Dynamic Flow Migration for Embedded Services in SDN/NFV-Enabled 5G Core Networks

Kaige Qu, *Student Member, IEEE*, Weihua Zhuang, *Fellow, IEEE*, Qiang Ye, *Member, IEEE*, Xuemin Shen, *Fellow, IEEE*, Xu Li, and Jaya Rao

***Abstract*— Software defined networking (SDN) and network function virtualization (NFV) are key enabling technologies in fifth generation (5G) communication networks for embedding service-level customized network slices in a network infrastructure, based on statistical resource demands to satisfy long-term quality of service (QoS) requirements. However, traffic loads in different slices are subject to changes over time, resulting in challenges for consistent QoS provisioning. In this paper, a dynamic flow migration problem for embedded services is studied, to meet end-to-end (E2E) delay requirements with time-varying traffic. A multi-objective mixed integer optimization problem is formulated, addressing the trade-off between load balancing and reconfiguration overhead. The problem is transformed to a tractable mixed integer quadratically constrained programming (MIQCP) problem. It is proved that there is no optimality gap between the two problems; hence, we can obtain the optimum of the original problem by solving the MIQCP problem with some post-processing. To reduce time complexity, a heuristic algorithm based on redistribution of hop delay bounds is proposed to find an efficient solution. Numerical results are presented to demonstrate the aforementioned trade-off, the benefit from flow migration in terms of E2E delay guarantee, as well as the effectiveness and efficiency of the heuristic solution.**

***Index Terms*— Service function chaining (SFC), network slicing, dynamic flow migration, end-to-end (E2E) delay, VNF state transfer, SDN/NFV-enabled 5G networks.**

I. Introduction

THE service-oriented fifth generation (5G) networks will support new use cases and diverse services with multi-dimensional performance requirements, which cannot be supported by the legacy *one-size-fits-all* network architecture [2]. Network slicing is a promising solution to accommodate the broad range of services over a common network infrastructure, and provides service-level performance guarantees [3]. The emerging software defined networking (SDN) and network function virtualization (NFV) technologies create new opportunities for a flexible and programmable network architecture, which supports network slicing [4]–[7]. SDN decouples network control from data plane into a centralized control module, which enables a global network view and facilitates centralized network management. End-to-end (E2E) data delivery paths can be established by an SDN controller, by configuring forwarding rules in SDN-enabled switches via southbound protocols such as Openflow [8]. NFV separates network functions from dedicated hardware to software instances, referred to as virtual network functions (VNFs), hosted in NFV nodes such as commodity servers and data centers. NFV enables cost-effective VNF placement and elastic VNF capacity scaling.

With new business models introduced in 5G networks, a tenant such as a service provider requests a set of network services in the form of service function chains (SFCs), from an infrastructure provider (InP). An SFC is composed of multiple VNFs in a predefined order, to fulfill a composite service in E2E data delivery. Each VNF supports a dedicated packet processing functionality such as intrusion detection system (IDS) and network address translation (NAT). Many VNFs are state-dependent, and states are updated together with packet header or payload processing, to guarantee accurate packet processing. For example, a virtual IDS belonging to an SFC keeps track of pattern matchings for accurate attack detection in subsequent packets. VNF states are stored and updated locally in associate VNFs. Packets processed by a VNF are transmitted to next VNF in the same SFC for further processing, generating traffic between consecutive VNFs, until the last VNF. We refer to an SFC flow as the aggregate traffic flow traversing an SFC, and an inter-VNF subflow as the traffic between two consecutive VNFs in the same SFC.

The InP customizes network services over the network infrastructure, generating service-level network slices. For each service (SFC), VNFs are embedded on NFV nodes, and inter-VNF subflows are routed over physical paths between the corresponding upstream and downstream VNFs. Here, a physical path is a series of SDN-enabled switches and links. The VNFs and subflows are allocated certain processing resources and transmission resources respectively, according to long-term traffic statistics and quality of service (QoS) requirements. This process is referred to as *SFC embedding* [9]–[11]. We call a logical abstraction of all embedded physical paths between two NFV nodes as a virtual link. After SFC embedding, processing resources of NFV nodes

Manuscript received June 27, 2019; revised November 20, 2019; accepted January 13, 2020. Date of publication January 23, 2020; date of current version April 16, 2020. This work was financially supported by research grants from Huawei Technologies Canada and from the Natural Sciences and Engineering Research Council (NSERC) of Canada. This article was presented in part at the IEEE ICC 2019 [1]. The associate editor coordinating the review of this article and approving it for publication was N. Pappas. *(Corresponding author: Kaige Qu.)*

Kaige Qu, Weihua Zhuang, and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: k2qu@uwaterloo.ca; wzhuang@uwaterloo.ca; sshen@uwaterloo.ca).

Qiang Ye is with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN 56001 USA (e-mail: qiang.ye@mnsu.edu).

Xu Li and Jaya Rao are with Huawei Technologies Canada, Inc., Ottawa, ON K2K 3J1, Canada (e-mail: xu.lica@huawei.com; jaya.rao@huawei.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2020.2968907

are virtualized and distributed among VNFs, and virtual links are established among NFV nodes and are allocated with transmission resources.

We consider a single tenant with multiple network slices under its management [3]. The slices are required to be mutually isolated from each other in terms of service performance, e.g., E2E delay [12], [13]. However, during the operation of network slices, traffic arrivals of each service fluctuate over time, due to dynamic user subscription and time-varying traffic flows from source nodes, which results in imbalanced resource usage on NFV nodes and virtual links from time to time. The mismatch between traffic load and resource availability is detrimental to both service performance and resource utilization. Allocating dedicated resources to each slice according to peak demands guarantees performance isolation at the cost of resource overprovisioning. For efficient resource utilization, resource sharing among slices is desired, i.e., processing resources at NFV nodes and transmission resources on virtual links are shared among traffic flows of multiple services. Due to the chaining nature of SFCs, one service can share resources with different services at different NFV nodes and virtual links traversed along its E2E path, making performance isolation challenging to achieve. Within an NFV-MANO reference architecture, a resource orchestrator (RO) is responsible for orchestrating resources among slices, on behalf of the tenant [3]. Each slice is subject to resource capacity scaling via a network service orchestrator (NSO) [3]. Hence, a dynamic resource management scheme is required for the RO to accommodate traffic dynamics and provide continuous QoS performance guarantees for multiple services.

In this paper, we study a delay-aware flow migration problem for delay-sensitive services in a processing resource limited network, to guarantee average delay isolation among services within maximal tolerable service downtime. The placement of VNFs is adjusted over time, with scaled processing resources allocated to each VNF. The associate states at migrated VNFs are transferred to target NFV nodes for consistency. For a subflow, if either its upstream or downstream VNF migrates to an alternative NFV node, it should be remapped to an alternative virtual link accordingly. The goal is to achieve efficient utilization of processing resources at NFV nodes with minimum transmission resource overhead incurred by state transfers. With the consideration that not every two NFV nodes are directly connected by virtual links, the number of extra virtual links required for flow rerouting is minimized to reduce signaling overhead. The problem is formulated as a multi-objective mixed integer optimization problem. For delay awareness, under the assumption of prior knowledge of time-varying traffic rates, average E2E delay requirements are included in constraints based on delay modeling. Due to several quadratic constraints, an optimal solution to the problem is difficult to obtain using solvers such as Gurobi [14]. We transform the original problem into a tractable mixed integer quadratically constrained programming (MIQCP) problem. Although the two problems are not equivalent, it is proved that there is a zero optimality gap between them. Given an MIQCP optimum, the optimum of the original problem is obtained through a proposed mapping algorithm. The MIQCP transformation, together with the mapping algorithm, gives an optimal solution, but time complexity is high due to NP-hardness of the MIQCP problem. Therefore, a low-complexity heuristic algorithm based on redistribution of hop delay bounds is proposed to obtain a sub-optimal solution to the original problem.

TABLE I
LIST OF IMPORTANT NOTATIONS

Parameters	
C_i	Processing capacity of NFV node $i \in \mathcal{I}$ in cycle/s
$D^{(r)}$	Average E2E delay requirement for SFC $r \in \mathcal{R}$
H_r	Number of VNFs in SFC $r \in \mathcal{R}$
$L_h^{(r)}$	The h-th subflow in SFC $r \in \mathcal{R}$
P_i^{rh}	Processing density of NFV node $i \in \mathcal{I}$ for VNF $V_h^{(r)}$
T_i	CPU polling period of the CPU in NFV node $i \in \mathcal{I}$
$V_h^{(r)}$	The h-th VNF in SFC $r \in \mathcal{R}$
W	Context switching time overhead in a CPU polling period
$Z_h^{(r)}(k)$	State size of VNF $V_h^{(r)}$ in time interval k
β_i^l	Whether $i \in \mathcal{I} \cup \mathcal{A}$ is the starting point of virtual link $l \in \mathcal{L}$
$\Delta^{(r)}$	Maximal tolerable service downtime for SFC $r \in \mathcal{R}$
$\lambda^{(r)}(k)$	Time-varying traffic rate in packet/s for SFC $r \in \mathcal{R}$
$\sigma^{(r)}$	Average packet size in bit for SFC $r \in \mathcal{R}$
φ_i^l	Whether $i \in \mathcal{I} \cup \mathcal{A}$ is the ending point of virtual link $l \in \mathcal{L}$
Decision variables	
$B_{i,j}^{rh}(k)$	Transmission resource overhead for transferring the state of VNF $V_h^{(r)}$ from NFV node i to NFV node j
$d_i^{rh}(k)$	The (dummy) processing delay on VNF $V_h^{(r)}$ at NFV node i
$m_{i,j}^{rh}(k)$	Whether the mapped NFV node for VNF $V_h^{(r)}$ changes from NFV node i to NFV node j
$x_i^{rh}(k)$	Whether VNF $V_h^{(r)}$ is mapped to NFV node i
$y_{ij}^{rh}(k)$	Whether subflow $L_h^{(r)}$ is mapped to an extra virtual link between $i \in \mathcal{I} \cup \mathcal{A}$ and $j \in \mathcal{I} \cup \mathcal{A}$
$w_i(k)$	Whether context switching happens at NFV node i
$\eta(k)$	The maximum NFV node loading factor
$\mu_i^{rh}(k)$	Processing rate allocated to VNF $V_h^{(r)}$ by NFV node i
$\tau_{i,j}^{rh}(k)$	The (dummy) delay to transfer the state of VNF $V_h^{(r)}$ from NFV node i to NFV node j

The rest of this paper is organized as follows. Section II gives an overview of background and related work. The system model is presented in Section III, and a delay-aware flow migration problem is formulated in Section IV. Section V presents the MIQCP problem transformation, and derives the optimality gap between the transformed problem and the original problem. A low-complexity heuristic algorithm is proposed in Section VI. Performance evaluation for both the MIQCP and heuristic solutions is presented in Section VII, and conclusions are drawn in Section VIII. A list of important notations is given in Table I.

II. BACKGROUND AND RELATED WORK

Traffic engineering (TE) has been extensively investigated, to find paths for date delivery from source to destination within link capacity [15], [16]. A cost function, such as a piece-wise linear increasing and convex function of link utilization, can be used to penalize high link utilization near capacity. The traditional TE ensures that no packets get sent across overloaded links, by minimizing link utilization costs. Flow migration, i.e., steering traffic flows of embedded SFCs through alternative NFV nodes and virtual links, is a TE approach for

elastic SFC provisioning [17]. Similarly, maximum loading on NFV nodes can be minimized, to achieve load balancing over processing resources. However, traditional TE methods cannot be directly applied due to the following reasons. First, candidate paths for an SFC flow must traverse through NFV nodes for processing. In traditional TE problems, a flow is a source-destination pair without a predefined sequence of intermediate processing nodes. Second, the transfer of VNF states should be considered, since simply rerouting in-progress flows on a state-dependent VNF to an alternative NFV node leads to state inconsistency, causing processing inaccuracy. Some frameworks such as OpenNF are proposed to solve the state inconsistency problem, by not only migrating packets of the rerouted flow but also transferring the associate VNF states [18]–[20]. In Co-Scalar [21], parallel state transfer is proposed for an SFC with multiple state-dependent VNFs. Instead of sequentially transferring the states of each VNF, Co-Scalar transfers all VNF states in parallel, thus greatly reducing latency at the cost of transmission resources.

