IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 17, 2021, accepted March 1, 2021, date of publication March 23, 2021, date of current version April 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068342

High Availability Deployment of Virtual Network Function Forwarding Graph 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
[2]Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University, Cairo 11517, Egypt
Corresponding author: Gamal A. Ebrahim (gamal.ebrahim@eng.asu.edu.eg)

ABSTRACT Network Function Virtualization (NFV) stands out quickly as a promising innovation in telecommunication networks. It leverages the concept of cloud technology and virtualization techniques. However, the continuity of cloud network services has become an essential availability requirement for NFV. Failure of Virtual Network Functions (VNFs) may cause critical quality assurance problems for such network services. VNF chaining and placement can be considered as the VNF Forwarding Graph (VNF-FG) mapped on a cloud provider infrastructure, also called NFV Infrastructure (NFVI). This mapping is addressed while neglecting the massive link utilization and bandwidth consumption that can be encountered during VNF recovery phase. In this paper, an Integer Linear Programming (ILP) approach is developed to model VNF-FG deployment problem while guaranteeing high availability against node/VM failures. Then, the Redundant VNF-FG Deployment (RVNF-FGD) heuristic algorithm is proposed that takes VNF migration into consideration. RVNF-FGD algorithm attempts to find a near-optimal solution that achieves a trade-off between availability and scalability with a reasonable convergence time. Thus, facilitating the practical deployment of the proposed approach. Simulation results confirm that RVNF-FGD algorithm is capable of simultaneously reducing link utilization and bandwidth consumption across the core layer of the cloud network. In addition, it reduces VNF-FG communication cost and overall energy consumption. The convergence time of RVNF-FGD algorithm is assessed by applying it to broader cloud network architectures. This assessment indicates the viability of the proposed approach in responding quickly to failures.

INDEX TERMS Network function virtualization (NFV), network function virtualization infrastructure (NFVI), software-defined networking (SDN), virtual network function (VNF), virtual network function forwarding graph (VNF-FG).

I. INTRODUCTION

Network Function Virtualization (NFV) [1]–[3] is an innovative network architecture paradigm. It leverages virtualization technology to separate software instances from hardware appliances. In other words, it decouples network functions from the underlying hardware to form Virtualized Network Functions (VNFs). VNF can be hosted on Virtual Machines (VMs), which, in turn, can be deployed on a general-purpose hardware. NFV can bring a variety of advantages to network operators along with other new technologies such as Software-Defined Networking (SDN) [4] and cloud computing. Adopting NFVs leads to several benefits. First, it simplifies the programming of networks on a need-based basis. Second, it expands networking capabilities. Third, it improves cost-efficiency. Fourth, it allows different network services to share the same physical infrastructure. On the other hand, SDN decouples the control plane from the underlying data plane. Therefore, it provides the possibility to control the routing of packets in a logically centralized manner. Hence, SDN can clearly lead to efficient utilization of network and computing resources [5]. Furthermore, SDN enables the inter-operability among multiple VNFs running on multiple VMs located across the datacenters. Consequently, it provides automation management and rapid deployment of dynamic network services in the cloud [6].

NFV adds a new dimension to the design, deployment, and management of heterogeneous network services using service function chaining concept [7]. Service function chain, also known as VNF Forwarding Graph (VNF-FG) [8], [9],

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque.

is defined as a virtualized network infrastructure consisting of multiple VNFs. These VNFs are interconnected through a predefined order to provide a specific network service to the end-user.

VNF-FG is required to handle real-time traffic of the streaming applications that makes up a large portion of current network traffic. Thus, the availability of cloud network services has become an essential requirement for NFV. Failure to provide the required level of availability for VNF-FGs may lead to critical quality assurance problems for these network services. Consequently, fines are incurred on network and service providers due to service interruption. Meanwhile, VNF placement problem has attracted the attention of many researchers due to its significant impact on the performance of NFV [10]–[12].

Efficient deployment of NFV placement approaches requires addressing several key challenges. In particular, the deployment of heterogeneous network functions for VNF-FGs over the cloud infrastructure. Hence, service providers face a range of trade-offs among different goals such as achieving service availability while minimizing link utilization and bandwidth consumption. Additionally, service providers need to reduce the energy consumption of active computing infrastructures. These goals are contradictory where achievement of service availability can lead to massive link utilization and bandwidth consumption that can be clearly encountered during VNF recovery phase. Meanwhile, optimizing link utilization and bandwidth consumption increase energy consumption due to spending more active computing resources to deploy VNF-FGs. A significant portion of energy consumption in computing nodes is converted into electrical utility cost for service providers [13]. Consequently, considering the amount of consumed energy as a goal for VNF-FG placement algorithms allows service providers to minimize their electrical utility cost.

This paper focuses on how to optimize VNF placement for VNF-FGs on the cloud provider infrastructure with a predefined level of availability. An Integer Linear Programming (ILP) approach is developed to model VNF-FG deployment problem while guaranteeing high availability against node/VM failures. Additionally, a variety of parameters are considered such as link utilization, bandwidth consumption, and overall energy consumption to achieve high availability while complying with convergence time requirements. In order to tune these parameters, the Redundant VNF-FG Deployment (RVNF-FGD) heuristic algorithm is introduced in this paper that takes into consideration these parameters in addition to VNF migration. Moreover, it could reduce the communication cost of VNF-FG. Indeed, there are several key objectives of RVNF-FGD algorithm such as achieving high availability, reducing link utilization, minimizing bandwidth consumption across the core layer of cloud network, reducing VNF-FG communication cost, and reducing the overall energy consumption.

The rest of this paper is organized as follows: Section II presents the related work followed by Section III that introduces the system model and problem formulation, where the details of the Integer Linear Programming model of the problem are presented. Meanwhile, Section IV presents the heuristic solution of the problem followed by its complexity analysis in Section V. Then, the results of the experimental study of the proposed solution are detailed in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK

European Telecommunications Standards Institute (ETSI) has introduced the definition of NFV and specifies its essential architecture and requirements [14]. ETSI operates on MANO framework [15] that allows VNFs to be deployed, managed, and run over NFV Infrastructure (NFVI), where VNFs are mapped to VMs located above the NFVI hypervisor layer. Another related work based on the ETSI architectural requirements is Cloud4NFV [16], [17], which provides a comprehensive management platform. Other MANO architectures and frameworks are proposed in [18]–[21].

Some of the earlier works [23]–[25] consider VNF placement for VNF-FGs as an extension of Virtual Network Embedding (VNE) problem [26], [27]. VNE problem is NP-hard [28], hence, VNF Forwarding Graph Embedding (VNF-FGE) problem is NP-hard too. Consequently, the optimal solution of the problem can only be applied to small instance sizes. However, there are different objectives and constraints for VNF placement for VNF-FGs and VNE [29]. VNE allocates only the requested virtual resources on the physical resources. In contrast, the requested resources and infrastructure may be virtual or physical in VNF-FGs.

VNF placement problem has recently attracted the attention of several research studies. For instance, early work in [30] studies the optimal placement of VNFs in hybrid scenarios. It assumes that some network functions are supported by dedicated physical hardware and others are virtualized on demand. Additionally, it introduces an ILP model with the goal of reducing the number of physical nodes. This leads to network size limitation due to the complexity of the ILP model. Meanwhile, the authors in [31] present a survey of recent research efforts on VNF placement, chaining, and migration, in addition to providing future directions and challenges. Moreover, an approach for offline batch embedding of multiple service chains is proposed in [32]. It tries to increase the profits or minimize the cost when all requests are included. Meanwhile, the study in [32] proposes a static VNF placement approach. The study in [33] tries to relocate the VNFs while reducing the control overhead in the management of service paths. It provides flexibility, adaptability, and ease of configuration. Moreover, the study in [34] proposes a framework for allocating VNF in 5G datacenters. It models the VNF placement problem in the context of user mobility as a multi-objective integer linear programming problem. It tries to reach a trade-off between the number of VNF relocations and the overall response time. Consequently, it contributes to improving the QoS for users. Additionally, the study in [35] addresses resource allocation with VNF-FG embedding. The problem of VNF placement and admission control in the network infrastructure is modeled as a Mixed Integer Linear Programming (MILP) problem. Meanwhile, the study in [36] proposes a VNF placement and routing algorithm for multicast service chain requests by merging multiple service paths. It mainly tries to optimize network resource utilization and reduce overall cost. Another facet of the problem is to address the inadequate VNF placement that leads to waste of resources and degradation in the performance of physical equipment. In this context, the study in [37] tackles this problem by proposing a framework for dynamic service function chaining and placement in the optimum physical node. Different from some of the previous algorithms, RVNF-FGD, proposed in this paper, dynamically allocates VNF-FG. It takes into consideration both server node resources and network resources. Consequently, it leads to the most effective use of physical devices.

Many of the recent research efforts on NFV architecture consider VNF placement problem with multiple objectives. These objectives include VNF placement cost, network resources cost, and license cost. For example, the study in [38] proposes a solution for the deployment and resource allocation of VNF-FG. It provides a cost model that addresses the trade-off between service efficiency and network cost. Meanwhile, the study in [39] proposes a strategy to place VNF-FGs with the goals of enhancing service performance, reducing VNF deployment cost, and meeting constraints such as CPU, memory, and disk capabilities. The study in [40] proposes a service chain placement model that reduces the cost of VNF placement and link utilization. Additionally, the study in [41] focuses on cost-effective VNF placement and traffic steering. Its main goal is reducing node running cost, VNF placement cost, and network infrastructure cost. Nevertheless, most of these research efforts do not provide effective methods for addressing these joint objectives [23], [42]. Despite that all the above mentioned solutions can deploy VNF-FGs, none of them provides a comprehensive analysis of the problem of deploying VNF-FGs with respect to communication cost. Meanwhile, RVNF-FGD cost function includes the cost of allocating bandwidth for virtual links, the energy cost of network devices, and the cost of migrating failed VNFs.

The studies in [42], [43] focus on the deployment of service function chain from a resource allocation perspective. The authors in [42] introduce a dynamic programming approach to solve VNF-FG deployment problem aiming to minimize operational expenditure and maximize network utilization. Meanwhile, the authors in [43] develop a virtual resource prediction strategy that can predict potential resource requirements of VNFs and control resource variability over already allocated VNFs. However, none of these studies considers the dynamic user's requirements of VNF and the deployment of VNF-FG. As a result, the problem of deploying VNF-FG has become much more complicated. Additionally, the trade-off between resource consumption and operational overhead should be balanced. To specifically address this problem, RVNF-FGD allocates VNF-FGs with the dynamic heterogamous demand of VNFs in terms of server node resources and network bandwidth.

Several research efforts target VNF-FG design and placement. For instance, the study in [44] considers VNF-FG design and placement with the aim of reducing resource consumption. However, it does not take into account the latency requirements of VNF-FGs. Additionally, the study in [25] proposes a heuristic algorithm to design VNF-FG. Then, an exact algorithm for VNF-FG embedding is provided that considers parameters such as latency, data rate capacity, and computational resources. Nevertheless, there is a lack of cooperation between VNF-FG design and placement. In fact, it is not practical for medium to large-scale scenarios because exact algorithms may result in excessive computational time for large networks. Consequently, a meta-heuristic approach for allocating NFV resources is provided in [45]. It provides online VNF-FG mapping and scheduling to reduce the flow execution time.

The deployment of VNF-FGs becomes a crucial and even more difficult problem for cloud providers when the user outsources his network functions to the cloud [46]. This issue has attracted more attention from academic and industrial communities. The studies in [47], [48] consider the VNF to be placed with the intention of decreasing the number of common active physical devices for VM placement. Moreover, the studies in [23], [49] ignore the relationship among VNFs in the VNF-FG. Meanwhile, the study in [50] discusses the VNF-FG placement problem taking into consideration the latencies across geographically scattered clouds in addition to the VNF response time. However, it does not discuss the overlapping among the VNF-FGs. Additionally, it does not make decisions on the allocation of CPUs and disregards the diverse demands of the different VNF-FGs. The study in [16] presents an approach to orchestrate and manage VNF-FG over distributed cloud infrastructures. Moreover, the study in [51] proposes a programmable management framework that considers the latencies of the VNF-FGs. It uses SDN to track the Service Level Agreement (SLA) and enhances the QoS of the VNF-FG in terms of availability and performance of service chains for each request. The studies in [52], [53] propose the VNF-FG embedding and routing algorithm in geographically distributed cloud environments. They model network topology as a multi-layer graph based on VNF-FG constraints. Meanwhile, the study in [54] provides an online strategy for VNF-FG placement in cloud datacenters. It creates active-active replicas for VNF-FGs. Additionally, it studies the effect of choosing the topology on the acceptance ratio.

Other than the study in [25], this paper provides a heuristic approach that can be applied in broader cloud network architectures. In contrast to the studies in [50], [23], [49], RVNF-FGD takes into account the ordered sequence among the VNFs in the VNF-FGs and the overlapping among them. In contrast to the study in [54], RVNF-FGD deploys the backup VNF on standby mode and activates it when a failure occurs. As a result, further reduction in the overall energy consumption is accomplished since the energy consumption of the standby server node is negligible. In other words, the server node that hosts active VNFs is on the active state while the server node that hosts standby VNFs is on the standby state.

Another significant paradigm of VNF placement is network traffic-aware placement. Several research efforts [24], [42] focus on optimizing network traffic cost while ignoring the consolidation policy. The study in [55] applies an exact strategy for VNF-FGE and routing over a network-enabled cloud to reduce physical network bandwidth consumption. Meanwhile, the study in [56] analyzes the problem of VNF-FG routing and migration to reduce energy consumption and reconfiguration costs. However, it does not consider the bandwidth consumption of VNF-FGs. The study in [29] discusses how to design VNF graphs to adapt to network topology. Additionally, some research efforts have proposed solutions to the efficient resource orchestration for VNF-FG requests in single-domain networks [57]. Meanwhile, a few research efforts have recently discussed the embedding of VNF-FG into multiple domains such as the study in [58]. It models VNF-FG allocation as a factor graph and each domain manages a portion of this graph that includes its networks. The study in [59] addresses VNF-FG placement that considers the constraints of cloud computing and physical network resources. It focuses on host demands and tries to reduce communication cost without considering the bandwidth requirements of VNF-FGs. Moreover, the study in [60] provides a heuristic approach to solve VNF-FG placement problem in WLANs. It tries to balance the total network load by using shortest path algorithms. In contrast, the approach adopted in this paper, RVNF-FGD, tries to solve VNF-FG placement problem. It addresses the huge bandwidth consumption detected during VNF recovery phase that has been neglected in previous studies [56], [59]. Additionally, it focuses on utilizing other physical cloud resources such as CPU, RAM, and storage to provide an integrated approach to deploy VNF-FGs.

VNF-FG failures can be caused by node/VM failures. A node can fail due to hardware errors (CPU, RAM, storage, power supply, NICs …etc.) Additionally, there are VM failures that can be classified as software failures. The study in [61] evaluates network failure events in datacenters. It considers node failure as a common type of failure due to the maintenance process. Meanwhile, the study in [62] discusses the characteristics of node failure and proves that they can often be related to hard disk events. Additionally, main node failures and VM failures are discussed in [63]. The continuity of a network service relies directly on the high availability of the hardware and software. As a result, the VNF-FG recomposition and reallocation should have no effect on the physical infrastructure to ensure a stable and reliable service. This is achieved by RVNF-FGD that guarantees high availability against node/VM failures in cloud computing environments.

There are few research efforts that investigate VNF placement from the point of view of availability in the NFV. The studies in [64], [65] focus on achieving certain level of availability for virtual datacenters. A virtual datacenter is a virtual network extension that provides on-demand computing, storage, and networking as applications. The study in [66] proposes an automated resilient VNF placement in the cloud using OpenStack. It is used to implement the service orchestrator technology. The study in [67] deploys VNG-FGs with sufficient availability in the NFVI. It focuses only on hardware failures. Current VNF redundancy methods are widely adopted to enhance the performance of VNF-FGs in cloud environments. However, several techniques [68]–[70] ignore the enormous problem of network resource utilization that could be faced when VNF service chain fails to recover. On the other hand, RVNF-FGD outperforms these techniques by considering the utilization of network resources when VNF-FGs fail and recover.

