

Received April 16, 2020, accepted May 9, 2020, date of publication May 18, 2020, date of current version June 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995384

Communication and Computing Cost Optimization of Meshed Hierarchical NFV Datacenters

BINAYAK KAR¹, (Member, IEEE), **ERIC HSIAO-KUANG WU²**, (Member, IEEE),
AND YING-DAR LIN³, (Fellow, IEEE)

¹Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

²Department of Computer Science and Information Engineering, National Central University, Chung-Li 32001, Taiwan

³Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan

Corresponding author: Binayak Kar (bkar@mail.ntust.edu.tw)


This work was supported in part by the H2020 Collaborative Europe/Taiwan Research Project 5G-CORAL under Grant 761586, and in part by the Ministry of Science and Technology (MOST), Taiwan.

ABSTRACT Telecommunication carriers of 5G-MEC are re-architecting their central offices and mobile base stations as datacenters with network function virtualization (NFV) technology. These datacenters (DCs) are known as edge datacenters that help network operators speed deployment and reduce costs. Previously, the use of NFV was limited to within a datacenter (DC) known as intra-DC. Recently, many studies have been conducted into the use of NFV across DCs, *i.e.*, inter-DC. However, these NFV inter-DC architectures have limited communication between DCs with either horizontal or vertical connectivity. In this paper, we propose a generic architecture of such edge NFV datacenters with both horizontal and vertical connectivity, and demonstrate the consequences of both vertical and horizontal connectivity between DCs in terms of communication and computing costs. We formulate a cost optimization problem with latency and capacity as constraints by estimating the traffic dispatch rate between DCs. We propose a vertical-horizontal communication (VHC) heuristic solution to the NP-hard problem. Compared to horizontal connectivity, our results show that vertical connectivity helps to reduce computing costs by 10-30%. However, both vertical and horizontal communications together can help to reduce such costs by 20-30% compared to only vertical communication.

INDEX TERMS Inter-DC connectivity, VNF placement, service chaining, communication, computing.

I. INTRODUCTION

Network function virtualization (NFV) [1] is a new alternative technology in the revolution of the communication network that has emerged as an appealing solution to transform dedicated hardware implementations to software instances running in a virtualized environment. In NFV, a requested service is implemented by a sequence of Virtual Network Functions (VNF) that can run on generic servers by leveraging virtualization technology. These VNFs are pitched with a predefined order. This is also known as Service Function Chaining (SFC) [2]. NFV is being adopted by telecommunication service providers (TSPs) to avoid the problems caused

The associate editor coordinating the review of this manuscript and approving it for publication was Zehua Guo .

by the application of traditional techniques over the years. NFV brings flexibility, easy deployment, dynamic adjustment on demand, and easy and faster up-gradation [4]. NFV offers a new way to design, deploy, and manage networking services by decoupling the network functions, such as network address translation, firewalls, intrusion detection, domain name service, etc., from dedicated hardware devices so they can run in software [1], [3]. Hence, TSPs such as AT&T are re-architecting their central offices as datacenters, popularly known as CORD (Central Office Re-architected as a Datacenter) [5], [6]. To improve the user-perceived service response time in 5G-MEC [7] architecture and deliver faster services, service providers are upgrading their base stations (BS) to NFV-enabled datacenters [8]. These NFV-enabled central offices and base stations are called edge datacenters

(EDC) [9]. Such NFV datacenters with software defined networking (SDN) help service providers speed up deployment and reduce costs [10].

Over the past few years, some research has been conducted on the integration of NFV and service chaining in datacenters [11], [12]. However, such integration of NFV is limited to within a datacenter, *i.e.*, intra-DC [13]. When the communication takes place between two virtual machines (VMs) in the same or different servers but in the same DC we called it intra-DC communication. However, if two VMs in two different servers and those two servers are in different DCs we called it inter-DC communication. In recent years, a few published research papers have extended NFV across the datacenters, *i.e.*, inter-DC [14]. Such inter-DC architectures focus on either horizontal [15] or vertical communication [14], [16]. In a multi-tier topology, the communication between two sibling is called horizontal communication (*example: communication between node 2.1 and node 2.2 in Figure 1*) whereas the communication between parent and child is called vertical communication (*example: communication between node 2.1 and node 3.1 in Figure 1*). However, in a single-tier topology, the communication is horizontal by default. The connectivity of DCs depends on various factors, such as the number of DCs, their location, capital expenditure, and so on. Similarly, the number of DCs also plays a significant role in the selection of particular connectivity architecture.

In the USA a service provider like AT&T has about 4700 central offices [5], while in Taiwan, a service provider like Chunghwa Telecom has only about 450. India, which is the second largest telecom subscriber, has around 1.1 million base stations that provide 86% of the coverage for the total population [18]. To date, there are two ways to connect these edge NFV datacenters (NFV-enabled CO and BS): a horizontal connection (where the DCs are connected to its siblings) or a vertical connection (where the DCs are connected to both parent and child DCs). However, there is an alternative connection, the combination of both horizontal and vertical connections, where the DCs can be connected to parent, child, and sibling DCs. The detailed architecture of this NFV inter-DC connectivity is discussed in Section II-A.

In NFV, a VNF runs as a virtual machine (VM) on a physical device of the DC to continuously serve packets belonging to one or more flows. By doing this, the computing cost is reduced when a single VNF is shared between different flows, reducing the number of active VMs. However, in inter-DC architecture, communication costs are still an issue. Although the intra-DC communication delay can be neglected (as it is relatively very low compared to the inter-DC communication delay [17]), as well as the communication costs within the datacenter [11], [23], however, in inter-DC, the distance between DCs and the location of the DCs will affect the inter-DC communication cost [19]. Communication costs depend mostly on the flow, its size, which and how many DCs it is traveling between. Again, the path of the flow depends on the service chain demand of the flow and available VNFs

of the DCs. While adopting certain connectivity for these NFV-DCs, a few questions arise immediately:

- 1) *Which connectivity helps more—vertical or horizontal?*
- 2) *Which capacity helps more on inter-DC architecture—the communication capacity between DCs, or computing capacity within a DC?*
- 3) *Does higher inter-DC communication capacity help reduce the required computing capacity?*

In this paper, we attempt to solve these issues. This novel contribution can be summarized as

- 1) First, we propose a generic inter-DC architecture, which can have one or multiple tiers. Each DC may have vertical and horizontal connectivity.
- 2) Second, we design a model to estimate the inter-DC traffic rate and formulate an optimization problem to minimize the cost of the network with capacity and delay as the constraints.
- 3) We propose a heuristic algorithm for the inter-DC network communication. By a MATLAB experiment, we demonstrate the performance of different types of networks.

The remainder of the paper is organized as follows. In Section II, we discuss inter-DC architecture and related works. We discuss system models and formulate optimization problems in Section III. A heuristic solution is presented in Section IV, and in Section V, we analyze the results, and draw conclusions in Section VI.

II. BACKGROUND

In this section, we first discuss possible inter-DC network topologies and their properties. In the second part, we will discuss work related to our paper.

A. GENERIC INTER-DC ARCHITECTURE

For communication between DCs, connectivity is a key factor [20], [21]. This connectivity can be either horizontal (*i.e.*, between siblings) or vertical/hierarchical (*i.e.*, between parent and child), or both. Again, in hierarchical connectivity, a parent DC may have multiple child DCs and a child DC can have multiple parents. Taking these factors into account, we have considered five different topologies such as the partial mesh (M), tree (T), fat-tree (FT) [22], tree with partial mesh (TwM), and fat-tree with partial mesh (FTwM) that demonstrate the degree of DC connectivity (*There are two kinds of degree of connectivity: (1) The horizontal degree of connectivity defines a node horizontally and directly connected to how many of its siblings (2) The vertical degree of connectivity means a node vertically and directly connected to how many of its parent nodes.*) in all possible scenarios. Of these topologies, the tree topology is used in [14] and partial mesh topology is used in [15] for datacenter connectivity. Again, the type of connectivity of the datacenters also influences the single point of failure problem of the network. If the connectivity topology is a single parent topology like tree topology or TwM topology, the single point failure is

TABLE 1. Degree of connectivity of datacenters.

| Topology | Degree of Connectivity (horizontal) | Degree of Connectivity (vertical) |
|--------------------------|-------------------------------------|-----------------------------------|
| Tree | 0 | 1 |
| Fat-tree | 0 | Multiple |
| Partial mesh | Multiple | 0 |
| Tree w/ partial mesh | Multiple | 1 |
| Fat-tree w/ partial mesh | Multiple | Multiple |

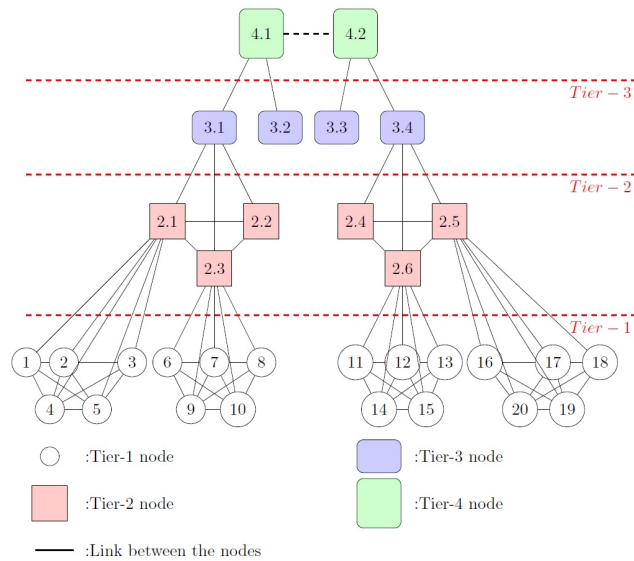


FIGURE 1. Multi-tier tree with partial mesh inter-DC architecture.

obvious. The single point failure possibility of the partial mesh topology is purely depending on the connectivity of the nodes. However, in topologies like FT topology or FTwM topology, where we have multiple path connectivity due to multiple parents, the single point failure problem is very rare. Table 1 shows the degree of horizontal and vertical connectivity of DCs for the different topologies. In the partial mesh topology, all DCs are in one tier and are connected horizontally in an arbitrary manner without any predefined structure. The tree (T), fat-tree (FT), tree with partial mesh (TwM), and fat-tree with partial mesh (FTwM) topologies are multi-tier structures. In these multi-tier topologies, the tier-1 nodes represent the NFV-enabled base stations and tier-2 nodes represent the CORD; each CORD has multiple base stations. For example, as shown in Figure 1, the base stations *i.e.*, tier-1 nodes which are directly connected to the node in tier-2 are under that CORD. Other top-tier nodes are NFV DCs that establish the connection between these edge DCs. The T and FT connect vertically, whereas, TwM and FTwM connect both vertically and horizontally. In T and TwM, each child will have a single parent, whereas, in FT and FTwM, a child may have multiple parents. In TwM and FTwM, siblings of the same parents are connected horizontally. In this paper, we have considered that sibling nodes are connected horizontally in tier-1 in TwM and FTwM and in tier-2 in TwM. An example of TwM topology is shown in Figure 1.

