Joint VNF Placement and Multicast Traffic Routing in 5G Core Networks

Omar Alhussein*, Phu Thinh Do*, Junling Li*, Qiang Ye*, Weisen Shi*,
Weihua Zhuang*, Xuemin (Sherman) Shen*, Xu Li†, and Jaya Rao†
*Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
†Huawei Technologies Canada Inc., Ottawa, ON, Canada
Email: {oalhussein, pt3do, j742li, q6ye, w46shi, wzhuang, sshen}@uwaterloo.ca, {Xu.LiCA, jaya.rao}@huawei.com

***Abstract*—The software defined networking (SDN) enabled network function virtualization (NFV) architecture emerges as a cost-effective solution for service customization in fifth generation (5G) networks. In this paper, a joint traffic routing and virtual network function (VNF) placement problem is studied for a multicast service request accommodated over a physical substrate network, where the multipath traffic routing is considered between embedded VNFs. The joint problem is formulated as a mixed integer linear programming (MILP) problem to minimize the provisioning cost of both VNFs and links, under the physical network resource constraints, flow conservation constraints, and VNF placement rules. Since the problem is NP-hard, low complexity heuristic algorithms, with the consideration of both the single-path and multipath routing cases, are proposed to determine an efficient solution. Simulation results are presented to demonstrate the effectiveness and accuracy of the proposed heuristic algorithms especially for a large-size network.**

***Index Terms*—5G networks, SDN, NFV, multicast services, VNF chains, VNF embedding, MILP, service customization.**

I. Introduction

The fifth generation (5G) networks [1] are envisioned to accommodate a dramatical growing demand for mobile communications at highly increased rates, due to the augmented types of new mobile broadband services with high traffic volume (e.g., video conferencing, virtual reality, and intelligent transportation systems) and the advent of miscellaneous and ubiquitous Internet-of-Things (IoT) devices [2]. However, the conventional network architecture is evolving in a cost-ineffective way. In the wireless network domain, more and more small-cell base stations (SBSs) are deployed underlaying the coverages of macro-cells to exploit the spatial multiplexing gain of radio resources, while various types of network servers providing specific functionalities (e.g., domain name system (DNS), transcoding, and intrusion detection system (IDS)) are placed in the core network for customized end-to-end (E2E) service provisioning [3], [4]. The placement of an increasing number of network elements inevitably expands both capital and operational expenditure (CapEx and OpEx).

To better utilize the network resources and reduce the infrastructure deployment cost, software-defined networking (SDN) [5] and network function virtualization (NFV) [6], [7] are two complementary innovative technologies for the evolvement of a new networking paradigm. The SDN decouples the control functions from network switches/servers to a logically centralized SDN control plane, while NFV abstracts network (or service) functions from each server as software instances, referred to as virtual network functions (VNFs), that can be flexibly placed at any general-purpose commodity server in the network. The SDN-enabled NFV architecture achieves service customization in a cost-effective way. With global network information (e.g., physical network topology, resources, and traffic statistics), the SDN control module determines the best routing path for traffic traversing a chain of VNFs to fulfill customized requirements of different service requests, and the NFV control module embeds different sets of VNFs at appropriate network locations (i.e., network servers) for high processing resource utilization and low deployment cost [1]. This process is referred to as *VNF chaining and embedding* [8].

In the core network, traffic from a service request is required to traverse a sequence of VNFs before arriving at the destination node. For example, a secured DNS request passes through a firewall function and DNS function sequentially for obtaining the IP address of an intended network server. Therefore, a service request can be described by a *VNF chain*, which comprises certain source and destination nodes and a sequence of VNFs interconnected by virtual links that have to be traversed with certain quality-of-service (QoS) requirement. To further improve resource utilization, service providers (SPs) are increasingly demanding service requests with multicast traffic, also referred to as *multicast service requests (or multicast VNF chains)*, to provide bandwidth efficiency through the use of packet replication at network edges[1] [9].

One of the fundamental research problems under the SDN-enabled NFV architecture is how to embed multicast service requests onto the physical substrate network, where VNFs are placed at commodity servers and virtual links are assigned to physical links (or paths) for traffic routing [10]–[12]. Instantiating large number of VNF instances at several commodity servers can achieve balanced traffic load at the expense of higher function provisioning cost, whereas fewer VNF instantiations can reduce the overall function provisioning cost with less load balancing and inefficient network resource exploitation. Therefore, our objective is to determine the optimal locations for VNF placement and the optimal

[1]For the case that multiple destination nodes in the core network require the same information contents, the source node transmits each packet only one time, and then packet replication occurs at edges close to the destinations.

routing paths among the embedded VNFs with load balancing, while minimizing the cost of provisioning a resultant multicast VNF chain. Existing studies present Steiner tree-based approaches [10], [13] for multicast routing, which cannot be directly extended to the joint VNF placement and routing problem, since some VNFs have to be deployed between terminal nodes. In [11], a joint VNF placement and routing problem is studied to minimize the link and server cost, for a scenario with a single server. In practice, the multicast streams may have to pass through multiple geographically dispersed servers. In [12], a joint VNF placement, routing, and spectrum assignment framework is considered for a fibre optical network. Similarly, it is assumed that an VNF is used for each source-destination pair. In [14], under the assumption that there are multiple servers that can host all types of VNFs, every source/destination pair needs to pass through only one server for all VNFs before arriving at each destination. In practice, some servers can only host several types of VNFs in a VNF chain, either because of limitations in available processing resources or subscription to only a subset of VNFs.