Network energy bills account for more than 10% of the running cost at Telecommunication Service Providers (TSPs) in some countries and 40-50% in other countries [71]. The reduced energy consumption is one of the NFV strong points of sale [72]. NFV tries not only to manage energy consumption but also to satisfy environmental standards. The introduction of NFV in cloud environments dramatically decreases energy consumption [73]. Cloud-based NFV provides energy efficiency, therefore it continues to attract research interests [74], [75]. Moreover, most energy-efficient VNF-FG deployment solutions ignore the effect of network topology on reducing energy consumption. The study in [76] proposes a heuristic solution for VNF-FG placement to reduce energy consumption. However, it does not guarantee the ordering of the VNFs in the VNF-FG. Meanwhile, the study in [56] proposes a consolidation algorithm for VNF placement, migration, and VNF-FG routing under changing workloads to reduce energy consumption. However, all physical routes must be included in the algorithm that may limit its scalability. The study in [77] proposes a resource allocation algorithm for VNF-FGs in SDN-based networks to reduce energy consumption and network reconfiguration. Additionally, the study in [78] proposes a VNF-FG deployment algorithm with the aim of minimizing energy consumption. It adopts a decomposition approach to decompose the problem into two smaller problems to achieve a quick and scalable algorithm. Different from these studies, RVNF-FGD not only optimizes the energy consumption of network resources, but also reduces the number of active server nodes. This is achieved by optimizing the activation times for the backup VNFs, which leads to a reduction in the overall energy consumption.

NFV goals can be met without the need for SDN mechanisms [3] as the common techniques used in many cloud datacenters. Nonetheless, the separation between the control plane and the data forwarding plane carried out in SDN contributes to simple, quick, and dynamic network management. It provides an efficient and flexible approach of inter-networking and chaining of VNFs. Hence, it enhances performance, configures network connectivity and bandwidth, and provides security and policy control [4], [6], [79]. When applied to NFV, SDN can help resolving the problem of complex resource management and providing intelligent service orchestration [80]. There are currently several research efforts that integrate SDN and NFV to complement each other. For instance, the study in [81] presents VNF placement framework to exploit SDN and cloud computing capabilities. NFV orchestrator controls both SDN controller and cloud controller to select the optimal location for the allocation of VNFs. The study in [82] proposes a VNF placement approach for VNF-FGs. It achieves load balancing over the core links while minimizing the energy consumption and the VNF-FG placement cost in software-defined cloud computing environments. Meanwhile, the study in [83] provides a solution for VNF placement and routing to tackle NFV resource allocation problem in SDN networks. Additionally, the study in [84] analyzes the impact of traffic steering mechanisms on the deployment efficiency of VNG-FGs using SDN-NFV cloud-based approach. Meanwhile, this paper mainly relies on SDN for redirecting traffic to backup VNFs when a failure occurs.

It has been observed that most of these mentioned research efforts are subject to certain limitations. Therefore, unlike these approaches, RVNF-FGD tries to achieve high availability in addition to addressing the massive bandwidth consumption that occurs during VNF recovery phase, which is ignored in previous studies. Hence, it can reduce link utilization and bandwidth consumption, particularly across cloud core layer. RVNF-FGD optimizes the energy consumption of both network and computing resources. This is achieved by adopting timers in the backup VNFs. Moreover, RVNF-FGD can reduce communication cost of VNF-FG. It considers the resilience of the VNF-FG instead of the resilience of a single VNF. Additionally, it takes into account both hardware and software failures. Moreover, the factors related to energy-efficient hardware are considered such as partially turning off specific hardware via VNF-FGs placement algorithm. Finally, RVNF-FGD is able to satisfy the latency restrictions imposed by network services. This is achieved by allocating and reallocating VNF-FGs in a reasonable amount of time.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, the problem of VNF-FG deployment to achieve high availability and scalability is modeled and mathematically formulated. In NFV, the performance of network services depends directly on the availability and reliability of both hardware and software. Moreover, the requirements of network services should guarantee service continuity and resilience to failure. Resilience to failure can be imposed by implementing an on-demand mechanism to reconfigure the VNF-FGs after failure. Therefore, improving the availability of NFV can ensure the stability of network services.

Network service requests consist of ordered sets of network functions connected together, which are modeled as VNF-FGs. SDN controller can periodically monitor network status, updates its details, and configures network devices. Moreover, it provides the ability to route the traffic through VNFs in a predefined order to satisfy VNF-FG requirements. When a failure occurs, the traffic is reconfigured to be routed to the backup VNFs that are also in a predefined order based on the decision made by the SDN controller. In NFV, VNF-FG may fail due to software or hardware failure. As a result, the entire VNF-FG is broken and, hence, the operation of the entire network service request may be delayed or interrupted. Node failures may lead to hardware failures, while VM instances can cause software failures.

The proposed approach in this paper focuses on both hardware and software failures. Furthermore, the most significant element of this work is to ensure the continuity of service in the event of failure. Hence, the availability should be considered when designing the deployment approach. This implies deploying backup VNFs that are retained on standby mode and activated when a node or VM failure occurs. The primary and backup VNFs are allocated on separate server nodes within the same subnet. Hence, they are connected using the lower network layer (the edge layer). This configuration avoids the hardware problems that lead to VNF failures. Otherwise, the network service will no longer be usable when a server node fails. If a primary VNF fails and there is no redundancy, then it will be migrated to a new VNF and the data will be migrated to the new VNF. This process is time-consuming and requires massive network resources. On the other hand, when using RVNF-FGD, failure of one of the primary VNFs leads to activation of the backup VNF and the traffic will be reassigned to the backup VNF. This is the optimal VNF-FG deployment configuration that would save the network bandwidth and reduce convergence time. Thus, it leads to speeding up the recovery phase. However, it may not be possible to accomplish this task because the subnet may not have adequate computing resources on the server nodes, or the required number of server nodes does not exist. Accordingly, it may become necessary to place the primary and backup VNFs on separate server nodes using different network layers (i.e., different subnets). The problem now becomes more complicated since the placement of VNF-FG may result in a varying consumption of link utilization and network bandwidth. Consequently, the performance of the network functions may be negatively affected. In order to address this problem, RVNF-FGD seeks to identify a near-optimal VNF-FG deployment while guaranteeing high availability. It considers different network architectures for cloud environments. Additionally, the objectives of RVNF-FGD are to reduce link utilization and minimize bandwidth consumption across cloud core layer. Hence, it can reduce VNF-FG communication cost and save significant delay during VNF recovery phase.

A. SYSTEM MODEL

The physical cloud network is modeled as a graph consisting of vertices and edges. Vertices represent physical nodes (switches and server nodes) while edges depict physical links among server nodes and switches or among switches.

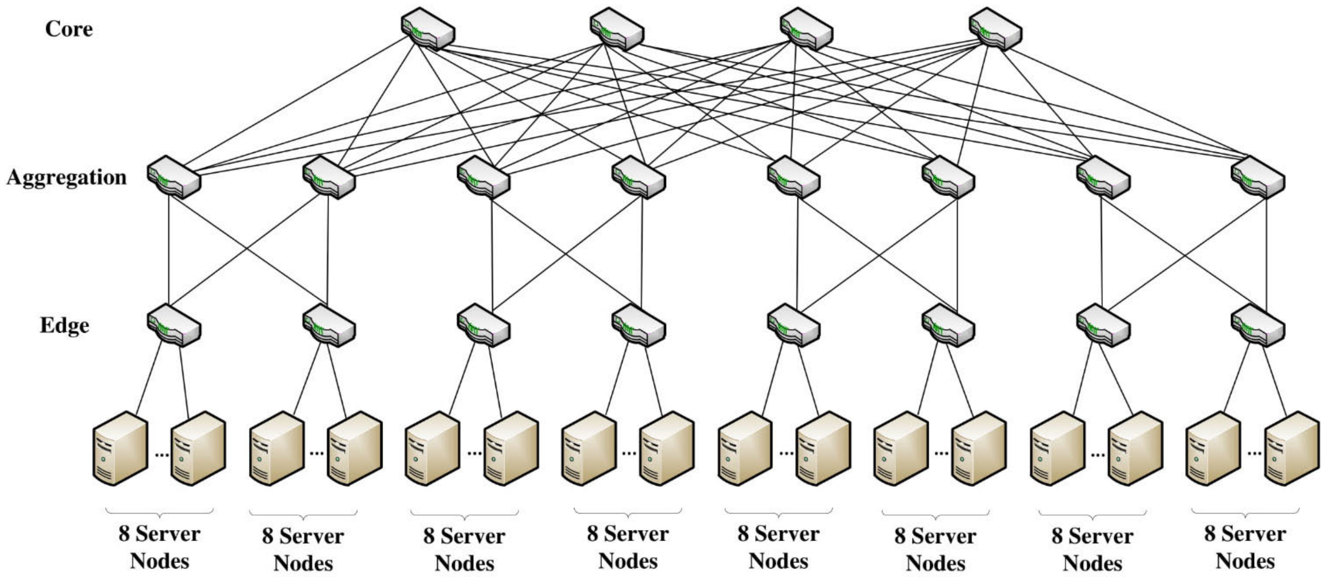


FIGURE 1. Fat-tree architecture.

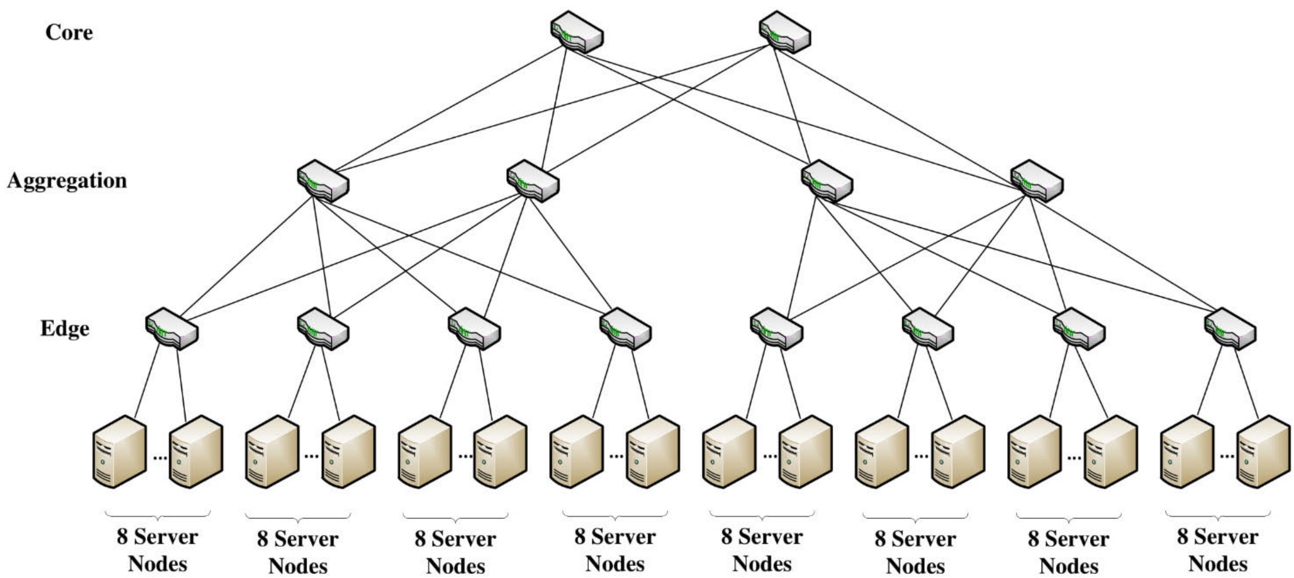


FIGURE 2. 3-Tier tree architecture.

Each link has a bandwidth capacity while each server node can host VMs that, in turn, can host VNFs. Each server node has specific capabilities of CPU, RAM, and storage. The proposed approach is responsible for determining the correct location of VNFs on the physical server nodes in the physical cloud network and chaining them via a physical route. Moreover, it assumes that each VNF-FG handles the traffic for the tenant who requests a particular service. The various cloud network architectures examined in this paper are as follows:

1) *Fat-Tree*, shown in Fig. 1, it is the most suitable architecture for cloud datacenter network. The upper network layers (core and aggregation layers) in this architecture transmit data more than the lower network layers (edge and aggregation layers). Hence, it might generate core bottlenecks [85].

2) *3-Tier Tree*, shown in Fig. 2, it consists of three tiers of switches, where core tier is the root of the tree, aggregation tier is in the middle, and edge tier represents the switches connecting the server nodes that act as the tree leaves [86].

3) *2-Tier Tree*, shown in Fig. 3, it is similar to 3-Tier Tree, but there is no aggregation tier. Therefore, there are no direct connections among switches at the same tier or between non-adjacent tiers [86].

4) *BCube*, shown in Fig. 4, it is a recursively defined architecture, where server nodes with multiple network ports are connected to multiple tier switches [87].

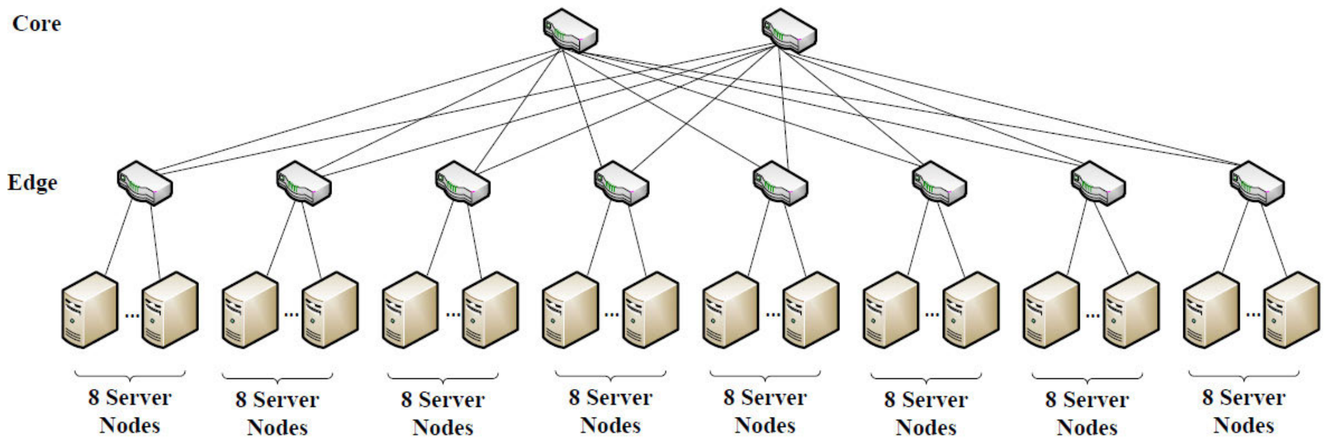


FIGURE 3. 2-Tier tree architecture.

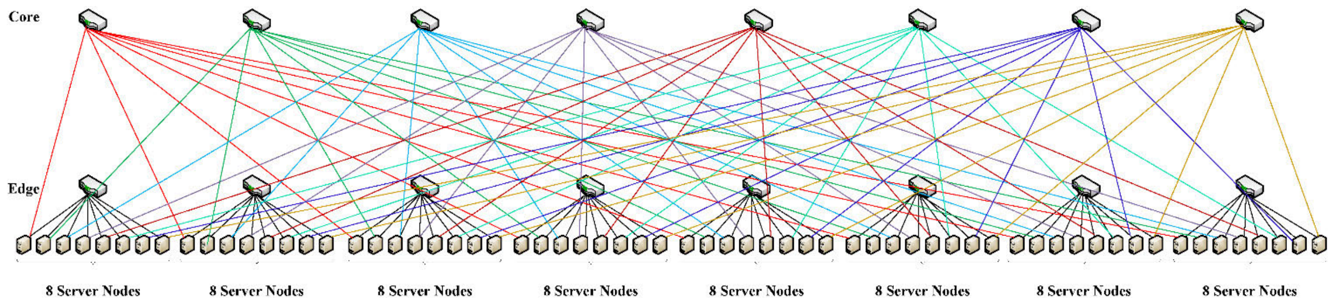


FIGURE 4. BCube architecture.