B. RELATED WORKS

1) INTRA-DC

A survey of the literature revealed that several works have reported on intra-DC network architecture [11]–[13], [23]–[26] and addressed different optimization issues separately. The service chain composition problem in NFV networks was discussed by D’Oro *et al.* in [23]. They proposed a distributed and privacy-preserving algorithm using the non-cooperative game theory in polynomial time. In [24], D’Oro *et al.* used the game theory to model for the interaction between a user’s demand and a server’s availability and response in which they focus on the distributed resource allocation and orchestration of a softwarized network. An Eigen-decomposition-based approach for the placement of network function chains was presented in [25]. Sun *et al.* in [26] proposed a reliability cost saving algorithm to condense the capital expenditures (CAPEX) and operational expenditures (OPEX) of telecommunication service providers, by reducing the reliability of the SFC deployments. In [12], Liu *et al.* discussed the optimal deployment of new service function chains and readjustment of the in-service chains dynamically. Bari *et al.* [13] solve the problem of determining the number of VNFs required and their placement to optimize operational expenses dynamically while adhering to service level agreements using an integer linear programming (ILP). Kar *et al.* proposed an *m/m/c* queuing model in [11] to dynamically optimize the energy consumption cost of the NFV datacenter network with the *minimum capacity* policy, where a certain amount of load is required to start the physical machine (PM), increasing the utilization of the PM, and avoiding frequent changes of the PM’s states. It uses VNF chaining to minimize energy consumption cost within a datacenter, where only computing cost is taken into consideration as the communication costs within a datacenter is minimal compared to computation costs. However, this paper focuses on the optimal deployment of service functions across datacenters to minimize the total cost, including both computation and communication costs.

2) INTER-DC

The service chaining across datacenter is still in its early stage. However, Gharbaoui *et al.*, in [35] shows experimental validation of an orchestration system for geographically distributed Edge/NFV clouds, supporting end-to-end latency-aware and reliable network service chaining. To do the service changing across datacenters, they set up their experiment on top of the Fed4FIRE+ experimentation platform with three datacenters. VirtPhy, a fully programmable NFV orchestration architecture for edge datacenters based on server-centric topologies, was discussed in [29]. It is mainly a distributed service function chaining scheme, which integrates NFV and SDN to benefit from the physical network topology and enable SFC in datacenter environments based on software switches. In [30], Chen *et al.* provided the first study on traffic dynamics among multiple

datacenters using the network traces collected at five major Yahoo! datacenters. The results show that Yahoo! employs a hierarchical way of deploying its datacenters. Yang *et al.* in [31] presented an optimal resource allocation method in an NFV-enabled Mobile Edge-Cloud environment. In this work, they addressed where and when to allocate the resources as well as how many resources could be allocated. A jointly optimized network delay and energy saving mechanism were studied in [32] and considered intra-and-inter datacenter VM placement issues. In this large-scale cloud system, they considered multiple medium-size DCs geographically distributed, connected via the backbone network. In [33], Bouet *et al.* considered a geo-clustering approach for mobile edge computing (MEC) resource optimization. In their paper, they presented a graph-based algorithm which enables identifying a section of MEC areas where traffic is consolidated at the edge of the MEC servers. Obadia *et al.* presented a novel game-theory approach for exploiting excessive resources, offering service function chains which point to a new business model and revenue opportunities for NFV operators in [34]. A low-cost VNF placement and the routing and spectrum assignment (RSA) on the multicast tree was discussed in [36], where both static network planning and dynamic network provisioning is addressed.

Gu *et al.* proposed a general model framework for inter-DC that describes the relationship of geo-distributed datacenters and formulate the communication cost minimization problem for big data stream processing (BDSP) in [19]. CARPO, a correlation aware power optimization scheme for datacenter networks was proposed by Wang *et al.* in [28], in which they dynamically consolidate traffic flows onto a small set of links and switches in a datacenter network and then shut down unused network devices for energy saving. Krishnaswamy *et al.* propose partitioning the VNF types according to their latency sensitivity [27] where the resources in the datacenters can be allocated hierarchically for NFV. In [14], Lin *et al.* propose hierarchical NFV/SDN-integrated architecture in which datacenters are organized into a multi-tree overlay network to collaboratively process user traffic flows. However, the results are yet to be optimized, and they have not considered inter-DC service chaining. The articles [14], [16] and [27] discuss layered architecture but only vertical connectivity, and not the horizontal connectivity issue. The major differences of this paper compared to the other work noted are:

- 1) First, all of the papers on inter-DC focus on either cost and latency or cost and capacity but not all three together, which is the key contribution of this paper.
- 2) Second, these papers address either communication cost or computing cost; however, in this paper, we address both the communication and computing costs.
- 3) Finally, these papers consider either horizontal or vertical connectivity, whereas in this paper, we are considering not only horizontal and vertical connectivity individually, but also vertical-horizontal connectivity

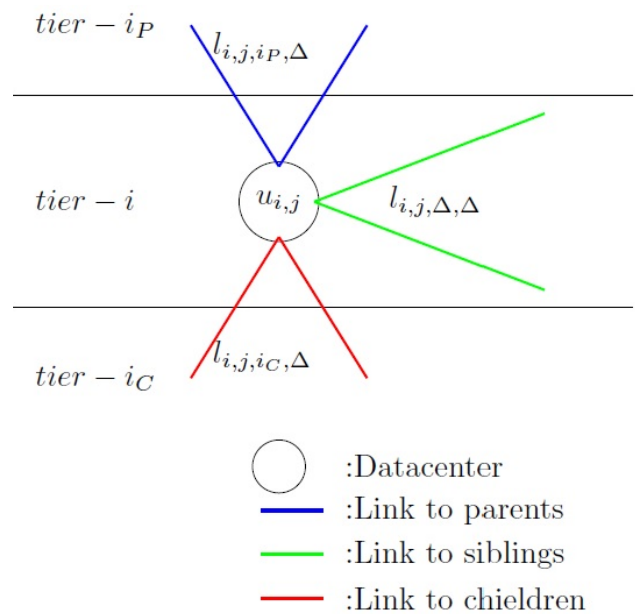


FIGURE 2. Inter-DC tier classification.

by considering topologies like TwM and FTwM where a node can communicate to its parent, child, and sibling directly.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we develop our system model and state our objective. In Table 2, we list the variables used to estimate the traffic rate between the DCs, both within and across the tiers, and formulate an optimization problem to minimize the total cost, which includes the communication cost and computing cost. The mathematical modeling of the VNF placement across DCs is a complex task. In this paper, to simplify our formulation, we assumed the datacenters are single server datacenters.

A. SYSTEM MODEL

We consider a generic inter-EDC architecture composed of multi-tier topology where DCs may have zero-to-multiple degrees of connectivity, both horizontally and vertically. U_i denotes the set of DCs in tier- i . $u_{i,j}$ denotes the j -th DC in i -th tier, and $\hat{\mu}_C^{i,j}$ is its capacity. \hat{U} denotes the set of DCs in tier-1 from where traffic originates and terminates. L denotes the set of links, and $l_{i,j,i',j'}$ denotes the link between $u_{i,j}$ and $u_{i',j'}$; if both ends of the links are in the same tier, then $j \neq j'$. The capacity of the link $l_{i,j,i',j'}$ is denoted as $\hat{\mu}_M^{i,j,i',j'}$. We classify the DCs according to tiers, where tier- i_P and tier- i_C (*i.e.*, tier- $(i + 1)$ and tier- $(i - 1)$) are the parent tier and child tier of tier- i . $l_{i,j,\Delta,\Delta}$, $l_{i,j,i_P,\Delta}$, and $l_{i,j,i_C,\Delta}$ stands for the links between DC $u_{i,j}$ and DCs in tier- i , tier- i_P , and tier- i_C , respectively, as shown in Figure 2. F and SC denote the set of network functions and service chains, respectively, and μ_C^n is the capacity of f^n . If $|F| = w$, and the length of a service chain is some integer between $[p, q]$, where $q \geq p$, then theoretically

TABLE 2. List of commonly used variables and notations.

| Notations | Descriptions |
|--|--|
| H | Number of tiers in the topology |
| $u_{i,j} \in U_i$ | U_i is the set of DCs in tier- i , and is j -th DC in i -th tier |
| $\hat{s}, \hat{d} \in \hat{U}$ | \hat{s} is DC $u_{1,s}$ and \hat{d} is DC $u_{1,d}$. $\hat{U} \subseteq U_1$ is the set of host DCs where traffic originates and terminates |
| $l_{i,j,i',j'} \in L$ | Set of links between DCs and link between $u_{i,j}$ and $u_{i',j'}$ if $i == i'$ then $j \neq j'$ |
| $f^n \in F$ | F is the set of VNFs, and f^n is the network function n |
| $\hat{g}_{i,j}^n$ | A binary variable. Value will be 1 if network function n running on DC $u_{i,j}$, 0 otherwise |
| $r_{i,j}^n$ | Number of VMs of network function running on DC $u_{i,j}$ |
| SC | Set of service chains |
| $\hat{\lambda}_{\hat{s},\hat{d},k} = \sum_t \lambda_{\hat{s},\hat{d},k,t}$ | $\hat{\lambda}_{\hat{s},\hat{d},k}$ is the flow with k -th service chain from DC $u_{1,s}$ to DC $u_{1,d}$, and $\lambda_{\hat{s},\hat{d},k,t}$ is the t -th flow with k -th service chain from DC $u_{1,s}$ to DC $u_{1,d}$, if $t == 0$, then no flow |
| $g_{\hat{s},\hat{d},k,t}^{i,j,i',j'}$ | A binary variable. Value will be 1, if t -th flow with k -th service chain from DC $u_{1,s}$ to DC $u_{1,d}$ passes through link $l_{i,j,i',j'}$, 0 otherwise |
| τ | Amount of traffic with k -th service chain from DC $u_{1,s}$ to DC $u_{1,d}$ |
| $P_{\hat{s},\hat{d},k} \in P_{\hat{s},\hat{d}}$ | $P_{\hat{s},\hat{d}}$ is the set of paths from DC $u_{1,s}$ to DC $u_{1,d}$, and $p_{\hat{s},\hat{d},k}$ is the path of a flow with k -th service chain from DC $u_{1,s}$ to DC $u_{1,d}$ |
| $ly_{i,j,i',j'}, Y_{\hat{s},\hat{d},k}$ | $ly_{i,j,i',j'}$ is the latency from DC $u_{i,j}$ to DC $u_{i',j'}$, and $Y_{\hat{s},\hat{d},k}$ is the tolerable latency of the flow with k -th service chain from DC $u_{1,s}$ to DC $u_{1,d}$ |
| $\mu_c^n, \hat{\mu}_c^{i,j}$ | Computing capacity of one instance of VM running network function n and computing capacity of DC $u_{i,j}$. |
| $\hat{\mu}_M^{i,j,i',j'}, \mu_M^{\hat{s},\hat{d},k,t}$ | Communication capacity of link $l_{i,j,i',j'}$, and communication capacity consumed by t -th flow with k -th service chain from DC $u_{1,s}$ to DC $u_{1,d}$ |
| δ_M, δ_C | Cost of one unit of communication capacity and computing capacity |
| π, π_M, π_C | Total cost, communication cost, and computing cost |
| $ta_{i,j}, td_{i,j}$ | Aggregate traffic arrival to DC $u_{i,j}$ and aggregate traffic departure from DC $u_{i,j}$ |
| $ta_{i,j}^i, ta_{i,j}^P, ta_{i,j}^{i_C}$ | Total traffic arrival rate from all DCs in the tier- i , tier- i_P and tier- i_C to DC $u_{i,j}$ |
| $td_{i,j}^i, td_{i,j}^P, td_{i,j}^{i_C}$ | Total traffic departure rate from DC $u_{i,j}$ to all DCs in the tier- i , tier- i_P and tier- i_C |
| $ti_{i,j}, tt_{i,j}$ | Total traffic initiated from DC $u_{i,j}$ and total traffic terminated at DC $u_{i,j}$ |

