electronics

MDPI

Article

Efficient Placement of Service Function Chains in Cloud Computing Environments

Marwa A. Abdelaal [1,*], Gamal A. Ebrahim [2,*] and Wagdy R. Anis [1]

[1] Electronics and Communications Engineering Department, Faculty of Engineering, Ain Shams University, Cairo 11517, Egypt; wagdy_anis@eng.asu.edu.eg
[2] Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University, Cairo 11517, Egypt
* Correspondence: marwa.ahmed@edf.edu.eg (M.A.A.); gamal.ebrahim@eng.asu.edu.eg (G.A.E.)

Abstract: The widespread adoption of network function virtualization (NFV) leads to providing network services through a chain of virtual network functions (VNFs). This architecture is called service function chain (SFC), which can be hosted on top of commodity servers and switches located at the cloud. Meanwhile, software-defined networking (SDN) can be utilized to manage VNFs to handle traffic flows through SFC. One of the most critical issues that needs to be addressed in NFV is VNF placement that optimizes physical link bandwidth consumption. Moreover, deploying SFCs enables service providers to consider different goals, such as minimizing the overall cost and service response time. In this paper, a novel approach for the VNF placement problem for SFCs, called virtual network functions and their replica placement (VNFRP), is introduced. It tries to achieve load balancing over the core links while considering multiple resource constraints. Hence, the VNF placement problem is first formulated as an integer linear programming (ILP) optimization problem, aiming to minimize link bandwidth consumption, energy consumption, and SFC placement cost. Then, a heuristic algorithm is proposed to find a near-optimal solution for this optimization problem. Simulation studies are conducted to evaluate the performance of the proposed approach. The simulation results show that VNFRP can significantly improve load balancing by 80% when the number of replicas is increased. Additionally, VNFRP provides more than a 54% reduction in network energy consumption. Furthermore, it can efficiently reduce the SFC placement cost by more than 67%. Moreover, with the advantages of a fast response time and rapid convergence, VNFRP can be considered as a scalable solution for large networking environments.

Keywords: network function virtualization (NFV); software-defined networking (SDN); service function chain (SFC); virtual network function (VNF); VNF placement

Citation: Abdelaal, M.A.; Ebrahim, G.A.; Anis, W.R. Efficient Placement of Service Function Chains in Cloud Computing Environments. *Electronics* **2021**, *10*, 323. https://doi.org/10.3390/electronics10030323

Academic Editor: Filipe Araujo
Received: 9 December 2020
Accepted: 25 January 2021
Published: 30 January 2021

1. Introduction

Network functions are commonly provided through middleboxes that are used in conventional networks. The functions provided by these middleboxes range from firewalls to proxies, and even to intrusion detection systems. These functions are implemented using dedicated hardware appliances that lead to inflexible management and costly capital expenditure (CAPEX) and operating expenditure (OPEX) [1,2].

The emergence of network function virtualization (NFV) [3,4] and software-defined networking (SDN) [5,6] provides the ability to lead to more agile and flexible networks. NFV is a viable networking paradigm for moving traffic processing from dedicated hardware appliances to flexible software applications, known as virtual network functions (VNFs) [7]. Each VNF is placed on commercial off-the-shelf (COTS) servers or on a virtual machine (VM) located in a virtualization environment, such as the cloud. NFV enables network operators to achieve a high degree of flexibility and cost-saving, resulting in simplified and extensible placement and management, the efficient utilization of network functions, and a reduction in power usage. Moreover, VNFs leverage the virtualization technology

to provide performance and security enhancements in the network [8]. Chaining and steering traffic flows through an ordered sequence of VNFs to comply with security and performance policies while delivering a service is commonly referred to as service function chain (SFC) [9,10]. Consequently, SFC shapes the VNF forwarding graph (VNF-FG) [11].

The NFV approach provides various benefits, such as the reduction of capital and operating expenditures [4]. These expenditures include license fees and redundant resources that provide a greater degree of automation during disaster recovery and software upgrades [12,13]. Additionally, using NFV leads to the ability to process high-bitrate traffic streams by promoting the concept of service chaining [14,15]. Hence, this leads to flexible support of real-time streaming applications that represent the majority of traffic in today's networks. At the same time, NFV alleviates the reliance on network devices such as firewalls and proxies to provide network functionality. Therefore, for many users, NFV is considered an economically appealing platform that allows users to pay only for the services they use, according to the pay-as-you-go payment model [16,17]. Due to these capabilities, many cloud service providers have recently migrated their network services and applications to NFV.

Meanwhile, SDN [5] is a networking technology that decouples network control from forwarding functions. It consolidates network control functions into a logically centralized controller. The capabilities of SDN and cloud technologies have been proven to be highly complementary to the NFV paradigm for handling the revolutionary increase of data traffic in today's networks [1]. An SDN controller can manage networks as well as network functions (typically VNFs). It handles this process whether these functions run on VMs in a single data center or in multiple data centers to manipulate traffic flows through appropriate network functions. Thus, the overall network service delivery platform becomes easily scalable. Additionally, the flexibility of cloud computing [18] makes it the best candidate for deploying VNFs on dedicated VMs. Furthermore, NFV and SDN are mutually beneficial to each other. For example, SDN can accelerate NFV deployment by offering a flexible and automated approach to chaining functions, configuring network connectivity and bandwidth, and automating the security and policy control [19–21]. Additionally, it can be used to support SFC to reduce both the management complexity and the operational cost. Meanwhile, handling SFC requests necessitates directing their traffic through an ordered sequence of VNFs. Since these SFC requests are dynamic, they cause differences in the network load [22]. Moreover, placing VNFs in cloud datacenters can minimize the deployment cost. However, it typically causes churns in the network load that increase network delay due to the distance between the users and the datacenters. Therefore, optimally placing heterogeneous VNFs while dynamically observing SFC requests is a challenging problem [23–25].

Several research efforts have been conducted for optimizing the placement of VNFs and steering traffic in the network to satisfy SFC requests. A set of objectives has been adopted, such as reducing the network cost [26] or reducing resource utilization [27]. The VNF placement problem has been proven to be NP-hard [28,29]. It can be solved using classic mathematical algorithms that require a prohibitive computational cost, due to the exponential time complexity with respect to network size [30]. Hence, it can lead to the slow deployment of VNFs in cloud datacenters. Meanwhile, several heuristic algorithms [31–33] were introduced to provide near-optimal solutions with small execution times. Therefore, they could be suitable for cloud datacenters that represent real environments of scalable networks.

There are limitations of physical resources (e.g., CPU, memory, storage, bandwidth, etc.) and other constraints of VNFs. Consequently, VNF placement in cloud datacenters should be optimized. The objectives of this optimization are to achieve load balancing and reduce energy consumption, resource consumption, and the overall cost. Furthermore, the time required to calculate the placement should be kept as short as possible to provide continuous service chains. Hence, the effective deployment of VNFs faces conflicting objectives, such as reducing link bandwidth utilization and minimizing energy consumption,

mainly because the consolidation of resource usage can cause congestion in the physical network. For example, minimizing the number of active servers leads to increasing the additional bandwidth used for all embedded SFC physical links [34]. Meanwhile, less effort has been made to address the network load balancing issue with VNF placement.

This paper addresses VNF placement for SFCs using replica concept in software-defined cloud computing environments. It introduces a novel approach to the VNF placement problem for SFCs called virtual network functions and their replica placement (VNFRP). It formulates the VNF placement problem as an integer linear programming (ILP) optimization problem. The objective of this ILP problem is to minimize the link bandwidth utilization, energy consumption, and SFC placement cost. The optimization of VNF placement is more challenging mainly because the centralized controller should have a complete picture of the sequence of network service requests and the physical resource constraints. Additionally, the processing of SFC requests takes into consideration not only the individual VNF constraints, but also the chaining order restrictions of the entire flow, along with the required virtual functions. Meanwhile, most of the current VNF placement research considers only a portion of the problem by optimizing either the host or network resources but does not provide a holistic view of computing the solution. Hence, this paper sheds light on placing VNFs for the service chain, which requires the allocation of both server resources and network resources. Server resources are used to host the VNFs, while network resources are used to route the traffic flow from one VNF to the next in the service chain.

The rest of this paper is organized as follows. Section 2 presents the related work, followed by Section 3, which details the system model and problem formulation. Meanwhile, Section 4 explains the heuristic solution of the problem, followed by Section 5, which describes the experimental results. Finally, the paper is concluded in Section 6.

2. Related Work

Despite NFV being embraced by both industry and academia at unparalleled levels, the development of NFV is still in its infancy with many open challenges. Service providers look at the details of implementing NFV to meet their intended goals. Meanwhile, the accomplishment of some of NFV objectives and whether implementation leads to the anticipated benefits are questionable. Many key challenges still need to be adequately addressed when seeking the efficient placement of virtualized network functions. These challenges include link bandwidth utilization, energy consumption, cost, resource allocation, violations of the service-level agreement (SLA), and the quality of service (QoS). Early research in this direction focuses solely on the optimal placement of VNFs in NFV-enabled hybrid environments [30]. In such environments, dedicated physical hardware and virtual network functions coexist, depending on the demand. The study in [30] addressed the VNF placement problem by formulating it as an ILP model, with the aim of reducing the number of physical nodes used. Given the complexity of ILP, therefore, the performance of this model has been addressed in small service provider scenarios. On the other hand, the studies in [2,35] modeled the VNF placement problem as a generalization of the virtual network embedding (VNE) problem [13]. However, the placement of VNFs for the service chain and VNE placement have different goals and constraints [35].

Several studies formulate the VNF placement problem as an optimization problem with different objective functions. Therefore, a variety of optimal and near-optimal solutions are proposed. For example, a heuristic approach was introduced in [29] for solving the VNF placement problem with an attempt to reduce the OPEX. This approach modeled the problem using two well-known NP-hard problems: the facility location problem and the generalized assignment problem. Then, it introduced rounding-based heuristics to solve the problem and achieve better results with regard to the OPEX. Meanwhile, the study in [36] formulated the VNF placement problem as a binary integer programming problem in packet optical datacenters. The main objective in this study was to minimize the overall conversions. The study proposed an alternative effective heuristic algorithm to

solve the problem. However, the model and the algorithm were limited to packet optical datacenters.

Many heuristic and metaheuristic algorithms have been proposed to reduce the computational complexity and NP-hardness of the VNF placement problem. The study in [24] proposed a resource allocation algorithm for VNFs based on genetic algorithms. The introduced placement algorithm outperformed ILP-based resource allocation for a large number of VNFs in service function chains. Furthermore, the proposed placement algorithm not only decreased memory and CPU utilization but also reduced energy consumption in the datacenter. Meanwhile, the study in [37] proposed a metaheuristic solution, with the aim of maximizing physical resource utilization and minimizing the number of active servers that host VNFs in the network. Another algorithm that approximately solved VNF placement problem was presented in [38]. However, it took only one network function into account while ignoring the chaining of service functions.

