End-to-End Delay Modeling for Embedded VNF Chains in 5G Core Networks

Qiang Ye, *Member, IEEE*, Weihua Zhuang, *Fellow, IEEE*, Xu Li, and Jaya Rao, *Member, IEEE*

***Abstract*—In this paper, an analytical end-to-end (E2E) packet delay modeling is established for multiple traffic flows traversing an embedded virtual network function (VNF) chain in fifth generation communication networks. The dominant-resource generalized processing sharing is employed to allocate both computing and transmission resources among flows at each network function virtualization (NFV) node to achieve dominant-resource fair allocation and high resource utilization. A tandem queueing model is developed to characterize packets of multiple flows passing through an NFV node and its outgoing transmission link. For analysis tractability, we decouple packet processing (and transmission) of different flows in the modeling and determine average packet processing and transmission rates of each flow as approximated service rates. An M/D/1 queueing model is developed to calculate packet delay for each flow at the first NFV node. Based on the analysis of packet interarrival time at the subsequent NFV node, we adopt an M/D/1 queueing model as an approximation to evaluate the average packet delay for each flow at each subsequent NFV node. The queueing model is proved to achieve more accurate delay evaluation than that using a G/D/1 queueing model. Packet transmission delay on each embedded virtual link between consecutive NFV nodes is also derived for E2E delay calculation. Extensive simulation results demonstrate the accuracy of our proposed E2E packet delay modeling, upon which delay-aware VNF chain embedding can be achieved.**

***Index Terms*—Bi-resource allocation, CPU and bandwidth resources, dominant-resource generalized processor sharing (DR-GPS), embedded virtual network function (VNF) chains, end-to-end (E2E) delay modeling, network function virtualization (NFV), rate decoupling, software-defined networking (SDN), tandem queueing model.**

I. Introduction

FUTURE communication networks are expected to provide customized delay-sensitive end-to-end (E2E) service deliveries [1] (e.g., video streaming and machine-to-machine communications) with fine-grained quality-of-service (QoS) for Internet-of-Things (IoT) [2]–[6]. Typical IoT application scenarios include remote control for smart homing, smart sensing [7], large-scale mobile social networking [8], industrial automation [9], high-definition video conferencing, and intelligent transportation systems [10], [11]. To support diversified applications and use cases, network servers providing different functions (e.g., classifiers, firewalls, and proxies) are required to be augmented in a large scale to fulfill customized service requirements in both wireless and core network domains. However, the increasingly densified network deployment significantly expands installation and operational cost of the infrastructure. Network function virtualization (NFV) [12]–[15] provides a promising solution to reduce the deployment cost and realize flexible function placement and service customization. With NFV, network functions are decoupled from function-specific servers, softwarized as virtual network functions (VNFs), and placed on general-purpose programmable servers (also called NFV nodes [13], [16]). Specifically, through a resource virtualization platform [17], computing resources (i.e., CPU cores) for task processing on each NFV node are virtualized as virtual CPU cores, upon which virtual machines (VMs) are installed. Then, VNFs are programmed on VMs of different NFV nodes at different network locations to achieve high resource utilization.

For the evolving network paradigm, the backbone core network consists of a combination of network switches and NFV nodes interconnected via high-speed wired transmission links. NFV nodes hosting and operating VNFs are introduced to improve service provisioning and resource utilization. A set of VNFs and the virtual links connecting them constitute a logic VNF chain, referred to as service function chain [18], representing a specific sequence of network functions that a traffic flow[1] requires to traverse for E2E service provisioning. All VNF chains are managed by a virtualization controller and are placed onto the physical substrate network, with each VNF embedded on an NFV node and virtual links represented by transmission links and network switches. This process is known as *VNF chain embedding* [13], [14], [19], [20]. Note that the virtualization controller is software-defined networking (SDN) enabled [21], with all control functions decoupled from the underlying physical network [22], [23]. Therefore, the controller has direct control (programmability) on all VNFs to enhance resource utilization via traffic balancing and VM migration [24], [25].

E2E packet delay of a delay-sensitive service flow traversing an embedded VNF chain is a main metric indicating the

Manuscript received April 10, 2018; revised June 13, 2018; accepted June 30, 2018. Date of publication July 9, 2018; date of current version February 25, 2019. This work was supported by research grants from Huawei Technologies Canada and from the Natural Sciences and Engineering Research Council (NSERC) of Canada. *(Corresponding author: Qiang Ye.)*

Q. Ye and W. Zhuang are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: q6ye@uwaterloo.ca; wzhuang@uwaterloo.ca).

X. Li and J. Rao are with Huawei Ottawa Research & Development Centre, Huawei Technologies Canada Inc., Ottawa, ON K2K 3J1, Canada (e-mail: xu.lica@huawei.com; jaya.rao@huawei.com).