we can have $|SC| = \frac{q(q+1)-p(p-1)}{2} * w!$ number of service chains¹ (considering duplication of network functions in the service chains). The notation $\lambda_{\hat{s},\hat{d},k,t}$ stands for the t -th flow with k -th service chain from \hat{s} to \hat{d} and $\hat{\lambda}_{\hat{s},\hat{d},k}$ is the summation of all traffic with k -th service chain from \hat{s} to \hat{d} . The set of paths between two host DCs, \hat{s} and \hat{d} is denoted by $P_{\hat{s},\hat{d}}$, and $p_{\hat{s},\hat{d},k}$ is the path of a flow with k -th service chain from \hat{s} to \hat{d} . The maximum tolerable latency of a flow with k -th service chain from \hat{s} to \hat{d} is presented by $Y_{\hat{s},\hat{d},k}$ and $ly_{i,j,i',j'}$ is the latency of the link $l_{i,j,i',j'}$.

B. EXAMPLE OF INTER-DC TRAFFIC FLOW

In this section, we will discuss the inter-DC traffic flow using one example. Figure 3 shows a 3-tier topology with 4 DCs in tier-1 and two DCs in tier-2 and tier-3 each. We have four flows i.e., $\lambda_{b,c,1,1}$, $\lambda_{b,c,1,2}$, $\lambda_{a,d,2,1}$, $\lambda_{a,d,2,2}$. The first two flows (i.e., $\lambda_{b,c,1,1}$, $\lambda_{b,c,1,2}$) are from DC $u_{1,2}$ ('b') to DC $u_{1,3}$ ('c') with service chain 1 and the last two flows (i.e., $\lambda_{a,d,2,1}$, $\lambda_{a,d,2,2}$) are from DC $u_{1,1}$ ('a') to DC $u_{1,4}$ ('d')

¹In theory, duplicating a service function (SF) so that it occurs more than once in a chain is possible. This is a case when Network Service Header (NSH) or Segment Routing (SR) is used as chaining methods. In SR, the Source Routing Header (SRH) contains a segment left index which is decremented each time we go through a SF. In NSH, the routing information is distributed in the Service Forward Function (of the switches), and the packer has a chain ID, and an index in its NSH header. So, in both cases, a SF can be hit several times, each one with a different index. Other service chaining methods or implementations may not support this capability though. For example, if service chaining is implemented by chaining the destination MAC address to be one of the next hops, then a SF cannot appear twice in the chain.

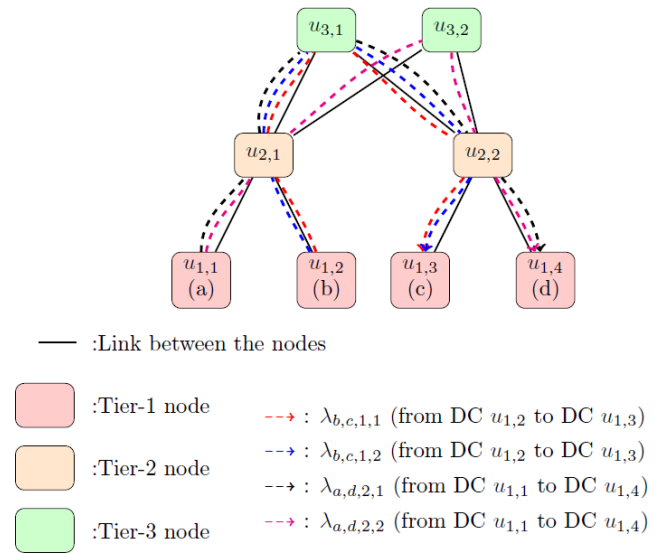


FIGURE 3. An example of inter-DC traffic flow.

with service chain 2, as shown in Figure 3. When a flow travels from source to destination across DCs, the path of the flow will be selected from of multiple available paths, based on the service chain of that flow and the available service function of the DCs, the available capacity of the DCs and the link capacity along the path. Let us assume the flows $\lambda_{b,c,1,1}$ and $\lambda_{b,c,1,2}$ travel along the path $u_{1,2} \rightarrow u_{2,1} \rightarrow u_{3,1} \rightarrow u_{2,2} \rightarrow u_{1,3}$. The flow $\lambda_{a,d,2,1}$ travels along the path $u_{1,1} \rightarrow u_{2,1} \rightarrow u_{3,1} \rightarrow u_{2,2} \rightarrow u_{1,4}$, and due to capacity

limitation in DC $u_{3,1}$, the flow $\lambda_{a,d,2,2}$ travels along the path $u_{1,1} \rightarrow u_{2,1} \rightarrow u_{3,2} \rightarrow u_{2,2} \rightarrow u_{1,4}$. Then the total traffic passing through link $l_{2,1,3,1}$ (i.e., from DC $u_{2,1}$ to DC $u_{3,1}$) is $\lambda_{b,c,1,1} + \lambda_{b,c,1,2} + \lambda_{a,d,2,1}$.

C. INTER-DC TRAFFIC ESTIMATION

In this section, we estimate the traffic rate between the datacenters both within a tier and across the tiers. The total traffic arrival rate at the DC from all DCs in the tier- i , tier- i_p (i.e., the parent tier of tier- i or tier- $(i+1)$), and tier- i_c (i.e., the child tier of tier- i or tier- $(i-1)$) are presented in Equations (1), (2) and (3), respectively, as

$$ta_{i,j}^i = \sum_{\forall l_{\Delta,\Delta,i,j}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i, \Delta, i, j} * \mu_M^{\hat{s}, \hat{d}, k, t}, \quad (1)$$

$$ta_{i,j}^{i_p} = \sum_{\forall l_{i_p, \Delta, i, j}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i_p, \Delta, i, j} * \mu_M^{\hat{s}, \hat{d}, k, t}, \quad (2)$$

$$ta_{i,j}^{i_c} = \sum_{\forall l_{i_c, \Delta, i, j}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i_c, \Delta, i, j} * \mu_M^{\hat{s}, \hat{d}, k, t}. \quad (3)$$

And the total traffic departure rate from the DC $u_{i,j}$ to all DCs in tier- i , tier- i_p , and tier- i_c are presented in Equations (4), (5) and (6), respectively, as

$$td_{i,j}^i = \sum_{\forall l_{i,j,i,\Delta}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i, j, i, \Delta} * \mu_M^{\hat{s}, \hat{d}, k, t}, \quad (4)$$

$$td_{i,j}^{i_p} = \sum_{\forall l_{i,j,i_p,\Delta}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i, j, i_p, \Delta} * \mu_M^{\hat{s}, \hat{d}, k, t}, \quad (5)$$

$$td_{i,j}^{i_c} = \sum_{\forall l_{i,j,i_c,\Delta}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i, j, i_c, \Delta} * \mu_M^{\hat{s}, \hat{d}, k, t}. \quad (6)$$

The aggregate traffic arrival rate ($ta_{i,j}$) at DC $u_{i,j}$ is the sum of the all incoming traffic to DC $u_{i,j}$ from all the DCs in tier- i , tier- i_p , and tier- i_c . By summing Equations (1), (2) and (3), we can estimate the aggregate traffic arrival rate at the DC $u_{i,j}$, shown in Equation (7) as

$$ta_{i,j} = \sum_{\forall l_{\Delta,\Delta,i,j}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{\Delta, \Delta, i, j} * \mu_M^{\hat{s}, \hat{d}, k, t}. \quad (7)$$

Similarly, Equation (8) gives the aggregate traffic departure rate ($td_{i,j}$) from DC $u_{i,j}$, which is sum of all departure traffics from DC $u_{i,j}$ to DCs in tier- i , tier- i_p , and tier- i_c can be estimated by summing Equations (4), (5) and (6) as

$$td_{i,j} = \sum_{\forall l_{i,j,\Delta,\Delta}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i, j, \Delta, \Delta} * \mu_M^{\hat{s}, \hat{d}, k, t}. \quad (8)$$

The above traffic estimation is applicable when the DC $u_{i,j}$ is not a host datacenter which that means neither any traffic is initiated nor terminated in that datacenter. If DC

$u_{i,j}$ is a host datacenter, then the aggregate arrival traffic with destination $u_{i,j}$ i.e., traffic terminate at DC $u_{i,j}$ ($tt_{i,j}$) and the aggregate departure traffic with source DC $u_{i,j}$ i.e., traffic initiated from $u_{i,j}$ ($t\hat{t}_{i,j}$) can be estimated by Equations (9) and (10), respectively, as

$$tt_{i,j} = \sum_{\forall l_{\Delta,\Delta,i,j}} \sum_{\forall \hat{s}, \hat{j} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{j}, k, t}^{\Delta, \Delta, i, j} * \mu_M^{\hat{s}, \hat{j}, k, t}, \quad (9)$$

$$t\hat{t}_{i,j} = \sum_{\forall l_{i,j,\Delta,\Delta}} \sum_{\forall \hat{j}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{j}, \hat{d}, k, t}^{i, j, \Delta, \Delta} * \mu_M^{\hat{j}, \hat{d}, k, t}. \quad (10)$$

Then the aggregate traffic departure rate from a host DC $u_{i,j}$ can be estimated as $ta_{i,j} + tt_{i,j} - t\hat{t}_{i,j}$.

