the case of type-b and type-d offered SFCs and when the number N_s of servers is equal to 2, 4 and 6. As you can observe from Figure 12, the SRCBA execution times of the SO are independent of N_s because the the Resource Orchestrator is able to abstract the NFVI-PoP network topology and exposes the NS orchestrator information about the cloud resources (number of available cores) only. Conversely, we also observe from Figure 12 how the SRCBA execution time increases as the number N_s of servers increases. For example, when the number N of type-d SFCs equals 1500, the SRCBA execution time equals 1246 ms, 9902 ms and 41,243 ms for N_s equal to 2, 4 and 6, respectively. The increase is according to the expression of the computational complexity $C_{TO} = O(NM_{max}N_s^2 log(M_{max}N_s))$ of the SRCBA algorithm in the TO solution [7]. Finally, we notice as the SRCBA execution time equals 1645 ms in SO solution, allowing for an execution time percentage reduction equal to 66%, 96% and 99% with respect to TO solution for number N_s of servers equal to 2, 4 and 6, respectively.

Next, we study the impact of the number of NFVIs on the SRCBA execution time. We report in Figure 13 the SRCBA execution time versus the number of SFC requests for the TO and SO solutions. We consider the NSFNET network with a number of NFVI-PoPs equal to 4, 7, 10 and 14 and located as shown in Figure 9. We report the SRCBA execution time for both TO and SO solutions and N_s = 4 servers are located in the NFVI-PoPs for the TO case. Type-b SFCs are considered. From Figure 13, we can observe the increase in the SRCBA execution time when the number of NFVI-PoPs is increased for both TO and SO solutions. The reduction of the SRCBA execution time is confirmed even varying the number N_{NFVI} of NFVI-PoPs; for example, when the number N of SFCs equals 3500, the percentage reduction in SRCBA execution time of the SO solution with respect to the TO one equals 91%, 98%, 99% and 99% for N_{NFVI} equal to 4, 7, 10 and 14, respectively.

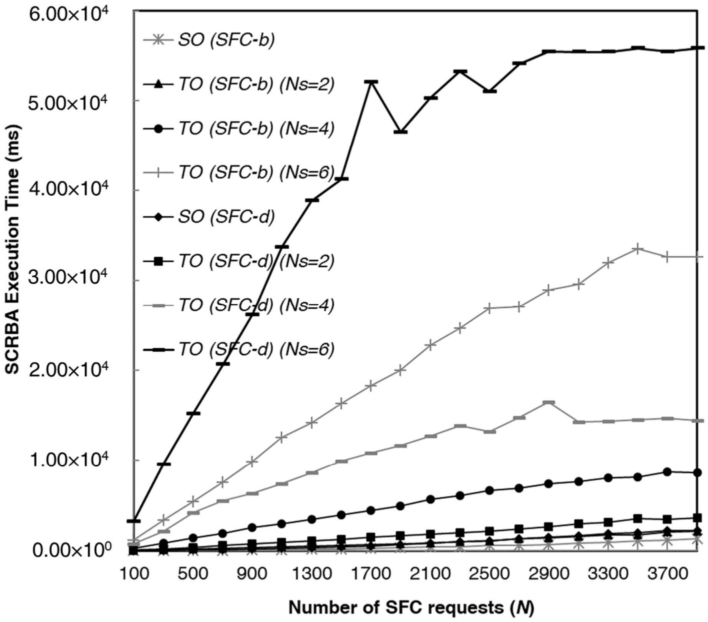


Figure 12. SRCBA execution time versus the number of SFC requests for the TO and SO solutions and when the Deutsche Telekom network is considered. The TO solution is investigated when the number N_s of servers in each NFVI-PoP is equal to 2, 4 and 6.