In this paper, we consider a joint VNF placement and traffic routing problem for a multicast service request to minimize the function and link provisioning cost, under the physical processing and bandwidth resource constraints, flow conservation constraints, and VNF placement rules. The problem is formulated as a mixed integer linear programming (MILP) problem. As the 5G core networks are geographically deployed in a large scale, more flexible embedding is considered in our problem formulation, where we allow for one-to-many and many-to-one VNF mappings. That is, many VNFs can be hosted at one commodity servers if allowable, and one VNF can be deployed in different commodity servers as VNF instances. In doing so, we do not impose any constraints on the locations of the multicast replication points, and the deployment of VNF instances can occur both before and after the replication points in the multicast topology. Our formulated problem also incorporates both single path and multipath traffic routing between the embedded VNFs. Since the formulated MILP problem is NP-hard, we devise low-complexity heuristic algorithms to obtain efficient solutions to the problem, based on a key-node preferred minimum spanning tree (KPMST) approach.

The rest of the paper is organized as follows. Section II presents the system model under consideration, which includes the representation of the physical network, VNFs, and multicast service requests. Section III addresses the joint VNF placement and routing problem for multicast services with multipath routing. Section IV proposes simple heuristic algorithms to address the complexity of the resultant MILP formulation. Simulation results are presented in Section V, and concluding remarks are drawn in Section VI.

II. System Model

A. Physical Substrate Network

Consider a physical substrate network, $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the set of nodes and links (as shown in Fig. 1). The nodes can be switches (represented by set $\mathcal{S}$) and commodity servers or data center (DC) nodes (namely NFV nodes, represented by set $\mathcal{M}$), *i.e.*, $\mathcal{N} = \mathcal{S} \cup \mathcal{M}$. Switches are capable of forwarding and replicating traffic, and NFV nodes are capable of hosting and operating VNFs. We assume that each NFV node has a forwarding capability, and has available CPU resources $C(n)$, $n \in \mathcal{M}$. Moreover, an NFV node is capable of provisioning a number of VNFs simultaneously as long as the available processing resources satisfy VNF processing requirements. Each physical link has a limited bandwidth $B(l)$, $l \in \mathcal{L}$.

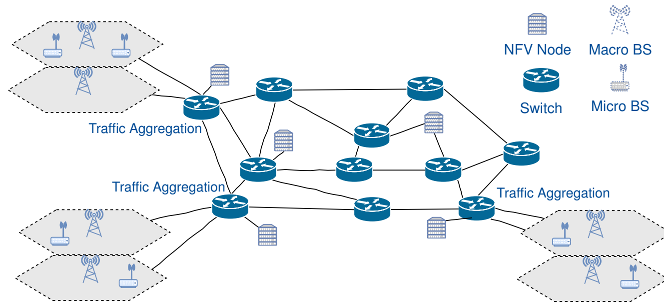

Fig. 1. An illustration of physical substrate network.

B. VNFs

We represent all the VNF types by set $\mathcal{P}$, where a specific type, $p \in \mathcal{P}$, resembles some virtual functionality (e.g., IDS, compression, proxy, and LTE packet gateway). We further associate NFV node $n \, (\in \mathcal{M})$ with a set of admittable VNF types using an indicator function $U(n, i) \in \{0, 1\}$, where $U(n, i) = 1$ if NFV node $n \, (\in \mathcal{M})$ can admit function $f_i \, (\in \mathcal{P})$.

C. Multicast VNF Chains

A multicast service, $r \in \mathcal{R}$, is described by a multicast VNF chain, represented by a weighted acyclic directed graph,

$$S_r = (s, \mathcal{D}, f_1, f_2, \dots f_{|\mathcal{V}|}, \overline{d}_r) \tag{1}$$

where s and $\mathcal{D}$ represent the source node and the set of destinations, $\mathcal{V} = \{f_1, f_2, \dots f_{|\mathcal{V}|}\}$ represents the set of functions that have to be traversed in an ascending order for every source/destination pair, and $\overline{d}_r$ is the data rate. We assume that function f_i, $i \in \{1, \dots, |\mathcal{V}|\}$, requires computing resources that is linearly proportional to the incoming data rate demand.