In fact, the relationship between virtual network functions and service chains is complicated. Some VNFs from different service chains can be shared during deployment, such as anti-virus, while others cannot, such as firewalls [34]. Moreover, some VNFs may modify the traffic between VNFs, such as firewalls, which may drop packets that do not comply with the security policy [39]. Thus, considerable attention has recently been paid to VNF placement and service chaining problems. The study in [35] introduced a mathematical model for VNF chaining and routing. The introduced approach constructed the VNF-FG and then mapped it to the physical resources. It took into consideration the limitation of network resources and the specific requirements of functions. The VNF-FG mapping problem is formulated as a mixed-integer quadratically constrained program (MIQCP). However, it faces scalability issues. Moreover, the approach introduced in [35] placed multiple VNFs in one node. On the other hand, the optimized consolidation policies outlined in some existing studies [34,40,41] are used to manage computing resources in the cloud. However, they ignore the interconnection among VNFs in the service chain inside the cloud datacenters. Additionally, these studies do not meet the performance isolation required in multitenant clouds.

A genetic algorithm for the placement of VNF chains was proposed in [42] to satisfy the SLA and QoS objectives. However, it did not address the issue of the dynamic placement of VNFs to balance the network operational cost and performance. The study in [43] designed and implemented a VNF traffic-aware chaining algorithm within the SDN controller. However, it considered only the unicast-oriented VNF chains, which are simpler than other types of VNF forwarding graphs. The study in [44] introduced an approach that deals with the deployment of SFCs over NFV architecture. This approach focuses deeply on the problem of VNF placement to form optimal SFC across geographically distributed clouds. Additionally, it formulates the problem as an ILP optimization problem to minimize the intercloud traffic and the response time in multicloud environments. Moreover, the total deployment cost and the SLA constraints are taken into consideration.

SDN and NFV are complementary technologies that can be applied together on different types of networks. Hence, there are several recent research efforts addressing the merging of both SDN and NFV. For instance, the studies in [45,46] introduced a control plane framework that enabled packet forwarding across a collection of network function instances. In addition, they provided a communication path for configuration and decision-making between each network role and the SDN controller. The SDN-based networking approach was introduced in [47]. In this approach, SDN provides dynamic management of traffic flows. On the other hand, NFV transfers the network entities into the cloud commodity hardware. Hence, this approach can achieve a lower OPEX. The approach in [48] studied VNF orchestration in virtualized SDN environments. However, it considered a limited number of use cases where user movement could be predictable. Therefore, more accurate and mobility-aware VNF migrations are required to improve the performance and reduce the operational cost.

Cloud-based NFV research has gradually increased due to the flexible requirements and elasticity of NFV-enabled cloud networks. One of the key frameworks for cloud-based NFV architecture is OpenStack, which was introduced in [49]. However, it does not meet some of the NFV requirements, as mentioned in [50]. The study in [51] presented a VNF deployment algorithm and a VNF migration algorithm in NFV-enabled cloud networks. These algorithms reduced the number of active physical nodes, the rejection rate of VNF requests, and the number of migrations.

Several research efforts to study the VNF placement problem, with the goal of discovering cost-effective placement, have been accomplished. The study in [52] introduced two algorithms to perform VNF placement while considering the budget, resource constraints, and processing capacity limitation of VNF hosting nodes. However, it did not take into account the bandwidth consumption. Meanwhile, the study in [53] considered the dynamic network load in SDN/NFV-enabled networks while tackling the VNF placement problem. It formulated the problem as a binary integer programming (BIP) problem with the goal of reducing the total cost. This cost included the VNF placement cost, VNF infrastructure (VNFI) running cost, and penalty of rejection for service function chain requests. However, it did not specify the network architecture for the server nodes hosting the VNFs. The study in [54] analyzed the cost-effective orchestration of SFCs in the inter-datacenter network. It provided an ILP model and developed a heuristic algorithm that guaranteed the reliability criteria and minimized the cost of service chain orchestration. Moreover, the study in [55] discussed the impact of NFV on the performance and cost of cloud-based networks. However, no general optimization model was considered.

The addressing of the VNF problem with regard to energy consumption has been introduced in several research works. The study in [56] discussed the problem of mapping and scheduling flows across available VNF instances to reduce energy consumption. However, it ignored routing problems, fault probability, and failure recovery. Meanwhile, the study in [57] addressed the problem of VNF placement and traffic steering to minimize energy consumption in NFV-enabled telecom networks. A power consumption model was presented, and the problem was formulated as an ILP model. Finally, a polynomial algorithm based on the Markov approximation approach was proposed to approximately solve the problem. The study in [58] formulated the service deployment problem as an ILP model, with the aim of minimizing the energy consumption of the deployed chains. A rapid and scalable polynomial algorithm was proposed to solve the designed ILP model. However, the proposed algorithm was designed for static traffic scenarios.

Furthermore, a limited number of research efforts addressed the replication of SFC in cloud-based NFV environments. The study in [59] proposed a migration-based reliability method for the service chain to ensure network service continuity via a replication strategy. It introduced a linear programming (LP) model to place VNFs on the physical computing servers. The main objective of the model was to minimize the overall cost of the network service while ensuring the reliability and performance required by the network service. However, this solution led to a longer recovery time. Additionally, the study in [60] modeled the placement of VNFs, with replications in mobile core networks considering multiple VNF instances. It tried to find a trade-off between the minimization of link utilization and CPU resource usage. However, the proposed solution had strict requirements for server computing resources, which reduced the computational efficiency in practical environments.

Most of the research efforts discussed in this section did not address the merging among different VNF placement objectives in cloud computing environments. This paper tries to jointly solve VNF placement for SFCs, considering VNF replicas in software-defined cloud computing environments. Hence, the goal of this research is to minimize the link bandwidth utilization, network energy consumption, and SFC placement cost. Additionally, a heuristic algorithm is proposed to place VNFs for the service chain, taking into consideration the load balancing, especially across the core links within the cloud. The proposed heuristic algorithm may place the VNFs of SFC in the same or separate nodes,

depending on the minimum link bandwidth utilization and SFC placement cost. Moreover, the proposed solution in this study can be used to achieve dynamic placement and provide a near-optimal solution within seconds.

3. System Model and Problem Formulation

It is important to pass the traffic of SFC requests through a sequence of VNFs in a specific order. The flexibility of placing VNFs at various network locations can achieve this goal. Moreover, path selection can affect resource consumption on network links. Providing the concept of replicas increases the number of admissible paths. Therefore, how to concatenate VNFs using replicas for each SFC request should be considered to achieve load balancing and reduce network resource bottlenecks, especially over the core links. Thus, the network energy consumption and SFC placement cost can be decreased. Hence, in this section, the system model is presented, in addition to formulating the mathematical representation of the problem.

3.1. System Model

The physical network is represented by a directed graph $G = (N, L)$, where N denotes the set of nodes and L denotes the set of links. N is given by the union of four sets of nodes in the network. The first set is N_E for the edge switch nodes, which are responsible for forwarding the data from servers to other neighboring nodes. The second set is N_D for the distribution switch nodes, which are responsible for data forwarding and serve as intermediate nodes between the edge and core switches. The third set is N_C for the core switch nodes, which are responsible for data forwarding among the distribution switch nodes. Finally, the set N_S is for the server nodes, which provide virtualized platforms to run VNFs while simultaneously processing the traffic of SFC requests. Each server node n_s has the resource capacity in terms of CPU, memory, and storage denoted by R_{CPU}, R_{RAM}, and R_{Stg}, respectively. Meanwhile, each link in the network is characterized by the bandwidth of the link B_L.

The set of service chains is represented by $S = \{s_1, s_2, ..., s_n\}$, where n is the number of service chains of VNFs that are provisioned in the physical network. Additionally, the set of VNFs in the service chain s is represented by $F = \{f_1, f_2, ..., f_m\}$, where m is the number of VNFs in the service chain. Each VNF $f \in F$ has a resource demand in terms of CPU, memory, storage, and bandwidth, denoted by $d_f = (d_{CPU}, d_{RAM}, d_{Stg}, d_{BW})$. Moreover, each network service request j is sent to the cloud as SFC $s_j = \{F_j, d_{CPUj}, d_{RAMj}, d_{Stgj}, d_{BWj}\}$, where F_j denotes the ordered sequence of VNFs requested in the SFC, with the first VNF in the chain being the source while the last VNF in the chain is the destination. Meanwhile, d_{CPUj}, d_{RAMj}, d_{Stgj}, and d_{BWj} denote the resource demand of VNFs belonging to s_j in terms of CPU, memory, storage, and bandwidth, respectively.

The traffic flow of the SFC request j is transferred according to its VNF-FG $G_j = (N_j, L_j)$. The VNF-FG is a directed graph where the directions of the links should satisfy the order constraint of the VNF request. In a VNF-FG, the parameters $f_{1j}, f_{2j} \in N_j$ denote two VNFs and $f_{1j}\, f_{2j} \in L_j$ denote the virtual link connecting VNF f_{1j} and f_{2j} in G_j. If $f_j \in F_j$ is assigned to server node $n_s \in N_S$, then L_j captures the placement constraints of the network functions—specifically, the bandwidth requirement—of the VNF-FG.

3.2. Problem Formulation

It is assumed in the context of formulating the problem that the service request has already been submitted to the cloud environment as SFC. Meanwhile, the SDN controller is designed to find the optimal placement of the VNFs with their replicas on the physical infrastructure. Optimal placement of the VNFs and their replicas can provide the optimal locations in cloud networks, which can be responsible for load balancing in the network. Ultimately, this leads to a reduction in network energy consumption. The optimal place-

ment of VNFs to satisfy an SFC request can be formulated as an ILP problem with a set of constraints, which can be mathematically expressed as follows:

$$\sum_{f_j \in F_j} d_{CPUj} \cdot X_{n_s}^{f_j} \leq R_{CPU}^{max} \tag{1}$$

$$\sum_{f_j \in F_j} d_{RAMj} \cdot X_{n_s}^{f_j} \leq R_{RAM}^{max} \tag{2}$$

$$\sum_{f_j \in F_j} d_{Stgj} \cdot X_{n_s}^{f_j} \leq R_{Stg}^{max} \tag{3}$$

where

$$X_{n_s}^{f_j} \in \{0, 1\}, \forall\, n_s \in N_S, \forall\, f_j \in F_j \tag{4}$$

$$\sum_{f_j \in F_j} \sum_{\forall n_s \in N_S} X_{n_s}^{f_j} = 1 \tag{5}$$

where Equations (1)–(3) represent the constraints of the server nodes for SFC request j. Equation (1) ensures that the total CPU utilization of all VNFs placed on the same server node n_s should not exceed its maximum CPU capability. Similarly, Equations (2) and (3) indicate that the total memory utilization and storage requirements of all VNFs consolidated on the same server node n_s should not exceed the maximum memory and storage capability of the server. Meanwhile, Equation (4) denotes a Boolean variable, which is equal to 1 when any VNF $f_j \in F_j$ belongs to SFC s_j placed on the server node $n_s \in N_S$; otherwise, it is 0. Moreover, Equation (5) ensures that each VNF f_j of SFC s_j can be merely placed on exactly one server node.