Dynamic VNF operations, including horizontal scaling, vertical scaling, and migration, are widely employed to provide elastic processing resource provisioning [22]. In this paper, both vertical scaling and migration are employed to provide elasticity under the assumption that the total number of VNF instances is unchanged. In existing studies on dynamic SFC embedding, the time-varying processing and transmission resource demands are assumed known a priori, based on which VNFs are placed on alternative NFV nodes, and inter-VNF subflows are rerouted to different physical paths [23]–[25]. QoS requirements are expressed in such a way that the time-varying resource demands are satisfied without violating the resource capacity. In [26], both resource capacity constraints and delay constraints are included in problem formulation, without load dependent queueing delay. In our study, queueing delay is considered and the delay isolation issue with time-varying traffic load is addressed. Existing studies take into consideration of the reconfiguration overhead for flow migration, which is modeled as a weighted number of reconfigured NFV nodes and physical links [24], or the total revenue loss due to throughput loss within a constant service downtime [25], or the time duration for all state transfers associated with flow migration [27]. One performance metric for migration is the maximum allowable downtime within certain time duration [28]. Under the assumption that the time interval for flow migration is much larger than state transfer time, we consider the total transmission resource overhead incurred by state transfers, within a maximal tolerable service downtime in one service interruption.

III. System Model

A. Physical Network

The physical network consists of links and SDN-enabled nodes, including switches and NFV nodes. Switches forward traffic from incoming links to outgoing links. Some switches act as edge switches for service access. NFV nodes have both forwarding and processing capabilities, each supporting multiple VNFs subject to resource constraints. There are a number of SFCs embedded in the physical network, each for one service request, as shown in Fig. 1.

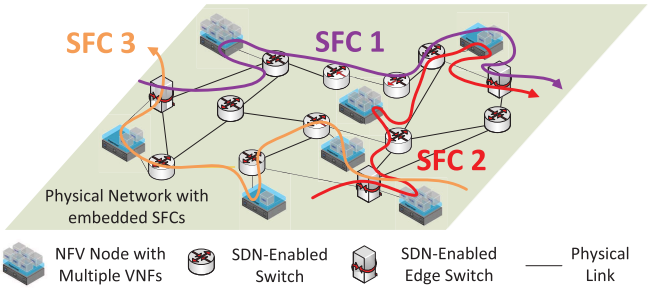


Fig. 1. A physical network with embedded SFCs.

B. Service Requests

A time-slotted system is considered, with integer k denoting the k-th time interval over which a flow migration plan remains unchanged. Let $\mathcal{R}$ denote the set of embedded service requests. A service request, $r \in \mathcal{R}$, is represented in the form of SFC. It originates from source node $a^{(r)}$ and traverses through H_r VNFs in sequence towards destination node $b^{(r)}$, with average E2E delay requirement $D^{(r)}$, maximal tolerable downtime $\Delta^{(r)}$ in one service interruption, average packet size $\sigma^{(r)}$ in bit, and time-varying traffic rate $\lambda^{(r)}(k)$ in packet/s. Under the assumption that flow migration is not frequent and the time interval is sufficiently large, traffic arrival of SFC $r \in \mathcal{R}$ during time interval k is modeled as Poisson with rate $\lambda^{(r)}(k)$ packet/s. Under the assumption that the time interval is much larger than $\max_{r\in\mathcal{R}} \Delta^{(r)}$, the experienced service downtime is much shorter than stable service operation time for any service. Let $\mathcal{H}_r = \{1, \cdots, H_r\}$, and denote the h-th ($h \in \mathcal{H}_r$) VNF in SFC r as $V_h^{(r)}$. Let $V_0^{(r)}$ and $V_{H_r+1}^{(r)}$ be dummy VNFs in SFC r, locating at source node $a^{(r)}$ and destination node $b^{(r)}$ respectively. Let $\mathcal{V}$ be a set containing all VNFs belonging to different SFCs, with $(r, h) \in \mathcal{V}$ denoting the h-th VNF in SFC r. Let $\mathcal{A}$ be a set of edge switches hosting all dummy VNFs. The h-th ($h \in \{0\} \cup \mathcal{H}_r$) inter-VNF subflow in SFC r, i.e., the subflow between upstream (dummy) VNF $V_h^{(r)}$ and downstream (dummy) VNF $V_{h+1}^{(r)}$, is denoted as $L_h^{(r)}$.

C. Abstraction of Virtual Resource Pool

In SFC embedding, we assume that each VNF is embedded to a single NFV node, every dummy VNF is hosted at one edge switch, and every subflow is allowed to be embedded to multiple physical paths between the locations of its upstream and downstream (dummy) VNFs. Fig. 1 shows a physical network with three embedded SFCs in single-path routing. We call a logical abstraction of all embedded physical paths between two NFV nodes or between one NFV node and one edge switch as a virtual link. A virtual resource pool is abstracted from the physical network with embedded SFCs, represented as a directed graph $G = \{\mathcal{I} \cup \mathcal{A}, \mathcal{L}\}$, where $\mathcal{I}$ is a set of all NFV nodes, $\mathcal{A}$ is a set of edge switches hosting dummy VNFs, and $\mathcal{L}$ is a set of virtual links. For virtual link $l \in \mathcal{L}$, we use

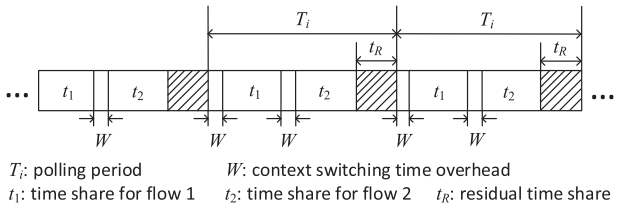


Fig. 2. A CPU polling scheme with two flows.

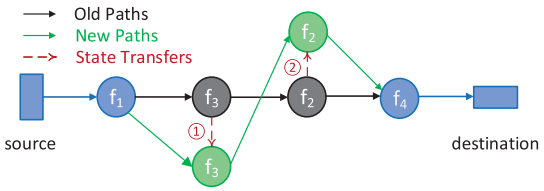


Fig. 3. An illustration for flow migration and state transfer.

binary parameters, $\{\beta_i^l\}$ and $\{\varphi_i^l\}$, to describe its location and direction, with $\beta_i^l = 1$ if $i \in \mathcal{I} \cup \mathcal{A}$ is its starting point and $\varphi_i^l = 1$ if $i \in \mathcal{I} \cup \mathcal{A}$ is its ending point. It is possible that G is not fully connected. Assume that there are sufficient transmission resources in the physical network. We can increase resources on existing virtual links, and find paths with enough resources for extra virtual links. Hence, we consider a processing resource limited virtual resource pool.

D. Processing Resource Sharing

The processing resource capacity C_i of NFV node $i \in \mathcal{I}$ is its maximum supporting CPU processing rate in cycle/s. For one packet/s of processing rate, the CPU resource demand on a certain NFV node depends on many factors, including the packet size, the type of function, the packet I/O scheme, and the virtualization technology [29]–[31]. We summarize all the factors into two categories: VNF dependent, and NFV node dependent. We define processing density of NFV node i for VNF $V_h^{(r)}$ as P_i^{rh} (in cycle/bit), which is the CPU resource demand (in cycle/s) of VNF $V_h^{(r)}$ on NFV node i for one bit/s of processing rate. Accordingly, $P_i^{rh}\sigma^{(r)}$ is the processing density of NFV node i for VNF $V_h^{(r)}$ in cycle/packet. A CPU polling scheme is employed for resource sharing among multiple flows, as illustrated in Fig. 2, in which two flow-specific processing queues are polled for service. Each queue gets a portion of CPU resources which is linear with its allocated CPU time share in a polling period. The polling scheme introduces multi-task context switching time overhead, due to extra CPU time spent on saving and loading contexts between every two consecutive tasks [32]. Here, processing packets from a certain processing queue is a task. The polling period T_i and the context switching time overhead W in NFV node i are constants. Under the assumption that resources are infinitely divisible, generalized processor sharing (GPS) is a benchmark resource allocation scheme to achieve QoS isolation and multiplexing gain among flows [12], [13]. Based on GPS discipline, the two flows are guaranteed minimum processing rates (in packet/s) of $\mu_1 = \frac{t_1}{T_i} \frac{C_i}{P_i^1 \sigma^{(1)}}$ and $\mu_2 = \frac{t_2}{T_i} \frac{C_i}{P_i^2 \sigma^{(2)}}$ respectively, where $t_1 + t_2 + 2W = T_i - t_R$, and t_R is the residual time share in a polling period. Define loading factor of NFV node i, denoted by η_i, as the percentage of allocated time shares plus context switching time overhead in a polling period of NFV node i.

E. Reconfiguration Overhead

When an SFC flow migrates at a state-dependent VNF, the VNF is remapped to an alternative NFV node, with the associate states transferred to the target NFV node. Fig. 3 illustrates the flow migration and associate state transfers, where two VNFs are remapped, and two state transfers are triggered correspondingly. Packet processing is halted during state transfer, incurring a service downtime. Let $Z_h^{(r)}(k)$ (in bit) be the time-varying state size of VNF $V_h^{(r)}$, whose value at time interval k is monitored by the SDN controller. For a state transfer at VNF $V_h^{(r)}$, $Z_h^{(r)}(k)$ is the product of state transfer delay and consumed transmission resources (in bit/s) [27]. For a remapped SFC with multiple state transfers, we use parallel state transfer in data plane in which all state transfers can take place simultaneously [21]. Then, the service downtime, which is the maximum state transfer delay along the E2E path, is much less than that of sequential state transfer, at the cost of transmission resources. Within a maximal tolerable service downtime, the total transmission resource overhead incurred by state transfers should be minimized. We assume that the transmission resource overhead for each state transfer is not less than $B_{min} = \frac{min(Z_h^{(r)}(k))}{max(\Delta^{(r)})}$.

For a subflow, if its upstream or downstream VNF migrates to an alternative NFV node, the subflow should be remapped to an alternative virtual link accordingly. After a successful remapping, transmission resources allocated to the subflow are released from the old virtual link. However, it is possible that NFV nodes in the virtual resource pool are not fully connected. Assume that the SDN controller can find physical paths with enough resources for an extra virtual link. Forwarding rule configuration along the physical paths incurs signaling overhead between the SDN controller and SDN-enabled switches.

In summary, the reconfiguration overhead due to flow migration is described in two parts: the total transmission resource overhead incurred by state transfers, and the total signaling overhead for configuring extra virtual links required for flow rerouting. The latter is assumed to be linear with the total number of extra virtual links required for flow rerouting.

IV. Problem Formulation

Problem 1: Consider a processing resource limited virtual resource pool. Assume that packet processing time at an NFV node for an SFC is exponentially distributed [33], [34]. During time interval k, traffic arrival of SFC $r \in \mathcal{R}$ is Poisson with rate $\lambda^{(r)}(k)$ packet/s. The rate can be predicted at the end of time interval $(k-1)$ based on measurements and historical information [17], [35]. A delay-aware flow migration problem is to 1) find the remapping between VNFs and NFV nodes in time interval k, based on the old mapping in time interval $(k-1)$, and 2) scale the processing resources allocated to VNFs, to satisfy average E2E delay requirements without

violating processing resource constraints. The objective, for time interval k, is to achieve load balancing among NFV nodes, with minimal reconfiguration overhead:

$$O(k) = \alpha_1 \eta(k) + \alpha_2 \sum_{(r,h)\in\mathcal{V}} \sum_{i,j\in\mathcal{I}} \frac{B_{i,j}^{rh}(k)}{B_{min}} + \alpha_3 \sum_{r\in\mathcal{R}} \sum_{h\in\{0\}\cup\mathcal{H}_r} \sum_{i,j\in\mathcal{I}\cup\mathcal{A}} y_{ij}^{rh}(k). \quad (1)$$

In objective function (1), there are several decision variables for time interval k: 1) continuous variable $\eta(k) \in [0,1]$ for maximum loading factor among all NFV nodes during interval k; 2) nonnegative continuous variable set $\boldsymbol{B}(k) = \{B_{i,j}^{rh}(k)\}$, with $B_{i,j}^{rh}(k)$ being the transmission resource overhead to transfer the state of VNF $V_h^{(r)}$ from NFV node $i \in \mathcal{I}$ during interval $(k-1)$ to NFV node $j \in \mathcal{I}$ during interval k; 3) binary variable set $\boldsymbol{y}(k) = \{y_{ij}^{rh}(k)\}$, with $y_{ij}^{rh}(k) = 1$ if subflow $L_h^{(r)}$ is mapped to an extra virtual link between i, $j \in \mathcal{I}\cup\mathcal{A}$ during interval k, and $y_{ij}^{rh}(k) = 0$ otherwise. Note that $\alpha_1, \alpha_2, \alpha_3$ are tunable weights to control the priority of the three components, with $\alpha_1+\alpha_2+\alpha_3 = 1$. In the right hand side of (1), the first term is the cost for imbalanced loading among NFV nodes since minimizing $\eta(k)$ achieves load balancing among all the NFV nodes, the second term is the cost for the overall normalized transmission resource overhead due to state transfers with a normalization ratio of $\frac{1}{B_{min}}$, the third term is the cost for extra virtual links required by flow rerouting. The normalization makes the three components in objective function (1) comparable, based on which α_1, α_2, α_3 can be selected on the same order of magnitude. A component in (1) is ignored if the corresponding weight is set to 0. If all weights are positive, all components in (1) are jointly optimized.