III. Problem Formulation

In order to formulate joint multipath-enabled VNF placement and routing for multicast services, we assume that there exists up to J multicast trees to deliver one multicast service from the source to destinations. In the special case of $J = 1$, the problem reduces to the joint problem with single path routing. Each tree emanates from the source and passes through the same set of traversed functions to the destinations.

Define binary variable $x_{li}^j \in \{0, 1\}$, where $x_{li}^j = 1$ indicates link $l \, (\in \mathcal{L})$ is used for forwarding traffic in multicast tree j

from f_i to f_{i+1} where $i \in \{1, \dots, |\mathcal{V}-1|\}$, $x_{l0}^j = 1$ indicates link l carries traffic from s to f_1, and $x_{l|\mathcal{V}|}^j = 1$ indicates link l carries traffic from $f_{|\mathcal{V}|}$ to any destination node $t\,(\in \mathcal{D})$; Define $y_{lit}^j \in \{0,1\}$, where $y_{lit}^j = 1$ indicates link l is used to direct traffic in multicast tree j from f_i to f_{i+1} for destination t, $y_{l0t}^j = 1$ indicates link l is used to direct traffic in tree j from s to f_1 for destination t, and $y_{l|\mathcal{V}|t}^j = 1$ indicates link l directs traffic in tree j from $f_{|\mathcal{V}|}$ to destination t. With the definitions of $\boldsymbol{x} = \{x_{li}^j\}$ and $\boldsymbol{y} = \{y_{lit}^j\}$, we have

$$y_{lit}^j \le x_{li}^j, \quad l \in \mathcal{L}, i \in \mathcal{S}_0^{|\mathcal{V}|}, j \in \mathcal{S}_1^J, t \in \mathcal{D} \tag{2}$$

where $\mathcal{S}_m^n \triangleq \{m, m+1, \dots, n\}$ with $m, n \in \mathbf{Z}_+$.

Let $z_{ni} \in \{0,1\}$ denote whether an *instance* of f_i is deployed on node n where $n \in \mathcal{N}, i \in \mathcal{S}_1^{|\mathcal{V}|}$. Under the assumption that packets traversing multicast trees are processed by the same instantiated functions, let $u_{nit} \in \{1,0\}$ indicate whether or not an instance of f_i is hosted at node n for traffic of all activated trees towards destination t, where $n \in \mathcal{N}, i \in \mathcal{S}_1^{|\mathcal{V}|}$, $t \in \mathcal{D}$, and $u_{n0t} = 0$. We now have a relationship constraint between $\boldsymbol{z} = \{z_{ni}\}$ and $\boldsymbol{u} = \{u_{nit}\}$ as

$$u_{nit} \le z_{ni}, n \in \mathcal{N}, i \in \mathcal{S}_1^{|\mathcal{V}|}, t \in \mathcal{D}. \tag{3}$$

To exploit the multipath property, we build J trees and each tree can provide a bandwidth of d_r^j for $j \in \mathcal{S}_1^J$. To meet the total required bandwidth of $\bar{d}_r$, we impose constraint

$$\sum\nolimits_{j=1}^{J} d_r^j = \bar{d}_r. \tag{4}$$

We incorporate the routing and placement constraints to ensure that traffic flows passing from the source to multiple destinations through an VNF chain on the path

$$\sum\nolimits_{(n,m)\in\mathcal{L}} y_{(n,m)it}^j - \sum\nolimits_{(m,n)\in\mathcal{L}} y_{(m,n)it}^j = \begin{cases} u_{n(i+1)t} - u_{nit}, & \text{tree } j \text{ is activated} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

for $n \in \mathcal{N}, i \in \mathcal{S}_0^{|\mathcal{V}|}, t \in \mathcal{D}$, where

$$u_{s0t} = 1,\; u_{n0t} = 0,\; \forall t \in \mathcal{D},\; n \in \mathcal{N}\backslash\{s\}, \tag{6}$$

$$u_{t(|\mathcal{V}|+1)t} = 1,\; u_{n(|\mathcal{V}|+1)t} = 0,\; \forall t \in \mathcal{D},\; n \in \mathcal{N}\backslash\mathcal{D}, \tag{7}$$

$$u_{sit} = 0,\; u_{tit} = 0,\; \forall i \in \mathcal{S}_1^{|\mathcal{V}|},\; t \in \mathcal{D}. \tag{8}$$

In our model, all variables related to deactivated trees (i.e., x_{li}^j, y_{lit}^j, and d_r^j) should be zero. Define $\pi^j \in \{0,1\}$ to indicate tree j is activated or not and impose constraints

$$x_{li}^j \le \pi^j,\; y_{lit}^j \le \pi^j, d_r^j \le \pi^j \bar{d}_r. \tag{9}$$

With variable π^j, we can rewrite (5) as

$$\sum_{(n,m)\in\mathcal{L}} y_{(n,m)it}^j - \sum_{(m,n)\in\mathcal{L}} y_{(m,n)it}^j = \pi^j \left(u_{n(i+1)t} - u_{nit}\right), \quad n \in \mathcal{N}, i \in \mathcal{S}_0^{|\mathcal{V}|}, t \in \mathcal{D}. \tag{10}$$

Since $y_{lit}^j \le x_{li}^j$ in (2), the constraint $y_{lit}^j \le \pi^j$ in (9) can be removed. Finally, we rewrite (9) as

$$x_{li}^j \le \pi^j,\; d_r^j \le \pi^j \bar{d}_r,\; l \in \mathcal{L}, i \in \mathcal{S}_0^{|\mathcal{V}|}, j \in \mathcal{S}_1^J. \tag{11}$$

Constraint (11) means that we consider x_{li}^j and d_r^j when the tree j is activated; Otherwise, we simply set these variables to zero.