The constraint in Equation (6) guarantees that the bandwidth of each virtual link $f_{1j}f_{2j}$ of the SFC request j (in the path between the two VNFs $f_{1\mathrm{j}}$ and $f_{2\mathrm{j}}$) does not exceed the maximum bandwidth of the physical path p:

$$\forall\, l \in L : \sum_{f_{1j}f_{2j} \in L_j} d_{BWj} Y_{j,l_1l_2}^{f_{1j}f_{2j}} \leq BW_p \tag{6}$$

where l_1l_2 is the physical link of the physical path p connecting the two server nodes n_{s1} and n_{s2}. BW_p is the maximum bandwidth of the physical path connecting the two server nodes n_{s1} and n_{s2} when deploying VNFs on VMs in the cloud environment. $Y_{j,l_1l_2}^{f_{1j}f_{2j}}$ denotes a Boolean variable, which is equal to 1 when the virtual link $f_{1j}f_{2j} \in L_j$ from $f_{1\mathrm{j}}$ to $f_{2\mathrm{j}}$ uses the physical link l_1l_2; otherwise, it is 0.

The constraint in Equation (7) guarantees that each virtual link between two adjacent VNFs f_{1j} and f_{2j} of SFC s_j can be mapped on exactly one physical link l_1l_2:

$$\forall s_j \in S : \sum_{f_{1j}, f_{2j} \in F_j} Y_{j,l_1l_2}^{f_{1j}f_{2j}} \leq 1, \forall\, n_{s1}, n_{s2} \in N_S\ ,\ f_{1j} \neq f_{2j} \tag{7}$$

Equation (8) ensures that the sequence order of the VNFs forming the service chain is preserved in the selected physical path p. This means that the function f_{mj} cannot be allocated in the server node n_s if the previous function $f_{(m-1)j}$ is not already assigned to any of the previous server nodes of the same path:

$$\left(\sum_{i=1}^{n_s-1} Z_{s_j,\, f_{(m-1)j}}^{i}\right) - Z_{s_j,\, f_{mj}}^{n_s} \geq R_{p,s_j},\ \forall\, n_s \in p, \forall f_{mj} \in F_j\ ,\ \forall s_j \in S \tag{8}$$

where $Z_{s_j,\, f_{mj}}^{n_s}$ is a Boolean variable, which is equal to 1 when the VNF f_{mj} from SFC s_j is allocated in server node n_s; otherwise, it is 0. R_{p,s_j} is a Boolean variable which is equal to 1 if the SFC s_j utilizes the physical path p; otherwise, it is 0.

To achieve the maximum load balancing that reduces the bandwidth consumption of the network using VNF replicas, an additional constraint is provided as follows:

$$\sum_{n_s}^{N_S} Z_{s_j,f_{mj}}^{n_s} \leq r_a \cdot Rep_{f_{mj}}, \forall s_j \in S, \ \forall \ f_{mj} \in F_j \tag{9}$$

where $Rep_{f_{mj}}$ is a Boolean variable, which is equal to 1 when the VNF f_{mj} from SFC s_j can be replicated (otherwise, it is 0), and r_a is the number of allowed replicas per service chain. In Equation (9), if the VNF f_{mj} can be replicated, then the allowed number of replicas is constrained by r_a. Otherwise, the VNF f_{mj} will be set once in the cloud for each service chain s_j. Based on these constraints, an effective model for the multi-objective VNF placement problem is proposed with the goal of minimizing link bandwidth utilization, network energy, and the overall SFC placement cost, as shown in the following equations:

$$\begin{aligned} &\textit{minimize} \\ &w_1 \cdot \sum_{\forall \ l \in L} U_l + w_1 \cdot \sum_{\forall l \in L} \sum_{\forall \ N} Q_l + w_1 \cdot \sum_{\forall \ f_j \in F_j} C_t \end{aligned} \tag{10}$$

where w_1, w_2, and w_3 are weighting parameters that represent the relative importance of each objective. They must satisfy Equation (11):

$$w_1 + w_2 + w_3 = 1 \tag{11}$$

The link utilization U_l is given by Equation (12):

$$U_l = \sum_{s_j \in S} \sum_{l_1 l_2 \in p} \frac{d_{BWj} \cdot Y_{j,l_1 l_2}^{f_{1j} f_{2j}} \cdot V_p^{l_1 l_2}}{BW_p} \tag{12}$$

where $V_p^{l_1 l_2}$ is a Boolean variable, which is equal to 1 if the physical link $l_1 l_2$ belongs to the physical path p; otherwise, it is 0. The costs C_S associated with the placement of SFC and the corresponding placement of the VNFs can be computed as follows:

$$C_S = \rho \cdot C_{NW} + \sigma \cdot C_{n_s} \tag{13}$$

$$C_{NW} = U_l \cdot \mu_l \tag{14}$$

$$C_{n_s} = \Big(\sum_{n_s \in N_S} \sum_{s_j \in S} \sum_{f_j \in F_j} \Big(\alpha \frac{d_{CPUj}}{R_{CPU}^{max}} + \beta \frac{d_{RAMj}}{R_{RAM}^{max}} + \gamma \frac{d_{Stg}}{R_{Stg}^{max}} \Big) \cdot X_{n_s}^{f_j} \Big) \cdot \mu_{n_s} \tag{15}$$

where C_{NW} represents the cost of network resources, C_{n_s} represents the cost of the server node resources, α, β, and γ represent the types of CPU, memory, and storage in different server nodes, respectively, μ_{n_s} is the cost of consuming one unit of the server node resources, and μ_l is the cost of consuming one unit of bandwidth via one cloud network link. The relative importance of network bandwidth and active server nodes are captured by ρ and σ. The higher the value of ρ, the more important the network bandwidth in the overall SFC placement cost.

The objective function in Equation (10) aims to minimize the utilization of physical links, which will result in a reduced consumption of network bandwidth. Additionally, it reduces network energy consumption and the overall cost of VNF placement.

Using ILP to find the optimal solution of this problem will lead to a scalability problem. Hence, the ILP model is only feasible for small networks. Moreover, due to the NP-hardness of the above problem, it cannot be solved in polynomial time. Hence, a heuristic solution is introduced in the next section to solve this problem. It has the advantage of offering a near-optimal solution for this optimization problem.

4. Heuristic Solution

In this section, the virtual network functions and their replica placement (VNFRP) algorithm is introduced. It is a heuristic algorithm that practically solves the VNF placement problem and provides a near-optimal solution, as shown in Figure 1. The proposed heuristic algorithm should decrease the computational overhead while providing sufficient quality in computing a near-optimal VNF placement using the replication concept. Additionally, to achieve the required goals, the proposed algorithm should be aware of the properties and capabilities of the VNFs and the physical resources.

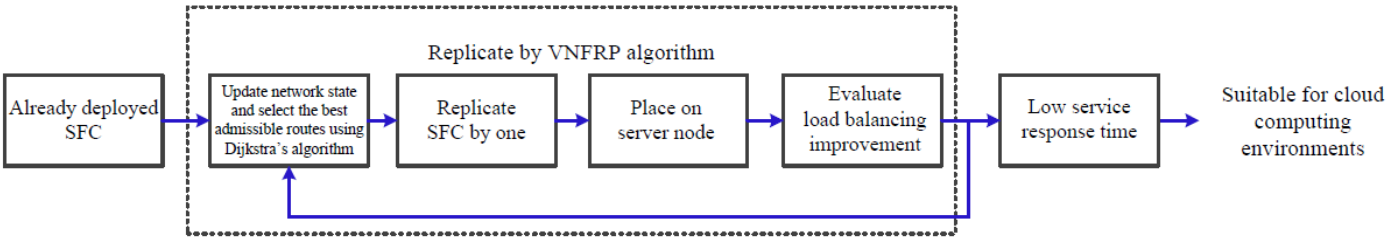


Figure 1. The architecture of the virtual network functions and their replica placement (VNFRP) algorithm.

The proposed algorithm assumes that the service chain requests are delivered dynamically based on customer needs. Each service chain is characterized by an ordered sequence of VNFs to be used in addition to the CPU, memory, storage, and bandwidth requirements for each VNF in the same service chain. The SDN controller periodically discovers the current state of the network where the previous SFCs are deployed, including the corresponding provisioned bandwidth for each VNF in the SFCs. Then, it selects the set of the best admissible routes using Dijkstra's shortest path algorithm [61]. The proposed algorithm is responsible for deploying the VNFs in the incoming SFC request while satisfying their resource requirements. In other words, a VNF service chain is mapped to the VMs hosted on the server nodes in the cloud datacenters. Meanwhile, it preserves the sequence order for the chosen admissible route, where the SDN controller is responsible for routing the traffic according to a predefined sequence in the cloud computing environment.

The proposed algorithm aims to replicate already-deployed SFCs to achieve load balancing and save network bandwidth and energy consumption. This means that the proposed algorithm replicates all VNFs composing the service chain with one replica. This must be performed under the condition that a new replica must be deployed on the route using lower network levels (lower network cost) compared with the unreplicated situation [61]. Then, depending on the predefined order of the VNFs in the SFC, the SDN controller routes the traffic through the original and the new replica. If the proposed algorithm improves load balancing effectively, then the consumed network bandwidth on the links is going to be reduced. The details of the proposed heuristic algorithm are shown in Algorithm 1. As shown in Algorithm 1, the proposed algorithm attempts to deploy the second replica while testing whether the load balancing is improved more than the previous state using only one replica. It adds the number of replicas incrementally until the improvement of load balancing becomes constant or the placement constraints cannot be met. The idea of deploying the original SFC and inserting allowed replicas is to find an iteration number of the available alternative parallel routes suitable for forwarding inter-VNF traffic; hence the decrease of the number of overloaded links and network bandwidth consumption. For all replicated chains, the proposed algorithm attempts to avoid intermediate core switches among the replicated chains as much as possible, mainly to reduce the network energy consumption.