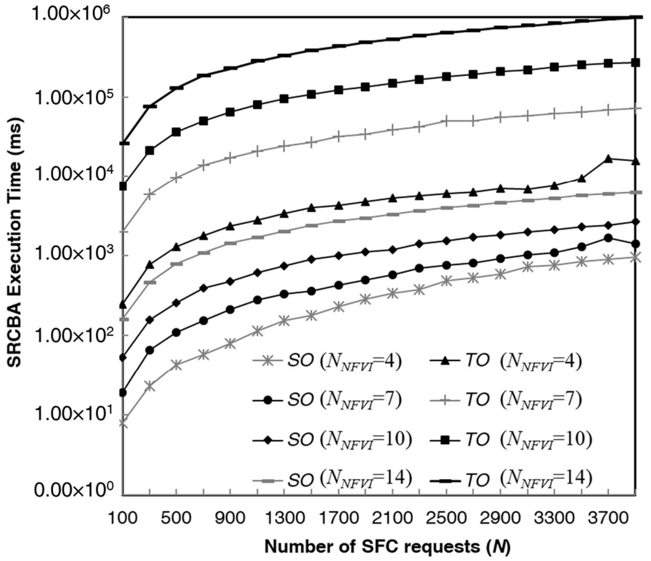


Figure 13. SRCBA execution time versus the number of SFC requests for the TO and SO solutions. The number N_{NVFI} of NFVIs is varied from 4 to 14 and the NFVI-PoPs are located as shown in Figure 9. The TO solution is investigated when the number N_s of servers in each NFVI-PoP is equal to 4.

We compare the TO and SO solution in terms of total cost given by the sum of the cloud and bandwidth resource costs. We report in Figure 14 the total cost as a function of the number N of SFCs. Type-b SFCs are considered and the results are provided for the NSFNET network with a number of NFVI-PoPs equal to 4, 7, 10 and 14. From Figure 14, we can observe similar costs for the TO and SO solutions; the reason is due to the operation mode of SRCBA that is analogue for the TO and SO cases.

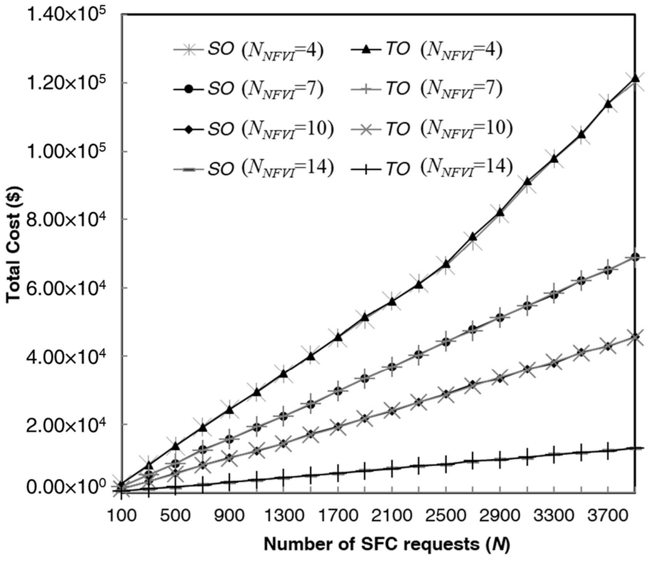


Figure 14. Total cost versus the number of SFC requests for the TO and SO solutions. The number N_{NVFI} of NFVIs is varied from 4 to 14 and the NFVIs are located as shown in Figure 9. The TO solution is investigated when the number N_s of servers in each NFVI-PoP is equal to 4.

Finally, notice how the aggregate knowledge of the resources of SO may lead to a high loss of SFC requests when the NFVI-PoP bandwidth and cloud resources are limited. This analysis is out of the scope of this work and we believe that the problem is of little interest since the NFVI-PoP resources are over-dimensioned with respect to the network ones.

7. Conclusions

A scalable and ETSI compliant NFV orchestration solution for Multi-Provider NFV environments has been proposed and investigated. An SR-based data/control plane for the proposed solution has also been introduced. We have shown how the proposed solution allows for a drastic reduction in execution time of the SFC Routing and SFC Routing and Cloud and Bandwidth resource Allocation algorithm executed by an NFV orchestrator. In the analyzed study case, we have verified a drastic reduction of the execution time with respect to a traditional orchestration solution without an appreciable degradation of performance in terms of total cost.

As future steps, we will compare the SR-based approach with the NSH one and we will realize an experimental testbed based on Cisco's Vector Packet Processing (VPP) technology [39], in order to demonstrate the feasibility and performance analysis of the proposed SR based scalable orchestrator architecture.

Author Contributions: V.E. and A.C. have organized and written the paper. F.G.L. and T.C. have implemented the code for the performance evaluation. M.P. have provided useful comments for the results section.

Funding: This research received no external funding.