B. PROBLEM FORMULATION

It is assumed in this paper that the network service requests are submitted to the cloud as VNF-FGs. RVNF-FGD tries to find a near-optimal placement of VNF-FGs while guaranteeing high availability against node/VM failures on the physical cloud infrastructure. It tries to reduce link utilization and bandwidth consumption across cloud core layer. The optimal placement of VNF-FGs can be achieved by allocating the primary and backup VNFs within the same subnet, which means connecting them using the lower network layers. The optimal deployment of VNF-FGs can be formulated as a multi-objective optimization problem with a set of constraints described mathematically in the following subsections.

1) RESOURCE CONSTRAINTS

The problem of VNF placement can be defined as the allocation of virtual resources on the candidate physical resources. The entire network is embedded, and the physical resources are spent only if all virtual resources can be allocated. The allocations of the virtual node and the virtual link are the two phases of VNF placement. Furthermore, each VNF is characterized by the amount of computing, storage, and bandwidth resources. Therefore, it must be assigned to a physical node that satisfies its requirements. Allocating virtual networks to the physical network should assume several goals such as reducing link utilization and bandwidth consumption across cloud core layer. Resource constraints in cloud environments can be mathematically described by (1).

$$\forall\ L_p : \sum_{i=1}^{f} B_a \leq B_t \tag{1}$$

where L_p, f, B_a, and B_t represent the physical link, the number of VNF-FGs, the bandwidth of the virtual link required to be allocated on the physical link, and the total available bandwidth on the physical link, respectively.

Multiple virtual links can be allocated on the same physical link. Equation (1) ensures that the bandwidth of the virtual link required to be allocated on a physical link must not exceed the total available bandwidth on the physical link. This implies that the bandwidth of each physical link should not be over-used. Meanwhile, equation (2) ensures that the total amount of physical resources allocated to the VNFs in VNF-FGs should not exceed the total amount of physical resources that can be provided by the physical node.

$$\sum\nolimits_{F \in f} (vnf_{N_{p_F}} \times R_F^{vnf}) \leq R_{N_p}, \quad \forall\ N_p \in CN \tag{2}$$

where F is the VNF, $vnf_{N_{p_F}}$ is the number of the VNFs created on the physical node N_p to provide the virtual network function F. R_F^{vnf} represents the required amount of physical resources for each VNF in the VNF-FG. R_{N_p} represents the total amount of physical resources that can be provided by the physical node. CN is the cloud network that provides connectivity among cloud-based services.

Equation (3) ensures that each virtual link can be allocated to only one physical route.

$$\forall\ L_v \in E_v : \sum_{R_p \in E_p} Y_{R_p}^{L_v} = 1 \tag{3}$$

where L_v denotes the virtual link, E_v is the set of virtual links in the VNF-FGs. R_p is the physical route. E_p denotes the set of all possible physical routes where the virtual link can be allocated. $Y_{R_p}^{L_v}$ is a binary variable, where $Y_{R_p}^{L_v} = 1$ if the virtual link is mapped to physical route R_p, otherwise, $Y_{R_p}^{L_v} = 0$. Meanwhile, equation (4) ensures that each VNF can be allocated to only one physical node.

$$\sum\nolimits_{N_p \in CN} Y_{N_p}^{N_v} = 1 \quad \forall\, N_v \in v_i,\ \ i = 1, 2, 3, 4, \ldots, f \tag{4}$$

where N_v is the virtual node that refers to the VNF in the VNF-FG v_i. $Y_{N_p}^{N_v}$ is a binary variable, where $Y_{N_p}^{N_v} = 1$ if the VNF N_v is allocated to physical node N_p, otherwise $Y_{N_p}^{N_v} = 0$.

2) AVAILABILITY

Deploying VNF-FGs while guaranteeing high availability against node/VM failures can be formulated as an optimization problem as follows:

$$mimimize\quad w_1.N_{v_i} + w_2.T_{v_i} + w_3.E_{v_i},\quad i = 1, 2, 3, 4, \ldots, f \tag{5}$$

$$where\quad w_1 + w_2 + w_3 = 1 \tag{6}$$

$$N_{v_i} = \sum_{l_p} L_p \cdot B_c \tag{7}$$

$$T_{v_i} = \sum_{l_p} L_p \cdot t \tag{8}$$

where w_1, w_2, and w_3 are weighting parameters representing the relative importance of each objective. N_{v_i} denotes the total link utilization (bandwidth consumption) for VNF-FG v_i. B_c is the link utilization in each network layer to deploy the VNF-FG requested by a certain tenant. T_{v_i} is the convergence time required for allocating and/or reallocating the VNFs of the VNF-FG. t denotes the convergence time for allocating and/or reallocating the VNFs of the VNF-FG at each network layer. E_{v_i} denotes the overall energy consumption to deploy the VNF-FG.

When a failure occurs, the convergence time for allocating and/or reallocating the VNFs of the VNF-FG is increased due to the delay occurred until the RVNF-FGD takes place. Equation (9) ensures that there are x VNFs of VNF-FG v_i. Meanwhile, equation (10) ensures that there are m backup VNFs of VNF-FG v_i allocated on different server nodes to avoid hardware failures. Additionally, equation (11) ensures that there are n backup VNFs of VNF-FG v_i allocated on the same server node to avoid software failures.

$$x = \sum N_{vip},\quad i = 1, 2, 3, 4, \ldots, f \tag{9}$$

$$m = \sum N_{vib},\quad i = 1, 2, 3, 4, \ldots, f \tag{10}$$

$$n = \sum N_{vib},\quad i = 1, 2, 3, 4, \ldots, f \tag{11}$$

Each x primary VNFs for each VNF-FG should be mapped to m backup VNFs on different server nodes when node failure occurs. The backup VNFs guarantee that the network service will not be down when a failure occurs. If the VMs fail, the x primary VNFs should be mapped to n backup VNFs on the same node. However, if this is not feasible, x primary VNFs of VNF-FG will be partially mapped to n backup VNFs on the same server node. The rest of VNFs on the same VNF-FG will be mapped to m backup VNFs on separate server nodes.

Meanwhile, VNF-FGs should be deployed in a way that reduces the overall energy consumption. It should meet both VNF-FG requirements and maintains a high-level of availability. Thus, in normal circumstances, the primary VNFs are on active state while the backup VNFs are on standby state. As a result, the server node hosting active VNFs is on active state while the server node hosting standby VNFs is on standby state. However, the server node might host both active and standby VNFs, therefore it will be partially on standby state. The energy consumption of the server node is negligible if all VNFs on the server node are on the standby state. However, if any VNF is on the active state, then the energy consumption is a fraction of the server node energy consumption. This energy consumption is determined by the actual measurements. On the other hand, when the server node operates on the active state with its full capabilities, the energy consumption is at its maximum rate.

To achieve more reduction in the overall energy consumption when a failure occurs, a timer is added to the backup VNF. It will be triggered only during predefined time intervals. SDN controller is responsible for initializing backup VNFs when it detects a need to handle the traffic. It adjusts the timers of the backup VNFs to run during these time periods and then returns them back to the standby state. Consequently, this process reduces the energy consumed, especially by the server nodes that contribute a significant portion of overall consumed energy. Equation (12) represents the overall energy consumption to deploy the VNF-FG.

$$E_{v_i} = \sum \left(Q_j^{t-1} + Q_j^t \right) .\delta .E_j,\quad j = 1, 2, 3, 4, \ldots, N_{pt} \tag{12}$$

where E_j denotes the energy consumption of server node j. Q_j^{t-1} specifies the number of server nodes operating under normal circumstances for all VNFs in the VNF-FG. Q_j^t specifies the number of server nodes operating after a failure occurs for all VNFs in the VNF-FG. δ is a parameter that defines the portion of the energy consumption of the server node when it is partially on standby state. $\delta = 1$ if the server node is on active state and $\delta = 0$ if the server node is on standby state. Meanwhile, N_{pt} represents the total number of server nodes.

If at least one server node or VM fails, then the corresponding backup VNF will be activated instead of its failed primary VNF. Hence, the server node hosting active VNFs will also be in active state. The server node might host both active and standby VNFs, therefore it will be partially in standby state. The backup VNFs are activated when there is a need to handle traffic and then returned back to the standby state. Only the server node hosting backup VNFs is activated during these periods.

Primary and backup VNFs for the same VNF-FG are allocated to separate server nodes. Mainly to achieve high availability against hardware failures that eventually lead to VNF failures. Equation (13) ensures that the primary and backup VNFs of the same VNF-FG are not allocated to the same physical node.

$$Y_{N_p}^{N_{vip}} + Y_{N_p}^{N_{vib}} \leq 1, N_{vip}, N_{vib} \in v_i, \quad i = 1, 2, 3, 4, \ldots, f \quad (13)$$

where $Y_{N_p}^{N_{vip}}$ and $Y_{N_p}^{N_{vib}}$ are binary variables, $Y_{N_p}^{N_{vip}} = 1$ if primary VNF N_{vip} of VNF-FG v_i is allocated to physical node N_p, otherwise $Y_{N_p}^{N_{vip}} = 0$. Similarly, $Y_{N_p}^{N_{vib}} = 1$ if backup VNF N_{vib} of VNF-FG v_i is allocated to physical node N_p, otherwise $Y_{N_p}^{N_{vib}} = 0$.

3) VNF-FG COMMUNICATION COST

VNF-FG deployment problem is formulated to reduce the communication cost of VNF-FG. The reduction is mainly attributed to the reduction in link utilization and bandwidth consumption across cloud core layer. This is achieved during VNF recovery phase. This objective can be mathematically formulated by (14) and (15).

$$minimize \left(\alpha \cdot C_{VNF}^{N} + \beta \cdot C_M + (1 - \alpha - \beta) \cdot C_B\right) \quad (14)$$

$$where \ \alpha \ll 1, \beta \ll 1 \quad (15)$$

where C_{VNF}^{N} is the cost incurred due to the energy consumed by network devices used to inter-communicate VNFs in each VNF-FG. C_M is the cost of migrating failed VNFs, and C_B represents the total cost of allocating the bandwidth for the virtual links embedded in the physical links. The cost of the allocated bandwidth is the dominant cost component in communication cost. Therefore, the parameters α and β are introduced to weigh the relative importance of bandwidth cost versus network devices and migration costs. Equation (16) represents the cost of network devices used to inter-communicate VNFs in each VNF-FG.

$$C_{VNF}^{N} = \sum_{F \in f} \rho \cdot E_R^F \cdot \sum_{L_v \in E_v} Y_{R_p}^{L_v}, \quad i = 1, 2, 3, 4, \ldots, f \quad (16)$$

where E_R^F denotes the network energy consumed for inter-communicating VNFs in each VNF-FG. ρ denotes the cost per unit power of network energy.

Migration cost C_M, which is represented by (17), is the difference between the total cost of bandwidth in the original location C_{BO} and the total cost of bandwidth in the migrated location C_{BF} after the failure occurred. Meanwhile, C_{BO} and C_{BF} for the original and migrated location are computed by (18).

$$C_M = C_{BO} - C_{BF} \quad (17)$$

$$C_B = \sum_{R_p} \delta \cdot |R_p| \cdot \left(B_a \cdot \sum_{r} Y_r^{E_p}\right) \quad (18)$$

where δ denotes the cost per unit bandwidth on the link, $|R_p|$ represents the number of physical routes, and r is the number of virtual links in the VNF-FG. $Y_r^{E_p}$ is a binary variable, where $Y_r^{E_p} = 1$ if the r^{th} virtual link in the VNF-FG is allocated to physical route $R_p \in E_p$ and $Y_r^{E_p} = 0$ otherwise.

Meanwhile, Equation (19) represents the profit gained by cloud service provider after deploying the VNF-FGs.

$$O = \sum_{f} o_f \cdot Y_f \quad (19)$$

where O is the profit gained by the cloud service provider, o_f denotes the profit that the cloud service provider earns after deploying f^{th} VNF-FG. Y_f is a binary variable, where $Y_f = 1$ if the VNF-FG of the tenant is served and $Y_f = 0$ otherwise.

The final optimization goal can be described as follows:

$$maximize \left(O - \left(C_{VNF}^{N} + C_B + C_M\right)\right) \quad (20)$$

Assume that the number of subnets with available server nodes is S. Hence, the number of candidate server nodes in each subnet can be expressed using the vector Z described in (21).

$$Z = [z_1, z_2, z_3, \ldots, z_S] \quad (21)$$

The solution to this problem is determined using following vectors

$$X = [x_1, x_2, x_3, \ldots, x_S] \quad (22)$$

$$M = [m_1, m_2, m_3, \ldots, m_S] \quad (23)$$

$$N = [n_1, n_2, n_3, \ldots, n_S] \quad (24)$$

Equation (22) represents the number of primary VNFs in each subnet. Meanwhile, equation (23) represents the number of backup VNFs when hardware failure occurs. Similarly, equation (24) represents the number of backup VNFs when software failure occurs. The solution to this ILP problem can be found by solving the following equations:

$$x_1 + x_2 + x_3 + \ldots + x_S = x \quad (25)$$

$$m_1 + m_2 + m_3 + \ldots + m_S = m \quad (26)$$

$$n_1 + n_2 + n_3 + \ldots + n_S = n \quad (27)$$

$$x_k + m_k \leq Z_k, \quad k = 1, 2, 3, \ldots, S \quad (28)$$

$$x_k + n_k \leq Z_k, \quad k = 1, 2, 3, \ldots, S \quad (29)$$

$$x_k \geq 0, \quad k = 1, 2, 3, \ldots, S \quad (30)$$

$$m_k \geq 0, \quad k = 1, 2, 3, \ldots, S \quad (31)$$

$$n_k \geq 0, \quad k = 1, 2, 3, \ldots, S \quad (32)$$

Equations (26), (28), and (31) are used when hardware failure occurs, while equations (27), (29), and (32) are used when software failure occurs. Solving these equations repeatedly for each subnet would be extremely time-consuming. This is attributed to the large number of subnets and server nodes in the cloud network in addition to the constraints that should be considered. Therefore, the problem is tackled using a heuristic algorithm as will be presented in the next section.

IV. HEURISTIC SOLUTION

The deployment of VNF-FG is known to be NP-hard [9]. Therefore, computing the optimal solution is going to be computationally intensive. The problem gets even worse due to the huge number of VNFs and subnets in addition to the limitations of computing resources in the cloud network environments. Hence, a heuristic approach is proposed in this section to tackle this problem. The proposed heuristic approach attempts to find a near-optimal solution, as well as, achieving a trade-off between availability, scalability, and acceleration as the problem grows. Hence, it achieves efficient allocation of VNF-FGs that form the tenant's demands on cloud infrastructure. Therefore, VNF-FG deployment problem is solved by adopting a near-optimal time-efficient solution named Redundant VNF-FG Deployment (RVNF-FGD). RVNF-FGD can be described by the following steps:

1. Calculate and choose the candidate server nodes that have adequate computing resources. Server nodes with high residual resources have a higher priority to meet the demand of primary VNFs in the VNF-FG.
2. Define all feasible physical routes that meet the bandwidth demand of primary virtual links in VNF-FG. These physical routes are used to transfer the traffic among the candidate server nodes selected in the previous step. SDN has a major role in this step. SDN controller periodically monitors network status, updates link bandwidth, and configures the network devices.
3. Sort all candidate server nodes based on their routes. The routes that use the links of the lowest network layer have higher priority (classified as the best routes) to transfer traffic.
4. Build the backup graphs where the corresponding primary VNF-FG is located if applicable. The backup graph consists of backup VNFs in the same subnet and on the same server nodes. This is achieved by using the same procedure in steps 1 to 3 for software failure with different values of m and n according to equations (10) and (11). Otherwise, backup graphs should be allocated in the same subnet on a separate server node where the corresponding primary VNF-FG is located. This is achieved by using the same procedure in steps 1 to 3 for hardware failures with different values of m according to equation (10).
5. Allocate the primary x VNFs of the VNF-FG at the top of candidate server nodes that are found in the set of server nodes computed in step 3. Additionally, allocate the primary virtual links of the VNF-FG on the best physical route defined in step 3 that connects these server nodes.
6. Allocate the backup m and/or n VNFs on candidate server nodes that are found in the backup graph created in step 4. This allocation depends on the type of failure against which the tenant's needs to ensure service availability. Additionally, allocate the backup virtual links of VNF-FG on the best physical routes defined in step 4 that connect these server nodes.
7. Check if the VNF-FG deployment with the availability criterion is met. If met, stop the algorithm and report its success. Otherwise, calculate new backup graphs in a different cloud network within the same cloud service provider (i.e., repeat steps 1 to 6).
8. After a cn number of investigations in cloud network within the same cloud service provider, if VNF-FG deployment with the availability criterion is not met, then stop the algorithm and report its failure.