In terms of constraints, we start from node mapping constraints. Define binary decision variable set $\boldsymbol{x}(k) = \{x_i^{rh}(k)\}$ for interval k, with $x_i^{rh}(k) = 1$ if (dummy) VNF $V_h^{(r)}$ is mapped to node $i \in \mathcal{I}\cup\mathcal{A}$ during interval k, and $x_i^{rh}(k) = 0$ otherwise. As VNF $V_h^{(r)}$ should be mapped to exactly one NFV node in $\mathcal{I}$, we have

$$\sum_{i\in\mathcal{I}} x_i^{rh}(k) = 1, \quad \forall (r,h) \in \mathcal{V}. \quad (2)$$

For dummy VNFs, i.e., source and destination nodes, their physical locations are fixed and confined by

$$x_{a^{(r)}}^{r0}(k) = 1, \quad r \in \mathcal{R} \quad (3a)$$
$$x_i^{r0}(k) = 0, \quad r \in \mathcal{R},\ i \in \mathcal{I}\cup\mathcal{A}\backslash a^{(r)} \quad (3b)$$
$$x_{b^{(r)}}^{r(H_r+1)}(k) = 1, \quad r \in \mathcal{R} \quad (3c)$$
$$x_i^{r(H_r+1)}(k) = 0, \quad r \in \mathcal{R},\ i \in \mathcal{I}\cup\mathcal{A}\backslash b^{(r)}. \quad (3d)$$

The next constraints are related to transmission resource overhead for state transfer. From interval $(k-1)$ to interval k, the set representing VNF to NFV node mapping changes from $\boldsymbol{x}(k-1) = \{x_i^{rh}(k-1)\}$ to $\boldsymbol{x}(k) = \{x_i^{rh}(k)\}$, where $\boldsymbol{x}(k-1)$ is known in interval k. We introduce binary decision variable set $\boldsymbol{m}(k) = \{m_{i,j}^{rh}(k)\}$ for interval k, with $m_{i,j}^{rh}(k) = 1$ if the mapped NFV node for VNF $V_h^{(r)}$ changes from NFV node $i \in \mathcal{I}$ during interval $(k-1)$ to NFV node $j \in \mathcal{I}$ during interval k, and $m_{i,j}^{rh}(k) = 0$ otherwise. Hence, there is a relationship constraint among $\{m_{i,j}^{rh}(k)\}$ $(i \neq j)$, $\{x_i^{rh}(k-1)\}$ and $\{x_j^{rh}(k)\}$, given by

$$m_{i,j}^{rh}(k) = x_i^{rh}(k-1)\, x_j^{rh}(k), \quad \forall (r,h)\in\mathcal{V},\ \forall i\in\mathcal{I},\ \forall j\in\mathcal{I}\backslash\{i\}. \quad (4)$$

Also, we have

$$m_{i,i}^{rh}(k) = 0, \quad \forall (r,h) \in \mathcal{V},\ \forall i \in \mathcal{I}. \quad (5)$$

According to the definition of $B_{i,j}^{rh}(k)$, we have

$$0 \le B_{i,j}^{rh}(k) \le m_{i,j}^{rh}(k)\mathbb{M}, \quad \forall (r,h) \in \mathcal{V},\ \forall i,j \in \mathcal{I} \quad (6)$$

where $\mathbb{M}$ is a big-$\mathbb{M}$ constant to guarantee that $B_{i,j}^{rh}(k) = 0$ if $m_{i,j}^{rh}(k) = 0$. Let $\boldsymbol{\tau}(k) = \{\tau_{i,j}^{rh}(k)\}$ be a positive continuous decision variable set for interval k, with $\tau_{i,j}^{rh}(k)$ denoting the (dummy) delay to transfer state of VNF $V_h^{(r)}$ from NFV node $i \in \mathcal{I}$ during interval $(k-1)$ to NFV node $j \in \mathcal{I}$ during interval k. It follows that

$$\tau_{i,j}^{rh}(k) = \frac{Z_h^{(r)}(k)}{B_{i,j}^{rh}(k) + \epsilon}, \quad \forall (r,h) \in \mathcal{V},\ \forall i,j \in \mathcal{I} \quad (7)$$

where $0 < \epsilon \ll 1$ is a constant to avoid $\tau_{i,j}^{rh}(k)$ being undetermined, and $\tau_{i,j}^{rh}(k)$ is a dummy delay only if $m_{i,j}^{rh}(k) = 0$. Moreover, $\tau_{i,j}^{rh}(k)$ has an upper bound

$$0 < \tau_{i,j}^{rh}(k) \le m_{i,j}^{rh}(k)\Delta^{(r)} + \left[1 - m_{i,j}^{rh}(k)\right] \frac{Z_h^{(r)}(k)}{\epsilon}, \quad \forall (r,h) \in \mathcal{V}, \quad \forall i,j \in \mathcal{I}. \quad (8)$$

If $m_{i,j}^{rh}(k) = 1$, the upper bound is the corresponding maximal tolerable service downtime $\Delta^{(r)}$; otherwise, it is $\frac{Z_h^{(r)}(k)}{\epsilon}$.

For constraints related to processing resource scaling, let $\boldsymbol{\mu}(k) = \{\mu_i^{rh}(k)\}$ be a nonnegative continuous decision variable set for interval k, with $\mu_i^{rh}(k)$ being the processing rate in packet/s allocated to VNF $V_h^{(r)}$ by NFV node $i \in \mathcal{I}$ during interval k. It should be lower bounded by $x_i^{rh}(k)\,\lambda^{(r)}(k)$ due to the queue stability requirement and upper bounded by $\frac{x_i^{rh}(k)C_i}{P_i^{rh}\sigma^{(r)}}$ due to the limited processing capacity, given by

$$x_i^{rh}(k)\lambda^{(r)}(k) \le \mu_i^{rh}(k) \le \frac{x_i^{rh}(k)C_i}{P_i^{rh}\,\sigma^{(r)}}, \quad \forall (r,h)\in\mathcal{V},\ \forall i\in\mathcal{I}. \quad (9)$$

Let $w_i(k)$ be a binary decision variable with $w_i(k) = 1$ if context switching happens at NFV node i during interval k, i.e., there are at least two VNFs mapped to NFV node i, and $w_i(k) = 0$ otherwise. The loading factor of NFV node i during interval k, denoted by $\eta_i(k)$, consists of two parts, with a maximum value equal $\eta(k)$, which is upper bounded by a predefined constant η_U $(0 < \eta_U \le 1)$, given by

$$\sum_{(r,h)\in\mathcal{V}} \left[\frac{P_i^{rh}\sigma^{(r)}\mu_i^{rh}(k)}{C_i} + \frac{w_i(k)x_i^{rh}(k)W}{T_i}\right] \le \eta(k) \le \eta_U, \quad \forall i\in\mathcal{I} \quad (10)$$

where both useful time and context switching time overhead of CPU resources in a polling period are taken into consideration.

The left hand side of (10) is the expression for $\eta_i(k)$. The value of $w_i(k)$ is confined by an inequality constraint of

$$\frac{\sum_{(r,h)\in\mathcal{V}} x_i^{rh}(k)-1}{|\mathcal{V}|} \le w_i(k) \le \frac{\sum_{(r,h)\in\mathcal{V}} x_i^{rh}(k)}{A}, \quad \forall i \in \mathcal{I} \tag{11}$$

where A is an arbitrary value from $(1, 2]$.

For average E2E delay constraints, let $\boldsymbol{d}(k) = \{d_i^{rh}(k)\}$ be a positive continuous decision variable set for interval k, with $d_i^{rh}(k)$ denoting the average (dummy) delay on the queue associated with VNF $V_h^{(r)}$ at NFV node i. With Poisson traffic arrival and exponential packet processing time, the processing system is an M/M/1 queue. Then, $d_i^{rh}(k)$ is given by

$$d_i^{rh}(k) = \frac{1}{\mu_i^{rh}(k) - x_i^{rh}(k)\lambda^{(r)}(k) + \epsilon}, \quad \forall (r,h) \in \mathcal{V},\ \forall i \in \mathcal{I} \tag{12}$$

where $d_i^{rh}(k)$ is a dummy delay only if $x_i^{rh}(k) = 0$. There is an upper bound constraint for $d_i^{rh}(k)$, explicitly showing its relationship with $x_i^{rh}(k)$:

$$0 < d_i^{rh}(k) \le x_i^{rh}(k) D^{(r)} + \left[1 - x_i^{rh}(k)\right] \frac{1}{\epsilon}, \quad \forall (r,h) \in \mathcal{V},\ \forall i \in \mathcal{I}. \tag{13}$$

For QoS satisfaction, the average E2E delay of SFC $r \in \mathcal{R}$ should not exceed upper bound $D^{(r)}$,

$$\sum_{h\in\mathcal{H}_r} \sum_{i\in\mathcal{I}} x_i^{rh}(k)\, d_i^{rh}(k) \le D^{(r)}, \quad \forall r \in \mathcal{R}. \tag{14}$$

To decide whether a subflow should be mapped to an extra virtual link, we consider two cases. In the first case, we have

$$y_{ij}^{rh}(k) = 1 - \sum_{l\in\mathcal{L}} \beta_i^l \varphi_j^l x_i^{rh}(k)\, x_j^{r(h+1)}(k), \quad \forall r \in \mathcal{R},\ \forall h \in \{0\} \cup \mathcal{H}_r,\ \forall i \in \mathcal{I} \cup \mathcal{A},\ \forall j \in \mathcal{I} \cup \mathcal{A} \backslash \{i\} \tag{15}$$

to ensure that $y_{ij}^{rh}(k)$ equal 0 if (dummy) VNF $V_h^{(r)}$ and (dummy) VNF $V_{h+1}^{(r)}$ are mapped to node $i \in \mathcal{I} \cup \mathcal{A}$ and node $j \in \mathcal{I} \cup \mathcal{A} \backslash \{i\}$ between which a virtual link exists. In the second case, we have

$$y_{ii}^{rh}(k) = 1 - x_i^{rh}(k) x_i^{r(h+1)}(k), \quad \forall r \in \mathcal{R},\ \forall h \in \{0\} \cup \mathcal{H}_r,\ \forall i \in \mathcal{I} \cup \mathcal{A} \tag{16}$$

to ensure that $y_{ii}^{rh}(k)$ equal 0 if (dummy) VNF $V_h^{(r)}$ and (dummy) VNF $V_{h+1}^{(r)}$ are mapped to the same node $i \in \mathcal{I} \cup \mathcal{A}$.

In summary, the optimization problem is

$$\min_{\substack{\eta(k),\boldsymbol{B}(k),\boldsymbol{y}(k),\boldsymbol{x}(k),\boldsymbol{m}(k),\\ \boldsymbol{\tau}(k),\boldsymbol{\mu}(k),\boldsymbol{w}(k),\boldsymbol{d}(k)}} \quad O(k) \tag{17a}$$

$$\text{s.t.} \quad (2) - (16) \tag{17b}$$

$$\boldsymbol{d}(k), \boldsymbol{\tau}(k) > 0, \tag{17c}$$

$$\boldsymbol{x}(k), \boldsymbol{w}(k), \boldsymbol{m}(k), \boldsymbol{y}(k) \in \{0, 1\}. \tag{17d}$$

Remark 1: Problem (17) *is non-convex due to constraints* (7)*,* (10)*,* (12)*,* (14)*,* (15)*, and* (16)*.*

V. Optimal MIQCP Solution

In problem (17), quadratic constraints (10), (14) (15), and (16) can be transformed to equivalent linear forms using the big-$\mathbb{M}$ method. Quadratic constraints (7) and (12) cannot be linearized due to product terms of two continuous variables, but they can be replaced by combinations of linear constraints and rotated quadratic cone constraints. The new problem with transformed and replaced constraints is an MIQCP problem, which is not equivalent to the original problem. In this section, we discuss the relationship between both problems.