D. OBJECTIVE FUNCTION AND CONSTRAINTS

In this section, we will use the notations given in Table 2 to formulate the optimization problem. In this problem, “by determining the computing capacity of the datacenters, communication capacity between DCs and traffic dispatch rate between DCs, our objective is to minimize the total cost in a generic inter-DC network, with given traffic arriving with a set of service chains in the given topology with deployed service functions, subject to constraints on the end-to-end delay.” Here, the total cost is the sum of computing cost and communication cost, i.e., $\pi = \pi_C + \pi_M$, where π_C and π_M are the computing cost and communication cost shown in Equations (11) and (12), respectively, as

$$\pi_C = \sum_{i=1}^H \sum_{j \in U_i} \sum_{n=1}^{|F|} \hat{g}_{i,j}^n * \mu_C^n * r_{i,j}^n * \hat{\delta}_C, \quad (11)$$

$$\pi_M = \sum_{\forall l_{i,j,i',j'}} \sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i, j, i', j'} * \mu_M^{\hat{s}, \hat{d}, k, t} * \delta_M. \quad (12)$$

Our objective is to minimize(π). The set of operational constraints to be noticed are

$$\sum l_{i,j,i',j'}^{\hat{s}, \hat{d}, k} \leq Y_{\hat{s}, \hat{d}, k}, \quad \forall l_{i,j,i',j'} \in P_{\hat{s}, \hat{d}, k}, \quad \forall P_{\hat{s}, \hat{d}, k} \in P_{\hat{s}, \hat{d}}, \quad (13)$$

$$\sum_{n=1}^{|F|} \hat{g}_{i,j}^n * \mu_C^n * r_{i,j}^n \leq \hat{\mu}_C^{i,j}, \quad \forall u_{i,j}, \quad (14)$$

$$\sum_{\forall \hat{s}, \hat{d} \in \hat{U}} \sum_{k=1}^{|\text{SC}|} \sum_{t=0}^{\tau} g_{\hat{s}, \hat{d}, k, t}^{i, j, i', j'} * \mu_M^{\hat{s}, \hat{d}, k, t} \leq \hat{\mu}_M^{i,j,i',j'}, \quad \forall l_{i,j,i',j'}. \quad (15)$$

1) Latency Constraint: The inequality in Equation (13) ensures the total sum of latency of a flow along the path must be less than or equal to the maximum tolerable latency of the flow.

2) Computing Capacity Constraint: The inequality in Equation (14) ensures the total sum of VM capacities in a DC must be less than or equal to the maximum capacity of that DC.

3) Communication Capacity Constraint: The inequality in Equation (15) is the communication capacity constraint. It shows that the sum of flow capacities through a link should not exceed that link's maximum capacity.

E. PROBLEM ANALYSIS

In this section, we will show that our proposed optimization problem can be NP-hard, by reducing the Network Testbed Mapping (NTM) problem [37], [38], (which is known to be NP-hard), to our problem in polynomial time. In the first step, we state the NTM problem. In the second step, we demonstrate that the NTM problem could be reduced to our problem.

1) Network Testbed Mapping (NTM) problem [38]: Given a network of switches, s_1, \dots, s_n with capacities C_1, \dots, C_n and inter-switch bandwidth capacities $B_{1,1}, \dots, B_{1,n}, B_{2,1}, \dots, B_{n,n}$, and a test network of node N_1, \dots, N_m with inter-node bandwidth requirements $b_{1,1}, \dots, b_{m,m}$. If there is an injective assignment $\mathcal{A} : N \rightarrow s$ such that:

$$|\mathcal{A}(u) = i| \leq C_i, \quad \forall i, 1 \leq i \leq n, \quad (16)$$

$$\sum_{\mathcal{A}(u)=1, \mathcal{A}(v)=j} b_{u,v} \leq B_{i,j}, \quad \forall i, j. \quad (17)$$

The mapping that satisfies Equations (16) and (17) is feasible where the summation is taken over all $\mathcal{A}(u), \mathcal{A}(v)$, satisfying the equalities.

2) NP-hard proof: Our problem has two parts: (1) The cost objective with latency and capacity constraints, and (2) Placement of VNF in the DCs to process the traffic with required service chains. The first part can be proved NP-hard as the capacitated set covering problem (CSCP) [39] is NP-hard. For the second part, if we map variables of the existing NTM NP-hard problem to the variables of our optimization problem, such as switches to Datacenters, switch capacities to Datacenter computing capacities, inter-switch bandwidth to inter-DC communication capacity, test network nodes to traffic with service chains, and inter-node bandwidth requirements to required capacity of traffic, we have,

$$\left\{ \begin{array}{l} (s_1, \dots, s_n) \rightarrow (u_{i,j}, \dots, u_{H,j}) \\ (C_1, \dots, C_n) \rightarrow \mu_C \\ (B_{1,1}, \dots, B_{1,n}, B_{2,1}, \dots, B_{n,n}) \rightarrow \hat{\mu}_M \\ (N_1, \dots, N_m) \rightarrow \hat{\lambda}_{\hat{s}, \hat{d}, k} \\ (b_{1,1}, \dots, b_{m,m}) \rightarrow \mu_{\hat{s}, \hat{d}, k, t} \end{array} \right\}. \quad (18)$$

With definition 5 of [11] and Equation (18), we can map and reduce the NTM NP-hard problem to our optimization problem in polynomial time by polynomial-time mapping reductions method [42]. Hence, our optimization problem is NP-hard.

IV. HEURISTIC APPROACH

In this paper, as we are neglecting the communication delay within the datacenters, hence, all the physical machines in a datacenter assumed to be one PM and all the VMs in a DC can be considered to be in one server. In such a scenario, the VNF

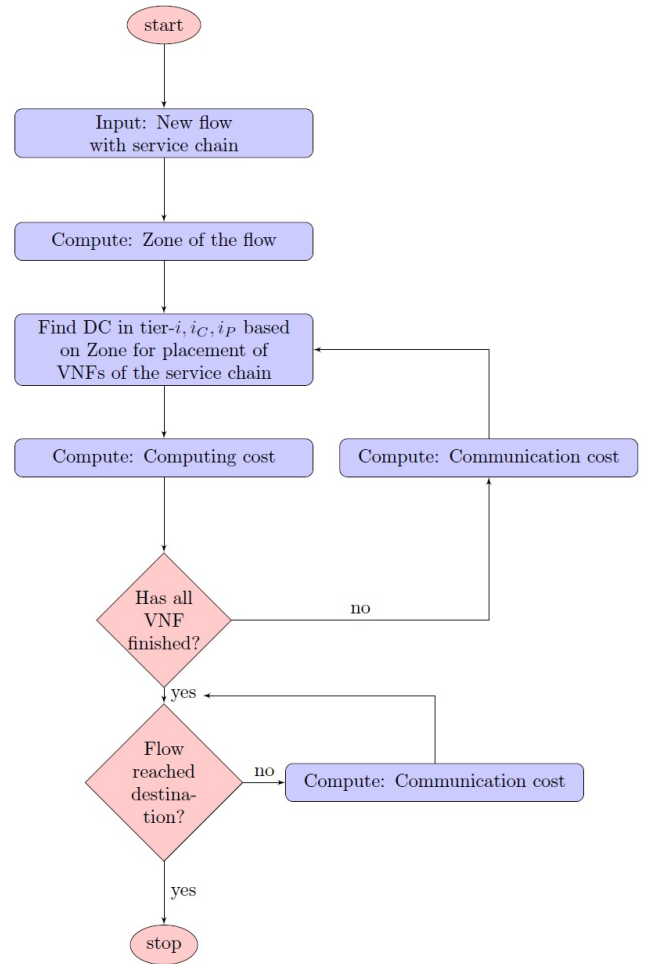


FIGURE 4. Vertical-horizontal communication algorithm.

placement approach of the inter-DC is similar to the approach of VNF placement of intra-DC. An illustrative example of traffic flow and VNF placement within the intra-DC was presented in [11] where each node represents one PM. In this paper, we followed a similar placement approach discussed in [11], but here each node represents one datacenter. Therefore, we neglect the intra-DC communication cost but consider the inter-DC communication cost.

In this section, we propose a heuristic algorithm which we have termed as vertical-horizontal communication (VHC) algorithm (shown in Figure 4) for VNF placement in the inter-DC network for both single and multi-tier topologies. We follow the physical and virtual path mapping as described in [11] and used the ²Assign and ³Release operations for the placement of VNFs. Assign operation is used to assign a flow to the VM to process its packets and Release operation is used to release a flow from the VM when processing of all packets of the flow completed by the VM. The computing cost is estimated based on how long the VM remains active in packet

²Definition 3 of [11].

³Definition 4 of [11].

processing and processing capacity of the respective VNFs. However, the communication cost is estimated based on how many flows and of what capacity are transferred from one datacenter to another, and as the communication cost doubles from one tier to the next tier above. We apply another operation *Zone*, described in Definition 1, to classify the zone of each flow according to its source and destination datacenters. To estimate the zones of the traffics, in a multi-tier topology, we named the nodes in each tier sequentially and classified them into multiple groups where all siblings of the same parents are considered a group (e.g., tier-1 of Figure 1 has four groups). When a flow is initiated, based on its destination node, we confirm its destination group and estimate its zone.

Definition 1 (Zone): For a traffic $\hat{\lambda}_{\hat{s},\hat{d},k}$ (source DC $u_{1,s}$ and destination DC $u_{1,d}$), the DC $u_{1,d}$ is in:

- [Zone-1] if $u_{1,d}$ is a sibling of $u_{1,s}$
- [Zone-2] if $u_{1,d}$ is a descendant of parent’s sibling of $u_{1,s}$
- [Zone-3] if $u_{1,d}$ is a descendant of grandparent’s sibling of $u_{1,s}$
- [Zone-H] if $u_{1,d}$ is a descendant of root-parent’s sibling of $u_{1,s}$.