We require that exactly one instance of function f_i is traversed for every source/destination pair, which can be expressed as

$$\sum\nolimits_{n\in\mathcal{M}} u_{nit} = 1,\; i \in \mathcal{S}_1^{|\mathcal{V}|}, t \in \mathcal{D}. \tag{12}$$

The function f_i is hosted at node n only when admissable and when the resources at node n are sufficient. We have

$$\sum\nolimits_{i=1}^{|\mathcal{V}|} z_{ni} C(f_i) \le C(n), n \in \mathcal{M}, \tag{13a}$$

$$z_{ni} U(n,i) = 1, \forall n \in \mathcal{M}, i \in \mathcal{S}_1^{|\mathcal{V}|} \tag{13b}$$

where $U(n,i) = 1$ indicates NFV node $n\,(\in \mathcal{M})$ can admit function instance $f_i\,(\in \mathcal{M})$; Otherwise, $U(n,i) = 0$.

Objectives: The objective is to minimize the function and link provisioning cost (i.e., embedding cost), in addition to balancing load of the substrate network resources in the long run, which is represented by the following function

$$\min \sum\nolimits_{l\in\mathcal{L}} \sum\nolimits_{j=1}^{J} \sum\nolimits_{i=0}^{|\mathcal{V}|} \alpha \left(\frac{d_r^j}{B(l)} + 1\right) x_{li}^j + \sum\nolimits_{i=1}^{|\mathcal{V}|} \sum\nolimits_{n\in\mathcal{M}} \beta \frac{C(f_i)}{C(n)} z_{ni}. \tag{14}$$

In (14), we minimize the total cost of utilizing physical links in all trees (minimize x_{li}^j) as well as to minimize the cost of operating VNF instances in the NFV nodes (minimize z_{ni}) with weighting coefficients α and β that reflects the importance level of each resource respectively, such that $\alpha, \beta > 0$ and $\alpha + \beta = 1$. The terms $d_r^j x_{li}^j / B(l)$ and $C(f_i) z_{ni} / C(n)$ guarantee the load balancing over physical links and nodes [15]. The highly utilized links and NFV nodes become more costly as $B(l), C(n) \to 0$. Moreover, the term '+1' in (14) minimizes number of hops in building trees from the source to destinations.

Let us denote the product term $d_r^j x_{li}^j$ in the objective function (14) by r_{li}^j as

$$r_{li}^j = x_{li}^j d_r^j. \tag{15}$$

The term r_{li}^j can be interpreted as the transmission rate over link l to deliver traffic from f_i to f_{i+1} in tree j. The total rate over link l is upper bounded by the available link bandwidth resources $B(l)$, i.e.,

$$\sum\nolimits_{j=1}^{J} \sum\nolimits_{i=0}^{|\mathcal{V}|} r_{li}^j \le B(l),\; l \in \mathcal{L}. \tag{16}$$

In summary, the optimization problem is formulated as

$$\min \sum_{l\in\mathcal{L}} \sum_{j=1}^{J} \sum_{i=0}^{|\mathcal{V}|} \alpha \left(\frac{r_{li}^j}{B(l)} + x_{li}^j\right) + \beta \sum_{i=1}^{|\mathcal{V}|} \sum_{n\in\mathcal{M}} \frac{C(f_i)}{C(n)} z_{ni} \tag{17a}$$

$$\text{s.t.} \quad (2)-(4), (10)-(13), (15), (16) \tag{17b}$$

$$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{\pi} \in \{0,1\},\; \boldsymbol{d}_r \succeq \mathbf{0},\; \boldsymbol{r} \succeq 0. \tag{17c}$$

In (17), the objective function and all constraints are linear except for constraints (10) and (15). In the next step, we transform these nonlinear constraints to linear ones such that a standard MILP solver can solve it. To do so, for the nonlinear constraint in (10), the bilinear term $\pi^j u_{nit}$ can be handled by introducing a new variable $w^j_{nit} = \pi^j u_{nit}$. We then linearize constraint (10) as

$$\sum_{(m,n)\in\mathcal{L}} y^j_{(m,n)it} - \sum_{(n,m)\in\mathcal{L}} y^j_{(n,m)it} = w_{nit} - w_{n(i+1)t}, \quad n \in \mathcal{N}, i \in \mathcal{S}_0^{|\mathcal{V}|}, t \in \mathcal{D}, j \in \mathcal{S}_1^J. \tag{18}$$