Algorithm 1 VNFRP

Input: $N, L, S, F_j, d_{CPUj}, d_{RAMj}, d_{Stgj}, d_{BWj}$
Constraint: $R_{CPU}^{max}, R_{RAM}^{max}, R_{Stg}^{max}, BW_p, r_a$
for $l = 1$ **to** L **do**
Compute the admissible routes using Dijkstra's shortest path algorithm
end for
for $s = 1$ **to** S **do**
for $f_j = 1$ **to** F_j **do**
Place f_j on server node
Route d_{BWj} over the admissible routes
$R \leftarrow 1$
While Choose number of replicas (R) **do**
Place f_j replicated on server node
for $d_{BWj} = 0$ **to** $\sum L_j$ **do**
Route d_{BWj} over all alternative parallel routes
if $R < r_a$ **then**
$R \leftarrow R + 1$
end if
end for
end while
end for
end for
end for
Output: The solution $X_{n_s}^{f_j}$, $Y_{j,l_1 l_2}^{f_{1j} f_{2j}}$, $Z_{s_j, f_{mj}}^{n_s}$, $Rep_{f_{mj}}$ of the original problem.

5. Experimental Results

In this section, the results of the conducted experimental studies for the proposed solution are presented. The CloudSimSDN-NFV simulator [62] was used in these studies. The results collected from the proposed algorithm are compared against the least full first (LFF) strategy [62]. The LFF strategy tries to place the VNF on the least loaded server node, which has the least amount of resources allocated in terms of computing energy and network bandwidth. In this strategy, VNFs are distributed across the server nodes in the cloud environment, where server nodes that host more VNFs have a lower priority than server nodes that host few VNFs or do not host any VNFs. This results in spreading the traffic across the links and switches of all network levels.

5.1. Simulation Setup

The proposed VNF placement algorithm was implemented using Java programming language and incorporated with the CloudSimSDN-NFV simulator. The physical network topology used in the simulation was a fat tree topology [63] with 64 server nodes, 104 bidirectional links, and the traffic has been generated with 769 demands, as shown in Figure 2. The bandwidth of each higher link (between the core and the distribution switches) was assumed to be 10 Gbps. Meanwhile, the bandwidth of each lower link (between the edge switches and the server nodes) was set to 1 Gbps. There were 66 service chains created in the system. Each service chain consisted of two VNF endpoints (source and destination) and one intermediate VNF. The SDN controller was primarily responsible for controlling the network resources and VNFs, analyzing the information gathered, and making adaptive solutions and comprehensive management in large-scale complex networks [1,4]. Therefore, it was rational to run the proposed algorithm on the SDN controller to improve the network performance based on the information gathered. Additionally, the implementation of the proposed algorithm on the SDN controller allowed for doing the calculation of the shortest forwarding candidate paths based on the smallest link weight [61] and placing the VNF replicas accordingly.

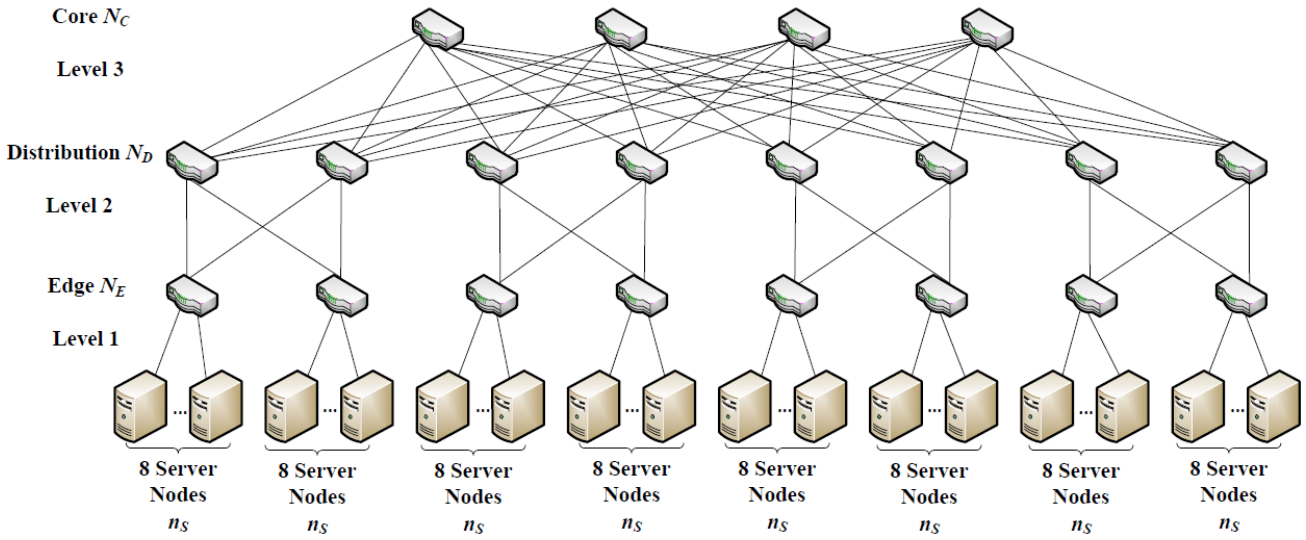


Figure 2. Physical network topology of the cloud computing environment adopted in the simulation.

Each server node had its available computing resources, while each physical link had its available bandwidth. Similarly, the required resources for each VNF in the service chain were set, and the required bandwidth of each virtual link in the service chain was set too. Moreover, the sum of the required resources should have never exceeded its maximum available capacity. It was assumed that the physical network contained a variety of service chains, while the required resources and bandwidth of each service chain were different. In order to ensure adequate service chain performance, it became essential to track the computing and network resources to place the VNFs and virtual links on the server nodes and the physical links accordingly.

5.2. Performance Evaluation

During the experimental studies, comprehensive simulations were examined through a series of different iterations. The number of replicas was increased by one for each iteration. Then, in each scenario, a set of traffic demands was generated, corresponding to the first iteration. The effectiveness of the proposed algorithm was analyzed in terms of four metrics:

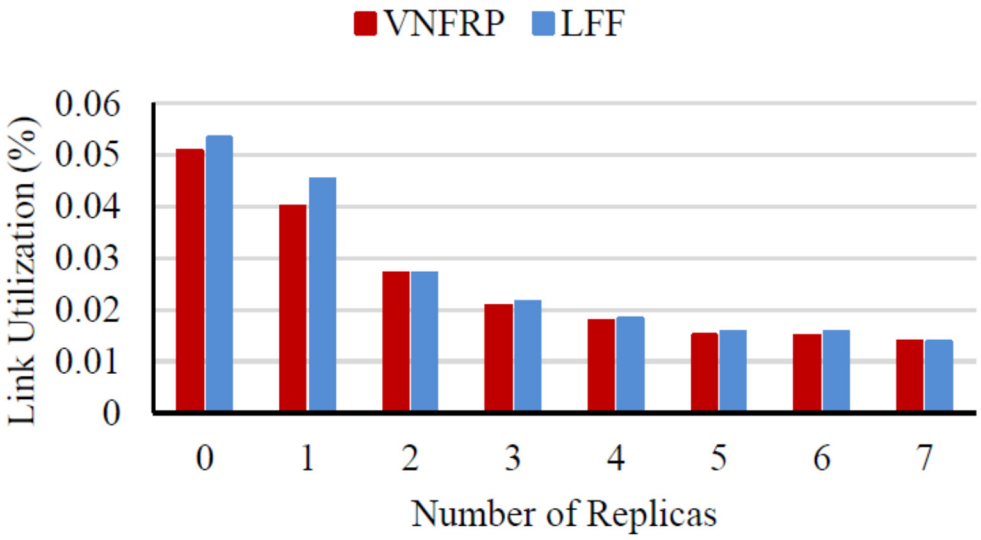
- Network load balancing and link bandwidth utilization;
- Network energy consumption;
- Service response time;
- SFC placement cost.

At the beginning of the experiments, where there was no replica, VNFs were randomly placed on any physical machine that could host VNFs. After the random placement of VNFs, the algorithm increased the number of replicas until reaching the allowed number of them. Meanwhile, the SDN controller searched for the smallest weight-admissible candidate paths that could traverse the VNFs in a predefined order for each traffic demand.

5.2.1. Network Load Balancing and Link Bandwidth Utilization

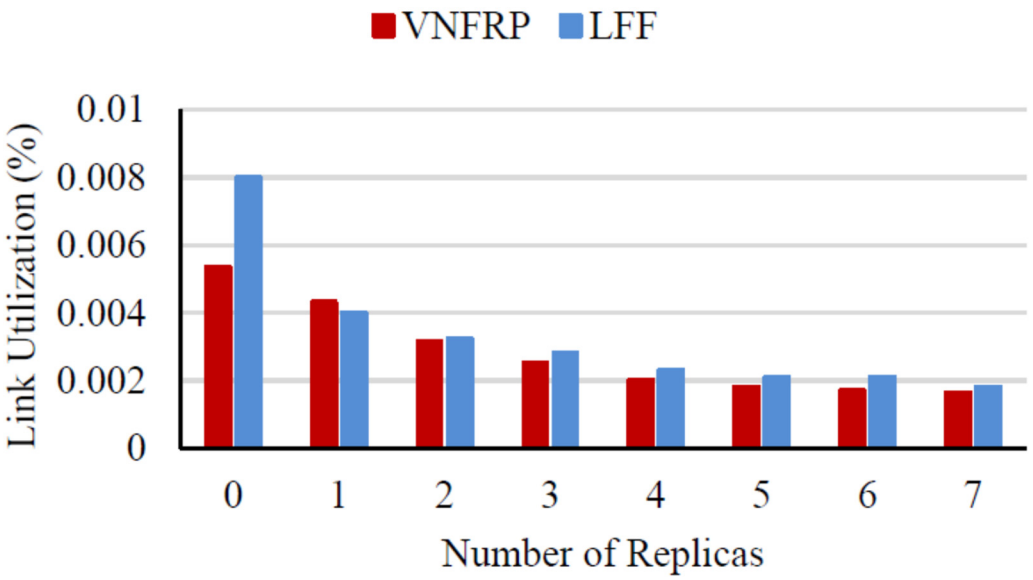
Link bandwidth utilization represents the percentage of the utilized link capacity across the physical links. The links are used to host the service chain virtual paths while meeting the required computing resources of VNFs in the service chain. The link bandwidth utilization was decreased while the network load balancing increased when changing the replica from 0 to 7 with traffic consisting of 769 demands. The load balancing ratio is represented by the percentage of the gap between the initial bandwidth utilization in the absence of a replica and the enhancement bandwidth utilization in each iteration when the replica is increased from 0 to 7 with the same constant traffic demands.