If at least one server node or VM fails, then the corresponding backup VNF is activated instead of its failed primary VNF. Consequently, the traffic of the failed primary VNF is going to be reassigned to its backup VNF and the data will be processed using the new VNF-FG. On the other hand, if the primary VNF fails due to a hardware failure, then all VNFs belonging to the primary VNF-FG and their corresponding backups will not be allowed to be on the same server. However, they can be allocated in the same subnet. Otherwise, they can be allocated in other subnets that are as close as possible to the subnet that hosts a portion of the backup VNF-FG. As a result, RVNF-FGD algorithm can reduce link utilization and bandwidth consumption across cloud core layer in the cloud network. Furthermore, the minimum number of core and aggregation hops can be used to connect the VNFs of the VNF-FG.

RVNF-FGD algorithm focuses on dynamic VNF-FG deployment and steering the traffic using SDN where availability, network architecture, and resource requirements are considered together. Algorithm 1 represents the details of RVNF-FGD algorithm.

V. COMPLEXITY ANALYSIS

In this section, the computational complexity of RVNF-FGD algorithm is analyzed. As shown in the previous section, RVNF-FGD algorithm searches for the candidate server nodes that have adequate computing resources. Then, it defines all feasible physical routes that meet the bandwidth demand of the primary virtual links in the VNF-FG. Next, it retrieves and sorts the candidate server nodes using the routes with the lowest network layer links in each subnet. Hence, the worst-case computational complexity due to searching for the candidate server nodes using the routes with the lowest network layer links in each subnet can be computed as follows:

$$C_{worst} = O\left(N_p + L_p \cdot \left(V + N_{vnf} \cdot N_p\right) + S + z_S\right) \quad (33)$$

where N_{vnf} is the total number of VNFs in all VNF-FGs, V is the total number of switches, S is the number of subnets, and z_S is the number of candidate server nodes per subnet.

Algorithm 1 RVNF-FGD Algorithm

1 **Input**: x number of $N_{vip} \in v_i$, m/n number of $N_{vib} \in v_i$, R_F^{vnf}, B_a, *failure type*;
2 **for** each N_p **in** target *CN* **do**
3 **if** $R_{N_p} \geq R_F^{vnf}$ **then**
4 calculate the candidate server nodes;
5 **else**
6 go to **Final_Step**;
7 **end if**
8 **end for**
9 **for** each L_p **in** *CN* **do**
10 **if** $B_t \geq B_a$ **then**
11 compute the admissible routes;
12 **else**
13 go to **Final_Step**;
14 **end if**
15 **end for**
16 **for** each subnet **in** *CN* **do**
17 add candidate server nodes to the subnet;
18 **end for**
19 retrieve and sort candidate server nodes connected using the routes with the lowest network layer links in each subnet;
20 **for** $i = 1$ *to* f **do**
21 **for** $j = 1$ *to* x **in** v_i **do**
22 **if** $R_F^{vnf} \leq R_{N_p}$ **in** server node **then**
23 allocate N_{vip} on the candidate server node;
24 **else if** $R_F^{vnf} \leq R_{N_p}$ **in** subnet **then**
25 allocate N_{vip} on a different candidate server node within the same subnet;
26 **else**
27 allocate N_{vip} on a different subnet;
28 **end if**
29 go to **Allocation_Step**;
30 **end for**
31 **if** *failure type* = *software failure* **then**
32 **for** $i = 1$ *to* n **in** v_i **do**
33 **if** $R_F^{vnf} \leq R_{N_p}$ **in** server node $\supseteq N_{vip}$ **then**
34 allocate N_{vib} on the same server node where N_{vip} is already allocated;
35 **else**
36 go to **step 41**;
37 **end if**
38 **end for**
39 **else**
40 **for** $i = 1$ *to* m **in** v_i **do**
41 **if** $R_F^{vnf} \leq R_{N_p}$ **in** server node $\not\supseteq N_{vip}$ **then**
42 allocate N_{vib} on a different server node where N_{vip} is not allocated within the same subnet;
43 **else**
44 allocate N_{vib} on a different subnet;
45 **end if**
46 **end for**
47 **end if**

Algorithm 1 *(Continued.)* RVNF-FGD Algorithm

48 **Allocation_Step:**
49 allocate all $L_v \in v_i$ over $R_p \in E_p$;
50 **end for**
51 go to **Output**;
52 **Final_Step:**
53 **while** $CN \leq cn$ **do**
54 go to cloud network within the same cloud service provider;
55 repeat steps from 2 to 53;
56 $CN \leftarrow CN + 1$;
57 **end while**
58 **Output:** The solution of VNF-FG deployment problem that guarantees high availability against node/VM failures;

If each VNF is assigned to a separate server, then the computational complexity can be computed as follows:

$$C_{worst} = O\left(N_p + L_p \cdot V + L_p \cdot N_{vnf} \cdot N_p + S \cdot x\right) \quad (34)$$
$$C_{worst} = O\left(N_p.\left(1 + L_p.N_{vnf}\right) + L_p.V + S.x\right) \quad (35)$$
$$C_{worst} = O\left(N_p.L_p.N_{vnf} + L_p.V + S.x\right) \quad (36)$$
$$C_{worst} = O\left(L_p.\left(N_p.N_{vnf} + V\right) + S.x\right) \quad (37)$$
$$C_{worst} = O\left(L_p\left(N_p.N_{vnf} + V\right)\right) \quad (38)$$

The number of VNFs in all VNF-FGs is the dominant factor compared to the number of VNFs of the VNF-FG. Additionally, the number of subnets is relatively small. Hence, their contributions to the computational complexity are generally assumed negligible compared to that of VNFs.

RVNF-FGD algorithm allocates the primary x VNFs of the VNF-FG to candidate server nodes. Then, it allocates the primary virtual links of the VNF-FG to the admissible physical routes that connect these server nodes. After that, it allocates the backup m and/or n VNFs to candidate server nodes. Additionally, it allocates the backup virtual links of the VNF-FG to the admissible physical routes that connect these server nodes. Hence, the computational complexity of allocating primary VNFs of the VNF-FG can be computed as follows:

$$C_{primary} = O\left(z_t + f \cdot x \cdot z_t\right) \quad (39)$$

where z_t is the number of candidate server nodes.

If each VNF is assigned to a separate server node, then this computational complexity can be computed as follows:

$$C_{primary} = O\left(x + f \cdot x \cdot x\right) \quad (40)$$
$$C_{primary} = O\left(x + f \cdot x^2\right) \quad (41)$$
$$C_{primary} = \left(f \cdot x^2\right) \quad (42)$$

The computational complexity of allocating backup VNFs of VNF-FG can be computed as follows:

$$C_{backup} = O\left(S \cdot z_S\right) \quad (43)$$

Meanwhile, the computational complexity is computed for a single cloud network. Consequently, RVNF-FGD computational complexity when the cloud service provider consists of *cn* cloud networks can be computed as follows:

$$C = O\left(\left(L_p.\left(N_p.N_{vnf} + V\right) + C_{primary} + C_{backup}\right).cn\right) \quad (44)$$

$C_{primary}$ and C_{backup} are very small compared to the first term in (44). Hence, they can be neglected, and the overall computational complexity of RVNF-FGD algorithm can be described as follows:

$$C = O\left(L_p.\left(N_p.N_{vnf} + V\right).cn\right) \quad (45)$$

Therefore, the computational complexity of RVNF-FGD algorithm is linear, which has a direct influence on its scalability.

VI. EXPERIMENTAL RESULTS

In this section, the conducted simulation study is detailed. First, the simulation setup is described, then performance evaluation results of RVNF-FGD algorithm are presented. A benchmark algorithm is designed to compare the proposed algorithm against it. The benchmark algorithm deploys VNF-FGs without redundancy based on the strategy outlined in [88].

A. SIMULATION SETUP

The experimental framework is implemented on CloudSim-SDN-NFV simulator [89]. The performance of RVNF-FGD algorithm is evaluated on different cloud network architectures as shown in Figs. 1, 2, 3, and 4. To make the experiments comparable for all cloud network architectures, each physical cloud network architecture consists of 64 server nodes, with 8 server nodes in each subnet. Each server node hosts at most five VNFs. The bandwidth capacity of each physical link in core and aggregation layer (upper layers) is set to 10 Gbps. Meanwhile, the bandwidth capacity of each physical link in the edge layer (lower layer) is set to 1 Gbps. Network requests are randomly generated with heterogeneous requirements to form VNF-FGs. Each VNF-FG consists of two end points (source and destination VNFs) with an intermediate VNF. Hence, the traffic enters the source VNF, then passes the intermediate VNFs and leaves the VNF-FG at the destination VNF.

B. PERFORMANCE EVALUATION RESULTS

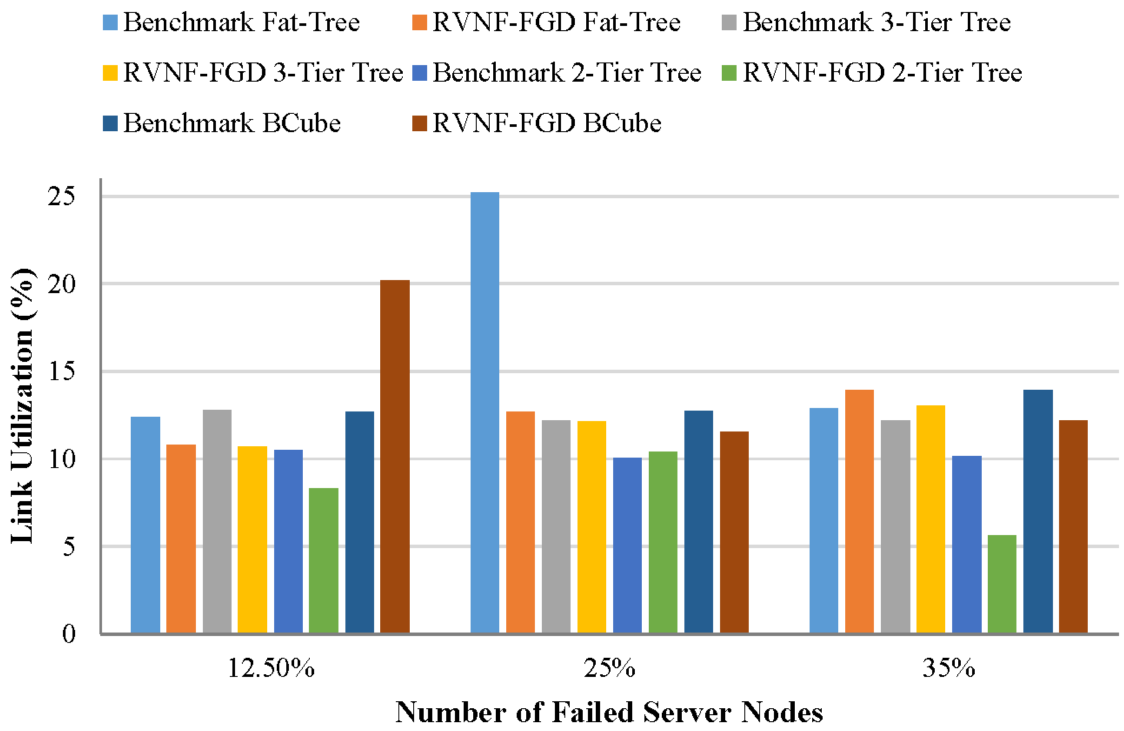
In order to evaluate the performance of RVNF-FGD algorithm, a VNF placement benchmark algorithm is updated on the basis of the strategy in [88]. This benchmark algorithm provides network-aware allocation to resolve the congestion problem at the core links in the datacenter network. It considers the cost of using the links inside the datacenter and the cost of inter-datacenter links. Hence, the benchmark algorithm is able to reduce link utilization of the upper-layer links. Consequently, it has a direct positive impact on reducing the potential congestion that might occur. This benchmark algorithm handles VNF deployment and migration without redundancy. The experiments include 37 network service requests that consist of 111 primary and their 111 backup VNFs. They are implemented on several cloud network architectures. Additionally, 3496 workloads are created to reflect traffic that passes through the VNF-FGs. Two different failure events are triggered, node failure and VM failure. Node failure can lead to hardware failures while VM failure can cause software failures. The percentage of failure is increased at each iteration that represents the number of failed nodes/VMs. In each iteration, the network service requests are generated corresponding to the first iteration. In this scenario, SDN controller is responsible for routing the traffic through the VNFs in a predefined order. When a failure occurs, the SDN controller reconfigures the traffic to be routed to the backup VNFs that are predefined too.

The key performance metrics used to evaluate RVNF-FGD algorithm against the benchmark algorithm are link utilization and bandwidth consumption, VNF-FG communication cost, overall energy consumption, and convergence time.

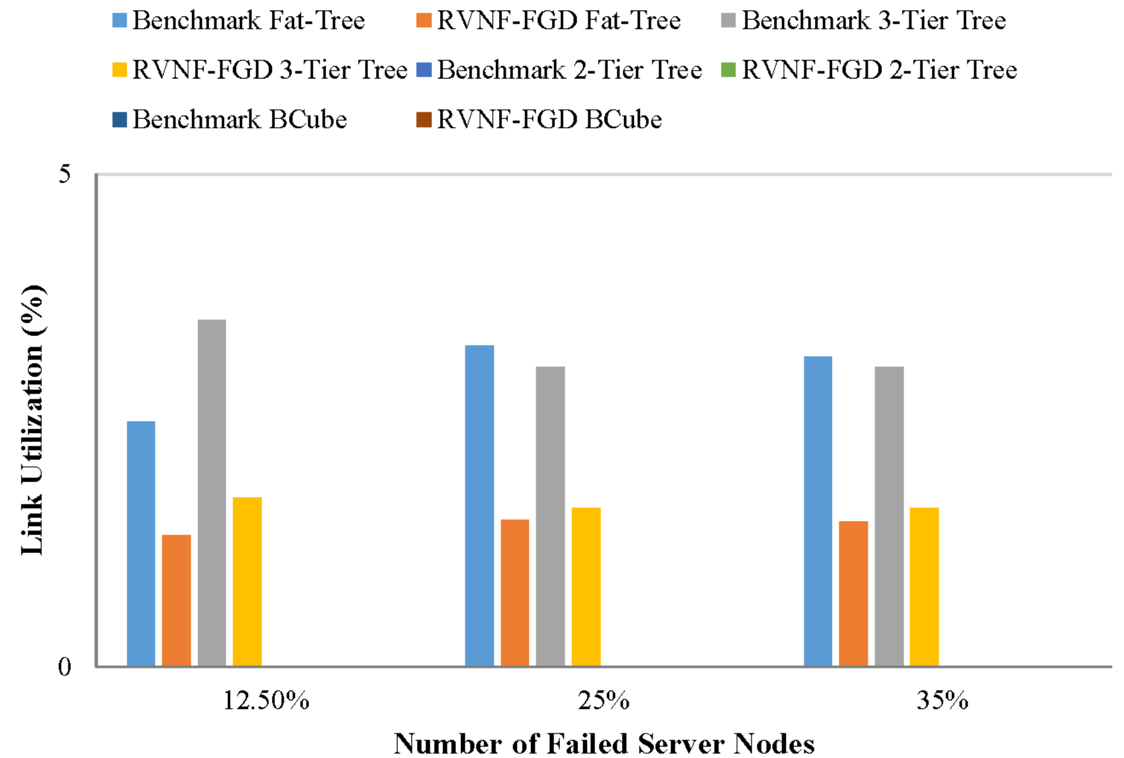
1) LINK UTILIZATION AND BANDWIDTH CONSUMPTION

RVNF-FGD algorithm attempts to find a near-optimal solution, as well as to achieve a trade-off between availability and scalability. Meanwhile, it should maintain link utilization and, hence, bandwidth consumption as low as possible. RVNF-FGD algorithm attempts to deploy primary and backup VNFs in the same subnet that results in transferring the traffic using edge layer links. Therefore, RVNF-FGD algorithm consumes less core and aggregation layer bandwidth than the benchmark algorithm. RVNF-FGD algorithm considers the ordering of VNFs and the cloud network architectures when deploying the VNF-FGs. Hence, it leads to a reduction in utilization of the core and aggregation layer links.