The corresponding relationship constraints between w^j_{nit}, π^j, and u_{nit} are given by

$$w^j_{nit} \leq \pi^j,\ w^j_{nit} \leq u_{nit},\ w^j_{nit} \geq \pi^j + u_{nit} - 1, \quad n \in \mathcal{N}, i \in \mathcal{S}_0^{|\mathcal{V}|+1}, t \in \mathcal{D}, j \in \mathcal{S}_1^J. \tag{19}$$

For nonlinear constraint (15), we introduce the big-M notation and rewrite (15) equivalently as

$$d^j_r - M(1 - x^j_{li}) \leq r^j_{li} \leq d^j_r,\ 0 \leq r^j_{li} \leq M x^j_{li} \tag{20}$$

where M is a large positive number. Since d^j_r is upper bounded by $\bar{d}_r$, r^j_{li} given by (15) is bounded above by $\bar{d}_r$. We thus can set $M = \bar{d}_r$.

As a result, the nonlinear optimization problem (17) can be rewritten in an MILP form as

$$\min_{\boldsymbol{\mathcal{X}}} \sum_{l\in\mathcal{L}} \sum_{j=1}^{J} \sum_{i=0}^{|\mathcal{V}|} \alpha\left(\frac{r^j_{li}}{B(l)} + x^j_{li}\right) + \beta \sum_{i=1}^{|\mathcal{V}|} \sum_{n\in\mathcal{M}} \frac{C(f_i)}{C(n)} z_{ni} \tag{21a}$$

$$\text{s.t. } (2) - (4), (11) - (13), (16), (17c), (18) - (20), \tag{21b}$$

where $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{\pi}, \boldsymbol{d}_r, \boldsymbol{r}\}$, and problem (21) can be solved by an MILP solver.

Note that the VNF chain embedding for a single multicast service is considered in the problem formulation. The embedding for multiple multi-service requests is left for future research.

IV. Heuristic Solutions

Even though the formulated problem in Section III can be solved for optimal solutions via an MILP solver, the computational complexity increases substantially as the network size grows. A low-complexity heuristic approach is needed to find the solutions more efficiently. The heuristic algorithm can be designed with the following objectives: 1) To minimize the cost of the multicast topology, including both processing cost at activated servers and the bandwidth cost over activated links; 2) To enable a flexible placement and routing. To achieve these objectives, the heuristic algorithm first builds a multicast routing tree between the source node and the destination nodes, and then places the VNFs along the built routing tree. Conversely, if the VNFs are placed first, the search space for optimal VNF placement significantly increases with the network scale as a network-wide search is required. Moreover, placing the VNFs first will hinder the one-to-many VNF mapping, where the embedded path from source to the last VNF will be identical for all source-destination pairs. In other words, the multicast replication would occur only after the last embedded VNF. Therefore, we first build a KPMST to find a multicast routing topology, while minimizing the cost of utilizing physical links for all source-destination paths, and then place the VNF instances along the multicast topology to reduce the search space of VNF placement and increase the function placement flexibility. In what follows, we first provide the solution for the special case of single path ($J = 1$, single tree) routing, and then extend it to the general case of multipath ($J \geq 2$, multiple trees).

A. Heuristic Algorithm for Single Path Routing

We design the single-service heuristic algorithm based on following considerations: (i) Different types of VNFs can run simultaneously on an NFV node; (ii) The traversed VNF types and their order should be considered for each source-destination pair; (iii) The objective is to minimize the provisioning cost of the multicast topology based on (14). According to the aforementioned design principles, a two-step heuristic algorithm is devised as follows: We first minimize the link provisioning cost by constructing a KPMST-based multicast topology that connects the source with the destinations; Then, we greedily perform VNF placement such that the number of VNF instances are minimized. The algorithm is presented in Algorithm 1. In what follows, we explain it in more detail. First, since bandwidth is a bottleneck resource type, we remove all physical links that have $B(l) \leq \bar{d}$. Second, to prioritize the NFV node selection in building the KPMST, we modify the link weights in the substrate network $\mathcal{G}$ to yeild $\mathcal{G}'$ as

$$\omega_l = \alpha(\frac{1}{B(l)} + 1) + \beta\frac{1}{\tilde{C}(m)}, \quad l = (n, m) \in \mathcal{L} \tag{22}$$