The VHC algorithm works as follows: When a new flow arrives with its service chain, we first compute the traffic *zone* of the flow based on its source and destination as described in Definition 1. An example of traffic zone estimation is presented in Figure 5. Figure 5 shows a four-tier tree topology, with one source node ‘S’ and three destination nodes ‘D1’, ‘D2’ and ‘D3’. For traffic from S to D1, the zone is Zone-1 as the destination node is a sibling of the source node. For traffic from S to D2, since D2 is the descendant of S’s parent’s sibling, the zone is Zone-2. Similarly, for traffic from S to D3, the zone is Zone-3, as D3 is a descendant of S’s grandparent’s sibling. For each VNF of the service chain, we estimate the appropriate DC for placement in the appropriate tier based on its traffic *zone*. The placement function in this paper is a modified version of the algorithm DPVC described in [11]. The DPVC focused on the minimization of the energy cost consumption in a datacenter by reducing the active physical machines and maximizing the utilization of the machines. In DPVC, only the computing cost was taken into consideration to achieve its objective. However, this heuristic approach takes both communication (inter-DC) and computing cost into consideration to minimize the total cost. Since, this paper considers each node a datacenter, we cannot turn OFF a DC completely. Therefore, we use only two operations (Assign and Release), out of the four operations (Assign, Release, Add, and Delete) used in DPVC. Since each node in DPVC is physical machine, the Add and Delete operations are used to turn ON and OFF the machines. In this heuristic approach, if the VNF for next service function of a flow’s service chain is available in the same DC, the flow will be directed to that VNF in that DC, otherwise, the flow will be directed to another DC towards its destination based on cheaper communication cost. If two paths have the same communication

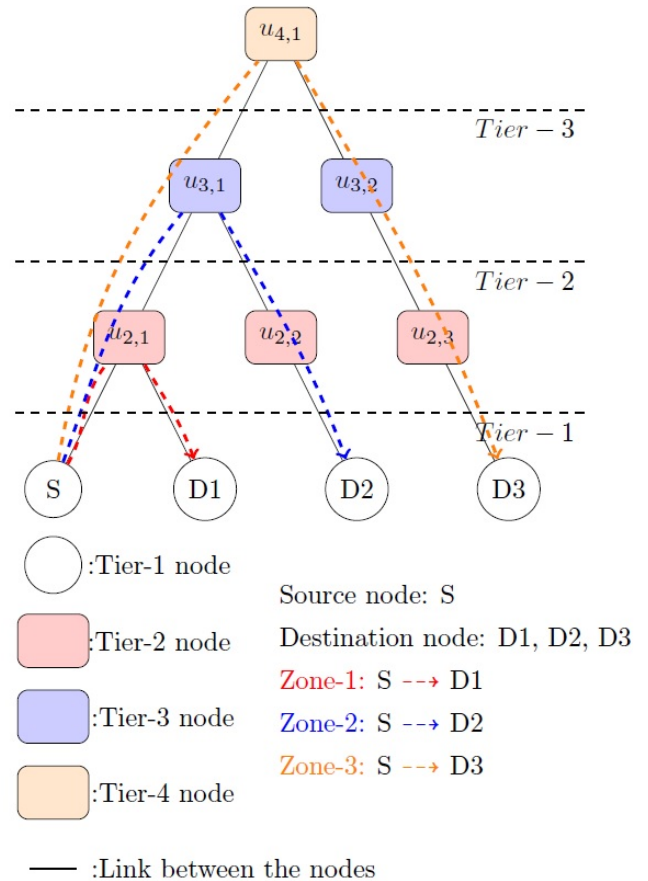


FIGURE 5. An example of estimating traffic zone.

cost, then the selection will be based on cheaper computing cost. After each iteration, we update the computing cost, and after each hop count of the flow, the communication cost is updated. The flow will terminate after all the packets of the flow have reached the destination after being processed by all VNFs of its service chain.

A. HEURISTIC COMPLEXITY ANALYSIS

Let there are N flows. The maximum length of each flow is K packets and the length of the assigned service chain is W (assumed to be the maximum length of the service chain). If Q is the minimal processing capacity of each function, and each flow passes through E number of edges to reach the destination, then by assuming that a VNF can process a single flow at a time and the lifetime of all flows are sequential (after the termination of one flow, next flow will be initiated), the worst case computing time of the algorithm is $O(E * N + N * \frac{WK}{Q})$. However, in practice, all VNFs can process multiple flows simultaneously and flows are not sequential. If there are x group of flows running together in the network and a VNF of a service chain shared by y number of flows, then the running time of the algorithm is $O(E * \log_x(N) + \frac{WK}{Q} * \log_y(N))$.

V. RESULTS

In this section, we discuss the experimental setup, which was used in this work to evaluate the performance of our

proposed algorithms on different topologies. In this experiment, we considered five different types of topologies: tree (T), fat-tree (FT), partial mesh (M), tree with partial mesh (TwM), and fat-tree with partial mesh (FTwM). Out of these topologies, the partial mesh topology was a single-tier topology and the others were multi-tier.

A. EXPERIMENT SETUP

We used MATLAB to compare the performance of our algorithm on different topologies. For this simulation, we considered randomly-generated flows (*i.e.*, source, destination, number of packets and service chain) as the input (maximum one flow per iteration) from a set of source DCs to a set of destination DCs, where for each flow, the source and destination nodes were not equal. All flows were initiated and terminated in the bottom tier of the topologies. The flows ranged from a minimum length of 10 packets to a maximum length of 1000 packets, and all packets were of equal size. We considered 10 types of network functions and each type had a different processing time. For each flow, the service chains were randomly generated of lengths consisting of a minimum one function to a maximum 10 functions without duplication of network functions in a service chain, *i.e.*, a chain of length 10 functions had to contain all types of network functions considered in this experiment. We considered three different cases, where the maximum number of VMs of equal capacities on the nodes were 10, 20, and 30. For the flow processing limit of the VMs, we also assumed three different values, *i.e.*, 5, 10 and 20 flows. A VM cannot process more flows than its capacity. When the number of input flow exceeds the capacity of the VM, a new VM will be added. The computing cost depended on how long the VNF remained active to process the packets of the flows. Again, this duration depended on the processing capacity of the VNF. For example: if we had two functions, A and B, with processing capacities of 10 and 25 packets per second, respectively, then processing a flow of 100 packets, the computing cost for A and B would be 10 and 4 units, respectively. When multiple flows shared a VNF, the cost was estimated based on when the first packet of the first arrival flow to that VNF started processing and when the last packet was served by that VNF. We considered this scenario since the property of NFV allowed us to consolidate multiple flows to process the packets in a single VNF [40] to reduce computing cost. The communication cost was estimated based on the number of packets transferred from one DC to another. Since the bottom tier nodes represent NFV-enabled base stations in our architecture, we considered the communication cost is one unit for transferring 800 packets from one DC to another DC in tier-1, as these DCs are relatively close to each other compared to the DCs in the top tiers. The cost is doubled if the communication takes place between the DCs in tier-1 and tier-2 or within the DCs in tier-2 (*i.e.*, between two CORDs). The cost is further doubled for the next top tier, and so on. In the multi-tier topologies, we considered four tiers, and we neglected the communication cost within a DC. In this

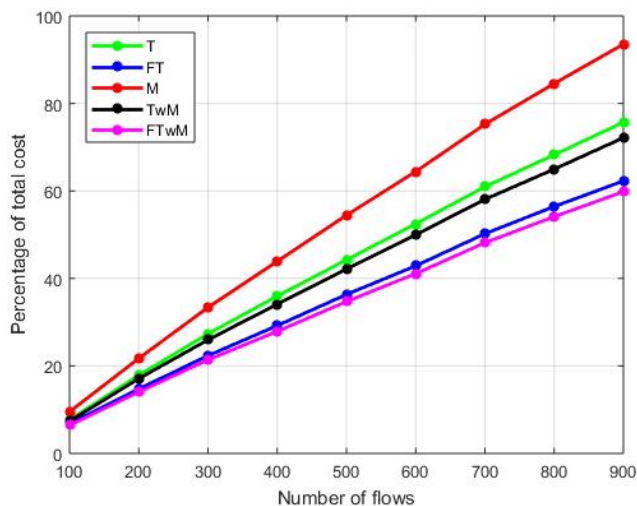


FIGURE 6. Total cost consumption of different networks per number of flows.

MATLAB experiment, we ran our algorithm ten times for each setup and calculated the average results and presented them in a normalized form.

B. HEURISTIC PERFORMANCE ANALYSIS

Here we demonstrate the performance of our heuristic algorithm on multiple types of single and multi-tier topologies. In this evaluation, we considered the scenario where the network functions are distributed over different DCs. For each service of the flow's service chain, our algorithm will find a more suitable VNF for placement so as to minimize the cost.

1) TOTAL COST ANALYSIS

Figure 6 presents the total cost of different networks. The total cost is the sum of communication and computing cost of a network after each iteration. We discuss the communication and computing cost in detail in the following subsections. As the result in Figure 6 shows, hierarchical topologies are more cost-efficient compared to a horizontal topology. In particular, the partial mesh topology has the highest cost compared to all other topologies. Figure 6 shows that the partial mesh topology consumes 15-20%, 20-25%, 25-35%, and 30-35% more cost than the T, TwM, FT, and FTwM topologies, respectively. However, in multi-tier topologies, both horizontal and vertical communications help to decrease costs compared to only vertical communication. For example, as shown in Figure 6, TwM and FTwM decrease costs more than the tree and fat-tree topology, respectively.

2) COMMUNICATION COST VS. COMPUTING COST

The communication and computing cost comparisons of different networks in Figure 7 shows that the ratio of communication cost is relatively more than the computing cost in all networks. Since the DCs of our networks are NFV-enabled, we can consolidate multiple flows demanding similar service into a single VNF, reducing computing costs significantly.

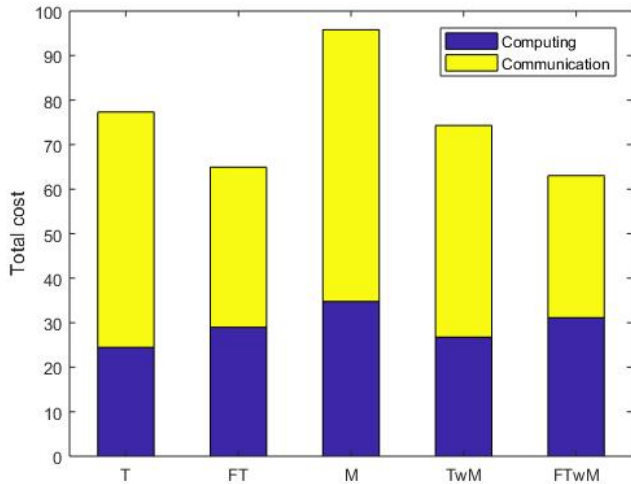


FIGURE 7. Comparison of communication and computing costs of different networks.

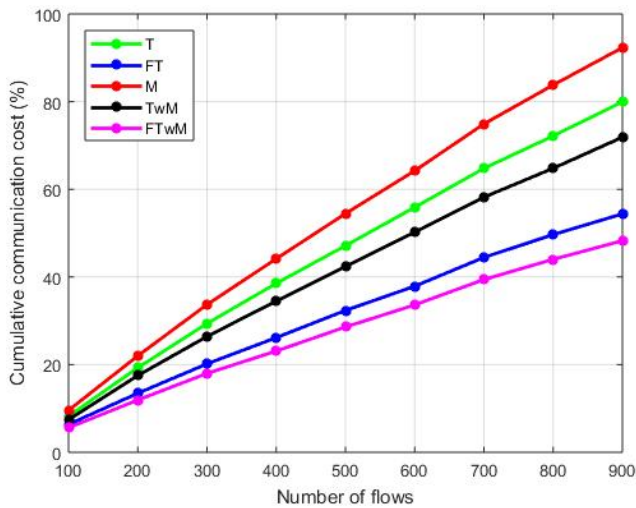


FIGURE 8. Cumulative communication cost per number of flows of different networks.