To measure the efficiency of reducing link utilization and bandwidth consumption in RVNF-FGD algorithm, the number of failed server nodes and VMs is increased across different cloud network architectures. Fig. 5 shows the average utilization of all network layer links in RVNF-FGD algorithm and the corresponding values for the benchmark algorithm. The number of failed server nodes increases from 12.5% to 35% of the total number of server nodes, while the number of failed VMs increases to 100% of the total number of VMs. Fat-Tree and 2-Tier Tree architectures, noted as the best cloud network architectures, guarantee the best performance in terms of link utilization compared to the other architectures. As shown in Fig. 5c, RVNF-FGD algorithm consumes 2.3% of link bandwidth while the benchmark algorithm consumes 4.8% of link bandwidth in the core layer of the Fat-Tree architecture at 35% of hardware failure. Therefore, RVNF-FGD algorithm exhibits low utilization of core and aggregation layer links resulted in consuming less bandwidth of the upper layers during the recovery phase.



(a)

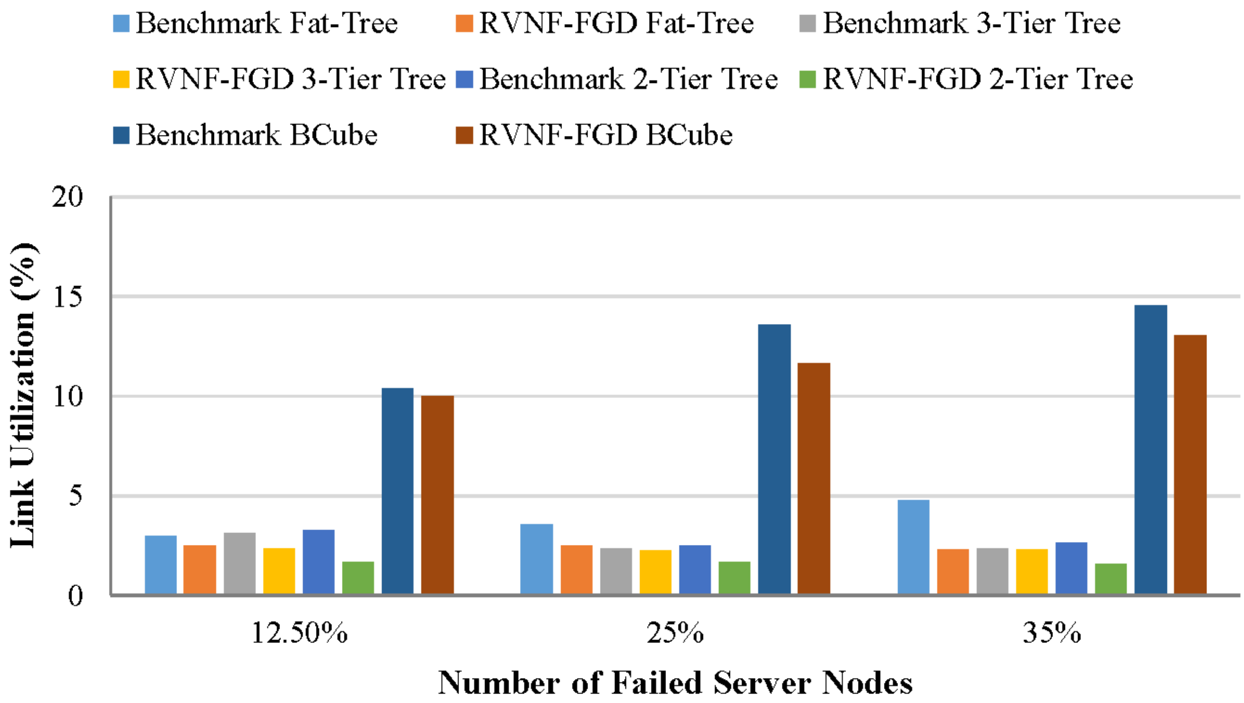


(b)

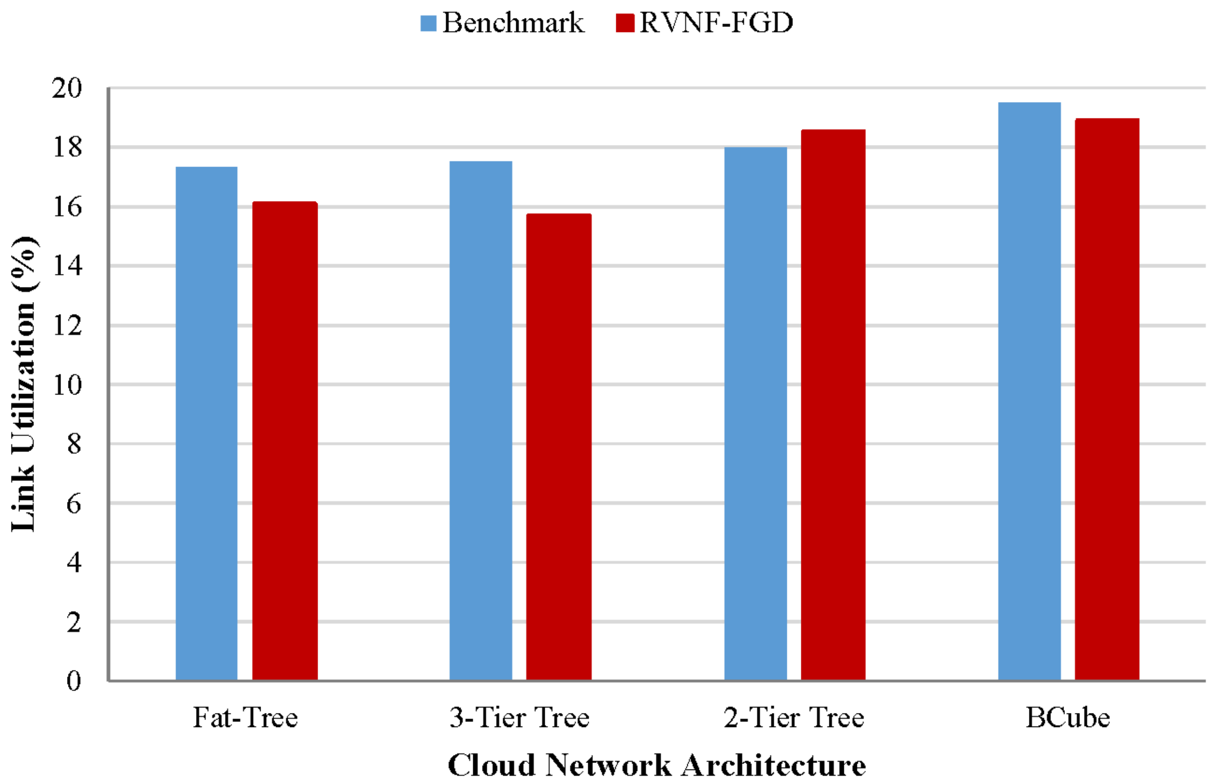
FIGURE 5. Link utilization for RVNF-FGD algorithm and the corresponding values for the benchmark algorithm at: a., d. Edge layer, b., e. Aggregation layer, and c., f. Core layer, when hardware failure that scales from 12.5% to 35% occurs for the cases (a), (b), and (c), and when 100% software failure occurs for the cases (d), (e), and (f).

2) VNF-FG COMMUNICATION COST

This metric reflects the cost of allocating bandwidth for virtual links embedded on the physical links, the cost of energy consumed for network devices, and the cost of migrating failed VNFs. Also, this metric reflects the importance of each cost component determined by the parameters α and β in



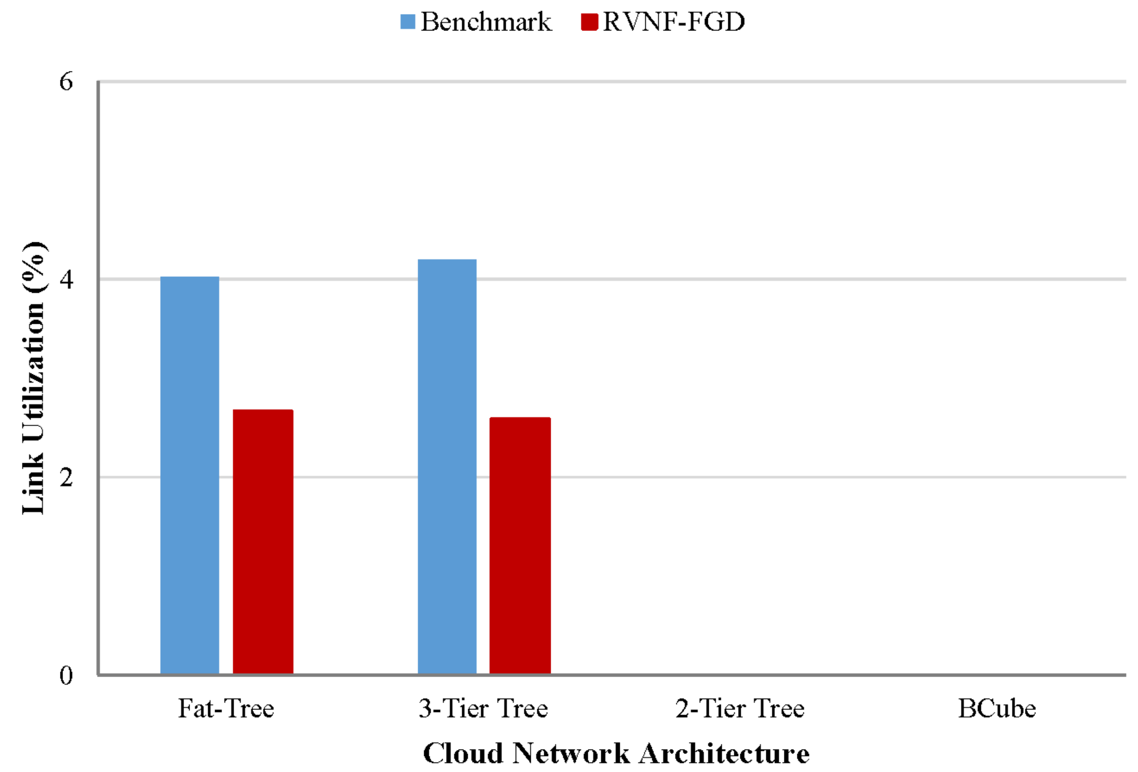
(c)



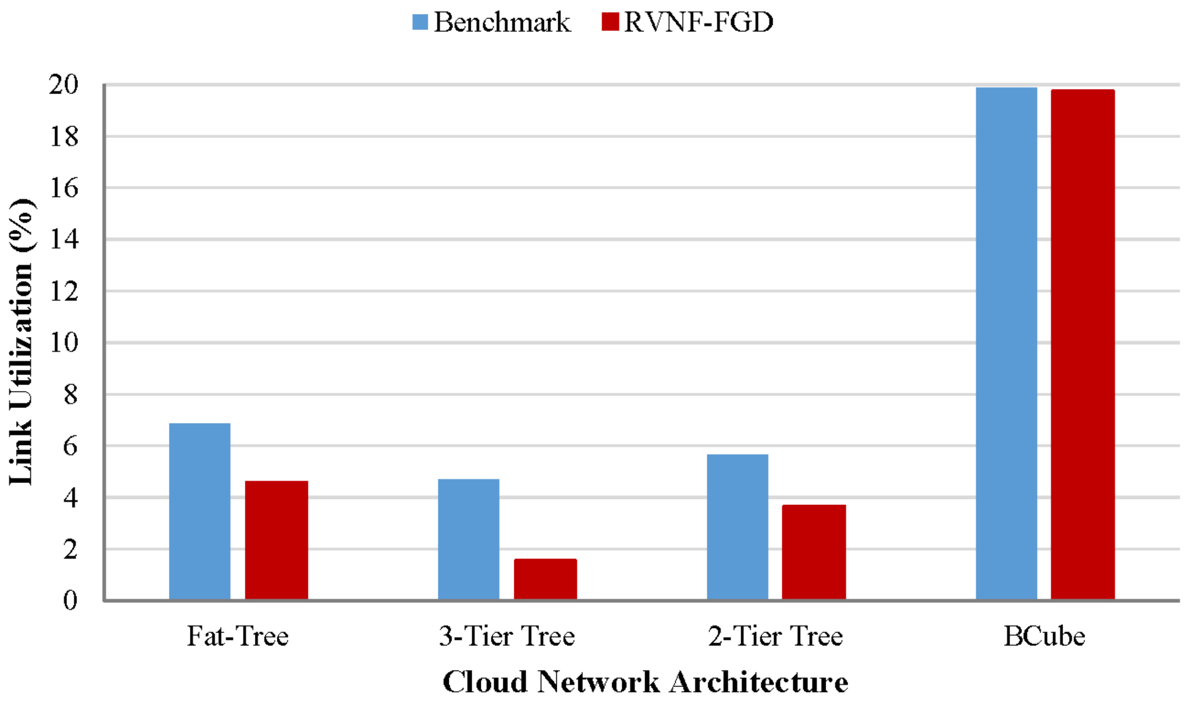
(d)

FIGURE 5. *(Continued.)* **Link utilization for RVNF-FGD algorithm and the corresponding values for the benchmark algorithm at: a., d. Edge layer, b., e. Aggregation layer, and c., f. Core layer, when hardware failure that scales from 12.5% to 35% occurs for the cases (a), (b), and (c), and when 100% software failure occurs for the cases (d), (e), and (f).**

(14) and (15). One of the main advantages of NFV is the significant reduction in overall running cost. Hence, one of the goals of RVNF-FGD algorithm is to reduce the communication cost required for chaining VNFs in each VNF-FG.



(e)



(f)

FIGURE 5. *(Continued.)* **Link utilization for RVNF-FGD algorithm and the corresponding values for the benchmark algorithm at: a., d. Edge layer, b., e. Aggregation layer, and c., f. Core layer, when hardware failure that scales from 12.5% to 35% occurs for the cases (a), (b), and (c), and when 100% software failure occurs for the cases (d), (e), and (f).**

In this experiment, both α and β are set to 0.1 to highlight the importance of the allocated bandwidth cost. As a result, the cost of allocated bandwidth becomes a major dominant cost relative to the other cost components in (14).