where $\tilde{C}(m)$ is set to a small value when m is a switch; Otherwise, $\tilde{C}(m) = C(m)$. Then, a key-NFV node is selected iteratively. We construct the metric closure of $\mathcal{G}'$ in $\mathcal{G}''$, where the metric closure is a complete weighted graph with the same node set $\mathcal{N}$, where the new link weights are given by shortest path distances with respect to modified weights ω_l. From the metric closure, we find the MST which connects the source node and destination nodes and the key-NFV node. The multicast routing topology $\mathcal{G}_v$ can be constructed by replacing the edges in $\mathcal{G}''$ with corresponding paths from $\mathcal{G}'$ wherever needed. We greedily place the VNFs from the source of the multicast topology to its destinations with the objective of minimizing the number of VNF instances. Consequently, the cost $\mathbb{C}(\mathcal{G}_v)$ of the new multicast topology as well as the number $\mathbb{A}(\mathcal{G}_v)$ of successfully embedded VNF instances are computed. In every iteration, a new key-NFV node is selected: If $\mathbb{A}(\mathcal{G}_v)$ is improved, we update the selected multicast topology with the new key-NFV node; If $\mathbb{A}(\mathcal{G}_v)$ is unchanged and $\mathbb{C}(\mathcal{G}_v)$ is improved, we also update the selected multicast topology. The objective is to jointly maximize the number of successfully allocated VNFs and improve the provisioning cost while we iterate over all candidate key-NFV nodes.

If a path cannot include all required VNFs (*i.e.*, $f_1, f_2, \ldots, f_{|\mathcal{V}|}$) after selecting a key-NFV node, we devise a corrective subroutine that places the missing VNF instances on the closest NFV node from the multicast topology, and the corresponding physical links are rerouted.

B. Heuristic Algorithm for Multipath Routing

The heuristic algorithm for the single path case can be extended to solve the multipath-enabled VNF chain embedding problem. Enabling multipath routing provides several advantages: Multipath routing are activated when the bandwidth requirement between two consecutive embedded VNFs cannot be satisfied (i.e., when $B(l) < \overline{d}_r, l \in \mathcal{L}$); Moreover, it reduces the overall link provisioning cost further compared with the single path case. The heuristic algorithm starts with applying the single path multicast heuristic algorithm to find NFV nodes where the functions are placed for each destination as in Algorithm 1. Consequently, we start with each embedded virtual segment (i.e., path between each pair of embedded VNFs) for each source-destination path, where a path splitting mechanism is triggered as follows:

Let $W_{i,i+1}^{t,k}$ be the kth path between two embedded VNFs (f_i,f_{i+1}) along the network substrate for destination $t(\in \mathcal{D})$, where the cardinality of all possible paths for each virtual segment is $K_{i,i+1}^{t}$. We first rank all candidate paths for each virtual segument in a descending order based on the amount of residual bandwidth resources. Then, we sequentially choose the paths from, such that the summation of all chosen paths' residual bandwidth meets the required bandwidth, namely $\overline{d}_r$. The bandwidth resources allocated on the k^{th} path ($W_{i,i+1}^{t,k}$) is then calculated as

$$R(W_{i,i+1}^{t,k}) = \frac{B_{\min}^{k}\overline{d}_r}{\sum_{k=1}^{K_{i,i+1}^{t}} B_{\min}^{k}}, \quad t \in \mathcal{D}, i \in \mathcal{S}_1^{|\mathcal{V}|} \qquad (23)$$

where $B_{\min}^{k}$ is the amount of minimum residual bandwidth resources for path $W_{i,i+1}^{t,k}$, i.e., $B_{\min}^{k} = \min_{l \in W_{i,i+1}^{t,k}} B(l)$.

V. Simulation Results

In this section, we present simulation results for both optimal and heuristic solutions to the joint multicast routing and VNF placement problem, considering the single path and multipath routing cases. Throughout the simulation, we set the weighting coefficients $\alpha = 0.6$, $\beta = 0.4$ and $C(f_i) = \overline{d}_r$. For solving the MILP, we use Gurobi solver with branch and bound mechanism. The simulated physical substrate is a mesh topology based network, with $|\mathcal{N}| = 100$ and $|\mathcal{L}| = 684$, as shown in Fig. 2. We set 25 vertices in the mesh topology as NFV nodes, and the bandwidth of physical links and processing resources of NFV nodes are normalized and are uniformly distributed from 50 to 200, i.e., $C(n)$, $B(l) \sim \mathcal{U}(50, 200)$. We compare the total cost obtained from both optimal and heuristic solutions for the single-path scenario as the number of destinations $|\mathcal{D}|$ or functions $|\mathcal{V}|$ increases.

As shown in Fig. 3, the total cost generally increases with $|\mathcal{D}|$ or $|\mathcal{V}|$. As $|\mathcal{D}|$ increases, both the cost obtained from both the optimal and heuristic solutions increase, with a gap that remains almost constant for the whole horizontal range. Adding a destination in general costs more than adding a VNF, since additional physical links and probably other instances are required. In the case of adding more functions, with sufficient network resources and under the assumption that NFV nodes can host multiple concurrent functions, it costs less to add one instance in the optimal Steiner tree. Therefore, for $|\mathcal{V}| = [3, 9]$, the gap between the optimal solution and heuristic algorithm solution remains almost constant.