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

References

1. Eramo, V.; Tosti, A.; Miucci, E. Server Resource Dimensioning and Routing of Service Function Chain in NFV Network Architectures. *J. Electr. Comput. Eng.* **2016**, *2016*, 7139852. [CrossRef]
2. Kar, B.; Wu, E.H.K.; Lin, Y.D. Energy Cost Optimization in Dynamic Placement of Virtualized Network Function Chains. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 372–386. [CrossRef]
3. Carpio, F.; Dhahri, S.; Jukan, A. VNF Placement with Replication for Load Balancing in NFV Networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017.
4. Chen, H.; Wang, X.; Zhao, Y.; Song, T.; Wang, Y.; Xu, S.; Li, L. MOSC: A method to assign the outsourcing of service function chain across multiple clouds. *Comput. Netw.* **2018**, *133*, 166–182. [CrossRef]
5. Wang, C.-C.; Lin, Y.-D.; Wu, J.-J.; Lin, P.-C.; Hwang, R.-H. Towards Optimal Resource Allocation of Virtualized Network Functions for Hierarchical Datacenters. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 1532–1544. [CrossRef]
6. Yi, B.; Wang, X.; Li, K.; Das, S.K.; Huang, M. A comprehensive survey of Network Function Virtualization. *Comput. Netw.* **2018**, *133*, 212–262. [CrossRef]
7. Bari, M.F.; Chowdhury, S.R.; Ahmed, R.; Boutaba, R.; Muniz, O.C.; Duarteyh, B. Orchestrating Virtualized Network Functions. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 725–739. [CrossRef]
8. De Sousa, N.F.S.; Perez, D.A.L.; Rosa, R.V.; Santos, M.A.S.; Rothenberg, C.E. Network Service Orchestration: A Survey. *arXiv* **2018**, arXiv:1803.06596.
9. Mijumbi, R.; Serrat, J.; Gorricho, J.; Bouten, N.; De Turck, F.; Boutaba, R. Network Function Virtualization: State-of-the-art and Research Challenges. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 236–262. [CrossRef]
10. Wang, Y.; Lu, P.; Lu, W.; Zhu, Z. Cost-Efficient Virtual Network Function Graph (vNFG) Provisioning in Multidomain Elastic Optical Networks. *J. Lightwave Technol.* **2017**, *35*, 2712–2723. [CrossRef]
11. ETSI. Network Function Virtualization (NFV); Management and Orchestration; Report on Architectural Options. ETSI GS NFV-IFA 009. Available online: https://www.etsi.org/ (accessed on 23 December 2018).
12. Cianfrani, A.; Listanti, M.; Polverini, M. Incremental Deployment of Segment Routing Into an ISP Network: A Traffic Engineering Perspective. *IEEE/ACM Trans. Netw.* **2017**, *25*, 3146–3160. [CrossRef]
13. Paolucci, F. Network Service Chaining Using Segment Routing in Multi-Layer Networks. *J. Opt. Commun. Netw.* **2018**, *10*, 582–592. [CrossRef]
14. Filsfils, C.; Nainar, N.K.; Pignataro, C.; Cardona, J.C.; Francois, P. The Segment Routing Architecture. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015.