By introducing an auxiliary nonnegative continuous decision variable set, $\boldsymbol{\theta}(k) = \{\theta_i(k)\}$, we linearize constraint (10) based on the big-$\mathbb{M}$ method with $\mathbb{M} = |\mathcal{V}|$, given by

$$\sum_{(r,h)\in\mathcal{V}} \frac{P_i^{rh}\sigma^{(r)}\mu_i^{rh}(k)}{C_i} T_i + \theta_i(k) W \le \eta(k) T_i, \quad \forall i \in \mathcal{I} \tag{18a}$$

$$\sum_{(r,h)\in\mathcal{V}} x_i^{rh}(k) - |\mathcal{V}|[1 - w_i(k)] \le \theta_i(k) \le \sum_{(r,h)\in\mathcal{V}} x_i^{rh}(k), \quad \forall i \in \mathcal{I} \tag{18b}$$

$$0 \le \theta_i(k) \le |\mathcal{V}| w_i(k), \quad \forall i \in \mathcal{I}. \tag{18c}$$

By introducing an auxiliary nonnegative continuous decision variable set, $\boldsymbol{\gamma}(k) = \{\gamma_i^{rh}(k)\}$, we linearize constraint (14) based on the big-$\mathbb{M}$ method with $\mathbb{M} = \frac{1}{\epsilon}$ as

$$\sum_{h\in\mathcal{H}_r} \sum_{i\in\mathcal{I}} \gamma_i^{rh}(k) \le D^{(r)}, \quad \forall r \in \mathcal{R} \tag{19a}$$

$$d_i^{rh}(k) - \frac{1}{\epsilon}\left[1 - x_i^{rh}(k)\right] \le \gamma_i^{rh}(k) \le d_i^{rh}(k), \quad \forall (r,h) \in \mathcal{V},\ \forall i \in \mathcal{I} \tag{19b}$$

$$0 \le \gamma_i^{rh}(k) \le \frac{1}{\epsilon} x_i^{rh}(k), \quad \forall (r,h) \in \mathcal{V}, \forall i \in \mathcal{I}. \tag{19c}$$

By introducing an auxiliary binary decision variable set, $\boldsymbol{\xi}(k) = \{\xi_{ij}^{rh}(k)\}$, we get an equivalent linear form of constraint (15) and constraint (16) for $\forall r \in \mathcal{R}, \forall h \in \{0\} \cup \mathcal{H}_r$, and $\forall i \in \mathcal{I} \cup \mathcal{A}$, given by

$$\xi_{ij}^{rh}(k) \le x_i^{rh}(k), \qquad \forall j \in \mathcal{I} \cup \mathcal{A} \tag{20a}$$

$$\xi_{ij}^{rh}(k) \le x_j^{r(h+1)}(k), \qquad \forall j \in \mathcal{I} \cup \mathcal{A} \tag{20b}$$

$$\xi_{ij}^{rh}(k) \ge x_i^{rh}(k) + x_j^{r(h+1)}(k) - 1, \quad \forall j \in \mathcal{I} \cup \mathcal{A} \tag{20c}$$

$$y_{ij}^{rh}(k) = 1 - \sum_{l\in\mathcal{L}} \beta_i^l \varphi_j^l \xi_{ij}^{rh}(k), \quad \forall j \in \mathcal{I} \cup \mathcal{A} \backslash \{i\} \tag{20d}$$

$$y_{ii}^{rh}(k) = 1 - \xi_{ii}^{rh}(k). \tag{20e}$$

Proposition 1: With linearized constraints (18a)*,* (19a) *and* (20a)*, problem* (17) *can be transformed to an MIQCP problem, if constraint* (7) *is replaced by*

$$\rho_{i,j}^{rh}(k) = B_{i,j}^{rh}(k) + \epsilon, \qquad \forall (r,h) \in \mathcal{V},\ i,j \in \mathcal{I} \tag{21a}$$

$$\rho_{i,j}^{rh}(k) \ge \epsilon, \qquad \forall (r,h) \in \mathcal{V},\ i,j \in \mathcal{I} \tag{21b}$$

$$\tau_{i,j}^{rh}(k) \rho_{i,j}^{rh}(k) \ge z_h^{(r)}(k)^2, \quad \forall (r,h) \in \mathcal{V},\ i,j \in \mathcal{I} \tag{21c}$$

$$z_h^{(r)}(k) = \sqrt{Z_h^{(r)}(k)}, \qquad \forall (r,h) \in \mathcal{V} \tag{21d}$$

where $\boldsymbol{\rho}_{i,j}^{rh}(k) = \{\rho_{i,j}^{rh}(k)\}$ *and* $\boldsymbol{z}_h^{(r)}(k) = \{z_h^{(r)}(k)\}$ *are auxiliary continuous decision variable sets, and if constraint* (12)

is replaced by

$$\pi_i^{rh}(k)=\mu_i^{rh}(k)-x_i^{rh}(k)\lambda^{(r)}(k)+\epsilon, \quad \forall(r,h)\in\mathcal{V},\ \forall i\in\mathcal{I} \tag{22a}$$

$$\pi_i^{rh}(k)\geq\epsilon, \quad \forall(r,h)\in\mathcal{V},\ \forall i\in\mathcal{I} \tag{22b}$$

$$d_i^{rh}(k)\pi_i^{rh}(k)\geq c^2, \quad \forall(r,h)\in\mathcal{V},\forall i\in\mathcal{I} \tag{22c}$$

$$c=1 \tag{22d}$$

where $\boldsymbol{\pi}(k)=\{\pi_i^{rh}(k)\}$ is an auxiliary continuous decision variable set and c is an auxiliary continuous decision variable. The optimality gap between the two problems is zero, i.e., an optimum of problem (17) *is either a unique optimum or one of multiple optimal solutions to the MIQCP problem. Given an MIQCP optimum ("$\star$"), an optimum of problem* (17) *("$*$") can be obtained by Algorithm 1.*

Proof: The fundamental difference between the MIQCP problem and problem (17) lies in "$\geq$" signs in rotated quadratic cone constraints (21c) and (22c). If both constraints are active in an MIQCP optimum, i.e., all the "$\geq$" signs achieve equality, the MIQCP optimum is also an optimum of problem (17). Next, we discuss how the "$\geq$" signs affect the optimum.

First, assume that there is an inactive constraint (21c) in an MIQCP optimum, i.e., $\tau_{i,j}^{rh\star}(k)\left[B_{i,j}^{rh\star}(k)+\epsilon\right]>Z_h^{(r)}(k)$. If $B_{i,j}^{rh\star}(k)=0$, it does not affect the objective value. Thus, we consider only the case with $B_{i,j}^{rh\star}(k)>0$. If $B_{i,j}^{rh\star}(k)$ is replaced by $B_{i,j}^{rh\circ}(k)$, with $B_{i,j}^{rh\circ}(k)<B_{i,j}^{rh\star}(k)$ and $\tau_{i,j}^{rh\star}(k)\left[B_{i,j}^{rh\circ}(k)+\epsilon\right]=Z_h^{(r)}(k)$, all constraints are still satisfied. The objective value is unchanged if $\alpha_2=0$, in which case $[\tau_{i,j}^{rh\star}(k),\ B_{i,j}^{rh\circ}(k)]$ is an optimal pair in another MIQCP optimum. Otherwise ($\alpha_2>0$), the objective value can be further reduced, inferring that the assumption must be false.

Second, assume that there is an inactive constraint (22c) in an MIQCP optimum, then $d_i^{rh\star}(k)\left[\mu_i^{rh\star}(k)-x_i^{rh\star}(k)\lambda^{(r)}(k)+\epsilon\right]>1$. Similarly, we consider only the constraint with $x_i^{rh\star}(k)=1$. There are four cases, depending on α_1 and $\eta_i(k)$.

Case 1: $\alpha_1>0$, and NFV node i is the only one with a loading factor of $\eta^\star(k)$ (i.e., dominating NFV node). If $\mu_i^{rh\star}(k)$ is replaced by $\mu_i^{rh\circ}(k)$, with $d_i^{rh\star}(k)[\mu_i^{rh\circ}(k)-x_i^{rh\star}(k)\lambda^{(r)}(k)+\epsilon]=1$, all constraints are satisfied but $\eta^\star(k)$ can be further reduced. Hence, the assumption must be false;

Case 2: $\alpha_1>0$ and $\eta_i(k)<\eta^\star(k)$ (i.e., non-dominating NFV node). If $d_i^{rh\star}(k)$ is replaced by $d_i^{rh\circ}(k)$, with $d_i^{rh\circ}(k)[\mu_i^{rh\star}(k)-x_i^{rh\star}(k)\lambda^{(r)}(k)+\epsilon]=1$, all constraints are satisfied with the objective value unchanged. Thus, $[d_i^{rh\circ}(k),\ \mu_i^{rh\star}(k)]$ is an optimal pair in another MIQCP optimum;

Case 3: $\alpha_1>0$, and there is more than one NFV node including NFV node i with loading factor $\eta^\star(k)$. There must be at least one of them satisfying an active constraint (22c). One such NFV node is selected as the dominating NFV node, and others are seen as non-dominating NFV nodes;

Case 4: $\alpha_1=0$, and $\eta(k)$ is not optimized. If we replace $\mu_i^{rh\star}(k)$ by $\mu_i^{rh\circ}(k)$, all constraints are satisfied and the objective value is unchanged. Thus, $[d_i^{rh\star}(k),\ \mu_i^{rh\circ}(k)]$ is an optimal pair in another MIQCP optimum.

Algorithm 1: Post-Processing to MIQCP Optimum

Input: $\eta^\star$, $\boldsymbol{B}^\star$, $\boldsymbol{y}^\star$, $\boldsymbol{x}^\star$, $\boldsymbol{m}^\star$, $\boldsymbol{\tau}^\star$, $\boldsymbol{\mu}^\star$, $\boldsymbol{w}^\star$, $\boldsymbol{d}^\star$.
Initialization ($*=\star$).
for $(r,h)\in\mathcal{V},\ i,j\in\mathcal{I}$ **do**
  if $\tau_{i,j}^{rh\star}(k)\rho_{i,j}^{rh\star}(k)>(z_h^{(r)}(k)^\star)^2$ *and* $B_{i,j}^{rh\star}(k)>0$, **then** $B_{i,j}^{rh*}(k)=\frac{Z_h^{(r)}(k)}{\tau_{i,j}^{rh\star}(k)}-\epsilon$
for $(r,h)\in\mathcal{V},\ i\in\mathcal{I}$ **do**
  if $d_i^{rh\star}(k)\pi_i^{rh\star}(k)>(c^\star)^2$ *and* $x_i^{rh\star}(k)==1$, **then**
    if $\alpha_1>0$, **then** $d_i^{rh*}(k)=\frac{1}{\mu_i^{rh\star}(k)-x_i^{rh\star}(k)\lambda^{(r)}(k)+\epsilon}$
    if $\alpha_1==0$, **then** $\mu_i^{rh*}(k)=\frac{1}{d_i^{rh\star}(k)}+x_i^{rh\star}(k)\lambda^{(r)}(k)-\epsilon$
if $\alpha_1==0$, **then** *calculate* $\eta^*(k)$
Output: η^*, $\boldsymbol{B}^*$, $\boldsymbol{y}^*$, $\boldsymbol{x}^*$, $\boldsymbol{m}^*$, $\boldsymbol{\tau}^*$, $\boldsymbol{\mu}^*$, $\boldsymbol{w}^*$, $\boldsymbol{d}^*$.

In summary, an MIQCP optimum with inactive constraints in (21c) and (22c) can always be mapped to another MIQCP optimum with active constraints in (21c) and (22c), without affecting other constraints and the objective value. The mapped MIQCP optimum is also the optimum of problem (17). The mapping algorithm is provided in Algorithm 1. □

Remark 2: The MIQCP problem is NP-hard.