Digital Object Identifier 10.1109/JIOT.2018.2853708

[1] A traffic (service) flow refers to an aggregation of packets belonging to the same service type for the same source and destination node pair in the backbone network.

embedding performance. Existing research works investigate how to achieve optimal embedding of VNF chains on the core network to minimize the deployment, operational, and delay violation cost under physical resource constraints and flow conservation constraints. E2E packet delay of each traffic flow is calculated as a summation of packet transmission delays on each physical link, without considering packet processing delay due to CPU processing on NFV nodes [13], [14], [26]. However, each packet from different traffic flows passing through an NFV node usually requires different amounts of CPU processing time on the NFV node and packet transmission time on the outgoing link in sequence [27], [28]. Depending on the type of VNF each flow traverses, some flows' packets with large headers bottleneck on CPU processing, whereas packets of other flows having large packet payload sizes demand more transmission time. In addition, the packet arrival process of a flow at an NFV node correlates with packet processing and transmission at its preceding NFV nodes, which makes E2E delay analysis difficult. Therefore, how to develop an analytical model to evaluate the delay that each packet of a flow experiences when passing through an embedded VNF chain, including packet queueing delay, packet processing delay on NFV nodes, and packet transmission delay on links, is a challenging research issue. For VNF chain embedding, there is a tradeoff between E2E delay satisfaction and reducing the network cost. Different VNF chains are often embedded on a common network path with multiple VNF instances operated on an NFV node to improve resource utilization and reduce the deployment and operational cost of network functions and links. On the other hand, sharing a set of physical resources with other flows on NFV nodes and links may degrade the delay of one individual flow. Thus, modeling E2E packet delay of each flow is essential to achieve delay-aware VNF chain embedding. Since traffic flows traversing each NFV node demonstrate discrepant "dominant" resource consumption on either CPU or link bandwidth, how to allocate the two resources among traffic flows to guarantee allocation fairness and achieve high resource utilization, referred to as *bi-resource allocation*, needs investigation and is required for E2E delay modeling.

In this paper, we employ dominant-resource generalized processor sharing (DR-GPS) [28] as the bi-resource allocation scheme among traffic flows sharing resources at each NFV node. The DR-GPS balances the tradeoff between service isolation and high resource utilization, and maintains dominant-resource fairness (DRF) [29] among flows. Then, we model packet delay of a flow passing through NFV nodes and physical links (and switches) of an embedded VNF chain. The contributions of this paper are twofold.

1) With DR-GPS, we establish a tandem queueing model to extract the process of each flow going through CPU processing and link transmission at the first NFV node. To remove the rate coupling effect among flows, we derive average processing rate as an approximation on instantaneous processing rate for each flow with the consideration of resource multiplexing. An M/D/1 queueing model is then developed for each decoupled processing queue to determine the average packet processing delay. Based on the analysis of packet departure process from each decoupled processing queue, the decoupled transmission rate is derived for each flow.
2) We analyze packet arrival process and then remove rate coupling effect of each flow traversing the subsequent NFV node. To eliminate the dependence of packet processing and packet transmission between consecutive NFV nodes, the arrival process of each flow at the subsequent NFV node is approximated as a Poisson process and an M/D/1 queueing model is employed to calculate the average delay for packet processing. It is proved that the average packet queueing delay for CPU processing based on the approximated M/D/1 queueing model is an improved upper bound over that calculated upon a G/D/1 queueing model.

The rest of this paper is organized as follows. Existing studies on E2E packet delay modeling for embedded VNF chains are reviewed in Section II. The system model under consideration is described in Section III. In Section IV, we present a bi-resource allocation scheme employed for flows traversing an NFV node. The E2E delay modeling for packet flows going through an embedded VNF chain is established in Section V. In Section VI, numerical results are presented to demonstrate the accuracy of the proposed delay analytical framework and its effectiveness for achieving delay-aware VNF chain embedding. Lastly, we draw the conclusions in Section VII. Main parameters and symbols throughout this paper are summarized in Table I.

II. Related Work

In most existing studies, E2E packet delay for an embedded VNF chain is modeled as a summation of transmission delays when every packet of a flow traverses each embedded physical link, without considering packet processing delay associated with VNFs. In [13] and [14], delay violation penalty is expressed as a function of E2E packet transmission delay for each traffic flow, which indicates the cost if the delay requirement for packets traversing an embedded VNF chain is violated. A delay violation cost minimization problem is then formulated to determine an optimal embedded physical network path that achieves minimum E2E packet transmission delay for each flow. In [26], the total delay when a packet is routed between consecutive virtual nodes consists of packet processing delay, packet queueing delay, and packet transmission delay, and is determined using network traffic measuring tools instead of analytical modeling. The E2E delay for packets going through each source–destination node pair in an embedded virtual network is then calculated to achieve QoS-aware multicast virtual network embedding. For embedded VNF chains, how CPU and bandwidth resources are shared among multiple flows traversing each NFV node determines the fractions of processing and transmission rates allocated to each flow, and thus affects E2E packet delay calculation. Multiresource sharing is studied in a data center (DC) environment [27]–[29], where dominant-resource fair allocation is adopted to equalize the dominant resource shares of multiple flows, while achieving high resource utilization. However, how to model E2E packet delay for each flow traversing a set of