However, the communication cost depends solely on the number of packets transferred between the DCs. Hence, the ratio of communication costs is relatively higher compared to computing costs.

3) COMMUNICATION COST ANALYSIS

The detailed communication cost analysis is presented in Figures 8, 9, 10 and 11. Figure 8 shows the percentage of cumulative communication cost per flow. The communication cost in multi-tier topologies is relatively low compared to single-tier topology. In particular, the T, FT, TwM, and FTwM topologies save by 10-15%, 35-40%, 15-25%, and 40-50%, respectively, of communication cost compared to partial mesh topology. However, between the multi-tier topologies, the communication cost of multi-tier, multiple parents topologies (FT, FTwM) is less than the multi-tier, single parent topologies (T, TwM). For example, Figure 8

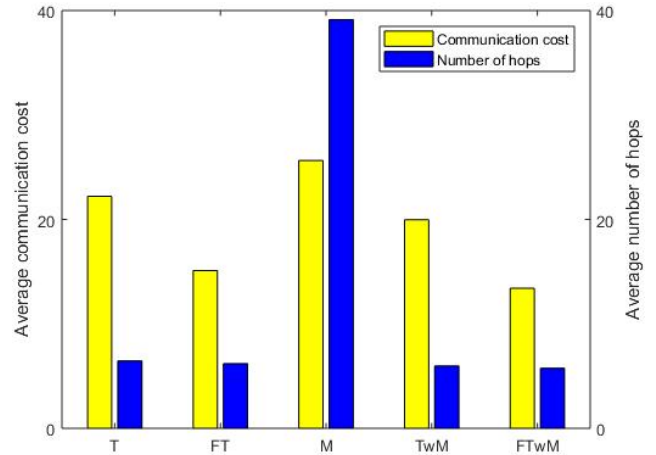


FIGURE 9. Comparison of average communication cost with number of hops.

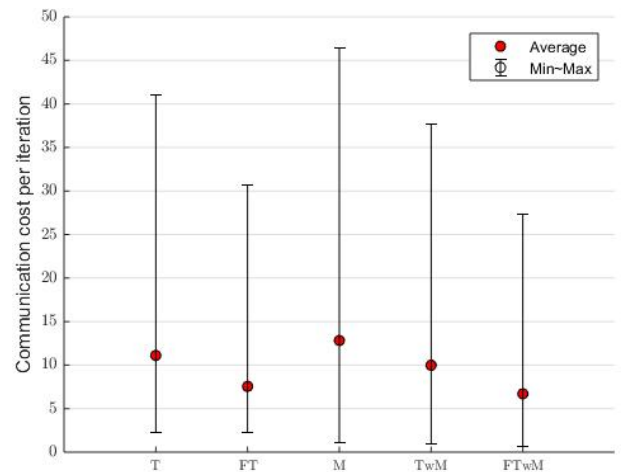


FIGURE 10. Max~min difference of communication cost per iteration.

shows that FT and FTwM reduce the communication cost by 20-30% compared to T and TwM, respectively, because, in FT and FTwM, an alternative path always exists in the same tier for the placement of VNF, which reduces the communication cost. However, if we do not have an appropriate DC for the placement of the VNF, we have to move to the next tier above. Again, as we described in the experimental setting, the communication cost is doubled when the packet flows from one tier to the next tier above. The results also reveal that the cost in vertical with horizontal communication being relatively less than only vertical communication. Figure 8 shows that the communication cost of TwM and FTwM is 5-15% less compared to T and FT, respectively, because, in TwM and FTwM, an option always exists to avoid the placement of VNFs in the top tiers to reduce communication cost.

Because communication costs depend on how many packets are transferred from one DC to another and the location of the sender and receiver DCs, the number of hops the packets

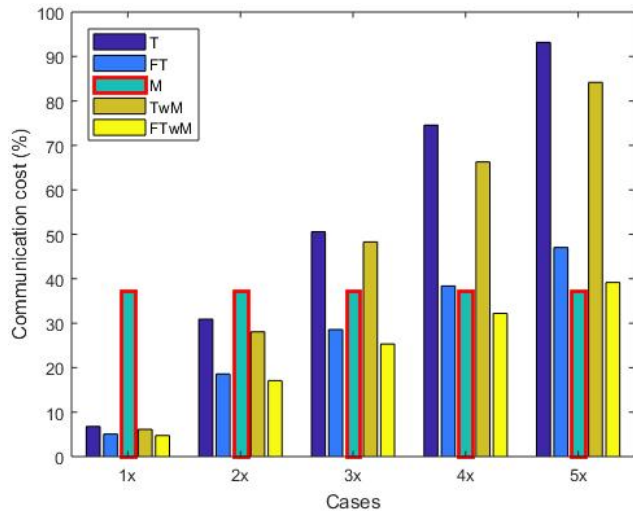


FIGURE 11. Change in percentage of communication cost with different unit values of communication cost from tier-1 to the next tier above.

of flow travel plays a key role in the communication cost estimates. Figure 9 gives the comparison of communication cost and number of hops. The left y-axis shows the average communication cost, and the right y-axis shows the average number of hops of the flows. The result shows that in all networks as the number of hops increases the cost also increases. The number of hops in the partial mesh is very high as it only follows horizontal communication. Although the communication cost in the partial mesh topology is highest compared to other topologies, its unit communication cost does however not change as in other networks, since all nodes of the partial mesh topology are in tier-1, and communication cost within tier-1 is the lowest.

Figure 10 shows the average-max-min chart of communication cost per iteration. The average communication cost in multi-tier, multiple parent topologies (FT and FTwM) is less than multi-tier, single parent topologies (T and TwM). However, the average communication cost of the partial mesh topology is relatively high compared to all multi-tier topologies. Similarly, the *max~min* difference of communication cost per iteration in the partial mesh topology is the highest and lowest in FTwM.

All previous results on communication cost are from cases where the communication cost of tier-2 is twice the communication cost of tier-1, and the value is doubled from one tier to the next tier above. Figure 11 shows a unique result where we have considered five different values of the unit cost of communication from tier-1 to the next tiers above. Tier-1 cost is equal in all case; however, it changes from tier-2 and upwards. In *case-1x*, we considered the communication cost to be equal in all tiers. *Case-2x* is the general case where the cost grows as 2x, 4x, 8x, and so on. In *case-3x*, it grows 3x, 6x, 12x, and so on; in *case-4x*: 4x, 8x, 16x, and so on; and in *case-5x*: 5x, 10x, 20x, and so on, in tier-2, tier-3, tier-4, and so on, respectively. As the results show, the communication cost of

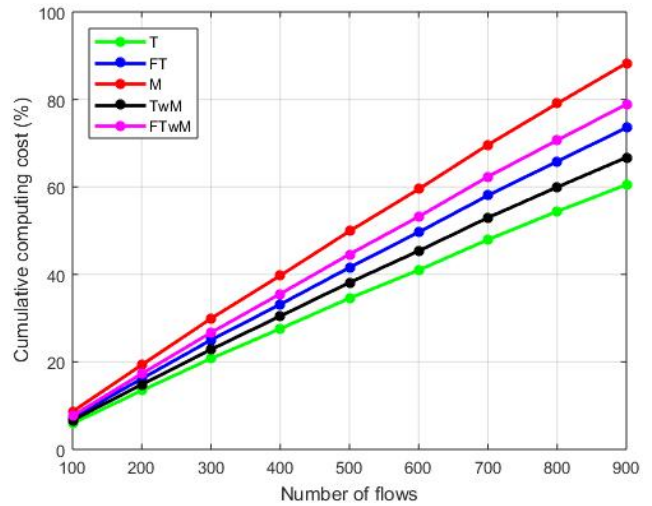


FIGURE 12. Cumulative computing cost per number of flows of different networks.

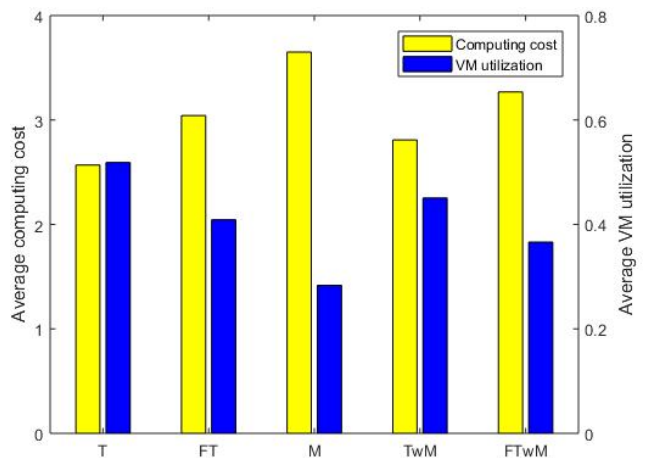


FIGURE 13. Comparison of average computing cost with average utilization of VMs.

partial mesh topology is the same in all 5 cases as all the nodes in this topology are in tier-1. The cost of T, FT, TwM, and FTwM topologies, in case-1x, and case-2x, is relatively less than partial mesh. In case-3x, the cost of T and TwM exceeds the limit of partial mesh, but the cost of FT and FTwM are still relatively less than partial mesh. The cost of FT and FTwM come closer to the cost of the partial mesh in case-4x and case-5x, respectively. The results in Figure 11 shows case-2x is the threshold for multi-tier, single parent topologies, and case-4x and case-5x are the threshold for FT and FTwM, respectively. The multi-tier multiple parents' topologies (FT, FTwM) save more communication costs, and fat-tree with partial mesh topology saves maximum costs compared to all other topologies considered in this work.

4) COMPUTING COST ANALYSIS

The detailed computing cost analysis is presented in Figures 12, 13, 14 and 15. Figure 12 shows the percentage of

cumulative computing cost per flow. The computing cost in multi-tier topologies is relatively low compared to single-tier topologies. However, computing costs in multi-tier, single parent topologies (T, TwM) is less than the multi-tier, multiple parent topologies (FT, FTwM). Figure 12 shows, both FT and FTwM have 10-15% higher computing cost than T and TwM, respectively, because in T and TwM, there is no alternative path for the flow. The VNFs have to be placed in the only parent DC or have to move to the next top tier. If the function is available on the parent node, all the VNFs will be placed in that node until the DC exceeds its capacity. This helps increase the utilization of the VMs and reduces the computing cost. The result in Figure 12 also shows that the computing cost in vertical communication is relatively less than in vertical with horizontal communication. In particular, T and FT save 5-10% on computing cost compared to TwM and FTwM, respectively. In TwM and FTwM, due to the horizontal connectivity between the siblings, an option always exists to place VNF in the same tier to reduce the communication cost by preventing some flows to proceed to the top tier. This reduces the utilization of the VMs and increases the computing cost.