Algorithm 1: Heuristic Algorithm for Joint VNF placement and routing

function JPR ($\mathcal{G}, S_r$);
Input : (1) Physical substrate $\mathcal{G}(\mathcal{N}, \mathcal{L})$
(2) VNF chain $S_r = (s, \mathcal{D}, f_1, f_2, \ldots, f_{|\mathcal{V}|}, \overline{d}_r)$
Let: $\mathcal{G}' = \{\mathcal{G} | B(l) \geq \overline{d}_r, l \in \mathcal{L}\}$
Let: $\mathbb{C}(\mathcal{X})$ be the cost function in topology $\mathcal{X}$ as in (14)
Let: $\mathbb{A}(\mathcal{X})$ be the number of allocated VNF instances in topology $\mathcal{X}$
$\mathbb{C}_{ref} = \infty$; $\mathbb{A}_{ref} = 0$
for $n \in \mathcal{M}$ **do**
  $\mathcal{G}'' \longleftarrow MetricClosure(\mathcal{G}', \{n, s, \mathcal{D}\})$;
  $\mathcal{G}_v^{temp}(\mathcal{N}_v, \mathcal{L}_v) \longleftarrow KruskalsMST(\mathcal{G}'')$;
  Let: W_t be a path from s to t in $\mathcal{G}_v^{temp}$
  for $t \in \mathcal{D}$ **do**
    Place functions from $\mathcal{V}$ sequentially on W_t subject to (13).
  if $\mathbb{A}(\mathcal{G}_v^{temp}) = \mathbb{A}_{ref}$ *and* $\mathbb{C}(\mathcal{G}_v^{temp}) < \mathbb{C}_{ref}$ **then**
    $\mathcal{G}_v \leftarrow \mathcal{G}_v^{temp}$; $\mathbb{A}_{ref} = \mathbb{A}(\mathcal{G}_v^{temp})$; $\mathbb{C}_{ref} = \mathbb{C}(\mathcal{G}_v^{temp})$
  else if $\mathbb{A}(\mathcal{G}_v^{temp}) > \mathbb{A}_{ref}$ **then**
    $\mathcal{G}_v \leftarrow \mathcal{G}_v^{temp}$; $\mathbb{A}_{ref} = \mathbb{A}(\mathcal{G}_v^{temp})$; $\mathbb{C}_{ref} = \mathbb{C}(\mathcal{G}_v^{temp})$
for $f \in \mathcal{V}$ **do**
  Let: P be $\cup_{t \in \mathcal{D}} W_t$ such that f is not hosted
  Let: P_c be longest common path before first branch in P
  Link nearest NFV node that can host f to P_c, and remove unnecessary edges;
Output: Embedded multicast topology $\mathcal{G}_v$

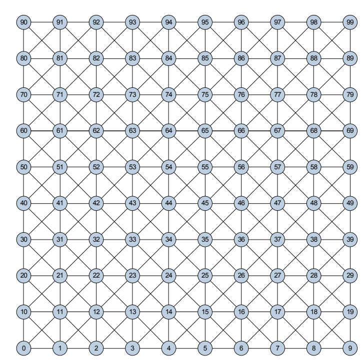

Fig. 2. Mesh topology with $|\mathcal{N}| = 100$ and $|\mathcal{L}| = 684$.

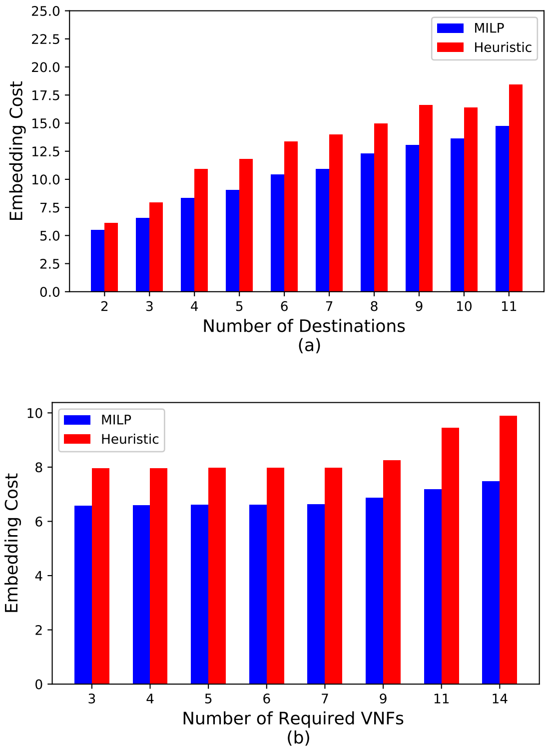

Fig. 3. Embedding cost with respect to (a) the number of destinations and (b) the number of functions.