Proof: To prove the NP-hardness, it is sufficient to consider a special case in which services with $D^{(r)}\to\infty$ are embedded in a fully-connected virtual resource pool. We also consider zero VNF state size, zero context switching time overhead, and sufficiently large processing resource capacity for each NFV node holding all VNFs without overloading [23], [25]. In such a case, the MIQCP problem can be reduced from a multiprocessor scheduling problem [36]. The multiprocessor scheduling problem minimizes the maximum load among a number of processors which are assigned with a number of tasks with different loads, which is proved to be NP-hard. □

VI. Heuristic Solution

Although problem (17) can be solved by the optimal MIQCP solution according to Proposition 1, the computational time is high due to NP-hardness of the MIQCP problem. In this section, we propose a low-complexity modular heuristic solution to problem (17). We consider only the case where all components in objective function (1) are jointly optimized, i.e., $\alpha_1,\alpha_2,\alpha_3>0$. In this case, we assume that one VNF migration is penalized more than imbalanced loading (i.e., $\eta(k)$ reaching its upper bound η_U). Then, the condition of $\alpha_1\eta_U<\alpha_2$ should be satisfied in the worst case, if all VNF migrations incur the same transmission resource overhead for state transfer and require no extra virtual links for flow rerouting. Accordingly, in the proposed algorithm, we first minimize the number of overloaded NFV nodes with loading factors greater than η_U, and make migration decisions at overloaded NFV nodes, after which $\eta(k)$ is equal to η_U. Afterwards, $\eta(k)$ is further reduced for load balancing. The algorithm is insensitive to α_1 but sensitive to $\frac{\alpha_2}{\alpha_3}$, due to reconfiguration

overhead aware migration decisions. Therefore, it provides a sub-optimal solution to problem (17) with $\alpha_1\eta_U < \alpha_2$.

A. Overview

The heuristic algorithm is to determine a migration and resource allocation plan for interval k in the presence of predicted traffic variations (i.e., from $\{\lambda^{(r)}(k-1)\}$ to $\{\lambda^{(r)}(k)\}$). We first find if and where NFV node resource overloading would happen due to traffic variations, based on three factors. The first is node mapping, denoted by $\{x_i^{rh}(k)\}$; the second is hop (VNF) delay bounds, denoted by $\{D^{rh}(k)\}$, with $\sum_{h\in\mathcal{H}_r} D^{rh}(k) = D^{(r)}$; the third is NFV node loading factor threshold, denoted by η_{th}. With current node mapping and hop delay bounds, i.e., $x_i^{rh}(k) = x_i^{rh}(k-1)$ and $D^{rh}(k) = \sum_{i\in\mathcal{I}} x_i^{rh}(k-1)d_i^{rh}(k-1)$, we calculate NFV node loading factors with traffic rates, $\{\lambda^{(r)}(k)\}$, based on the M/M/1 queueing model. By comparing the NFV node loading factors with threshold η_{th} (initial value set as η_U), a set of overloaded NFV nodes is identified as potential bottlenecks.

1) Reconfiguration Overhead Reduction: Even if potential bottlenecks are identified, it is possible that migration is not necessary. For a given η_{th} value, how an E2E delay requirement is decomposed into hop delay bounds makes a difference on the number of overloaded NFV nodes. By making hop delay bounds less stringent on overloaded NFV nodes and more stringent on underloaded NFV nodes, it is possible to reduce the number of overloaded NFV nodes. The basic idea is as follows. If an SFC traverses both overloaded and underloaded NFV nodes, loading factors of the underloaded ones are increased to η_{th}, by shrinking corresponding hop delay bounds, and loading factors of the overloaded ones are decreased, by enlarging corresponding hop delay bounds. The strategy is referred to as delay scaling. Delay scaling is performed iteratively, until there is no SFC traversing both overloaded and underloaded NFV nodes. The iterative delay scaling procedure with given threshold, η_{th}, is referred to as redistribution of hop delay bounds.

If the number of overloaded NFV nodes is reduced to zero after redistribution of hop delay bounds, no migration is required. Otherwise, migration is necessary to overcome resource overloading. Migration decisions are made sequentially, i.e., only a pair of variables in set $\{x_i^{rh}(k)\}$ is updated in one migration decision, each followed by a redistribution of hop delay bounds, until no more migration is required.

With alternate migration decision and redistribution of hop delay bounds, reconfiguration overhead is greedily reduced in two ways. One is the potential reduction of overloaded NFV nodes by redistribution of hop delay bounds. The other is consideration of reconfiguration overhead in migration decision.

2) Load Balancing: If no potential bottlenecks are detected or all detected potential bottlenecks are removed by migration and redistribution of hop delay bounds, load balancing is the only remaining objective. NFV node loading factors are gradually balanced by iterative redistribution of hop delay bounds with threshold updating. The threshold, η_{th}, is updated from binary search, until it reaches sufficient precision.

More details on redistribution of hop delay bounds are given in Subsection VI-B, with pseudo code presented in Algorithm 2. Migration decision is discussed in Subsection VI-C, and threshold updating is discussed in Subsection VI-D. Finally, the heuristic algorithm is presented in Algorithm 3.

Algorithm 2: Redistribution of Hop Delay Bounds

Input: η_{th}, $\{x_i^{rh}(k)\}$, $\{D^{rh}(k)\}$
Calculate $\{\eta_i(k)\}$, $\mathcal{O}$, $\mathcal{U}_U$, $\mathcal{U}_O$, $\mathcal{Q}$, $\{f_1^{(r)}\}$, $\{f_2^{(r)}\}$.
Loop index $n = 0$.
while $\mathcal{U}_O \neq \varnothing$ *and* $\sum_{r\in\mathcal{R}} f_1^{(r)} > 0$ **do**
  $\{\delta_i(k)\} = 1$; $\{\beta^{(r)}(k)\} = 1$.
  if $n == 0$, **then**
    Vertical delay scaling at NFV nodes in $\mathcal{U}_U$.
    Horizontal delay scaling for SFC category III at NFV nodes in $\mathcal{U}_O$.
    Update $\{\eta_i(k)\}$.
  Vertical delay scaling at NFV nodes in $\mathcal{U}_O$.
  Horizontal delay scaling for SFC category II at NFV nodes in $\mathcal{O}$.
  Update $\{\eta_i(k)\}$, $\mathcal{O}$, $\mathcal{U}_U$, $\mathcal{U}_O$, $\mathcal{Q}$, $\{f_1^{(r)}\}$, $\{f_2^{(r)}\}$.
  $n = n + 1$
Output: $\{D^{rh}(k)\}$, $\{\eta_i(k)\}$, $\mathcal{O}$, $\mathcal{O}_1$, $\{f_1^{(r)}\}$.

B. Redistribution of Hop Delay Bounds

Classification: With given threshold η_{th} and a given set of $\{D^{rh}(k)\}$, the loading factor of NFV node $i \in \mathcal{I}$ in the presence of traffic variations is calculated as

$$\eta_i(k) = \sum_{(r,h)\in\mathcal{V}} \left(\frac{P_i^{rh}\sigma^{(r)}\mu_i^{rh}(k)}{C_i} + \frac{w_i(k)x_i^{rh}(k)W}{T_i} \right) \quad (23)$$

where $w_i(k)$ is calculated from (11) and $\mu_i^{rh}(k)$ is given by

$$\mu_i^{rh}(k) = \left(\lambda^{(r)}(k) + \frac{1}{D^{rh}(k)} \right) x_i^{rh}(k). \quad (24)$$

Three sets of NFV nodes are identified: $\mathcal{O} = \{i \in \mathcal{I} | \eta_i(k) > \eta_{th}\}$ consisting of overloaded NFV nodes, $\mathcal{U} = \{i \in \mathcal{I} | \eta_i(k) < \eta_{th}\}$ for underloaded NFV nodes, and $\mathcal{Q} = \{i \in \mathcal{I} | \eta_i(k) = \eta_{th}\}$. Let binary variable $X_i^{(r)}(k)$ indicate whether SFC r traverses NFV node i during interval k, with $X_i^{(r)}(k) = 1$ if $\sum_{h\in\mathcal{H}_r} x_i^{rh}(k) > 0$, and $X_i^{(r)}(k) = 0$ otherwise. Let $f_1^{(r)}$ be a binary flag indicating whether SFC r traverses any overloaded NFV nodes, with $f_1^{(r)} = 1$ if $\sum_{i\in\mathcal{O}} X_i^{(r)}(k) > 0$, and $f_1^{(r)} = 0$ otherwise. Set $\mathcal{U}$ is divided into two subsets, i.e., $\mathcal{U} = \mathcal{U}_U \cup \mathcal{U}_O$, where $\mathcal{U}_U = \{i \in \mathcal{U} | \sum_{r\in\mathcal{R}} X_i^{(r)}(k) f_1^{(r)} = 0\}$ is a set of underloaded NFV nodes on which no SFCs traverse other overloaded NFV nodes, and $\mathcal{U}_O = \{i \in \mathcal{U} | \sum_{r\in\mathcal{R}} X_i^{(r)}(k) f_1^{(r)} > 0\}$ is a set of underloaded NFV nodes on which at least one SFC traverses other overloaded NFV nodes. Let $f_2^{(r)}$ be a binary flag indicating whether SFC r traverses any NFV nodes in $\mathcal{U}_O$, with $f_2^{(r)} = 1$ if $\sum_{i\in\mathcal{U}_O} X_i^{(r)}(k) > 0$, and $f_2^{(r)} = 0$ otherwise. Accordingly, SFCs are classified into four categories: SFC category I with $f_1^{(r)} = 1$ and $f_2^{(r)} = 0$, SFC category II with

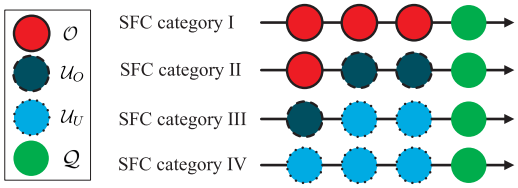


Fig. 4. Four SFC categories based on NFV node loading factors.

$f_1^{(r)} = f_2^{(r)} = 1$, SFC category III with $f_1^{(r)} = 0$ and $f_2^{(r)} = 1$, and SFC category IV with $f_1^{(r)} = f_2^{(r)} = 0$, as shown in Fig. 4.

Update and Iteration: Define two sets of delay scaling factors, with initial values of 1, vertical delay scaling factors $\{\delta_i(k)\}$ and horizontal delay scaling factors $\{\beta^{(r)}(k)\}$. A two-step delay scaling strategy is proposed as follows.

1) Step I - Delay scaling for SFC Category III: Hop delay bounds for SFC category III are relaxed on NFV nodes in $\mathcal{U}_O$, to release resources for SFC category II, by making hop delay bounds more stringent on NFV nodes in $\mathcal{U}_U$.

First, the loading factor of NFV node $i \in \mathcal{U}_U$ is increased to η_{th}, by shrinking hop delay bounds for SFC category III on NFV node i by a positive factor, $\delta_i(k)$, less than 1, as derived in Appendix and given by

$$\delta_i(k) = \frac{\sum_{(r,h)\in\mathcal{V}} \frac{P_i^{rh}\sigma^{(r)}}{D^{rh}(k)} x_i^{rh}(k) f_2^{(r)}}{[\eta_{th} - \eta_i(k)]\, C_i + \sum_{(r,h)\in\mathcal{V}} \frac{P_i^{rh}\sigma^{(r)}}{D^{rh}(k)} x_i^{rh}(k) f_2^{(r)}}. \quad (25)$$

The preceding delay scaling, called vertical delay scaling, is applied to multiple SFCs belonging to category III on NFV node $i \in \mathcal{U}_U$. Then, hop delay bounds for SFC r in category III on NFV nodes in $\mathcal{U}_O$ are relaxed by a factor, $\beta^{(r)}(k)$, larger than 1, given by

$$\beta^{(r)}(k) = \frac{D^{(r)} - \sum_{h\in\mathcal{H}_r} \left(D^{rh}(k) \sum_{i\in\mathcal{U}_U\cup\mathcal{Q}} x_i^{rh}(k) \right)}{D^{(r)} - \sum_{h\in\mathcal{H}_r} \left(D_p^{rh}(k) \sum_{i\in\mathcal{U}_U\cup\mathcal{Q}} x_i^{rh}(k) \right)} \quad (26)$$

where $D_p^{rh}(k)$ is the old value of $D^{rh}(k)$ before vertical delay scaling. The preceding delay scaling, called horizontal delay scaling, is applied to multiple hops in an SFC affected by vertical delay scaling. Based on (23), $\{\eta_i(k)\}$ is updated.

2) Step II - Delay Scaling for SFC Category II: More resources available at NFV nodes in $\mathcal{U}_O$ from Step I are allocated to SFC category II. First, vertical delay scaling is applied to NFV nodes in $\mathcal{U}_O$ to increase their loading factors to η_{th}, through scaling hop delay bounds for SFC category II on it by a vertical scaling factor. Then, horizontal delay scaling is applied to SFC category II, by relaxing hop delay bounds for SFC r in category II on NFV nodes in $\mathcal{O}$ by a horizontal scaling factor. The scaling factors are similar to that in Step I, and details are omitted.