15. Eramo, V.; Lavacca, F.G.; Catena, T.; Polverini, M.; Cianfrani, A. Proposal and Investigation of a Scalable NFV Orchestrator Based on Segment Routing Data/Control Plane. In Proceedings of the 14th International Conference on Network and Service Management, Rome, Italy, 5–9 November 2018.
16. Feng, H.; Llorca, J.; Tulino, A.M.; Raz, D.; Molisch, A.F. Approximation Algorithms for the NFV Service Distribution Problem. In Proceedings of the INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, Atlanta, GA, USA, 1–4 April 2017.
17. Xie, Y.; Liu, Z.; Wang, S.; Wang, Y. Service Function Chaining Resource Allocation: A Survey. *arXiv* **2016**, arXiv:1608.00095.
18. Herrera, J.G.; Botero, J.F. Resource Allocation in NFV: A Comprehensive Survey. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 518–532. [CrossRef]
19. Li, X.; Qian, C. A Survey of Network Function Placement. In Proceedings of the 2016 13th IEEE AnnualConsumer Communications and Networking Conference (CCNC), Las Vegas, CA, USA, 15–16 March 2016; pp. 1–6.
20. Ghaznavi, M.; Khan, A.; Shahriar, N.; Alsubhi, K.; Ahmed, R.; Boutaba, R. Elastic Virtual Network Function Placement. In Proceedings of the 4th IEEE conference on Cloud Networking (CloudNet) 2015, Niagara Falls, ON, Canada, 5–7 October 2015.
21. Ghaznavi, M.; Shahriar, N.; Ahmed, R.; Boutaba, R. Service Function Chaining Simplified. *arXiv* **2016**, arXiv:1601.00751. [CrossRef]
22. Wang, X.; Wu, C.; Le, F.; Liu, A.; Li, Z.; Lau, F. Online VNF Scaling in Datacenters. *arXiv* **2016**, arXiv:1604.01136.
23. Liu, J.; Lu, W.; Zhou, F.; Lu, P.; Zhu, Z. On Dynamic Service Function Chain Deployment and Readjustment. *IEEE Trans. Netw. Serv Manag.* **2017**, *14*, 543–553. [CrossRef]
24. Pei, J.; Hong, P.; Xue, K.; Li, D. Efficiently Embedding Service Function Chains with Dynamic Virtual Network Function Placement in Geo-distributed Cloud System. *IEEE Trans. Parallel Distrib. Syst.* **2019**. [CrossRef]
25. Cappanera, P.; Paganelli, F.; Paradiso, F. VNF placement for service chaining in a distributed cloud environment with multiple stakeholders. *Comput. Commun.* **2019**, *133*, 24–40. [CrossRef]
26. ETSI. Network Functions Virtualisation: Introductory; White Paper; SDN and OpenFlow World Congress. Available online: https://portal.etsi.org/nfv/nfv (accessed on 23 December 2018).
27. Desmouceaux, Y.; Pfister, P.; Tollet, J.; Townsley, M.; Clausen, T. The Power of Choices in Load Balancing with Segment Routing. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems, Atlanta, GA, USA, 5–8 June 2017.
28. Duchene, F.; Lebrun, D.; Bonaventure, O. SRv6Pipes: Enabling in-network bytestream functions. In Proceedings of the IFIP Networking 2018, Zurich, Switzerland, 14–16 May 2018.
29. Lebrun, D.; Bonaventure, O. Implementing IPv6 Segment Routing in the Linux Kernel. In Proceedings of the Applied Networking Research Workshop, Prague, Czech Republic, 15 July 2017.
30. AbdelSalam, A.; Clad, F.; Filsfils, C.; Salsano, S.; Siracusano, G.; Veltri, L. Implementation of virtual network function chaining through segment routing in a linux-based NFV infrastructure. In Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, 3–7 July 2017.
31. Abdelsalam, A.; Salsano, S.; Clad, F.; Camarillo, P.; Filsfils, C. SERA: SEgment Routing Aware Firewall for Service Function Chaining scenarios. In Proceedings of the IFIP Networking, Zurich, Switzerland, 14–16 May 2018.
32. Desmouceaux, Y.; Townsley, M.; Clausen, T.H. Zero-Loss Virtual Machine Migration with IPv6 Segment Routing. In Proceedings of the 2018 14th International Conference on Network and Service Management (CNSM), Rome, Italy, 5–9 November 2018.
33. Eramo, V.; Lavacca, F.G. Processing and Bandwidth Resource Allocation in Multi-Provider NFV Cloud Infrastructures interconnected by Elastic Optical Networks. In Proceedings of the 2018 20th International Conference on Transparent Optical Networks, Bucharest, Romania, 1–5 July 2018.
34. Fei, X.; Liu, F.; Xu, H.; Jin, H. Adaptive VNF Scaling and Flow Routing with Proactive Demand Prediction. In Proceedings of the INFOCOM 2018-IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018.
35. Cheng, X.; Wu, Y.; Min, G.; Zomaya, A.Y. Network Function Virtualization in Dynamic Networks: A Stochastic Perspective. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2218–2232. [CrossRef]

36. Filsfils, C.; Garvia, P.C.; Leddy, J.; Voyer, D.; Matsushima, S.; Li, Z. SRv6 Network Programming. Available online: https://tools.ietf.org/html/draft-filsfils-spring-srv6-network-programming-06 (accessed on 9 February 2019).
37. Lin, P.-C.; Lin, Y.-D.; Wu, C.-Y.; Lai, Y.-C.; Kao, Y.-C. Balanced Service Chaining in Software-Defined Networks with Network Function Virtualization. *Computer* **2016**, *49*, 68–76. [CrossRef]
38. Eramo, V.; Lavacca, F.G. Computing and Bandwidth Resource Allocation in Multi-Provider NFV Environment. *IEEE Commun. Lett.* **2018**, *22*, 2060–2063. [CrossRef]
39. Evolving the Mobile Core to Being Cloud Native. White Paper. Available online: https://ieeexplore.ieee.org/document/8463557 (accessed on 7 February 2019).