TABLE I
MAIN PARAMETERS AND SYMBOLS

Symbol	Definition
I	Set of traffic flows traversing a common embedded physical network path
N_z	z th NFV node along one embedded VNF chain
n_z	Number of network switches and physical links between N_z and N_{z+1}
λ_i	Packet arrival rate of traffic flow i at N_1
$\tau_{i,1}/\tau_{i,2}$	Time consumption of each packet of flow i for CPU processing/link transmission at N_1
$C_{i,1}/C_{i,2}$	Maximum packet processing/transmission rate allocated to flow i at N_1
$c_{i,1}/c_{i,2}$	Allocated packet processing/transmission rate out of $C_{i,1}/C_{i,2}$ to flow i
$h_{i,1}/h_{i,2}$	Fraction of CPU/link bandwidth resources allocated to flow i at N_1
B	One of the flow combinations for a backlogged flow set out of I
$\mu_{i,1}/\mu_{i,2}$	Decoupled packet processing/transmission rate for flow i at N_1
$\varrho_{i,1}$	Non-empty probability of flow i' s processing queue at N_1
$\mu'_{i,1}/\mu'_{i,2}$	Decoupled packet processing/transmission rate for flow i at N_2
$\varrho'_{i,1}$	Non-empty probability of flow i' s processing queue at N_2
Y_i	Inter-departure time of successive packets of flow i from the decoupled processing at N_1
Z_i	Packet inter-departure time for flow i passing though link transmission at N_1
$D_{i,z}$	Average packet delay for flow i passing through z th NFV node
$D^{(f)}_{i,z}$	Total packet transmission delay for flow i traversing n_z switches and links

DC nodes needs further investigation [28]. An analytical E2E delay model for packets passing through each embedded VNF chain is established in [30], where an independent M/M/1/K queueing model is employed to evaluate packet delay at each VNF. However, packet interarrival time at a subsequent VNF correlates with the processing and transmission rates at its preceding VNF, making packet delay calculation unique at each VNF. In addition, the assumption of exponential distribution on packet processing time needs justification. Existing studies establish analytical delay models for packets of a traffic flow going through a set of OpenFlow network switches in SDN [31], [32]. Since control functions are migrated from each network switch to the SDN controller where processing resources are consumed for making routing decisions, the network switches are simplified with only packet forwarding functions. E2E packet delay for a flow passing through a sequence of network switches is determined based on an M/M/1 queueing network modeling.

Overall, developing an accurate analytical E2E delay model for each traffic flow traversing an embedded VNF chain is challenging and of importance for achieving delay-aware VNF chain embedding.

III. SYSTEM MODEL

A. Embedded VNF Chains

An aggregated data traffic flow from the wireless network domain, belonging to an identical service type, is required to traverse a sequence of VNFs in the core network to fulfill certain service requirements. Each VNF is embedded and operated on an NFV node, and each virtual link represents a set of transmission links and network switches. Multiple VNF chains can be embedded on a common network path to improve resource utilization and reduce network deployment and operational cost. We consider a set, I, of traffic flows traversing different logic VNF chains over a common embedded physical path. In Fig. 1, two flows i and $j\,(\in I)$, representing two logic VNF chains $f_1 \rightarrow f_3$ and $f_1 \rightarrow f_2$, respectively, traverse one embedded network path and share the same physical resources. At the service level, we have flow i traversing a firewall function and a domain name system (DNS) function sequentially to fulfill a secured DNS service request, and flow j traversing a firewall function and an intrusion detection system (IDS) for secured E2E data streaming. At the network level, flow i goes through the first NFV node N_1 operating VNF f_1 and transmission link L_0, and are then forwarded by n_1 network switches $\{R_1, \ldots, R_k, \ldots, R_{n_1}\}$ and n_1 transmission links $\{L_1, \ldots, L_k, \ldots, L_{n_1}\}$ in between before reaching the second NFV node, N_2, operating VNF f_3; flow j traverses the same physical path but passes through VNF f_2 at the second NFV node N_2. After passing through N_2, traffic flows i and j are forwarded by a sequence of n_2 switches and n_2 links (not depicted in Fig. 1 for brevity) to reach the destination node in the core network. For a general case, we have the set I of flows traversing and sharing an embedded physical path, with m NFV nodes denoted by $N_z\ (z = 1, 2, \ldots, m)$ and n_z pairs of network switches and physical links forwarding traffic between NFV nodes N_z and N_{z+1} before reaching the destination node. With NFV, different VNFs can be flexibly orchestrated and installed at appropriate NFV nodes to enhance traffic balancing and reduce deployment cost of network infrastructure. When a traffic flow passes through an NFV node, each packet of the flow first requires a CPU time for packet processing, after which the processed packet is allocated link bandwidth resources for transmission [28]. The total amount of CPU time is assumed infinitely divisible [27], [28] on each NFV node and needs to be properly shared among traffic flows passing through, and the bandwidth resources on transmission links are also shared among traffic flows.