Since the computing cost depends on how many VMs are utilized for flow processing and for how long, the utilization of the active VMs plays a key role in estimating computing cost, because, if the utilization of the VMs decreases, the number of VMs to process the flows will increase causing the computing cost to increase. Here, the utilization of VM means that if the VM has the capacity to process a maximum of five flows together and it is processing three flows, then the utilization of VM is 60%. Figure 13 shows a comparison of computing costs and utilization of the VMs. The left y-axis is the average computing cost, and the right y-axis is the average utilization of VMs in the DCs. The results for all networks show that as the number of utilization of VMs decreases, the computing cost increases, because of the decrease in utilization, the number of active VMs increase. The computing cost of tree topology is lowest when its utilization of VMs is highest among all topologies, whereas the computing cost of partial mesh topology is highest when its utilization is lowest.

Figure 14 shows the average-max-min chart of computing cost per iteration. The average computing cost in multi-tier, single parent topologies (T, TwM) is less than multi-tier, multiple parent topologies (FT, FTwM). However, the average computing cost of the partial mesh topology is relatively high compared to all multi-tier topologies. Similarly, the *max~min* difference of computing cost per iteration in the partial mesh topology is the highest and lowest in the tree topology.

Figure 15 shows computing cost variation with the change of flow-sharing limits in the VNFs. Here we have considered all three values (5, 10 and 20) separately and compared the changes in computing cost. The results show that in all topologies with an increase in the flow-sharing limits of the VNFs the computing cost decreases: when VMs limits increase, more flows with similar service function demands

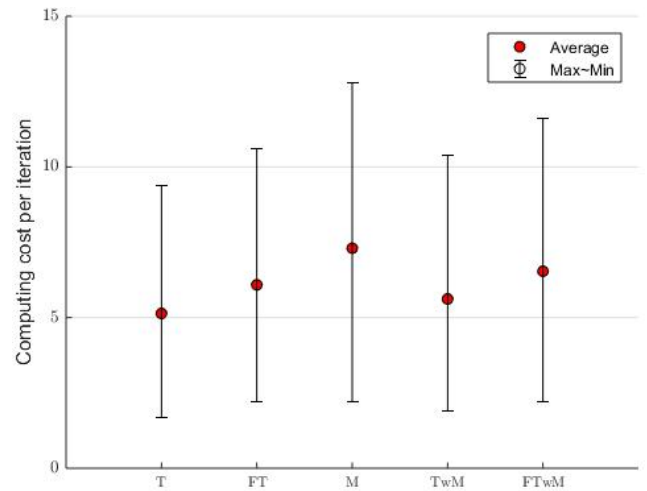


FIGURE 14. *Max~min* difference of computing cost per iteration.

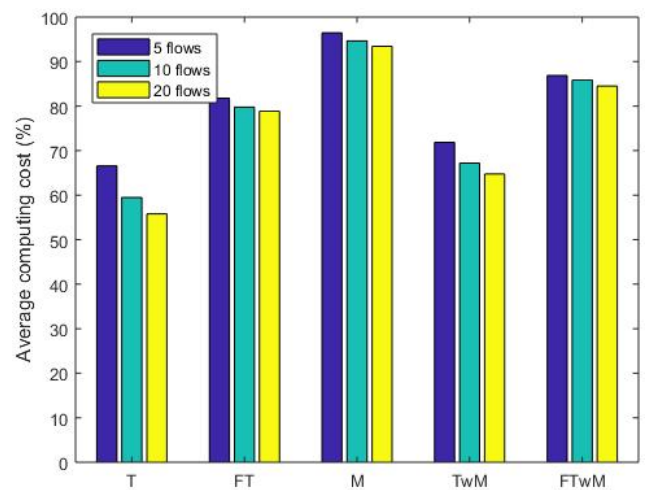


FIGURE 15. Percentage of computing cost decreases with the increases in VM's flow-sharing limit.

can share a common VM for their packet processing. As a result, the number of active VMs in the network is reduced, causing a decrease in computing cost.

5) FLOW-DROP WITH HOP CONSTRAINTS

Here we estimate the end-to-end delay in terms of the number of hops from the ingress to the egress node, where each node represents one datacenter, *i.e.*, in this experiment, the inter-DC communication delay has been taken into consideration and intra-DC communication delay has been neglected. We have coined the term flow-drop for delay estimates.

Flow-drop means is used for when a flow is unable to reach its destination as a result of time-to-live (TTL) or hop limit. The TTL is a mechanism that limits the lifespan of a flow in the network within which the flow has to reach its destination. If it fails to reach the destination within the time limit, we consider it a *flow-drop*. Figure 16 shows that as the TTL increases the amount of *flow-drop* decreases.

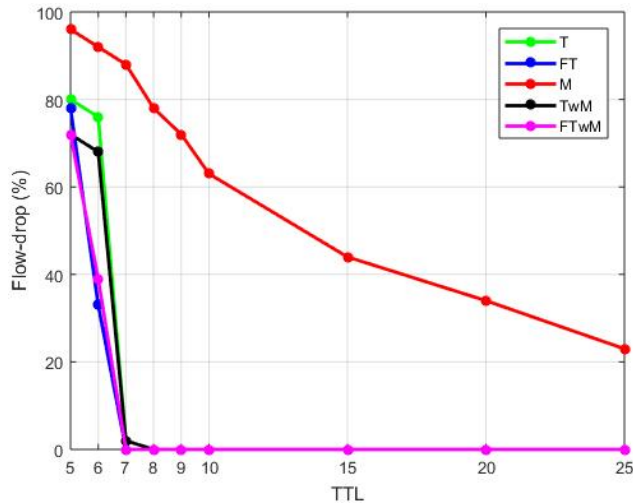


FIGURE 16. The percentage of flow-drop with hop constraints.

It also decreases faster in multi-tier topologies because of their hierarchical architecture where we can reach any destination with few hops. However, in a single-tier topology, with its horizontal architecture, the hop count is relatively higher than others. Although the *flow-drop* decreases in a partial mesh network, it is not as sharp a drop as in the other four networks.

C. PERFORMANCE COMPARISON OF THE HEURISTIC

In this section, we will compare the performance of the VHC algorithm with some other algorithms. We compare our VHC algorithm with random [43] and first-fit [44] placement algorithms. In the random placement algorithm (RAN), we randomly select a node with sufficient capacity for the placement of the function to process the packets of the flows. In the first-fit placement algorithm (FF), we select the first node with available capacity for the placement of the function.

Figure 17 shows the total cost comparison of the VHC algorithm with other algorithms for all five topologies that we have considered in our experiments where the ‘-comm’ and ‘-comp’ represent the communication cost and computing cost, respectively. The result shows in all algorithms percentage of total cost in partial mesh topology is very high compared to all other topologies due to its single-tier architecture. As in the partial mesh topologies, the number of hop count from source node to destination node is relatively very high which causes to increase the communication cost. But in all other four multi-tier topologies (T, FT, TwM, and FTwM), the difference of the communication cost may not be much due to nearly equal hop counts as shown in Figure 9. However, the computing cost in our HVC algorithm is low compared to other algorithms. In the VHC algorithm, the flow will be placed in the active VMs and the VMs try to accommodate more flows for maximum utilization of its available capacities. Whereas in FF, the flow will be placed in the first available VM having required function and in RAN, the placement is

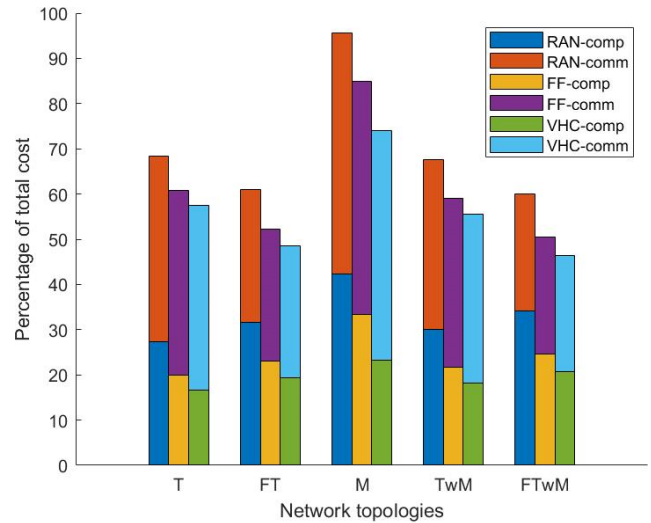


FIGURE 17. Performance comparison of the VHC algorithm with other existing algorithms.

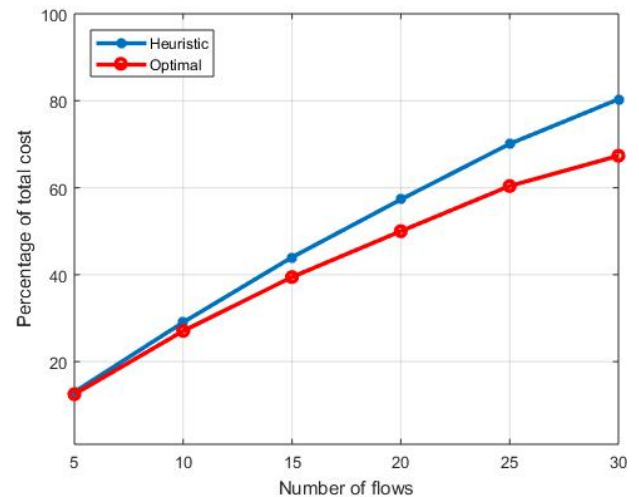


FIGURE 18. Performance comparison between heuristic and optimal solutions.

done randomly. As a result, the utilization of VMs in FF and RAN decreases and they used more VMs to process the flow which results in high computing cost.

D. OPTIMALITY OF THE HEURISTIC

In this section, we check the optimality of the heuristic. We implemented the optimization problem in AMPL (A Modeling Language for Mathematical Programming) [41] to solve the formulation. However, when a network is large, its time cost factor is not acceptable. At the same time, we need to discuss the performance of the heuristic solution in large networks. Simulations of a small network were run to compare the performance of the optimal solution and the heuristic solution. We conducted this experiment on a partial mesh topology. The result in Figure 18 shows that with low traffic, the performance of the heuristic is closer to the optimal results. Although with an increase in traffic, the total

cost consumption of the network increases compared to the optimal result, the computation time in the optimal solution is relatively much higher, at least 70–80 times that of the heuristic solution.