After the delay scaling procedures, NFV node loading factors, NFV node classification and SFC categories are updated. If $\mathcal{U}_O \neq \varnothing$ and $\sum_{r\in\mathcal{R}} f_1^{(r)} > 0$, i.e., there is at least one SFC in category II, it is possible to further reduce the number of overloaded NFV nodes through Step II, so Step II is performed iteratively until the condition is violated. The outputs of Algorithm 2 are shown in line 14, where $\mathcal{O}_1 = \{i \in \mathcal{O} | \sum_{r\in\mathcal{R}} X_i^{(r)}(k) = 1\}$ denotes a set of overloaded NFV nodes traversed by a single SFC.

C. Migration Decision

When one migration is required, a migration decision procedure is to select one bottleneck NFV node, one SFC to migrate, and one target NFV node. Migration decisions are made greedily to reduce the reconfiguration overhead. First, a candidate bottleneck NFV node set, $\mathcal{B}$, with $|\mathcal{B}| = |\mathcal{R}|$, is determined. For an SFC, the traversed NFV node with largest hop delay bound is selected as a candidate bottleneck. Then, bottleneck NFV node, i_b, is determined in three cases. In the first case with $\mathcal{B} \cap \mathcal{O} \neq \varnothing$, an NFV node in $\mathcal{B} \cap \mathcal{O}$ with the largest number of SFCs is selected, given by

$$i_b = \underset{i\in\mathcal{B}\cap\mathcal{O}}{\operatorname{argmax}} \sum_{r\in\mathcal{R}} X_i^{(r)}(k). \quad (27)$$

In the second case with $\mathcal{B} \cap \mathcal{O} = \varnothing$ and $\mathcal{O} \backslash \mathcal{O}_1 \neq \varnothing$, i_b is an NFV node in $\mathcal{O} \backslash \mathcal{O}_1$ with the largest loading factor,

$$i_b = \underset{i\in\mathcal{O}\backslash\mathcal{O}_1}{\operatorname{argmax}} \eta_i(k). \quad (28)$$

In the third case with $\mathcal{B} \cap \mathcal{O} = \mathcal{O} \backslash \mathcal{O}_1 = \varnothing$, an NFV node in $\mathcal{B}$ whose SFCs traverse the largest number of overloaded NFV nodes is selected, given by

$$i_b = \underset{i\in\mathcal{B}}{\operatorname{argmax}} \sum_{r\in\mathcal{R}} \left(X_i^{(r)}(k) \sum_{j\in\mathcal{O}} X_j^{(r)}(k) \right). \quad (29)$$

Next, an SFC to migrate from i_b and a target NFV node to accommodate the migrated SFC are jointly selected to minimize the reconfiguration overhead, i.e., the weighted sum of normalized transmission resource overhead for state transfer and number of extra virtual links for flow rerouting. In this way, α_2 and α_3 are considered in the heuristic algorithm. If there are multiple choices, an SFC with the largest resource demand is migrated to the closest target NFV node.

D. Coordination With Threshold Update

Let binary variable, κ, indicate whether migration is required to overcome resource overloading. It is set as 0 initially and updated iteratively. Let ζ be a step size to update η_{th}, with initial value ζ^0 and updated before each η_{th} update. A precision, ε, for ζ is set as a stop condition.

After initialization in Algorithm 3 (lines 2-3), a redistribution of hop delay bounds is performed to check whether migration is required. Based on outputs of Algorithm 2, κ, η_{th} and ζ are updated in three cases, as shown in lines 8-12.

1) Update for κ and η_{th}: In the first case, there are no remaining overloaded NFV nodes, i.e., $\sum_{r\in\mathcal{R}} f_1^{(r)} = 0$. Then $\kappa = 0$, and η_{th} is reduced by a step. In the other two cases, there are still overloaded NFV nodes but $\mathcal{U}_O = \varnothing$, meaning that delay scaling is not sufficient to deal with resource overloading on NFV nodes, but either at least one migration or increasing η_{th} by a step is required, depending on the η_{th} value. In the second case with $\eta_{th} = \eta_U$, at least one migration is needed, i.e., $\kappa = 1$, and η_{th} should remain η_U to check whether more migrations are required after a migration decision is made. In the third case with $\eta_{th} < \eta_U$, no more migrations are required since η_{th} has been reduced by at least one step in previous updates, thus $\kappa = 0$ and η_{th} is increased

Algorithm 3: Heuristic Algorithm for Problem (17)

Input: Step size ζ^0, precision ε
Initialize: $\{x_i^{rh}(k)\}$, $\{D^{rh}(k)\}$
Let: $\kappa = 0$, $\eta_{th} = \eta_U$, $\zeta = \zeta^0$
while $\kappa == 1$ *or* $\zeta > \varepsilon$ **do**
  if $\kappa == 1$, **then**
    Update $\{x_i^{rh}(k)\}$ according to the migration decision making procedure.
  Update $\{D^{rh}(k)\}$, $\{\eta_i(k)\}$, $\mathcal{O}$, $\mathcal{O}_1$, $\{f_1^{(r)}\}$ according to Algorithm 2.
  if $\sum_{r\in\mathcal{R}} f_1^{(r)} == 0$, **then**
    if $\zeta \neq \zeta^0$, **then** $\zeta = \zeta/2$
    $\eta_{th} = \eta_{th} - \zeta$, $\kappa = 0$
  else if $\eta_{th} == \eta_U$, **then** $\kappa = 1$
  else $\zeta = \zeta/2$, $\eta_{th} = \eta_{th} + \zeta$, $\kappa = 0$
Output: Sub-optimal solution to problem (17).

by a step. With the updates for κ and η_{th}, redistribution of hop delay bounds is performed iteratively until no more migrations are required and the precision of ζ reaches ε.

2) Update for ζ: Step size ζ plays a key role in guaranteeing algorithm convergence. For example, a constant ζ equal to ε guarantees precision but makes the algorithm slow to converge due to a potential oscillation of η_{th} around its optimal value. Therefore, we employ the following strategy to update ζ. If the outputs of Algorithm 2 fall into the second case where a migration is required, ζ remains a constant equal to ζ^0. After all migrations are performed, the outputs of Algorithm 2 correspond to the first case, and η_{th} should be reduced by a constant step size ζ equal to ζ^0. Until the first time that the outputs of Algorithm 2 fall into the third case, ζ starts to be reduced by half before each η_{th} update.

E. Complexity Analysis

We first analyze the time complexity of Algorithm 2. Delay scaling for SFC category III is performed once, using at most $O(\sum_{i\in\mathcal{I}}|\mathcal{V}|)$ time. Delay scaling for SFC category II is performed iteratively until there are no new NFV nodes in $\mathcal{U}_O$ or there are no overloaded SFCs. The worst case happens when each round of delay scaling for SFC category II transforms a single NFV node in $\mathcal{O}$ to a new NFV node in $\mathcal{U}_O$, consuming $O(\sum_{i\in\mathcal{O}}\sum_{i\in\mathcal{I}}|\mathcal{V}|)$ time. Thus, the complexity of Algorithm 2 is $O(\sum_{i\in\mathcal{O}}\sum_{i\in\mathcal{I}}|\mathcal{V}|)$, upper bounded by $O(|\mathcal{I}|^2|\mathcal{V}|)$. The complexity of the migration decision procedure is dominated by the selection of SFC to migrate in the third case, which requires a running time of $O(|\mathcal{I}||\mathcal{R}|^2)$. In Algorithm 3, at most $|\mathcal{V}|$ sequential migration decisions are performed followed by $[\frac{1}{\zeta^0} + log_2(\frac{\zeta^0}{\epsilon})]$ iterations of threshold updating. In each iteration, hop delay bounds are readjusted. Therefore, the worst case running time of Algorithm 3 is $|\mathcal{V}|[O(|\mathcal{I}||\mathcal{R}|^2) + O(|\mathcal{I}|^2|\mathcal{V}|)] + [\frac{1}{\zeta^0} + log_2(\frac{\zeta^0}{\epsilon})]O(|\mathcal{I}|^2|\mathcal{V}|)$, which is simplified to $O(|\mathcal{I}|^2|\mathcal{V}|^2)$ when $|\mathcal{R}|^2 < |\mathcal{I}||\mathcal{V}|$.

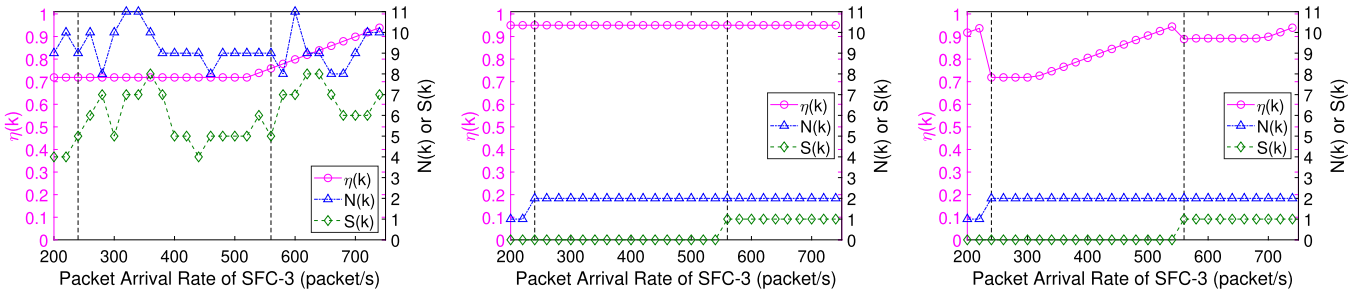
VII. Performance Evaluation

In this section, simulation results are presented to evaluate the MIQCP and heuristic solutions for the delay-aware flow migration problem. Two time intervals are considered: $(k-1)$ and k, representing the current and next time intervals respectively. We use two mesh networks with 64 NFV nodes and 256 NFV nodes to represent the virtual resource pool. Virtual links exist only between neighboring NFV nodes. In the 64-node network, we consider fixed SFC mapping for time interval $(k-1)$, with three SFCs initially mapped to the virtual resource pool. Specifically, SFC 3 shares two NFV nodes with SFC 1 and one of them also with SFC 2. In the 256-node network, we consider different numbers of SFCs, with [3, 5] VNFs in each one, randomly distributed in the network during time interval $(k-1)$. In the CPU polling scheme, we set a ratio of 0.01 between the context switching time and the polling period. The upper bound, η_U, for $\eta(k)$, is 0.95. The average E2E delay requirement for each SFC is 0.02 *s*, and the maximal tolerable service downtime is 0.005 *s*. For VNF states, the size is a constant, equal to 10 *bytes*, thus requiring at least a transmission rate B_{min} of 16 *kbit/s* for a state transfer. Under the simulation setup, the total normalized transmission resource overhead for state transfer is equal to the total number of migrations. For the weights, we set $\alpha_2 = 2\alpha_3$. Then, $\alpha_1 < \frac{2}{3\eta_U+2} = 0.4123$ should be satisfied to penalize migration more than imbalanced loading. In this case, 0.4123 is the worst-case boundary for α_1, to guarantee the penalization preference if α_1 is less than the boundary. We implement both the MIQCP and heuristic solutions in python. We use NetworkX to simulate the network scenario, and Gurobi python interface to solve the MIQCP problem.

A. Load Balancing and Reconfiguration Overhead Trade-Off

We use the 64-node network with three initially mapped SFCs to evaluate the performance of the MIQCP solution with varying traffic load under three sets of weights in (1), and investigate the trade-off between load balancing and reconfiguration overhead. For traffic load during interval k, we have $\lambda^{(1)}(k) = 600$ packet/s, $\lambda^{(2)}(k) = 200$ packet/s, and vary $\lambda^{(3)}(k)$ from 200 packet/s to 740 packet/s. Beyond 740 packet/s, the problem becomes infeasible due to processing resource constraints and average E2E delay constraints. Performance metrics are the maximum NFV node loading factor, $\eta(k)$, the number of migrations, $N(k)$, and the number of extra virtual links, $S(k)$, for flow rerouting. We explore three sets of weights. For $\{\alpha_1, \alpha_2, \alpha_3\} = \{1, 0, 0\}$, the reconfiguration overhead is not optimized but load balancing is the focus, corresponding to a load balancing flow migration (LBFM) strategy. For $\{\alpha_1, \alpha_2, \alpha_3\} = \{0, \frac{2}{3}, \frac{1}{3}\}$, $\eta(k)$ is not optimized but reconfiguration overhead reduction is emphasized, corresponding to a minimum overhead flow migration (MOFM) strategy. For $\{\alpha_1, \alpha_2, \alpha_3\} = \{0.4, 0.4, 0.2\}$, both load balancing and reconfiguration overhead reduction are important, corresponding to a hybrid flow migration (HFM) strategy. Fig. 5 shows performance of three strategies with the increase of $\lambda^{(3)}(k)$.