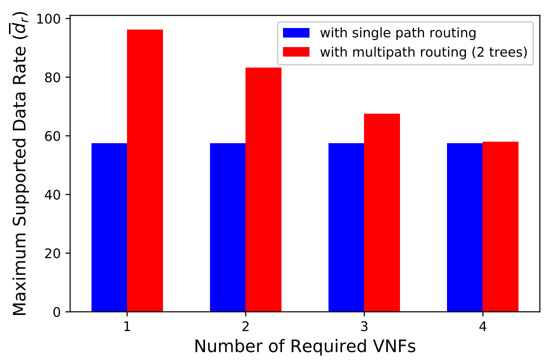

Fig. 4. Maximum supported data rate ($\bar{d}_r$) for the optimal single path and multipath routing formulations.

Finally, we analyze the performance of the optimal formulation for the multipath routing. Fig. 4 shows the advantage of multipath over single path routing. We use the mesh topology with 25 NFV nodes and with resources $C(n)$, $B(l) \sim \mathcal{U}(10, 100)$. We plot the maximum supported data rate ($\bar{d}_r$), for which the problem is still feasible. We limit the routing to 2 paths ($J = 2$) for the multipath case. As shown in Fig. 4, the multipath routing allows for supporting higher data rates, which achieves a higher acceptance ratio by supporting more service requests. By increasing the number of functions, the maximum supported data rate decreases. As $|\mathcal{V}|$ increases, the processing cost becomes more significant and the number of candidate NFV nodes and paths decrease.

VI. Conclusion

In this paper, we study a joint traffic routing and VNF placement problem for a multicast service request supported by a physical substrate network under the SDN-enabled NFV architecture. A flexible optimization problem is formulated to minimize function and link provisioning cost, under the physical resource constraints and flow conservation constraints. Our problem formulation is flexible as it allows for one-to-many and many-to-one VNF mapping. Moreover, it enables multipath routing by constructing multiple trees in delivering multicast service. The problem is formulated in an MILP, and thus can be solved to obtain an optimal solution as a benchmark. To reduce the computational complexity in solving the problem, heuristic algorithms are proposed, considering both single path and multipath routing, to achieve fairly competitive performance. Simulation results are presented to demonstrate the effectiveness and accuracy of the proposed heuristic algorithms especially for a large-scale size network.

References

[1] H. Zhang, S. Vrzic, G. Senarath, N.-D. Dao, H. Farmanbar, J. Rao, C. Peng, and H. Zhuang, "5G wireless network: MyNET and SONAC," *IEEE Netw.*, vol. 29, no. 4, pp. 14–23, Jul 2015.

[2] Q. Ye and W. Zhuang, "Distributed and adaptive medium access control for Internet-of-Things-enabled mobile networks," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 446–460, Apr 2017.

[3] Q. Ye, J. Li, K. Qu, W. Zhuang, X. Shen, and X. Li, "End-to-end quality of service in 5G networks – Examining the effectiveness of a network slicing framework," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 65–74, June 2018.

[4] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded vnf chains in 5G core networks," *IEEE Internet of Things J.*, to appear, doi: 10.1109/JIOT.2018.2853708.

[5] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36–43, Jul 2013.

[6] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov. 2014.

[7] N. Zhang, P. Yang, S. Zhang, D. Chen, W. Zhuang, B. Liang, and X. Shen, "Software Defined Networking Enabled Wireless Network Virtualization: Challenges and Solutions," *IEEE Netw.*, vol. 31, no. 5, pp. 42–49, 2017.

[8] J. Li, N. Zhang, Q. Ye, W. Shi, W. Zhuang, and X. Shen, "Joint resource allocation and online virtual network embedding for 5G networks," in *Proc. IEEE GLOBECOM' 17*, Dec. 2017, pp. 1–6.

[9] B. Yi, X. Wang, M. Huang, and A. Dong, "A Multi-stage Solution for NFV-enabled Multicast over the Hybrid Infrastructure," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 2061–2064, 2017.

[10] C. Qiao, H. Yu, J. Fan, W. Zhong, X. Cao, X. Gao, Y. Zhao, and Z. Ye, "Virtual network mapping for multicast services with max-min fairness of reliability," *J. Opt. Commun. Netw.*, vol. 7, no. 9, pp. 942–951, Sep 2015.

[11] S. Q. Zhang, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia, "Routing algorithms for network function virtualization enabled multicast topology on SDN," *IEEE Trans. Netw. Serv. Management*, vol. 12, no. 4, pp. 580–594, Dec 2015.

[12] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *J. Lightwave Techn.*, vol. 34, no. 14, pp. 3330–3341, July 2016.

[13] J. Blendin, J. Ruckert, T. Volk, and D. Hausheer, "Adaptive software defined multicast," in *Proc. IEEE NetSoft'15*, Apr 2015, pp. 1–9.

[14] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Approximation and online algorithms for NFV-enabled multicasting in SDNs," in *Proc. IEEE ICDCS' 17*, Jun 2017, pp. 625–634.

[15] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, 2012.