Figure 3 shows the average link bandwidth utilization of each level when the VNFRP algorithm was adopted and the corresponding values for the LFF strategy [62]. Both algorithms were evaluated using different VNF replicas. Initially, when no replicas were used, the VNFRP algorithm randomly placed VNFs on any physical machine that could host VNFs, which resulted in increased bandwidth utilization across all links. More replicas produced better improvements in minimizing link bandwidth utilization, which gradually resulted in much better load balancing, as shown in Figure 4. It is obvious that the decrease in link bandwidth utilization was induced by the increase in network load balancing. The explanation behind such a phenomenon is that as the number of VNF replicas increases, the number of admissible paths over the lower links is proportionally increased. Thus, the remaining available bandwidth will increase as well, which increases the volume of traffic that can pass. The load balancing ratios for all replicas were computed for both the VNFRP algorithm and the LFF strategy, and they are represented in Figure 5. The load balancing ratio decreased significantly in the VNFRP algorithm and the LFF strategy as the number of replicas increased. Nevertheless, in the VNFRP algorithm, the load balancing ratio decreased uniformly on all levels from 80% to 27%, while in the LFF strategy, it decreased differently on each level from 85% to 22.7%. The reduction in the load balancing ratio proves the fact that the improvement of load balancing was relatively high at the first replica and gradually decreased. Moreover, the maximum reduction in link utilization in the VNFRP algorithm was observed in the lowest network level, while it was observed in the highest network level in the LFF strategy. This observation is attributed to the fact that the VNFRP algorithm has a greater probability to choose server nodes that have fewer remaining resources to host VNFs of SFC when compared with the LFF strategy. Meanwhile, the VNFRP algorithm tried to reduce the link bandwidth utilization; hence, its advantage became more pronounced as the number of replicas increased.

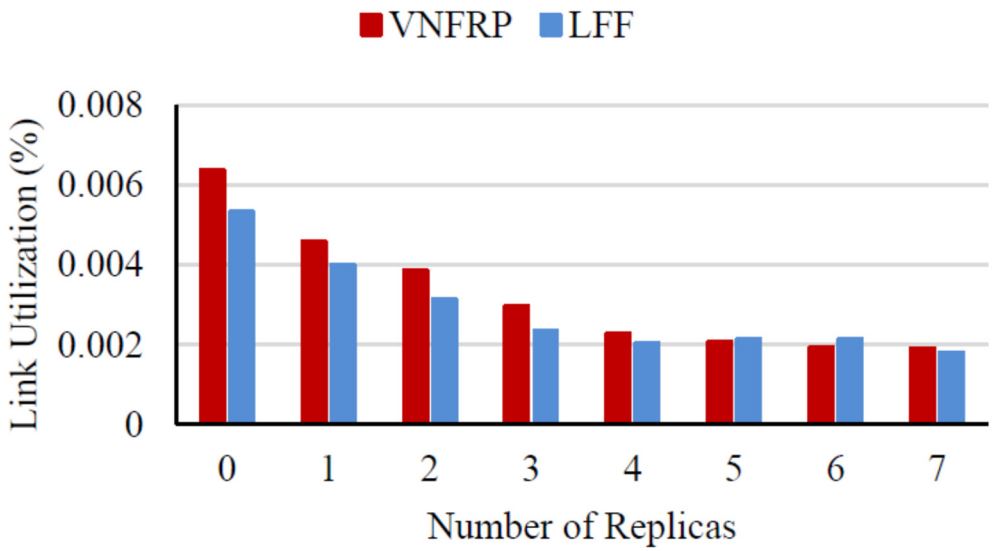


(**a**)

Figure 3. *Cont.*

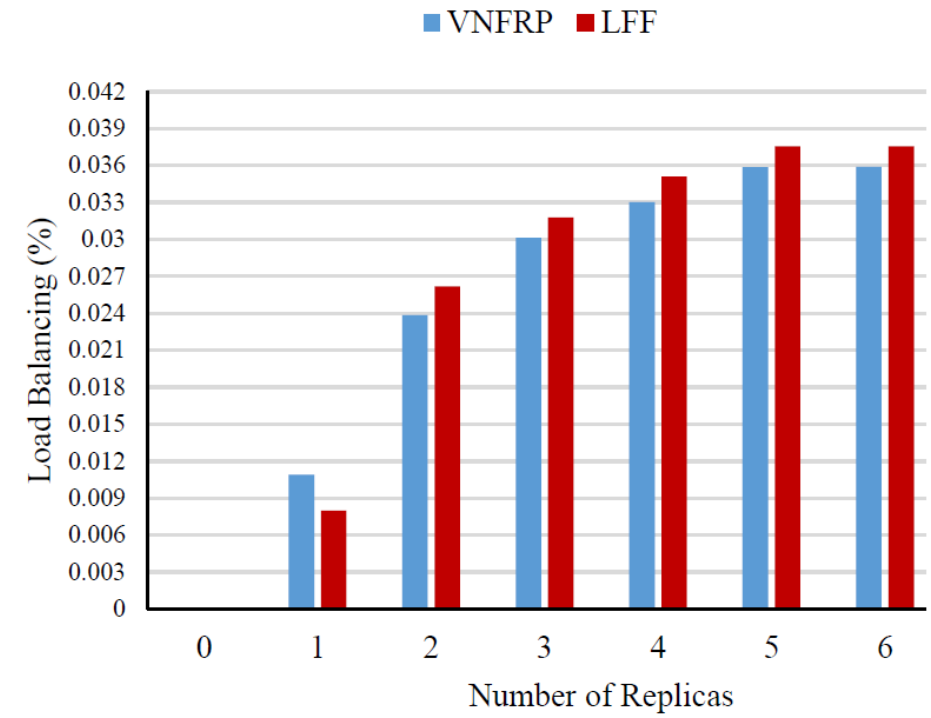


(**b**)

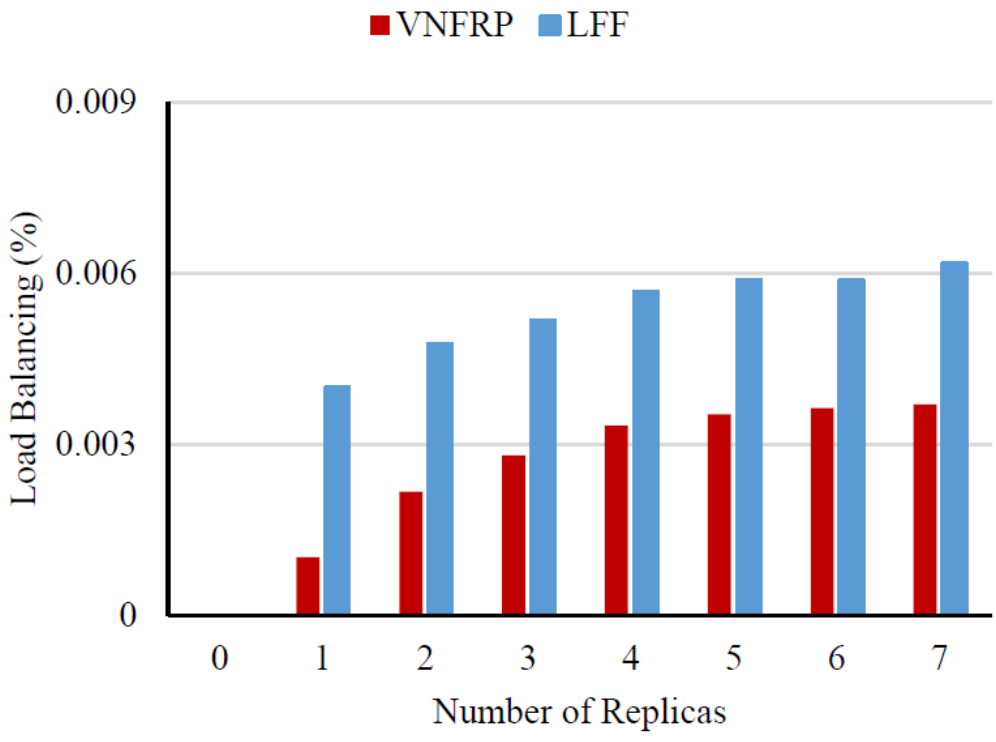


(**c**)

Figure 3. Link utilization when adopting the VNFRP algorithm and the corresponding values when adopting the least full first (LFF) strategy: (**a**) Level 1, (**b**) Level 2, and (**c**) Level 3.

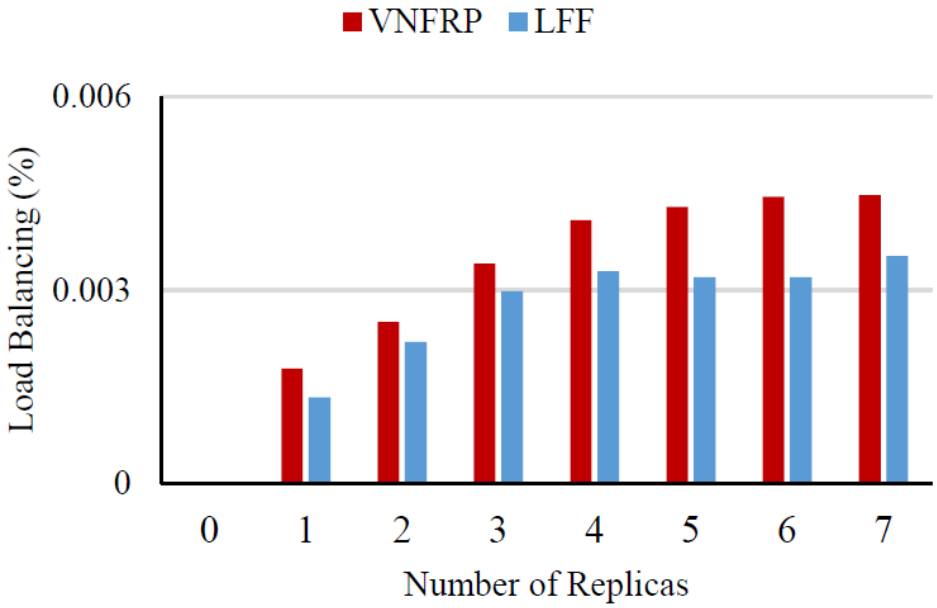


(**a**)



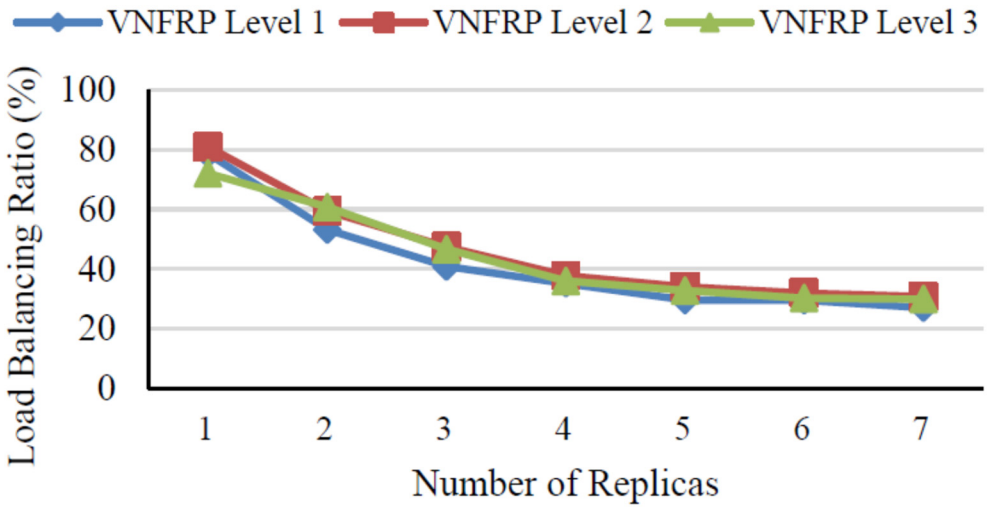
(**b**)