LBFM strategy: It is observed that $\eta(k)$ is dominated by SFC 1 when $\lambda^{(3)}(k)$ is relatively small, showing a flat trend first, but turns to be dominated by SFC 3 with the increase of $\lambda^{(3)}(k)$. Both $N(k)$ and $S(k)$ are high and vary with the



(a) LBFM strategy (b) MOFM strategy (c) HFM strategy

Fig. 5. Performance of three flow migration strategies with the increase of traffic load $\lambda^{(3)}(k)$, for $\eta_U = 0.95$.

traffic load, since they are not optimized. SFCs separate from each other even at a relatively low traffic load to balance traffic loads in the virtual resource pool.

MOFM strategy: Both $N(k)$ and $S(k)$ show a step-wise increasing trend with the increase of $\lambda^{(3)}(k)$. However, $\eta(k)$ is fixed at η_U, since it is not optimized.

HFM strategy: A trade-off among performance metrics is observed. With the increase of $\lambda^{(3)}(k)$, $\eta(k)$ drops sharply when $N(k)$ or $S(k)$ is increased by 1. When $N(k)$ and $S(k)$ stay stable, $\eta(k)$ shows either a linear increasing or a flat trend. Compared with LBFM and MOFM strategies, HFM strategy approaches to the lower bounds of $N(k)$ and $S(k)$ determined by the MOFM strategy, while keeping $\eta(k)$ at a medium level.

B. Average End-to-End Delay Performance

We carry out packet-level simulations using network simulator OMNeT++ to evaluate average E2E delay of SFC 3 with and without flow migration, under the same network and SFC settings as in Subsection VII-A. For average traffic rates, we set $\lambda^{(1)}(k) = \lambda^{(2)}(k) = 200$ packet/s and increase $\lambda^{(3)}(k)$ from 200 packet/s. To verify the effectiveness and accuracy of our flow migration model in the presence of traffic burstiness, not only Poisson but also MMPP packet arrivals are simulated. For each traffic arrival pattern, we collect sufficient packet delay information to estimate the average E2E delay. We use a two-state MMPP model with same transition rate between states and an average rate of $\lambda^{(3)}(k)$. We use "MMPP-q_1-q_2" to represent the MMPP traffic model, where q_1 and q_2 are ratios between state-dependent rates and $\lambda^{(3)}(k)$, with $q_1 + q_2 = 2$. For example, for "MMPP-1.6-0.4" traffic model with $\lambda^{(3)}(k) = 500$ packet/s, the state-dependent rates are 800 packet/s and 200 packet/s respectively.

Fig. 6 shows the average E2E delay without flow migration, in which a flat trend is observed, followed by an exponential increasing trend, for Poisson traffic arrival with increasing rate from 200 packet/s to 425 packet/s. The flat trend corresponds to feasible traffic rates for E2E delay guarantee with local processing resource scaling. Beyond 360 packet/s, local resources are not sufficient, resulting in an exponential increase of E2E delay. Fig. 7 shows the average E2E delay with flow migration. We observe that the E2E delay requirement is satisfied for Poisson traffic with rate in [200, 700] packet/s, inferring that more traffic can be accommodated

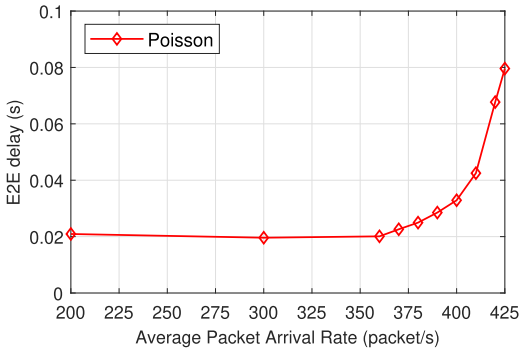


Fig. 6. Average E2E delay without flow migration.

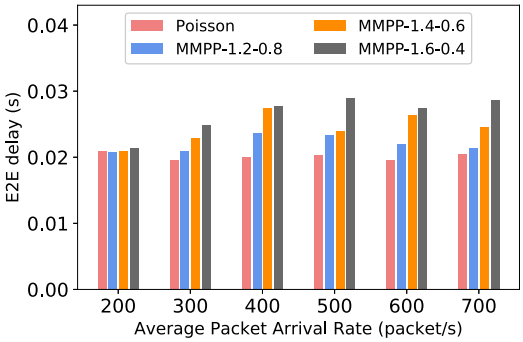


Fig. 7. Average E2E delay comparison with flow migration.

from services which originally share some NFV nodes on their E2E paths, with joint flow migration and processing resource scaling. At a certain average rate, the E2E delay performance degrades with more traffic burstiness. However, even "MMPP-1.6-0.4" for average rate in [400, 700] packet/s with flow migration performs much better than Poisson traffic arrival for average rate larger than 360 packet/s without flow migration, indicating that our flow migration model can accommodate some traffic burstiness without a significant degradation on E2E delay.

C. Comparison Between MIQCP and Heuristic Solutions

1) Cost Sensitivity to Different Weights: Under the 64-node network setup with three SFCs, we compare the MIQCP and heuristic solutions in terms of their cost sensitivity to different weights in (1). With $\alpha_2 = 2\alpha_3$ and $\alpha_1 + \alpha_2 + \alpha_3 = 1$, three cost metrics including the maximum NFV node loading factor, $\eta(k)$, the reconfiguration overhead, $2N(k) + S(k)$,

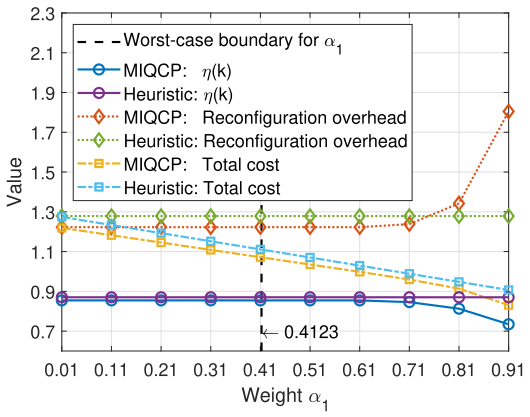


Fig. 8. Costs with respect to weight α_1 in objective function, for $\alpha_2 = 2\alpha_3$.

and the total cost, $\alpha_1\eta(k) + (1-\alpha_1)(2N(k) + S(k))$, are evaluated. The first two costs are partial costs. Although the heuristic solution is in principle insensitive to α_1, we use the same definition of total cost for a fair comparison. The three cost metrics with respect to α_1 for both the MIQCP and heuristic solutions are given in Fig. 8. In both solutions, the total cost approaches to the reconfiguration overhead, for α_1 close to 0, and approaches to the maximum NFV node loading factor, for α_1 close to 1. For the heuristic solution, we observe constant partial costs with respect to α_1, which is consistent with the design principle. For the MIQCP solution, both partial costs show a stable trend for small and medium values of α_1 in a range larger than the theoretical worst-case range $(0, 0.4123)$. For large values of α_1, the reconfiguration overhead of the MIQCP solution increases with α_1, while the maximum NFV node loading factor decreases with α_1, since much more penalization is placed on imbalanced loading than migrations.

2) Cost Efficiency and Time Efficiency: We use a 256-node mesh network to compare the cost and time efficiency between the MIQCP and heuristic solutions. The comparison is performed with fixed weights in (1), under the condition of $\alpha_1\eta_U < \alpha_2$. Specifically, we have $\{\alpha_1, \alpha_2, \alpha_3\} = \{0.4, 0.4, 0.2\}$. Eight groups of experiments are implemented with 10, 15, 20, 25, 30, 35, 40 and 45 SFCs respectively. The total cost and running time for each group are evaluated at different traffic rates from 200 to 740 packet/s. In each experiment, all SFCs have the same traffic rate, denoted by $\lambda(k)$. The initial step size, ζ^0, and the precision, ε, are set to 0.1 and 0.0001 respectively. Fig. 9 shows the average total cost with respect to the number of SFCs ($|\mathcal{R}|$). It shows that the total cost obtained from both solutions increases with $|\mathcal{R}|$. Adding more SFCs tends to increase the number of overloaded NFV nodes, especially when the traffic rate is high and the added SFCs share some NFV nodes with others. Hence, more migrations tend to be triggered with more SFCs added, incurring more cost. Fig. 10 shows the average running time with respect to $|\mathcal{R}|$. We can see an almost exponential increasing trend for the running time of the MIQCP solution. In comparison, the time complexity of the heuristic solution is much less, and the increasing trend is much less significant.

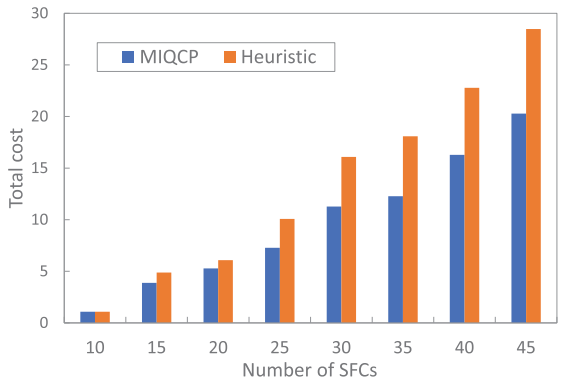


Fig. 9. Total cost with respect to the number of SFCs.

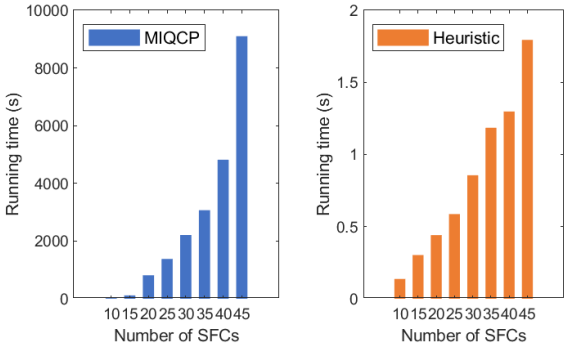


Fig. 10. Running time with respect to the number of SFCs.

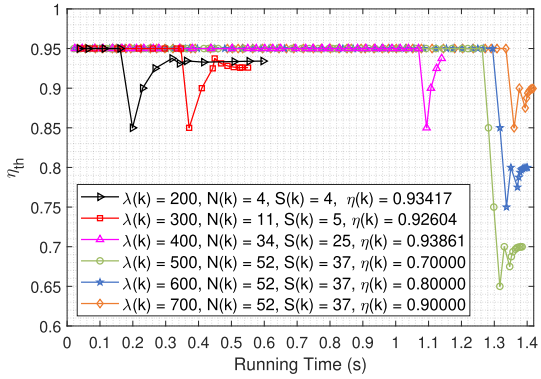


Fig. 11. Threshold update in the heuristic algorithm, for $\eta_U = 0.95$.

D. Convergence of Heuristic Algorithm

To evaluate convergence of the proposed heuristic algorithm, we plot the updating process of the threshold η_{th}, in the 45-SFC experiments with different traffic rates $\lambda(k)$, as shown in Fig. 11, in which $\eta(k)$ is the maximum NFV node loading factor after convergence. On each threshold updating curve corresponding to a specific traffic rate, we see that η_{th} first remains η_U due to several sequential migration decisions at the beginning and then drops with the initial step size of 0.1 until a turning point at the lower bound. After the turning point, the step size is reduced by half with each iteration until it is below the required precision 0.0001. With the increase of traffic rate to 500 packet/s, more migrations happen to gradually decouple the SFCs from each other, and more extra virtual links are observed. When the traffic rate grows larger than 500 packet/s, all SFCs are completely decoupled, with no resource sharing on NFV nodes, thus $N(k)$ and $S(k)$ are

stabilized but $\eta(k)$ increases. When the traffic rate is smaller than 500 packet/s, $\eta(k)$ is close to η_U, while less migrations and extra virtual links are observed, showing a trade-off between load balancing and reconfiguration overhead.