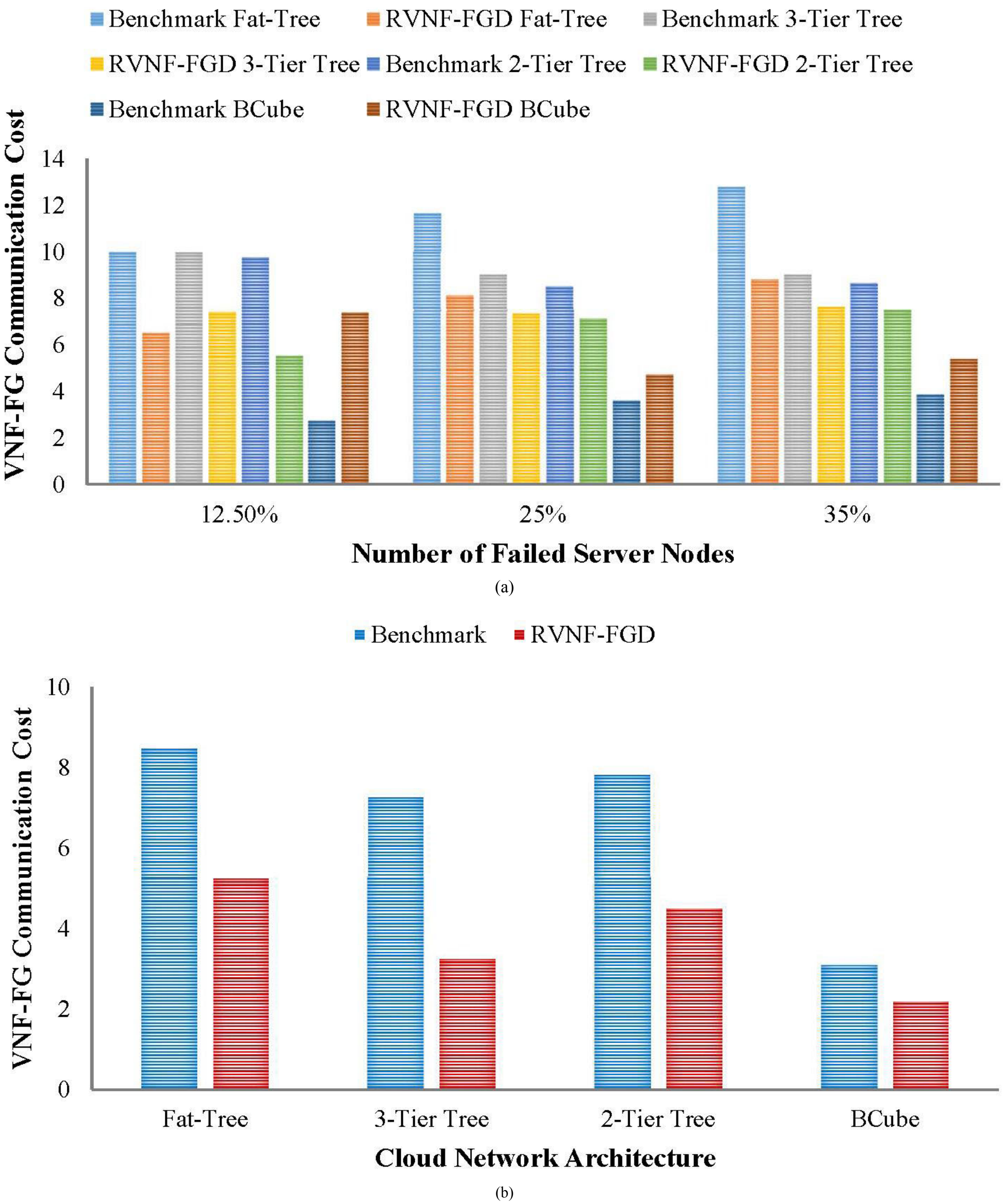


FIGURE 6. VNF-FG communication cost for RVNF-FGD algorithm and the corresponding values for the benchmark algorithm when: a. Hardware failure that scales from 12.5% to 35% occurs. b. 100% software failure occurs.

Fig. 6 shows the communication cost when using RVNF-FGD algorithm and the benchmark algorithm across various cloud network architectures. As shown in Fig. 6a, when the number of failed server nodes increases in Fat-Tree architecture, the communication cost increases. This observation is attributed to the fact that more backup VNFs must be deployed and communicated during recovery phase, which increases the migration cost. Fig. 6 also shows that

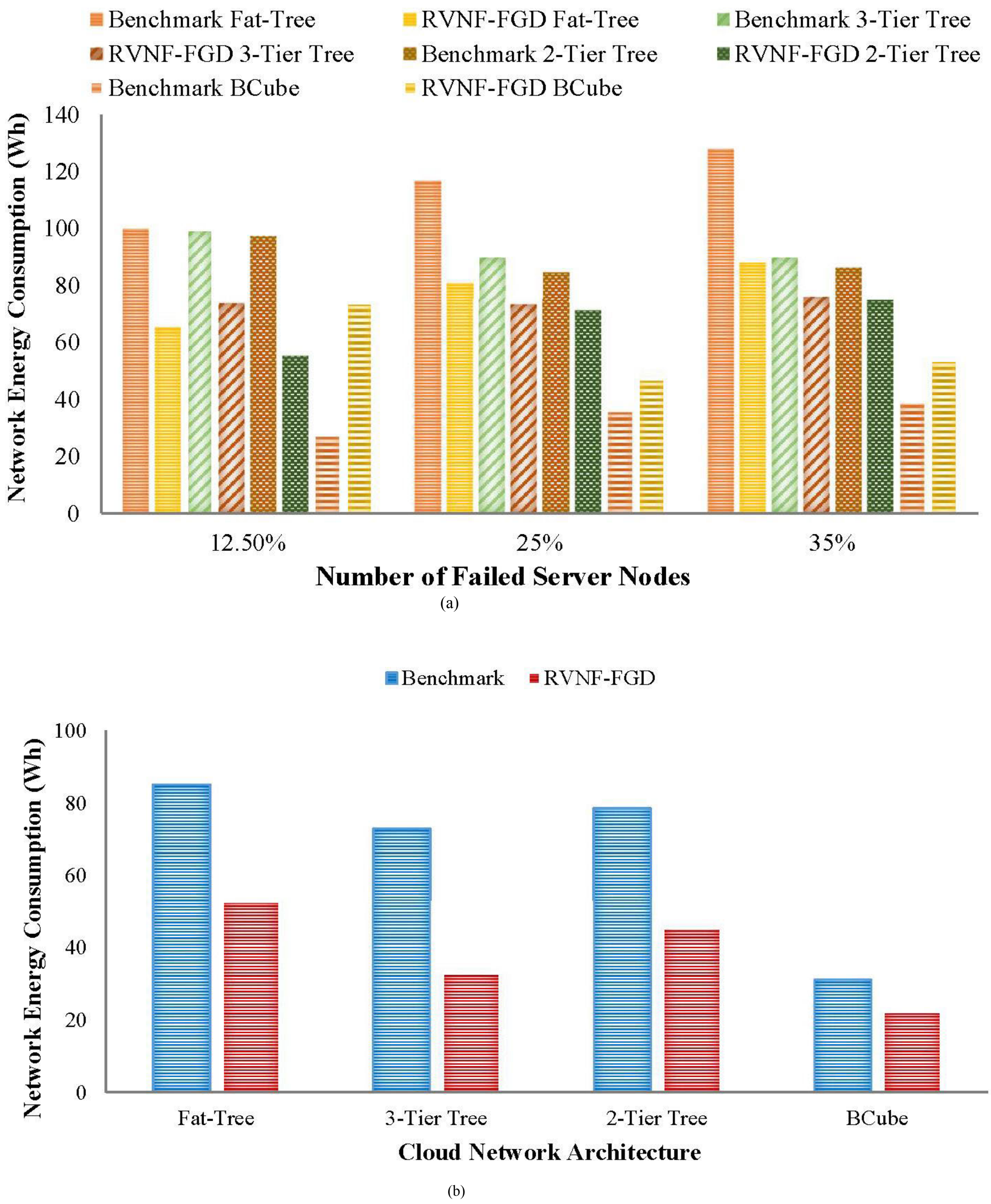
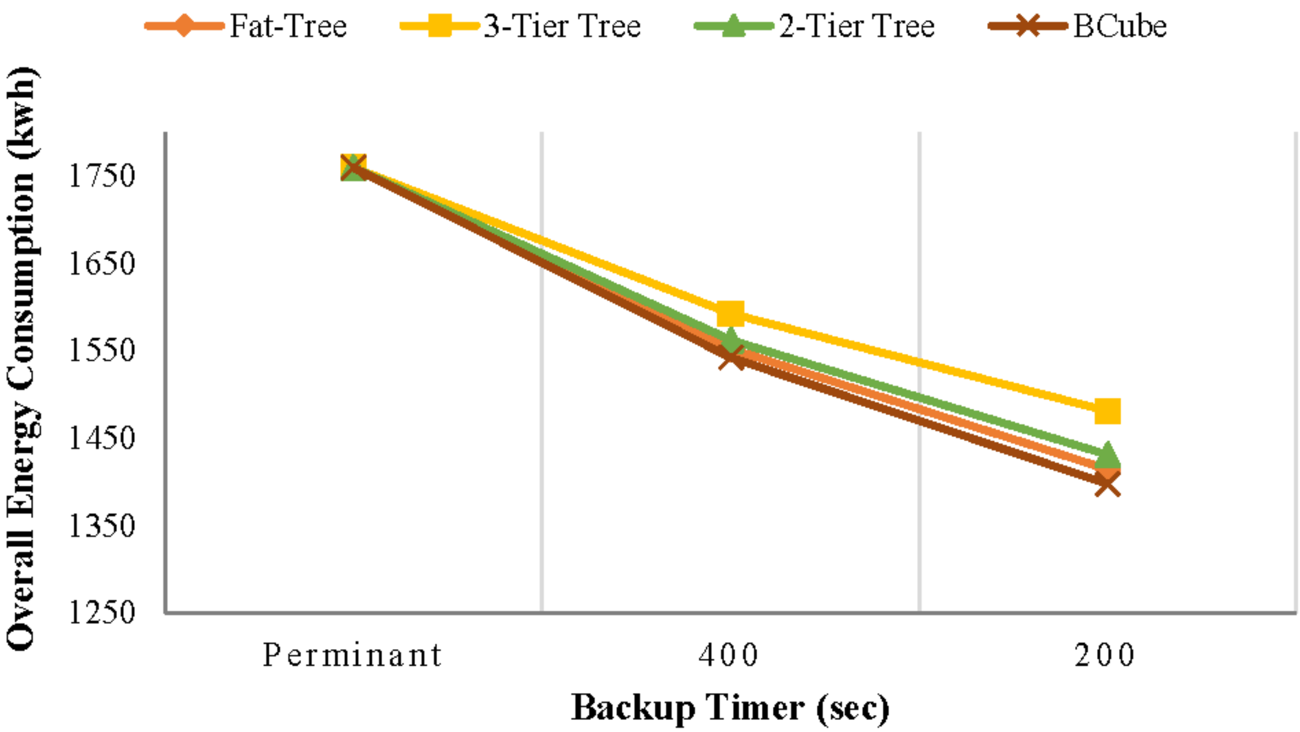
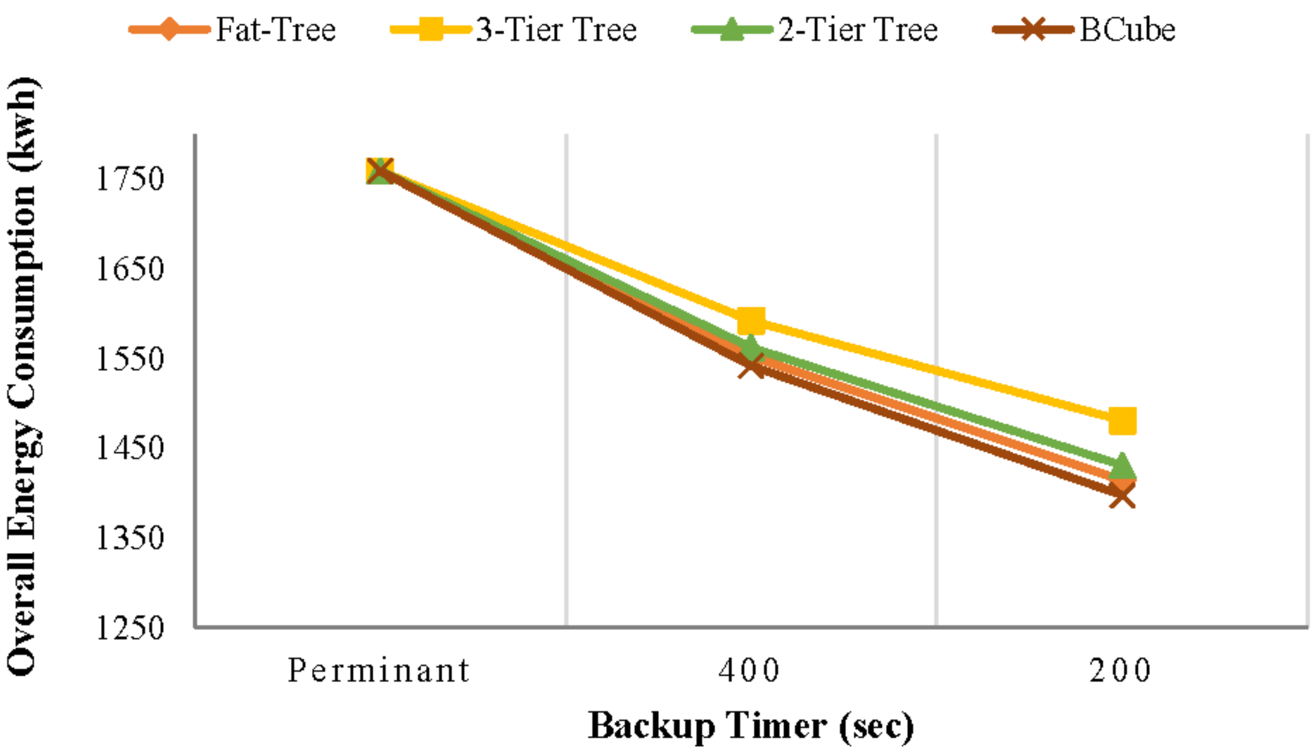


FIGURE 7. Network energy consumption for RVNF-FGD algorithm and the corresponding values for the benchmark algorithm when: a. Hardware failure that scales from 12.5% to 35% occurs. b. 100% software failure occurs.