Figure 4. *Cont.*



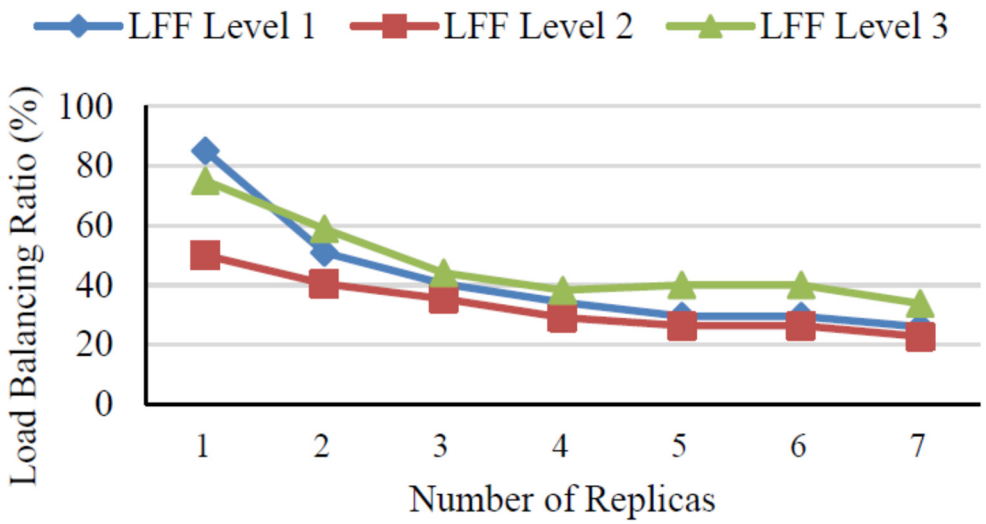
(**c**)

Figure 4. Load balancing when adopting the VNFRP algorithm and the corresponding values when adopting the LFF strategy: (**a**) Level 1, (**b**) Level 2, and (**c**) Level 3.



(**a**)

Figure 5. *Cont*.



(**b**)

Figure 5. Load balancing ratio: (**a**) When adopting the VNFRP algorithm, and (**b**) The corresponding values when adopting the LFF strategy.

5.2.2. Network Energy Consumption

The proposed algorithm places SFC and its replicas by utilizing the resources of the nodes that communicate with the smallest possible number of active network nodes (the smallest weight). Moreover, the SDN controller switches off the unused hosts and switches to reduce energy consumption. The network energy consumption was measured by using a built-in monitoring tool for energy consumption included in the CloudSimSDN-NFV simulator. A Java interface was defined to record the energy consumption of each server node and each switch over a specified period of time, based on the linear models represented in [64,65].

The network energy consumption was almost the same for both the VNFRP algorithm and the LFF strategy at the initial placement without replicas, as shown in Figure 6. This was due to the deployment of VNFs across the cloud network in the LFF strategy contributing to the utilization of more physical hosts. Meanwhile, VNFs were spread randomly around the cloud network in the proposed algorithm, resulting in more physical hosts to be used. Additionally, in both placement algorithms, when the network traffic volumes increased, more network energy was consumed. Figure 6 also shows that the network energy consumption of the cloud network could be saved when the replica was applied to both placement algorithms. The increased number of VNF replicas increased the number of admissible paths that had the smallest possible number of active network nodes, which were used to connect the VNFs on different server nodes. Meanwhile, Figure 6 shows that the improvement of the network energy consumption was higher in the VNFRP algorithm compared with the LFF strategy. The VNFRP algorithm was clearly focused on optimizing link utilization more than server node utilization. Hence, the VNFRP algorithm could reduce the network energy consumption by more than 54% when the number of replicas was increased from 0 to 7. Meanwhile, it decreased the energy consumption by 6% on average compared with the LFF strategy. Moreover, the difference between VNFRP and LFF in network energy consumption became higher as the number of replicas increased. As depicted in Figure 6, the gap in network energy consumption between VNFRP and LFF increased remarkably from about 152 Wh to 612 Wh when the number of replicas was increased.

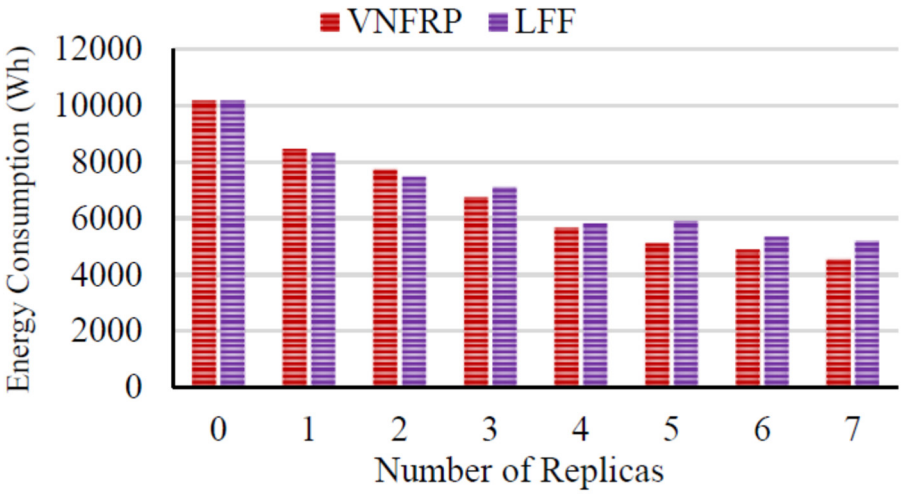


Figure 6. Network energy consumption in the cloud network when adopting the VNFRP algorithm and the corresponding values when adopting the LFF strategy.

5.2.3. Service Response Time

The service response time was measured as the average end-to-end delay of all requests submitted to the cloud network. It included the processing time of the application server nodes and network delays along the SFCs. The service response time was correlated with the number of replicas and requests. To demonstrate the effectiveness of the VNFRP algorithm, the number of replicas was increased while the number of requests was fixed. As shown in Figure 7, when the number of replicas increased, the service response time decreased, which resulted in a rapid convergence of the VNFRP algorithm and the LFF strategy. Additionally, Figure 7 shows that the service response time of the LFF strategy was slightly longer than the corresponding one in the VNFRP algorithm, but it was still acceptable. Moreover, the observed difference was not significant between the LFF strategy and the VNFRP algorithm in each replica except for the initial placement, where the replica was not applied. The service response time could be reduced in the VNFRP algorithm by more than 16% in the best case and by more than 4% in the worst case, compared with the LFF strategy, as shown in Figure 7. This occurred because VNFs were deployed over server nodes connected to the lower network levels. Therefore, it led to routing the traffic among the VNFs of the SFC through these small weight shortest paths, which resulted in a corresponding reduction in end-to-end delay. Additionally, when the number of replicas was more than 4, the service response time of both VNFRP and LFF was nearly constant. With the advantages of the fast response time and rapid convergence, the VNFRP algorithm had an effective impact on practical networks that had thousands of nodes and VNFs. Therefore, the VNFRP algorithm could be applied to NFV-enabled environments with heterogeneous nodes and workloads.

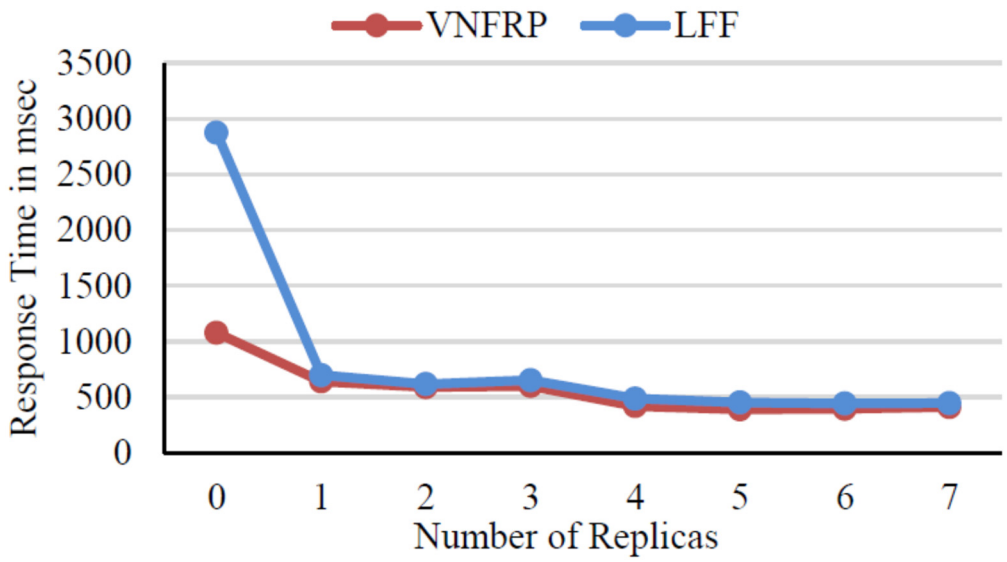


Figure 7. Service response time when adopting the VNFRP algorithm and the corresponding values when adopting the LFF strategy.

5.2.4. SFC Placement Cost

All VNFs of each SFC and its replicas occupied one node or a set of nodes that were connected through the smallest weight. Hence, they could utilize a lot of active nodes with an increased number of replicas, as shown in Figure 8. Meanwhile, Figure 9 shows the overall SFC placement cost, as defined in Equation (13). As shown in this figure, it was affected by applying the replica concept. The parameters ρ and σ were adjusted to 10 and 0.1, respectively, to capture the relative importance of the cost of network resources and the cost of server node resources. When ρ was higher than σ, the cost of network resources became more relevant than the cost of server node resources. It is clear that the LFF strategy provided a higher SFC placement cost compared with the VNFRP algorithm. Furthermore, the difference in SFC placement cost between the VNFRP algorithm and the LFF strategy increased when the number of replicas was increased. As shown in Figure 9, the VNFRP algorithm reduced the SFC placement cost by more than 67%, while the LFF strategy reduced the SFC placement cost by no more than 36% when the number of replicas increased from 0 to 7. Meanwhile, the average SFC placement cost gap between the VNFRP algorithm and the LFF strategy increased from 14% to 57%. As shown in Figure 8, the increase in the number of active server nodes was relatively the same in both VNFRP and LFF. Therefore, the cost of network resources only affected the SFC placement cost. As a result, the VNFRP algorithm could be implemented to allocate sophisticated SFCs consisting of multiple VNFs more efficiently than the LFF strategy. Moreover, the results in Figure 9 demonstrate that when the number of replicas increased, the SFC placement cost decreased. Additionally, more flexibility and efficiency were guaranteed in the VNFRP algorithm than the LFF strategy in terms of SFC placement and network bandwidth utilization. This is attributed to the fact that shorter paths could be used to direct VNF traffic in the VNFRP algorithm. Figure 9 shows that it is efficient and significant in the context of VNF and SFC placement problems to jointly and dynamically control the network and the server node resources.