VIII. Conclusion

In this paper, we study a delay-aware flow migration problem for embedded SFCs sharing a common physical network in SDN/NFV-enabled 5G communication systems. A mixed integer optimization problem is formulated based on an abstraction of virtual resource pool, addressing the trade-off between load balancing and reconfiguration overhead. The problem is non-convex and difficult to solve using optimization solvers. Hence, we reformulate a tractable MIQCP problem based on which the optimum of the original problem can be obtained. Numerical results show that the proposed model accommodates more traffic from services, in comparison with an SFC configuration without flow migrations. Moreover, a flow migration strategy with similar priority in load balancing and migration reduction achieves medium load balancing, as compared with flow migration strategies with a priority on either goal. Nevertheless, it achieves approximately as good performance in terms of the reconfiguration overhead as a flow migration strategy which aims at migration reduction. This result indicates the benefit of joint consideration of the two goals. A performance comparison between the MIQCP and heuristic solutions demonstrates the effectiveness and time efficiency of the heuristic solution. Under the assumption that flow migrations do not take place frequently and a time interval is sufficiently large, traffic arrival of SFCs is modeled as Poisson with different rates across time intervals. Correspondingly, the M/M/1 queueing model is employed in both the MIQCP and heuristic solutions, for performance modeling and prediction with predicted traffic variations. In reality, traffic can be highly dynamic especially in 5G new use cases. Motivated by the limitation of Poisson traffic model, we are working on an extension of this work for adaptive flow migration, using promising model-free techniques such as machine learning.

Appendix
Derivation of $\delta_i(k)$

Let $\eta_i^{rh}(k)$ denote a ratio between resources occupied by VNF $V_h^{(r)}$ and resource capacity of NFV node i, given by

$$\eta_i^{rh}(k) = \frac{P_i^{rh}\sigma^{(r)}}{C_i}\left(\lambda^{(r)}(k) + \frac{1}{D^{rh}(k)}\right) x_i^{rh}(k). \quad \text{(A1)}$$

VNF set $\mathcal{V}$ is divided into two subsets, i.e., $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, where $\mathcal{V}_1 = \{(r,h) \in \mathcal{V} | x_i^{rh}(k) = f_2^{(r)} = 1\}$ is a set of VNFs belonging to SFC category III on NFV node i, and $\mathcal{V}_2$ is a set of all other VNFs. Vertical delay scaling is applied to only VNFs in $\mathcal{V}_1$. Before delay scaling, resource usage at NFV node i is composed of three parts, given by

$$\eta_i(k) = \sum_{(r,h)\in\mathcal{V}_1\cup\mathcal{V}_2} \eta_i^{rh}(k) + \sum_{(r,h)\in\mathcal{V}_1\cup\mathcal{V}_2} \frac{w_i(k)x_i^{rh}(k)W}{T_i}. \quad \text{(A2)}$$

The ratio of resources occupied by VNFs in $\mathcal{V}_1$ before vertical delay scaling is given by

$$\sum_{(r,h)\in\mathcal{V}_1} \eta_i^{rh}(k) = \sum_{(r,h)\in\mathcal{V}_1} \frac{P_i^{rh}\sigma^{(r)}}{C_i}\left(\lambda^{(r)}(k) + \frac{1}{D^{rh}(k)}\right). \quad \text{(A3)}$$

For a vertical delay scaling by a positive coefficient $\delta_i(k)$ to increase loading factor of NFV node i from $\eta_i(k)$ to η_{th}, we have the following relationship among parameters, given by

$$\eta_{th} - \sum_{(r,h)\in\mathcal{V}_2} \eta_i^{rh}(k) - \sum_{(r,h)\in\mathcal{V}_1\cup\mathcal{V}_2} \frac{w_i(k)x_i^{rh}(k)W}{T_i} = \sum_{(r,h)\in\mathcal{V}_1} \frac{P_i^{rh}\sigma^{(r)}}{C_i}\left(\lambda^{(r)}(k) + \frac{1}{\delta_i(k)D^{rh}(k)}\right). \quad \text{(A4)}$$

Subtracting (A3) from (A4) and arranging items, we obtain

$$\delta_i(k) = \frac{\sum_{(r,h)\in\mathcal{V}_1} \frac{P_i^{rh}\sigma^{(r)}}{D^{rh}(k)}}{\left[\eta_{th} - \eta_i(k)\right] C_i + \sum_{(r,h)\in\mathcal{V}_1} \frac{P_i^{rh}\sigma^{(r)}}{D^{rh}(k)}} \quad \text{(A5)}$$

which is equivalent to (25).

References

[1] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li, and J. Rao, "Delay-aware flow migration for embedded services in 5G core networks," in *Proc. IEEE Int. Conf. Commun.(ICC)*, May 2019, pp. 1–6.

[2] *Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, Int. Telecommunication Union (ITU), document Rec. ITU-R M.2083-0, Radiocommunication Study Groups, I. Vision, 2015.

[3] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.

[4] X. Li, P. Djukic, and H. Zhang, "Zoning for hierarchical network optimization in software defined networks," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–8.

[5] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 101–109, Jul. 2017.

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

[7] V.-G. Nguyen, A. Brunstrom, K.-J. Grinnemo, and J. Taheri, "SDN/NFV-based mobile packet core network architectures: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1567–1602, 3rd Quart., 2017.

[8] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[9] O. Alhussein *et al.*, "Joint VNF placement and multicast traffic routing in 5G core networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[10] X. Chen, W. Ni, I. B. Collings, X. Wang, and S. Xu, "Automated function placement and online optimization of network functions virtualization," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1225–1237, Feb. 2019.

[11] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Efficient NFV-enabled multicasting in SDNs," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2052–2070, Mar. 2019.

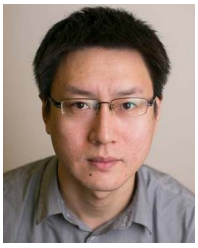
[12] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, Feb. 2019.

[13] 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, Jun. 2018.

[14] (2018). *Gurobi Optimizer Reference Manual*. Accessed: Jan. 15, 2020. [Online]. Available: http://www.gurobi.com
[15] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE Conf. Comput. Commun. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, Nov. 2002, pp. 519–528.
[16] J. Rexford, "Route optimization in IP networks," in *Handbook of Optimization in Telecommunications*. Boston, MA, USA: Springer, 2006, pp. 679–700.
[17] H. Tang, D. Zhou, and D. Chen, "Dynamic network function instance scaling based on traffic forecasting and VNF placement in operator data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 3, pp. 530–543, Mar. 2019.
[18] A. Gember-Jacobson *et al.*, "OpenNF: Enabling innovation in network function control," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, Aug. 2014, pp. 163–174.
[19] M. Peuster and H. Karl, "E-State: Distributed state management in elastic network function deployments," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Jun. 2016, pp. 6–10.
[20] L. Nobach, I. Rimac, V. Hilt, and D. Hausheer, "Statelet-based efficient and seamless NFV state transfer," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 964–977, Dec. 2017.
[21] B. Zhang, P. Zhang, Y. Zhao, Y. Wang, X. Luo, and Y. Jin, "Co-Scaler: Cooperative scaling of software-defined NFV service function chain," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2016, pp. 33–38.
[22] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba, "Elastic virtual network function placement," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 255–260.
[23] 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.
[24] W. Rankothge, F. Le, A. Russo, and J. Lobo, "Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 343–356, Jun. 2017.
[25] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017.
[26] L. Guo, J. Pang, and A. Walid, "Dynamic service function chaining in SDN-enabled networks with middleboxes," in *Proc. IEEE 24th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2016, pp. 1–10.
[27] J. Xia, D. Pang, Z. Cai, M. Xu, and G. Hu, "Reasonably migrating virtual machine in NFV-featured networks," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, Dec. 2016, pp. 361–366.
[28] F. Zhang, G. Liu, X. Fu, and R. Yahyapour, "A survey on virtual machine migration: Challenges, techniques, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1206–1243, 2nd Quart., 2018.
[29] M. Shin, S. Chong, and I. Rhee, "Dual-resource TCP/AQM for processing-constrained networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 435–449, Apr. 2008.
[30] L. Rizzo, M. Carbone, and G. Catalli, "Transparent acceleration of software packet forwarding using netmap," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2471–2479.
[31] S. Garzarella, G. Lettieri, and L. Rizzo, "Virtual device passthrough for high speed VM networking," in *Proc. ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, May 2015, pp. 99–110.
[32] P. Emmerich, D. Raumer, S. Gallenmüller, F. Wohlfart, and G. Carle, "Throughput and latency of virtual switching with Open vSwitch: A quantitative analysis," *J. Netw. Syst. Manage.*, vol. 26, no. 2, pp. 314–338, Apr. 2018.
[33] D. Bhamare, M. Samaka, A. Erbad, R. Jain, L. Gupta, and H. A. Chan, "Optimal virtual network function placement in multi-cloud service function chaining architecture," *Comput. Commun.*, vol. 102, pp. 1–16, Apr. 2017.
[34] F. Ben Jemaa, G. Pujolle, and M. Pariente, "Analytical models for qos-driven vnf placement and provisioning in wireless carrier cloud," in *Proc. 19th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, 2016, pp. 148–155.
[35] Z. Luo, C. Wu, Z. Li, and W. Zhou, "Scaling geo-distributed network function chains: A prediction and learning framework," *IEEE J. Select. Areas Commun.*, vol. 37, no. 8, pp. 1838–1850, Aug. 2019.
[36] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman, 1978.

Kaige Qu (Student Member, IEEE) received the B.S. degree in communication engineering from Shandong University, Jinan, China, in 2013, and the M.S. degrees in integrated circuits engineering and electrical engineering from Tsinghua University, Beijing, China, and KU Leuven, Leuven, Belgium, in 2016. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her research interests include resource allocation in SDN/NFV-enabled networks, 5G and beyond, and machine learning for future networking.

Weihua Zhuang (Fellow, IEEE) has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, since 1993, where she is currently a Professor and a Tier I Canada Research Chair of wireless communication networks. She is a fellow of the Royal Society of Canada, the Canadian Academy of Engineering, and the Engineering Institute of Canada. She is an Elected Member of the Board of Governors and VP Publications of the IEEE Vehicular Technology Society. She was a recipient of the 2017 Technical Recognition Award from the IEEE Communications Society Ad Hoc and Sensor Networks Technical Committee, and several best paper awards from IEEE conferences. She was the Technical Program Chair/Co-Chair of the IEEE VTC Fall 2016 and Fall 2017, the Editor-in-Chief of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2013, and an IEEE Communications Society Distinguished Lecturer from 2008 to 2011.



Qiang Ye (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2016. He had been with the Department of Electrical and Computer Engineering, University of Waterloo, as a Post-Doctoral Fellow and then a Research Associate from December 2016 to September 2019. He has been an Assistant Professor with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN, USA, since September 2019. His current research interests include 5G networks, software-defined networking and network function virtualization, network slicing, artificial intelligence and machine learning for future networking, protocol design, and end-to-end performance analysis for the Internet of Things.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, social networks, 5G and beyond, and vehicular ad hoc and sensor networks. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the R.A. Fessenden Award in 2019 from IEEE, Canada, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015, and Education Award in 2017 from the IEEE Communications Society. He has also received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award five times from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom'16, the IEEE Infocom'14, the IEEE VTC'10 Fall, the IEEE Globecom'07, the Symposia Chair for the IEEE ICC'10, the Tutorial Chair for the IEEE VTC'11 Spring, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He was the Editor-in-Chief of IEEE INTERNET OF THINGS JOURNAL from 2017 to 2019. He is the Vice President on Publications of the IEEE Communications Society.

Xu Li received the B.Sc. degree from Jilin University, China, in 1998, the M.Sc. degree from the University of Ottawa in 2005, and the Ph.D. degree from Carleton University in 2008, all in computer science. He worked as a Research Scientist (with tenure) with Inria, France. He is currently a Senior Principal Researcher with Huawei Technologies Canada, Inc. He contributed extensively to the development of 3GPP 5G standards through more than 90 standard proposals. He has published more than 100 refereed scientific articles and is holding more than 40 issued U.S. patents. His current research interests are focused in 5G and beyond.

Jaya Rao received the B.S. and M.S. degrees in electrical engineering from the University of Buffalo, New York, in 2001 and 2004, respectively, and the Ph.D. degree from the University of Calgary, Canada, in 2014. From 2004 to 2010, he was a Research Engineer with Motorola, Inc. Since joining Huawei in 2014, he has worked on research and design of CIoT, URLLC- and V2X-based solutions in 5G New Radio. He is currently a Senior Research Engineer with Huawei Technologies Canada, Inc., Ottawa. He has contributed for Huawei at 3GPP RAN WG2, RAN WG3, and SA2 meetings on topics related to URLLC, network slicing, mobility management, and session management. He was a recipient of the Best Paper Award at IEEE WCNC 2014.