RVNF-FGD algorithm outperforms the benchmark algorithm in terms of communication cost for all tested architectures except for BCube. This is mainly because the network energy cost of this architecture for RVNF-FGD algorithm is higher

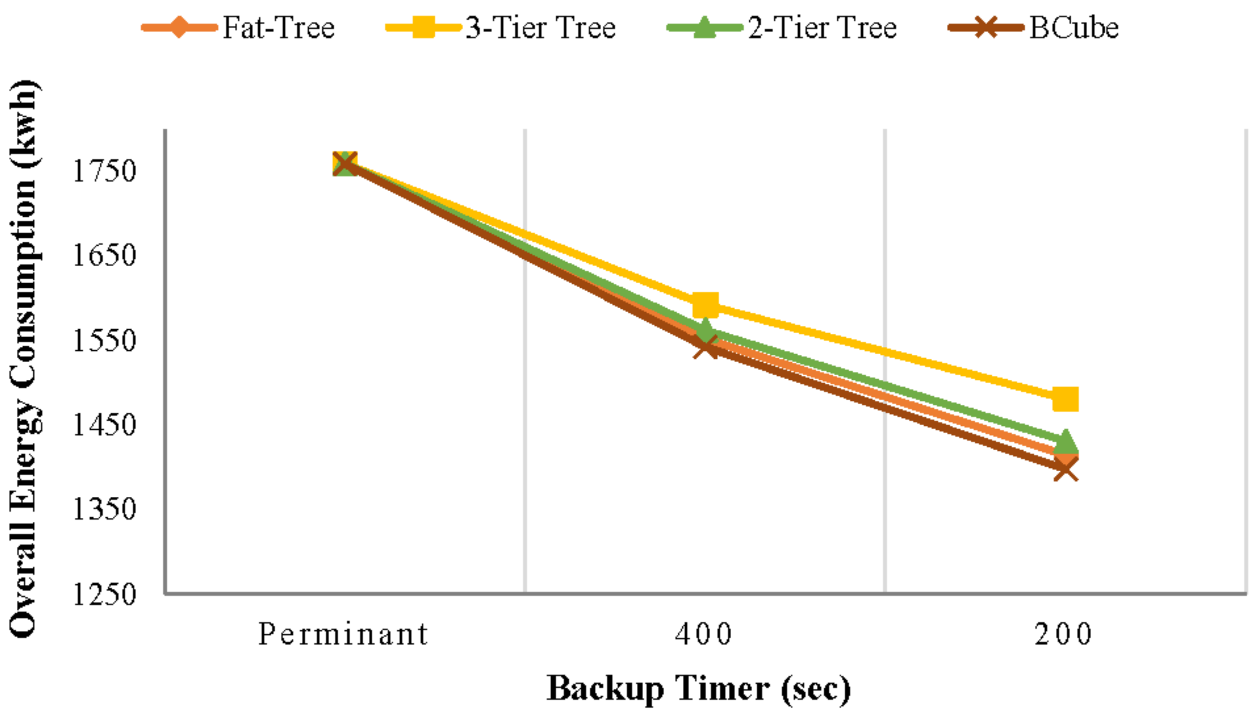


(a)

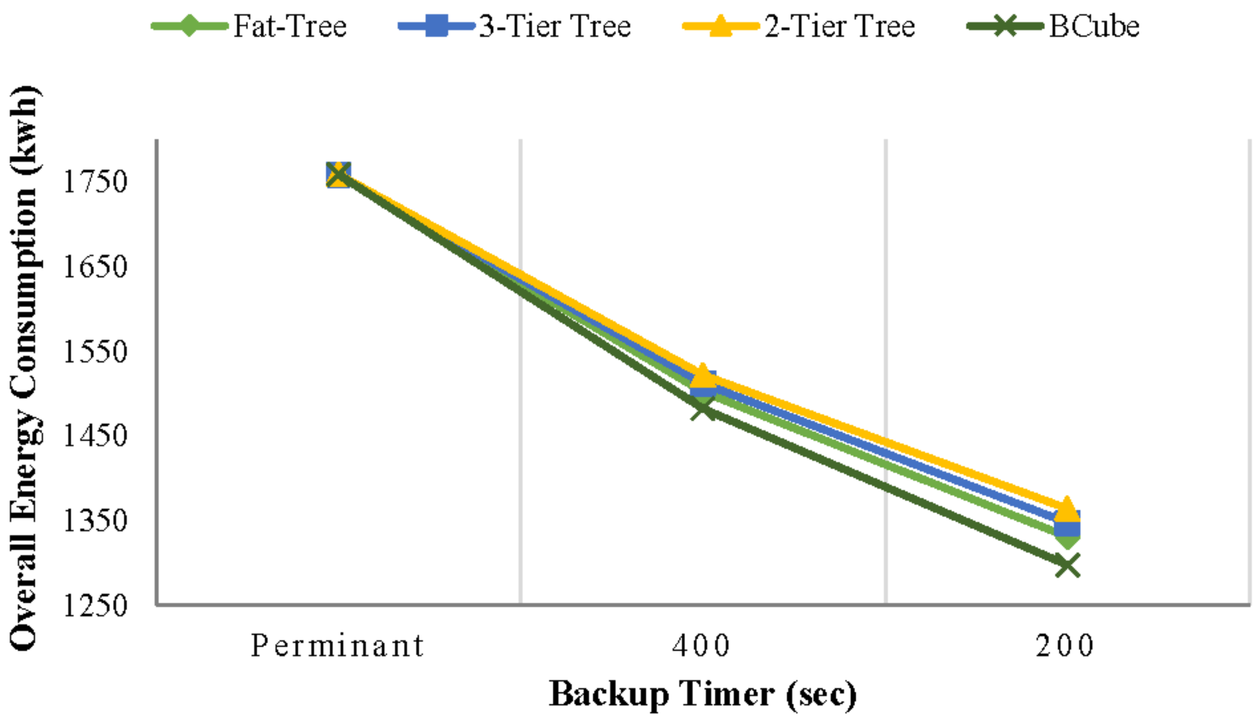


(b)

FIGURE 8. Overall energy consumption for RVNF-FGD algorithm when adding timers to backup VNFs when: a. 12.5% hardware failure occurs. b. 25% hardware failure occurs. c. 35% hardware failure occurs. d. 100% software failure occurs.



(c)



(d)

FIGURE 8. ***(Continued.)*** **Overall energy consumption for RVNF-FGD algorithm when adding timers to backup VNFs when: a. 12.5% hardware failure occurs. b. 25% hardware failure occurs. c. 35% hardware failure occurs. d. 100% software failure occurs.**

compared to the benchmark algorithm. 2-Tier Tree architecture has the smallest cost compared to the other cloud network architectures in the case of hardware failure as shown in Fig. 6a. The reason behind this observation is that the 2-Tier Tree architecture does not contain the aggregation layer that is subject to bottlenecks similar to the core layer.

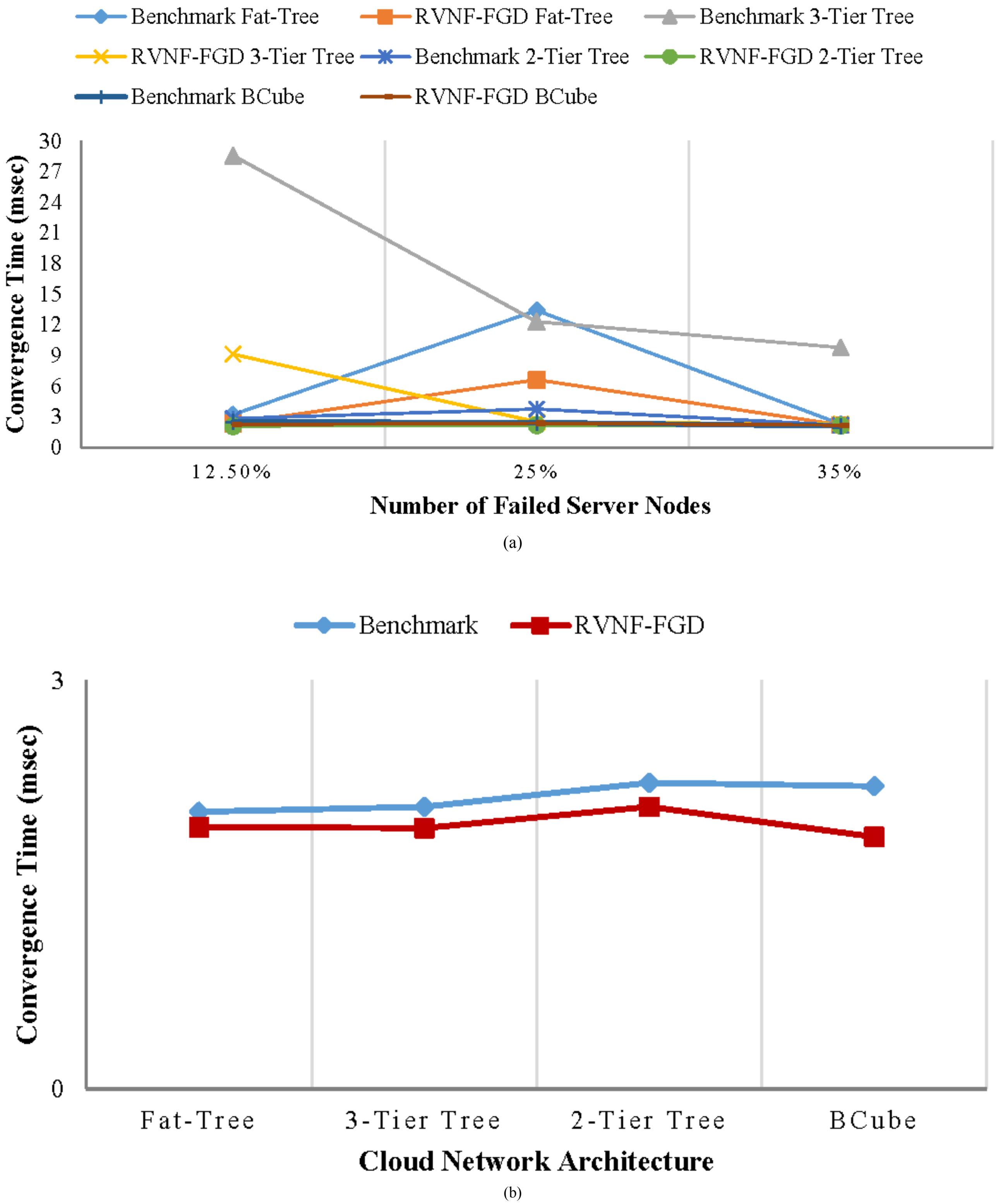


FIGURE 9. Convergence time for RVNF-FGD algorithm and the corresponding values for the benchmark algorithm when: a. Hardware failure that scales from 12.5% to 35% occurs. b. 100% software failure occurs.

3) OVERALL ENERGY CONSUMPTION

Energy consumption is partially attributed to the server nodes that host the VNFs. Additionally, network energy consumption is resulted from routing the traffic within the network. RVNF-FGD algorithm utilizes the lowest network layer links thus reducing overall energy consumption. Additionally, it reduces the number of active server nodes by reducing the activation times for the backup VNFs. This task is handled by the SDN controller. By implementing timers in the backup VNFs, RVNF-FGD algorithm demonstrates its efficiency in reducing computing energy consumption and thus overall energy consumption in the cloud network.

Fig. 7 compares the network energy consumption when using RVNF-FGD algorithm and the benchmark algorithm. As shown in this figure, RVNF-FGD algorithm achieves significant network energy saving compared to the benchmark algorithm. It saves up to 43.2% of network energy at 12.5% of hardware failure over the 2-Tier Tree architecture as shown in Fig. 7a. BCube architecture has the disadvantage that the network energy consumed when adopting RVNF-FGD algorithm is higher than the corresponding values in the benchmark algorithm. This is mainly because in BCube architecture, the server nodes are connected to both core and edge switches simultaneously. Meanwhile, in the other architectures, the server nodes are connected to edge switches only. Moreover, RVNF-FGD algorithm often consumes less computing energy and can even minimize overall energy consumption, which is mainly attributed to the backup VNF timer as shown in Fig. 8.

4) CONVERGENCE TIME

As the number of server nodes increases in cloud computing environments, the number of parameters and limitations to be addressed increases non-linearly. Therefore, current VNF placement approaches are unable to handle broader cloud network architectures in a reasonable amount of time. One of the motivations for combining NFV with cloud infrastructure is to be able to dynamically create and delete VNFs for various changes in tenant's requirements. In such scenarios, a less time-consuming approach is preferable.

This subsection provides a comprehensive analysis of the convergence time required to deploy the VNF-FGs with high availability. The algorithm is first tested when there is no failure and the number of backup VNFs is zero. Then, when failure occurs, the backup VNFs are increased to satisfy the availability requirements. Therefore, the number of failed server nodes/VMs is scaled to demonstrate the efficiency of the proposed approach.

When the number of failed server nodes/VMs is increased, RVNF-FGD algorithm has the advantage of reducing the convergence time and scales much better than the benchmark algorithm. Thus, it tries to provide near-optimal allocation and reallocation in a reasonable amount of time. Moreover, the chaining of VNFs using lower network layer links decreases data processing latency. Hence, RVNF-FGD algorithm can be able to satisfy the latency restrictions imposed by the network services.

The convergence times of RVNF-FGD algorithm for various cloud network architectures is shown in Fig. 9 with the corresponding values computed for the benchmark algorithm. The number of failed server nodes ranges from 12.5% to 35% of the total number of server nodes. Meanwhile, the number of failed VMs increases to 100% of the total number of VMs. As shown in Fig. 9a, RVNF-FGD algorithm takes 9 milliseconds to deploy VNF-FGs in the worst case for the 3-Tier Tree architecture. When software failure occurs, the convergence time of RVNF-FGD algorithm may increase across different cloud network architectures. However, it is still acceptable and less than the corresponding values for the benchmark algorithm. For example, as shown in Fig. 9b, the convergence time of RVNF-FGD algorithm for BCube architecture is 1.8 milliseconds compared to 2.2 milliseconds of the benchmark algorithm. Hence, RVNF-FGD algorithm can maintain the continuity of network services and saves more time when imposing the high availability requirements.

VII. CONCLUSION

NFV adopts virtualization technology to provide network services configured as VNF-FGs. Since network services must always be running, it is important to ensure the availability of services with the minimum amount of scare resources. This paper introduces Redundant VNF-FG Deployment approach named RVNF-FGD algorithm. Its main objective is to guarantee the availability of network services against node/VM failures in cloud computing environments. This process faces a set of conflicting goals such as reducing link utilization across cloud core layer, reducing VNF-FG communication cost, and reducing overall energy consumption. A comprehensive evaluation of RVNF-FGD algorithm is presented taking into consideration the different performance metrics and cloud network architectures. Evaluation results show that RVNF-FGD algorithm outperforms the benchmark algorithm in terms of link utilization. Moreover, RVNF-FGD algorithm consumes 2.3% of link utilization while the benchmark algorithm consumes 4.8% of link utilization at 35% of hardware failure when Fat-Tree architecture is adopted. Additionally, it outperforms the benchmark algorithm in communication cost during recovery phase. RVNF-FGD algorithm is able to achieve the minimum cost when using 2-Tier Tree architecture compared to the other cloud network architectures in the event of hardware failure. Furthermore, the results show that RVNF-FGD algorithm can achieve a trade-off between the redundancy and the overall energy consumption through implementing the backup VNF timer. RVNF-FGD algorithm is able to save up to 43.2% of network energy at 12.5% of hardware failure in 2-Tier Tree architecture. Likewise, the convergence time of RVNF-FGD algorithm is assessed by applying this approach to broader cloud network architectures. The convergence time of RVNF-FGD algorithm for BCube architecture is 1.8 milliseconds at 100% software failure compared to 2.2 milliseconds for the benchmark algorithm for the same percentage of software failure. These results indicate the viability of the proposed approach in responding quickly to hardware and software failures.

REFERENCES

[1] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[2] J. de Jesus Gil Herrera and J. F. B. Vega, "Network functions virtualization: A survey," *IEEE Latin Amer. Trans.*, vol. 14, no. 2, pp. 983–997, Feb. 2016.

[3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[4] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[5] B. Zhang, P. Zhang, Y. Zhao, Y. Wang, X. Luo, and Y. Jin, "Co-scaler: Cooperative scaling of software-defined NFV service function chain," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Palo Alto, CA, USA, Nov. 2016, pp. 33–38.

[6] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, Feb. 2013.

[7] P. Quinn and T. Nadeau, *Problem Statement for Service Function Chaining*, document RFC 7498, Internet Engineering Task Force, Apr. 2015.

[8] O. Houidi, O. Soualah, W. Louati, and D. Zeghlache, "Dynamic VNF forwarding graph extension algorithms," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 3, pp. 1389–1402, Sep. 2020.

[9] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[10] W. Miao, G. Min, Y. Wu, H. Huang, Z. Zhao, H. Wang, and C. Luo, "Stochastic performance analysis of network function virtualization in future Internet," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 613–626, Mar. 2019.

[11] Z. Luo and C. Wu, "An online algorithm for VNF service chain scaling in datacenters," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1061–1073, Jun. 2020.

[12] A. Nadjaran Toosi, J. Son, Q. Chi, and R. Buyya, "ElasticSFC: Auto-scaling techniques for elastic service function chaining in network functions virtualization-based clouds," *J. Syst. Softw.*, vol. 152, pp. 108–119, Jun. 2019.

[13] A. Khosravi, L. L. H. Andrew, and R. Buyya, "Dynamic VM placement method for minimizing energy and carbon cost in geographically distributed cloud data centers," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 2, pp. 183–196, Apr. 2017.

[14] ETSI. *Network Functions Virtualisation (NFV)*. Accessed: Jan. 2021. [Online]. Available: https://www.etsi.org/technologies/nfv

[15] ETSI. (Dec. 2014). *ETSI GS NFV-MAN 001 V1.1.1*. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf

[16] J. Soares, C. Goncalves, B. Parreira, P. Tavares, J. Carapinha, J. P. Barraca, R. L. Aguiar, and S. Sargento, "Toward a telco cloud environment for service functions," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 98–106, Feb. 2015.

[17] J. Soares, M. Dias, J. Carapinha, B. Parreira, and S. Sargento, "Cloud4NFV: A platform for virtual network functions," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Luxembourg, U.K., Oct. 2014, pp. 288–293.