VI. CONCLUSION

In this paper, we analyzed the connectivity issue in an inter-DC network. We proposed a generic inter-DC NFV network architecture and estimated the traffic rate between the DCs. We formulated an optimization problem to minimize the total cost, *i.e.*, communication and computing cost of the network, which we proved to be NP-hard. We proposed a heuristic algorithm for vertical and horizontal communication between DCs with VNF placement and service chaining in different network types. The heuristic results show that both vertical and horizontal connectivity is required to reduce costs. Vertical connectivity helps reduce the computing cost significantly compared to horizontal connectivity. However, horizontal connectivity with vertical connection plays a significant role in reducing communication costs.

In particular, FTwM and TwM have 5-15% less communication cost than FT and T, respectively. FTwM reduces communication cost by 40-45%, 30-35%, 20-25%, and 5-10% compared to M, T, TwM, and FT, respectively, as a result of its multi-tier, multiple parents' horizontal and vertical connectivity, which helps increase communication with fewer hop counts. However, the tree topology reduces computing cost by 25-30%, 15-20%, 10-15%, and 5-10% compared to M, FTwM, FT, and TwM, respectively, as a result of its multi-tier, single parent vertical connectivity which maximizes the utilization of the VMs of the DCs. Although the communication cost in tier-1 is reduced, a partial mesh network still has a higher communication because of the high hop count from the ingress to the egress node, and it also has a higher computing cost because of poor utilization of VMs. In terms of total cost, which includes both communication cost and computing cost, FTwM saves more compared to all topologies.

In summary, the results demonstrate that although vertical connectivity helps to reduce cost compared to horizontal connectivity, it is better to consider horizontal connectivity between siblings in multi-tier architecture to reduce cost even further.

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] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 7–13.
- [3] V. Eramo, E. Mucci, 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.
- [4] A. U. Rehman, R. L. Aguiar, and J. P. Barraca, "Network functions virtualization: The long road to commercial deployments," *IEEE Access*, vol. 7, pp. 60439–60464, 2019.
- [5] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, "Central office re-architected as a data center," *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 96–101, Oct. 2016.
- [6] *CORD: Re-Inventing Central Offices for Efficiency and Agility*. Accessed: Jun. 2019. [Online]. Available: <https://www.opencord.org/>
- [7] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.
- [8] White Paper. *NFV Everywhere: A Micro-Datacenter in-a-Box for the Network Edge*. Accessed: Jun. 2019. [Online]. Available: <https://networkbuilders.intel.com/docs/nfv-everywhere-a-micro-datacenter-in-a-box-for-the-network-edge.pdf>
- [9] Telecommunications Industry Association (TIA) Position Paper, "Edge data centers," Tech. Rep. 2018. Accessed: Jun. 2019. [Online]. https://www.tiaonline.org/wp-content/uploads/2018/10/TIA_Position_Paper_Edge_Data_Centers-18Oct18.pdf
- [10] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.
- [11] B. Kar, E. H.-K. Wu, and Y.-D. Lin, "Energy cost optimization in dynamic placement of virtualized network function chains," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 372–386, Mar. 2018.
- [12] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 543–553, Sep. 2017.
- [13] 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.
- [14] Y.-D. Lin, C.-C. Wang, C.-Y. Huang, and Y.-C. Lai, "Hierarchical CORD for NFV datacenters: Resource allocation with cost-latency tradeoff," *IEEE Netw.*, vol. 32, no. 5, pp. 124–130, Sep. 2018.
- [15] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Sep. 2013.
- [16] C.-C. Wang, Y.-D. Lin, J.-J. Wu, P.-C. Lin, and R.-H. Hwang, "Towards optimal resource allocation of virtualized network functions for hierarchical datacenter," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1532–1544, Dec. 2018.
- [17] W. Cerroni, L. Foschini, G. Y. Grabarnik, L. Shwartz, and M. Tortonesi, "Estimating delay times between cloud datacenters: A pragmatic modeling approach," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 526–529, Mar. 2018.
- [18] *Trai Telecom Subscription Data as on 30th*. Press Release, New Delhi, India, Sep. 2017. [Online]. Available: www.trai.gov.in
- [19] L. Gu, D. Zeng, S. Guo, Y. Xiang, and J. Hu, "A general communication cost optimization framework for big data stream processing in geo-distributed datacenters," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 19–29, 2016.
- [20] I. Karimov, M. Kamilov, E. Park, J. Song, H. Kim, and S. Han, "Managing alternative parent peers for providing fast reconnection between peers," in *Proc. 11th IEEE Int. Conf. Adv. Commun. Technol.*, vol. 3, Feb. 2009, pp. 1566–1571.
- [21] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (DCN): Infrastructure and operations," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 640–656, 1st Quart., 2017.
- [22] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity datacenter network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.
- [23] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "Exploiting congestion games to achieve distributed service chaining in NFV networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 407–420, Feb. 2017.
- [24] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "A game theoretic approach for distributed resource allocation and orchestration of software-defined networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 721–735, Mar. 2017.
- [25] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 533–546, Sep. 2016.
- [26] J. Sun, G. Zhu, G. Sun, D. Liao, Y. Li, A. K. Sangaiah, M. Ramachandran, and V. Chang, "A reliability-aware approach for resource efficient virtual network function deployment," *IEEE Access*, vol. 6, pp. 18238–18250, 2018.

- [27] D. Krishnaswamy, R. Kothari, and V. Gabale, "Latency and policy aware hierarchical partitioning for NFV systems," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2015, pp. 205–211.
- [28] X. Wang, X. Wang, K. Zheng, Y. Yao, and Q. Cao, "Correlation-aware traffic consolidation for power optimization of datacenter networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 992–1006, Apr. 2016.
- [29] C. K. Dominicini, G. L. Vassoler, L. F. Meneses, R. S. Villaca, M. R. N. Ribeiro, and M. Martinello, "VirtPhy: Fully programmable NFV orchestration architecture for edge data centers," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 817–830, Dec. 2017.
- [30] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu, "A first look at inter-data center traffic characteristics via yahoo! Datasets," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1620–1628.
- [31] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 475–488, Mar. 2018.
- [32] B. Kantarci, L. Foschini, A. Corradi, and H. T. Mouftah, "Inter-and-intra data center VM-placement for energy-efficient large-scale cloud systems," in *Proc. IEEE Globecom Workshops*, Dec. 2012, pp. 708–713.
- [33] M. Bouet and V. Conan, "Mobile edge computing resources optimization: A geo-clustering approach," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 2, pp. 787–796, Jun. 2018.
- [34] M. Obadia, M. Bouet, V. Conan, L. Iannone, and J.-L. Rougier, "Elastic network service provisioning with VNF auctioning," in *Proc. 28th Int. Teletraffic Congr. (ITC)*, Sep. 2016, pp. 340–348.
- [35] M. Gharbaoui, C. Contoli, G. Davoli, G. Cuffaro, B. Martini, F. Paganelli, W. Cerroni, P. Cappanera, and P. Castoldi, "Experimenting latency-aware and reliable service chaining in next generation Internet testbed facility," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2018, pp. 1–4.
- [36] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *J. Lightw. Technol.*, vol. 34, no. 14, pp. 3330–3341, Jul. 15, 2016.
- [37] R. Ricci, C. Alfeld, and J. Lepreau, "A solver for the network testbed mapping problem," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 65–81, Apr. 2003.
- [38] R. McGeer, D. G. Andersen, and S. Schwab, "The network testbed mapping problem," in *Proc. Int. Conf. Testbeds Res. Infrastruct.* Berlin, Germany: Springer, 2010, pp. 383–398.
- [39] J. R. Current and J. E. Storbeck, "Capacitated covering models," *Environ. Planning B, Planning Design*, vol. 15, no. 2, pp. 153–163, Jun. 1988.
- [40] E. Hernandez-Valencia, S. Izzo, and B. Polonsky, "How will NFV/SDN transform service provider opex?" *IEEE Netw.*, vol. 29, no. 3, pp. 60–67, May 2015.
- [41] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. 2nd ed. Belmont, CA, USA: Duxbury Press, 2003.
- [42] M. Sipser, *Introduction to the Theory of Computation*, 2nd ed. Boston, MA, USA: Cengage Learning, 2012.
- [43] C. Rose and M. Hluchyj, "The performance of random and optimal scheduling in a time-multiplex switch," *IEEE Trans. Commun.*, vol. COM-35, no. 8, pp. 813–817, Aug. 1987.
- [44] X. Tang, Y. Li, R. Ren, and W. Cai, "On first fit bin packing for online cloud server allocation," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2016, pp. 323–332.



BINAYAK KAR (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University (NCU), Taiwan, in 2018. He was a Postdoctoral Research Fellow in computer science with National Chiao Tung University (NCTU), Taiwan, from 2018 to 2019. He is currently an Assistant Professor of computer science and information engineering at the National Taiwan University of Science and Technology (NTUST), Taiwan. His research interests include network softwarization, cloud/edge/fog computing, optimization, queueing theory, and network security.



ERIC HSIAO-KUANG WU (Member, IEEE) received the B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1989, and the M.S. and Ph.D. degrees in computer science from the University of California at Los Angeles, in 1993 and 1997, respectively. He is currently a Professor of computer science and information engineering with the Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan. His research interests include wireless networks, mobile computing, and broadband networks.



YING-DAR LIN (Fellow, IEEE) is currently a Chair Professor of computer science at National Chiao Tung University (NCTU), Taiwan. He received the Ph.D. degree in computer science from the University of California at Los Angeles (UCLA), in 1993. He was a Visiting Scholar at Cisco Systems, San Jose, from 2007 to 2008, the CEO at Telecom Technology Center, Taiwan, from 2010 to 2011, and the Vice President of the National Applied Research Labs (NARLabs), Taiwan, from 2017 to 2018. He co-founded L7 Networks Inc., in 2002, later acquired by D-Link Corporation. He also founded and directed Network Benchmarking Lab (NBL), in 2002, which reviewed network products with real traffic and automated tools, also an approved test lab of the Open Networking Foundation (ONF), and O'Prueba Inc., a spin-off company, in 2018. He published a textbook *Computer Networks: An Open Source Approach* (with Ren-Hung Hwang and Fred Baker) (McGraw-Hill, 2011). His research interests include network security, wireless communications, and network softwarization. His work on multi-hop cellular was the first along this line, and has been cited over 900 times and standardized into the IEEE 802.11s, the IEEE 802.15.5, the IEEE 802.16j, and 3GPP LTE-Advanced. He is an IEEE Distinguished Lecturer from 2014 to 2017, and an ONF Research Associate from 2014 to 2017. He received the K. T. Li Breakthrough Award, in 2017, and the Research Excellence Award, in 2017 and 2020. He has served or is serving on the editorial boards of several IEEE journals and magazines, and is the Editor-in-Chief of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS (COMST).

...