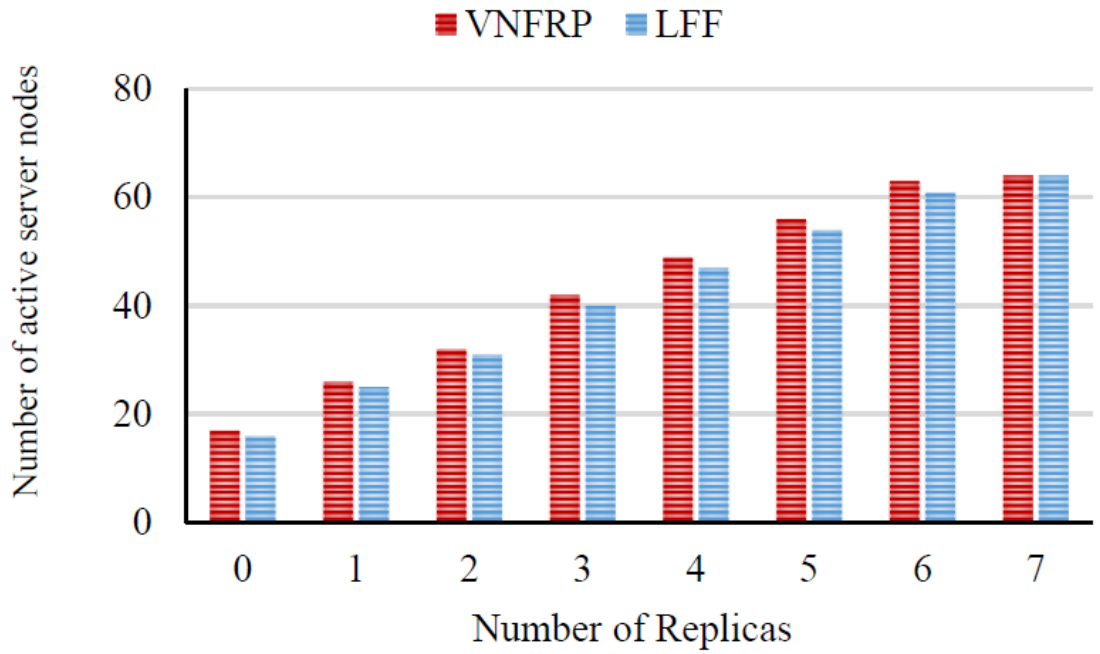


Figure 8. Number of active server nodes when adopting the VNFRP algorithm and the corresponding values when adopting the LFF strategy.

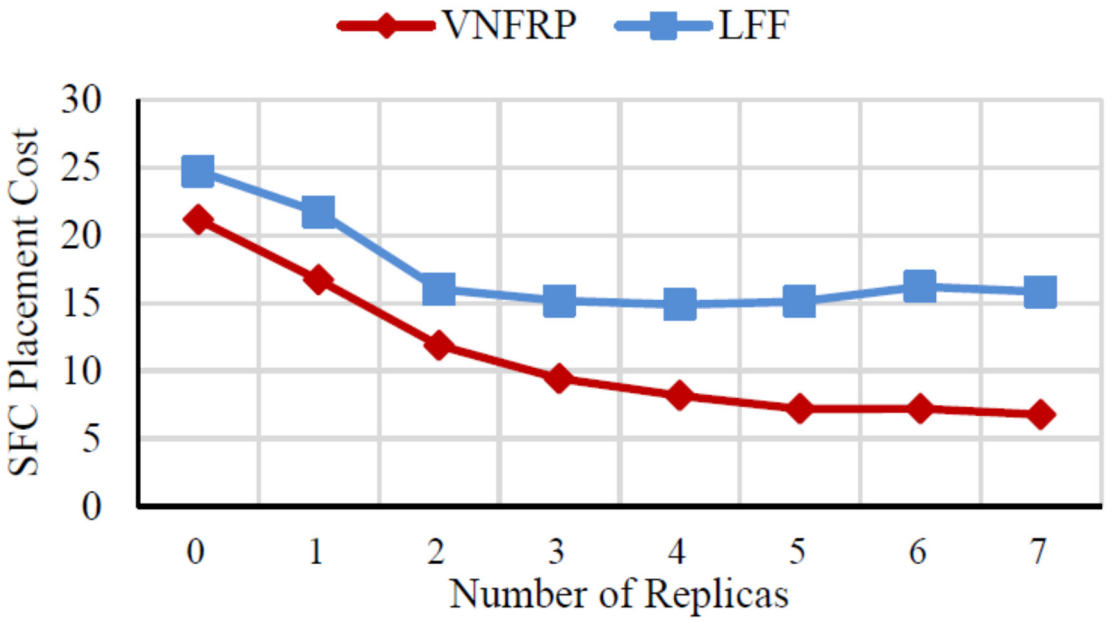


Figure 9. Service function chain (SFC) placement cost when adopting the VNFRP algorithm and the corresponding values when adopting the LFF strategy.

6. Conclusions

This paper studied the problem of VNF placement for SFCs using an effective replica technique in software-defined cloud computing environments. The problem was first formulated using an ILP model with the objective to reduce the link bandwidth utilization, energy consumption, and SFC placement cost. Then, a heuristic algorithm, named VNFRP, was designed to find a near-optimal solution for this problem. The VNFRP algorithm exploits the SDN controller knowledge about the smallest weight-admissible candidate paths to traverse the traffic of VNFs in a predefined order for each SFC and its replicas. The

experimental results revealed the effectiveness of the VNFRP algorithm. It could achieve an improvement in load balancing by 80% when replicas were increased. Moreover, the VNFRP algorithm provided better network performance in terms of network energy consumption. Additionally, it could efficiently reduce the VNF placement cost. Furthermore, its service response time could be kept sufficiently low, which makes it applicable for large networking environments. While these findings are encouraging, this work can be seen as a significant move toward SFC placement, mainly because it deals with the volatility of network traffic among VNFs in cloud computing environments and their requirements. This is accomplished using an efficient replica strategy that provides a new methodology for load balancing.

Exploring the performance of the VNFRP algorithm across different network architectures can be considered as one of the main future perspectives. Moreover, tuning the parameters of the VNFRP algorithm for complex traffic models that have more variability and uncertainty could be another potential area of interest.

Author Contributions: Conceptualization, M.A.A., G.A.E. and W.R.A.; formal analysis, M.A.A.; methodology, M.A.A.; supervision, G.A.E. and W.R.A.; validation, G.A.E. and W.R.A.; writing—original draft, M.A.A.; writing—review and editing, G.A.E. and W.R.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Mijumbi, R.; Serrat, J.; Gorricho, J.-L.; Bouten, N.; De Turck, F.; Boutaba, R. Network function virtualization: State-of-the-art and research challenges. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 236–262. [CrossRef]
2. Herrera, J.G.; Botero, J.F. Resource allocation in NFV: A comprehensive survey. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 518–532. [CrossRef]
3. Laghrissi, A.; Taleb, T. A survey on the placement of virtual resources and virtual network functions. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1409–1434. [CrossRef]
4. Han, B.; Gopalakrishnan, V.; Ji, L.; Lee, S. Network function virtualization: Challenges and opportunities for innovations. *IEEE Commun. Mag.* **2015**, *53*, 90–97. [CrossRef]
5. Kreutz, D.; Ramos, F.M.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2014**, *103*, 14–76. [CrossRef]
6. Nunes, B.A.A.; Mendonca, M.; Nguyen, X.-N.; 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]
7. Shiomoto, K. Research challenges for network function virtualization-re-architecting middlebox for high performance and efficient, elastic and resilient platform to create new services. *IEICE Trans. Commun.* **2018**, *101*, 96–122. [CrossRef]
8. Sherry, J.; Hasan, S.; Scott, C.; Krishnamurthy, A.; Ratnasamy, S.; Sekar, V. Making middleboxes someone else's problem: Network processing as a cloud service. *ACM SIGCOMM Comput. Commun. Rev.* **2012**, *42*, 13–24. [CrossRef]
9. Halpern, J.; Pignataro, C. *Service function chaining (SFC) architecture, RFC 7665*; Internet Engineering Task Force (IETF) Service Function Chaining (SFC) Working Group (WG): Fremont, CA, USA, 2015.
10. Bhamare, D.; Jain, R.; Samaka, M.; Erbad, A. A survey on service function chaining. *J. Netw. Comput. Appl.* **2016**, *75*, 138–155. [CrossRef]
11. Cao, J.; Zhang, Y.; An, W.; Chen, X.; Sun, J.; Han, Y. VNF-FG design and VNF placement for 5G mobile networks. *Sci. China Inf. Sci.* **2017**, *60*, 040302. [CrossRef]
12. Li, D.; Hong, P.; Xue, K.; Pei, J. Availability aware VNF deployment in datacenter through shared redundancy and multi-tenancy. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 1651–1664. [CrossRef]
13. Fischer, A.; Botero, J.F.; Beck, M.T.; De Meer, H.; Hesselbach, X. Virtual network embedding: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1888–1906. [CrossRef]
14. Kulkarni, S.; Arumaithurai, M.; Ramakrishnan, K.; Fu, X. Neo-NSH: Towards scalable and efficient dynamic service function chaining of elastic network functions. In Proceedings of the 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, 7–9 March 2017; pp. 308–312. [CrossRef]
15. Reddy, V.S.; Baumgartner, A.; Bauschert, T. Robust embedding of VNF/service chains with delay bounds. In Proceedings of the 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, 7–10 November 2016; pp. 93–99. [CrossRef]