[18] V. F. Garcia, L. D. C. Marcuzzo, A. Huff, L. Bondan, J. C. Nobre, A. Schaeffer-Filho, C. R. P. dos Santos, L. Z. Granville, and E. P. Duarte, "On the design of a flexible architecture for virtualized network function platforms," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[19] A. Bravalheri, A. S. Muqaddas, N. Uniyal, R. Casellas, R. Nejabati, and D. Simeonidou, "VNF chaining across multi-PoPs in OSM using transport API," in *Proc. Opt. Fiber Commun. Conf. (OFC)*, San Diego, CA, USA, 2019, p. 4.

[20] A. M. Alwakeel, A. Alnaim, and E. B. Fernández, "A pattern for NFV management and orchestration (MANO)," in *Proc. 8th Asian Conf. Pattern Lang. Programs*, Tokyo, Japan, Mar. 2019, pp. 1–12.

[21] W. Shen, M. Yoshida, T. Kawabata, K. Minato, and W. Imajuku, "VConductor: An NFV management solution for realizing end-to-end virtual network services," in *Proc. 16th Asia–Pacific Netw. Operations Manage. Symp.*, Hsinchu, Taiwan, Sep. 2014, pp. 1–6.

[22] K. Giotis, Y. Kryftis, and V. Maglaris, "Policy-based orchestration of NFV services in software-defined networks," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., Apr. 2015, pp. 1–5.

[23] M. Bouet, J. Leguay, and V. Conan, "Cost-based placement of vDPI functions in NFV infrastructures," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., Apr. 2015, pp. 490–506.

[24] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Istanbul, Turkey, Apr. 2014, pp. 2402–2407.

[25] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Luxembourg, U.K., Oct. 2014, pp. 7–13.

[26] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding LC-VNE algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3648–3661, Dec. 2016.

[27] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quart., 2013.

[28] E. Amaldi, S. Coniglio, A. M. C. A. Koster, and M. Tieves, "On the computational complexity of the virtual network embedding problem," *Electron. Notes Discrete Math.*, vol. 52, pp. 213–220, Jun. 2016.

[29] J. Cao, Y. Zhang, W. An, X. Chen, J. Sun, and Y. Han, "VNF-FG design and VNF placement for 5G mobile networks," *Sci. China Inf. Sci.*, vol. 60, no. 4, pp. 1–15, Apr. 2017.

[30] H. Moens and F. D. Turck, "VNF-P: A model for efficient placement of virtualized network functions," in *Proc. 10th Int. Conf. Netw. Service Manage. (CNSM) Workshop*, Rio de Janeiro, Brazil, Nov. 2014, pp. 418–423.

[31] B. Yi, X. Wang, K. Li, S. K. Das, and M. Huang, "A comprehensive survey of network function virtualization," *Comput. Netw.*, vol. 133, pp. 212–262, Mar. 2018.

[32] M. Rost and S. Schmid, "Service chain and virtual network embeddings: Approximations using randomized rounding," 2016, *arXiv:1604.02180*. [Online]. Available: http://arxiv.org/abs/1604.02180

[33] S. Kulkarni, M. Arumaithurai, K. K. Ramakrishnan, and X. Fu, "Neo-NSH: Towards scalable and efficient dynamic service function chaining of elastic network functions," in *Proc. 20th Conf. Innov. Clouds, Internet Netw. (ICIN)*, Paris, France, Mar. 2017, pp. 308–312.

[34] P. Roy, A. Tahsin, S. Sarker, T. Adhikary, M. A. Razzaque, and M. M. Hassand, "User mobility and quality-of-experience aware placement of virtual network functions in 5G," *Comput. Commun.*, vol. 150, pp. 367–377, Jan. 2020.

[35] M. A. Tahmasbi Nejad, S. Parsaeefard, M. A. Maddah-Ali, T. Mahmoodi, and B. H. Khalaj, "VSPACE: VNF simultaneous placement, admission control and embedding," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 542–557, Mar. 2018.

[36] N. Kiji, T. Sato, R. Shinkuma, and E. Oki, "Virtual network function placement and routing model for multicast service chaining based on merging multiple service paths," in *Proc. IEEE 20th Int. Conf. High Perform. Switching Routing (HPSR)*, Xi'an, China, May 2019, pp. 1–6.

[37] S. I. Kim and H. S. Kim, "Method for VNF placement for service function chaining optimized in the NFV environment," in *Proc. 11th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Zagreb, Croatia, Jul. 2019, pp. 721–724.

[38] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, 2016.

[39] M. A. Khoshkholghi, J. Taheri, D. Bhamare, and A. Kassler, "Optimized service chain placement using genetic algorithm," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Paris, France, Jun. 2019, pp. 472–479.

[40] N. Hyodo, T. Sato, R. Shinkuma, and E. Oki, "Virtual network function placement for service chaining by relaxing visit order and non-loop constraints," *IEEE Access*, vol. 7, pp. 165399–165410, 2019.

[41] Y. Liu, J. Pei, P. Hong, and D. Li, "Cost-efficient virtual network function placement and traffic steering," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[42] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network function placement for NFV chaining in packet/optical datacenters," *J. Lightw. Technol.*, vol. 33, no. 8, pp. 1565–1570, Apr. 15, 2015.

[43] R. Mijumbi, S. Hasija, S. Davy, A. Davy, B. Jennings, and R. Boutaba, "Topology-aware prediction of virtual network function resource requirements," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 1, pp. 106–120, Mar. 2017.

[44] Z. Ye, X. Cao, J. Wang, H. Yu, and C. Qiao, "Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization," *IEEE Netw.*, vol. 30, no. 3, pp. 81–87, May 2016.

[45] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, London, U.K., Apr. 2015, pp. 1–9.

[46] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, and S. Ratnasamy, "Making middleboxes someone else's problem: Network processing as a cloud service," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Helsinki, Finland, Aug. 2012, pp. 13–24.

[47] Y. Zhang and N. Ansari, "Heterogeneity aware dominant resource assistant heuristics for virtual machine consolidation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Atlanta, GA, USA, Dec. 2013, pp. 1297–1302.

[48] A. Beloglazov and R. Buyya, "OpenStack neat: A framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds," *Concurrency Comput., Pract. Exper.*, vol. 27, no. 5, pp. 1310–1333, Apr. 2015.

[49] V. Sekar, N. Egi, S. Ratnasamy, M. K. Reiter, and G. Shi, "Design and implementation of a consolidated middlebox architecture," in *Proc. 9th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, San Jose, CA, USA, Apr. 2012, pp. 323–336.

[50] D. Bhamare, M. Samaka, A. Erbad, R. Jain, L. Gupta, and H. A. Chan, "Optimal virtual network function placement in multi-cloud service function chaining architecture," *Comput. Commun.*, vol. 102, pp. 1–16, Apr. 2017.

[51] H. Hawilo, M. Jammal, and A. Shami, "Network function virtualization-aware orchestrator for service function chaining placement in the cloud," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 643–655, Mar. 2019.

[52] J. Pei, P. Hong, K. Xue, and D. Li, "Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 10, pp. 2179–2192, Oct. 2019.

[53] P. Cappanera, F. Paganelli, and F. Paradiso, "VNF placement for service chaining in a distributed cloud environment with multiple stakeholders," *Comput. Commun.*, vol. 133, pp. 24–40, Jan. 2019.

[54] G. Moualla, T. Turletti, and D. Saucez, "Online robust placement of service chains for large data center topologies," *IEEE Access*, vol. 7, pp. 60150–60162, 2019.

[55] A. Gupta, M. F. Habib, P. Chowdhury, M. Tornatore, and B. Mukherjee, "On service chaining using virtual network functions in network-enabled cloud systems," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Kolkata, India, Dec. 2015, pp. 1–3.

[56] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017.

[57] S. Khebbache, M. Hadji, and D. Zeghlache, "Virtualized network functions chaining and routing algorithms," *Comput. Netw.*, vol. 114, pp. 95–110, Feb. 2017.

[58] P. T. A. Quang, A. Bradai, K. D. Singh, G. Picard, and R. Riggio, "Single and multi-domain adaptive allocation algorithms for VNF forwarding graph embedding," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 98–112, Mar. 2019.

[59] Z. Abbasi, M. X. M. , M. Shirazipour, and A. Takacs, "An optimization case in support of next generation NFV deployment," in *Proc. 7th USENIX Workshop Hot Topics Cloud Comput.*, Santa Clara, CA, USA, Jul. 2015, pp. 1–3.

[60] R. Riggio, A. Bradai, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and T. Ahmed, "Virtual network functions orchestration in wireless networks," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, Barcelona, Spain, Nov. 2015, pp. 108–116.

[61] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," in *Proc. ACM SIGCOMM*, Toronto, ON, Canada, Aug. 2011, pp. 350–361.

[62] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *Proc. 1st ACM Symp. Cloud Comput.*, Indianapolis, IN, USA, 2010, pp. 193–204.

[63] W. Deng, F. Liu, H. Jin, X. Liao, H. Liu, and L. Chen, "Lifetime or energy: Consolidating servers with reliability control in virtualized cloud datacenters," in *Proc. 4th IEEE Int. Conf. Cloud Comput. Technol. Sci. Proc.*, Taipei, Taiwan, Dec. 2012, pp. 18–25.

[64] M. G. Rabbani, M. F. Zhani, and R. Boutaba, "On achieving high survivability in virtualized data centers," *IEICE Trans. Commun.*, vol. E97.B, no. 1, pp. 10–18, 2014.

[65] Q. Zhang, M. F. Zhani, M. Jabri, and R. Boutaba, "Venice: Reliable virtual data center embedding in clouds," in *Proc. IEEE Conf. Comput. Commun.*, Toronto, ON, Canada, Apr. 2014, pp. 289–297.

[66] M. Scholler, M. Stiemerling, A. Ripke, and R. Bless, "Resilient deployment of virtual network functions," in *Proc. 5th Int. Congr. Ultra Mod. Telecommun. Control Syst. Workshops (ICUMT)*, Almaty, Kazakhstan, Sep. 2013, pp. 208–214.

[67] P. Torkamandi, S. Khorsandi, and B. Bakhshi, "Availability aware SFC embedding in NFV: A clustering approach," in *Proc. 27th Iranian Conf. Electr. Eng. (ICEE)*, Yazd, Iran, Apr. 2019, pp. 1922–1928.

[68] J. Fan, M. Jiang, and C. Qiao, "Carrier-grade availability-aware mapping of service function chains with on-site backups," in *Proc. IEEE/ACM 25th Int. Symp. Qual. Service (IWQoS)*, Vilanova Geltrú, Spain, Jun. 2017, pp. 1–10.

[69] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "GREP: Guaranteeing Reliability with Enhanced Protection in NFV," in *Proc. 2015 ACM SIGCOMM Workshop Hot Topics Middleboxes Netw. Function Virtualization*, London, U.K., Aug. 2015, pp. 13–18.

[70] J. Xu, J. Tang, K. Kwiat, W. Zhang, and G. Xue, "Survivable virtual infrastructure mapping in virtualized data centers," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Honolulu, HI, USA, Jun. 2012, pp. 196–203.

[71] P. Richard and T. Klein. (Jan. 2015). *G.W.A.T.T. New Bell Labs Application Able to Measure the Impact of Technologies Like SDN & NFV on Network Energy Consumption*. [Online]. Available: https://media-bell-labs-com.s3.amazonaws.com/pages/20150114_1907/GWATT_WhitePaper.pdf

[72] R. Mijumbi, "On the energy efficiency prospects of network function virtualization," 2015, *arXiv:1512.00215*. [Online]. Available: http://arxiv.org/abs/1512.00215

[73] A. Montazerolghaem, M. H. Yaghmaee, and A. Leon-Garcia, "Green cloud multimedia networking: NFV/SDN based energy-efficient resource allocation," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 3, pp. 873–889, Sep. 2020.

[74] L. M. M. Zorello, M. G. T. Vieira, R. A. G. Tejos, M. A. T. Rojas, C. Meirosu, and T. C. Melo de Brito Carvalho, "Improving energy efficiency in NFV clouds with machine learning," in *Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD)*, San Francisco, CA, USA, Jul. 2018, pp. 710–717.

[75] D. Krishnaswamy, R. Krishnan, D. Lopez, P. Willis, and A. Qamar, "An open NFV and cloud architectural framework for managing application virality behaviour," in *Proc. 12th Annu. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2015, pp. 746–754.

[76] A. Marotta, F. D'Andreagiovanni, A. Kassler, and E. Zola, "On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures," *Comput. Netw.*, vol. 125, pp. 64–75, Oct. 2017.

[77] M. M. Tajiki, S. Salsano, L. Chiaraviglio, M. Shojafar, and B. Akbari, "Joint energy efficient and QoS-aware path allocation and VNF placement for service function chaining," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 374–388, Mar. 2019.

[78] B. Farkiani, B. Bakhshi, and S. A. MirHassani, "A fast near-optimal approach for energy-aware SFC deployment," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1360–1373, Dec. 2019.

[79] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.

[80] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[81] A. Leivadeas, M. Falkner, I. Lambadaris, and G. Kesidis, "Optimal virtualized network function allocation for an SDN enabled cloud," *Comput. Standards Interfae*, vol. 54, pp. 266–278, Nov. 2017.

[82] M. A. Abdelaal, G. A. Ebrahim, and W. R. Anis, "Efficient Placement of Service Function Chains in Cloud Computing Environments," *Electronics*, vol. 10, no. 3, pp. 1–22, Jan. 2021.

[83] M. Hamann and M. Fischer, "Path-based optimization of NFV-resource allocation in SDN networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[84] H. Hantouti, N. Benamar, T. Taleb, and A. Laghrissi, "Traffic steering for service function chaining," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 487–507, 1st Quart., 2019.

[85] C. Talavera and J. Santisteban, "Design of network infrastructure of a cloud data center for use in health sector," in *Proc. 7th Latin Amer. Workshop Commun.*, Arequipa, Peru, Nov. 2015, pp. 30–35.

[86] R. Pries, M. Jarschel, D. Schlosser, M. Klopf, and P. Tran-Gia, "Power consumption analysis of data center architectures," in *Green Communications and Networking* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 51, J. J. P. C. Rodrigues, L. Zhou, M. Chen, and A. Kailas, Eds. Berlin, Germany: Springer, Oct. 2011, pp. 114–124.
[87] V. Asghari, R. Farrahi Moghaddam, and M. Cheriet, "Performance analysis of modified BCube topologies for virtualized data center networks," *Comput. Commun.*, vol. 96, pp. 52–61, Dec. 2016.
[88] M. A. Abdelaal, G. A. Ebrahim, and W. R. Anis, "A scalable network-aware virtual machine allocation strategy in multi-datacentre cloud," *Int. J. Cloud Comput.*, vol. 8, no. 2, pp. 183–206, May 2019.
[89] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

MARWA A. ABDELAAL received the B.Sc. degree in electronics and electrical communication engineering from Assiut University, Assiut, Egypt, in 2004, and the M.Sc. degree in electrical engineering (electronics and communications engineering) from the Faculty of Engineering, Ain Shams University, Cairo, Egypt, in 2018. She is currently a Ph.D. Student with the Electronics and Communications Engineering Department, Faculty of Engineering, Ain Shams University. Her research interests include SDN-enabled cloud computing, resource provisioning, efficient datacenters, and network function virtualization.

GAMAL A. EBRAHIM received the B.Sc. and M.Sc. degrees from the Faculty of Engineering, Ain Shams University, Cairo, Egypt, in 1994 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Miami, Coral Gables, FL, USA, in 2004. He is currently a Professor with the Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University. His current research interests include computer networks, cloud computing, distributed multimedia communication, and intelligent systems.

WAGDY R. ANIS received the B.Sc. degree from the Electronics and Communications Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt, in 1973, the M.Sc. degree from the Electronics and Communications Engineering Department, in 1977, and the Ph.D. degree in electronics engineering from Louvain University, Belgium, in 1985. He is currently a Professor Emeritus with the Electronics and Communications Engineering Department, Faculty of Engineering, Ain Shams University.