16. Yang, S.; Pan, L.; Wang, Q.; Liu, S.; Zhang, S. Subscription or pay-as-you-go: Optimally purchasing iaas instances in public clouds. In Proceedings of the 2018 IEEE International Conference on Web Services (ICWS), San Francisco, CA, USA, 2–7 July 2018; pp. 219–226. [CrossRef]
17. Zhang, S.; Yuan, D.; Pan, L.; Liu, S.; Cui, L.; Meng, X. Selling reserved instances through pay-as-you-go model in cloud computing. In Proceedings of the 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, 25–30 June 2017; pp. 130–137. [CrossRef]
18. Prajapati, A.G.; Sharma, S.J.; Badgujar, V.S. All about cloud: A systematic survey. In Proceedings of the 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, India, 5 January 2018; pp. 1–6. [CrossRef]
19. vineel Anvith, P.; Gunavathi, N.; Malarkodi, B.; Rebekka, B. A Survey on Network Functions Virtualization for Telecom Paradigm. In Proceedings of the 2019 TEQIP III Sponsored International Conference on Microwave Integrated Circuits, Photonics and Wireless Networks (IMICPW), Tiruchirappalli, India, 22–24 May 2019; pp. 302–306. [CrossRef]
20. Nguyen, V.-G.; Brunstrom, A.; Grinnemo, K.-J.; Taheri, J. SDN/NFV-based mobile packet core network architectures: A survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1567–1602. [CrossRef]
21. Chiosi, M.; Wright, S.; Erfanian, J.; Smith, B.; Briscoe, B.; Reid, A.; Willis, P. Network functions virtualisation: Network operator perspectives on industry progress. In Proceedings of the SDN and OpenFlow World Congress, Dusseldorf, Germany, 14–17 October 2014.
22. 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.* **2018**, *30*, 2179–2192. [CrossRef]
23. Liu, Y.; Zhang, H.; Guan, H.; Wang, Y. A method for adaptive resource adjustment of dynamic service function chain. *IEEE Access* **2018**, *6*, 69988–70004. [CrossRef]
24. Rankothge, W.; Le, F.; Russo, A.; Lobo, J. Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms. *IEEE Trans. Netw. Serv. Manag.* **2017**, *14*, 343–356. [CrossRef]
25. Zhang, B.; Hwang, J.; Wood, T. Toward online virtual network function placement in software defined networks. In Proceedings of the 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), Beijing, China, 20–21 June 2016; pp. 1–6. [CrossRef]
26. Liu, Y.; Pei, J.; Hong, P.; Li, D. Cost-efficient virtual network function placement and traffic steering. In Proceedings of the ICC 2019–2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6. [CrossRef]
27. Schneider, S.; Dräxler, S.; Karl, H. Trade-offs in dynamic resource allocation in network function virtualization. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, UAE, 9–13 December 2018; pp. 1–3. [CrossRef]
28. Bari, F.; Chowdhury, S.R.; Ahmed, R.; Boutaba, R.; Duarte, O.C.M.B. Orchestrating virtualized network functions. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 725–739. [CrossRef]
29. Cohen, R.; Lewin-Eytan, L.; Naor, J.S.; Raz, D. Near optimal placement of virtual network functions. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, China, 26 April–1 May 2015; pp. 1346–1354. [CrossRef]
30. Moens, H.; De Turck, F. VNF-P: A model for efficient placement of virtualized network functions. In Proceedings of the 10th International Conference on Network and Service Management (CNSM) and Workshop, Rio de Janeiro, Brazil, 17–21 November 2014; pp. 418–423. [CrossRef]
31. Askari, L.; Hmaity, A.; Musumeci, F.; Tornatore, M. Virtual-network-function placement for dynamic service chaining in metro-area networks. In Proceedings of the 2018 International Conference on Optical Network Design and Modeling (ONDM), Dublin, Ireland, 14–17 May 2018; pp. 136–141. [CrossRef]
32. Soualah, O.; Mechtri, M.; Ghribi, C.; Zeghlache, D. A green VNFs placement and chaining algorithm. In Proceedings of the NOMS 2018–2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–5. [CrossRef]
33. Mechtri, M.; Ghribi, C.; Zeghlache, D. Vnf placement and chaining in distributed cloud. In Proceedings of the 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 27 June–2 July 2016; pp. 376–383. [CrossRef]
34. Bouet, M.; Leguay, J.; Combe, T.; Conan, V. Cost-based placement of vDPI functions in NFV infrastructures. *Int. J. Netw. Manag.* **2015**, *25*, 490–506. [CrossRef]
35. 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. [CrossRef]
36. Xia, M.; Shirazipour, M.; Zhang, Y.; Green, H.; Takacs, A. Network function placement for NFV chaining in packet/optical datacenters. *J. Lightwave Technol.* **2015**, *33*, 1565–1570. [CrossRef]
37. Tavakoli-Someh, S.; Rezvani, M.H. Multi-objective virtual network function placement using NSGA-II meta-heuristic approach. *J. Supercomput.* **2019**, *75*, 6451–6487. [CrossRef]
38. Sang, Y.; Ji, B.; Gupta, G.R.; Du, X.; Ye, L. Provably efficient algorithms for joint placement and allocation of virtual network functions. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9. [CrossRef]
39. Gu, S.; Li, Z.; Wu, C.; Huang, C. An efficient auction mechanism for service chains in the NFV market. In Proceedings of the IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9. [CrossRef]

40. Sekar, V.; Egi, N.; Ratnasamy, S.; Reiter, M.K.; Shi, G. Design and implementation of a consolidated middlebox architecture. In Proceedings of the Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12), San Jose, CA, USA, 25–27 April 2012; pp. 323–336.
41. Beloglazov, A.; Buyya, R. OpenStack Neat: A framework for dynamic consolidation of virtual machines in openstack clouds—A blueprint. In Proceedings of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Melbourne, Australia, 14 August 2012; pp. 1–18.
42. Rankothge, W.; Ma, J.; Le, F.; Russo, A.; Lobo, J. Towards making network function virtualization a cloud computing service. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 89–97. [CrossRef]
43. Ma, W.; Medina, C.; Pan, D. Traffic-aware placement of NFV middleboxes. In Proceedings of the 2015 IEEE Global Communication Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–6. [CrossRef]
44. Bhamare, D.; Samaka, M.; Erbad, A.; Jain, R.; Gupta, L.; Chan, H.A. Optimal virtual network function placement in multi-cloud service function chaining architecture. *Comput. Commun.* **2017**, *102*, 1–16. [CrossRef]
45. Gember-Jacobson, A.; Viswanathan, R.; Prakash, C.; Grandl, R.; Khalid, J.; Das, S.; Akella, A. OpenNF: Enabling innovation in network function control. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 163–174. [CrossRef]
46. Kawashima, R. vNFC: A virtual networking function container for SDN-enabled virtual networks. In Proceedings of the 2012 Second Symposium on Network Cloud Computing and Applications, London, UK, 3–4 December 2012; pp. 124–129. [CrossRef]
47. Sama, M.R.; Contreras, L.M.; Kaippallimalil, J.; Akiyoshi, I.; Qian, H.; Ni, H. Software-defined control of the virtualized mobile packet core. *IEEE Commun. Mag.* **2015**, *53*, 107–115. [CrossRef]
48. Deric, N.; Varasteh, A.; Basta, A.; Blenk, A.; Pries, R.; Jarschel, M.; Kellerer, W. Coupling VNF orchestration and SDN virtual network reconfiguration. In Proceedings of the 2019 International Conference on Networked Systems (NetSys), Munich, Germany, 18–21 March 2019; pp. 1–3. [CrossRef]
49. OpenStack. Available online: http://docs.openstack.org/ (accessed on 4 October 2020).
50. ETSI, OpenStack Liason Statement: NFV Requirements. Available online: https://wiki.openstack.org/w/images/c/c7/NFV(14)000154r2_NFV_LS_to_OpenStack.pdf (accessed on 4 October 2020).
51. Fu, J.; Li, G. An Efficient VNF Deployment Scheme for Cloud Networks. In Proceedings of the 2019 IEEE 11th International Conference on Communications Software and Networks (ICCSN), Chongqing, China, 12–15 June 2019; pp. 497–502. [CrossRef]
52. Sallam, G.; Ji, B. Joint placement and allocation of virtual network functions with budget and capacity constraints. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 523–531. [CrossRef]
53. Pei, J.; Hong, P.; Pan, M.; Liu, J.; Zhou, J. Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks. *IEEE J. Sel. Areas Commun.* **2019**, *38*, 263–278. [CrossRef]
54. Zhong, X.; Wang, Y.; Qiu, X. Cost-aware service function chaining with reliability guarantees in NFV-enabled Inter-DC network. In Proceedings of the 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Washington, DC, USA, 8–12 April 2019; pp. 304–311.
55. Ananth, M.; Sharma, R. Cost and performance analysis of network function virtualization based cloud systems. In Proceedings of the 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, India, 5–7 January 2017; pp. 70–74. [CrossRef]
56. Assi, C.; Ayoubi, S.; El Khoury, N.; Qu, L. Energy-aware mapping and scheduling of network flows with deadlines on VNFs. *IEEE Trans. Green Commun. Netw.* **2018**, *3*, 192–204. [CrossRef]
57. Zhang, X.; Xu, Z.; Fan, L.; Yu, S.; Qu, Y. Near-Optimal Energy-Efficient Algorithm for Virtual Network Function Placement. *IEEE Trans. Cloud Comput.* **2019**. [CrossRef]
58. Farkiani, B.; Bakhshi, B.; Mirhassani, S.A. A Fast Near-Optimal Approach for Energy-Aware SFC Deployment. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 1360–1373. [CrossRef]
59. Carpio, F.; Jukan, A. Improving reliability of service function chains with combined VNF migrations and replications. *arXiv* **2017**, arXiv:1711.08965.
60. Carpio, F.; Bziuk, W.; Jukan, A. Replication of virtual network functions: Optimizing link utilization and resource costs. In Proceedings of the 2017 40th Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 22–26 May 2017; pp. 521–526. [CrossRef]
61. Abdelaal, M.A.; Ebrahim, G.A.; Anis, W.R. A scalable network-aware virtual machine allocation strategy in multi-datacentre cloud computing environments. *Int. J. Cloud Comput.* **2019**, *8*, 183–206. [CrossRef]
62. Son, J.; He, T.; Buyya, R. CloudSimSDN-NFV: Modeling and simulation of network function virtualization and service function chaining in edge computing environments. *Softw. Pract. Exp.* **2019**, *49*, 1748–1764. [CrossRef]
63. Al-Fares, M.; Loukissas, A.; Vahdat, A. A scalable, commodity data center network architecture. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 63–74. [CrossRef]
64. Wang, X.; Yao, Y.; Wang, X.; Lu, K.; Cao, Q. CARPO: Correlation-aware power optimization in data center networks. In Proceedings of the 2012 IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 1125–1133. [CrossRef]
65. Pelley, S.; Meisner, D.; Wenisch, T.F.; Vangilder, J.W. Understanding and abstracting total data center power. In Proceedings of the WEED 2009: Workshop on Energy-Efficient Design, Austin, TX, USA, 20 June 2009.