B. Traffic Model

Packet arrivals of flow i at the first NFV node, N_1, are modeled as a Poisson process with arrival rate λ_i. For resource requirements of a flow, we define *time profile* for flow i passing through N_1 as a 2-D time vector $[\tau_{i,1}, \tau_{i,2}]$, indicating that every packet of flow i requires $\tau_{i,1}$ time for CPU processing and $\tau_{i,2}$ time for transmission if all CPU time on N_1 and link bandwidth resources on L_0 are allocated to flow i [28]. Correspondingly, the rate vector $[C_{i,1}, C_{i,2}]$ (in the unit of

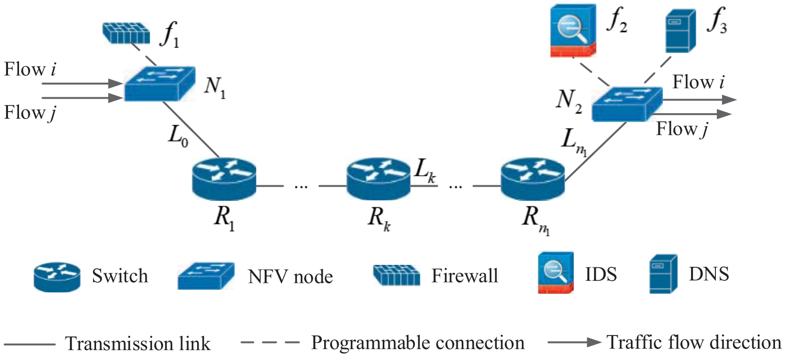

Fig. 1. Two embedded VNF chains sharing a common physical network path.

packet per second) for flow i is the reciprocal of the time profile, $C_{i,1} = (1/\tau_{i,1})$ and $C_{i,2} = (1/\tau_{i,2})$. Service flows with different packet structures often have discrepant time profile when passing through an NFV node. For example, when going through a firewall function, service flows carrying small packets with a large header size, such as DNS request packets, are more CPU time demanding, whereas other data traffic flows, e.g., video traffic, with a large packet size will require more time for packet transmissions but less time on packet processing. Therefore, service flows always have a more critical resource consumption on either CPU time or link bandwidth, which is referred to as *dominant resource*. With a given set of maximum available CPU resources and link bandwidth resources on an NFV node and its outgoing link, time profile for different traffic flows sharing one NFV node can be very different. The maximum available CPU time on N_1 and maximum link bandwidth on L_0 are shared among flows. Suppose flow i is allocated packet processing rate $c_{i,1}$ out of the maximum processing rate $C_{i,1}$, and packet transmission rate $c_{i,2}$ out of the maximum transmission rate $C_{i,2}$. Note that when the CPU time is shared among multiple flows, there can be some overhead time as the total CPU resources switch among flows for different processing tasks. Existing studies show that this CPU switching overhead only becomes obvious when the traffic of each flow is saturated with a high percentage of CPU utilization (i.e., CPU cores are frequently interrupted for switching tasks) [18], [33]. Considering only nonsaturation traffic, we assume that the allocated processing rate $c_{i,1}$ for flow $i\,(\in I)$ varies linearly with its occupied fraction of CPU time (i.e., the useful fraction of CPU usage). Thus, we denote the fraction of CPU resources allocated to flow i by $h_{i,1} = (c_{i,1}/C_{i,1})$ and the fraction of link bandwidth resources by $h_{i,2} = (c_{i,2}/C_{i,2})$.

IV. Bi-Resource Allocation Discipline

A. Dominant-Resource Generalized Processor Sharing

When different service flows multiplex at a common NFV node, we want to determine how CPU and link bandwidth resources should be shared among the traffic flows to achieve high utilization on each resource type and, at the same time, maintain a (weighted) fair allocation among the services. Since service flows can have different dominant resources, the bi-resource sharing becomes more challenging than single resource allocation, to balance between high resource utilization and fair allocation for both resource types.

The generalized processor sharing (GPS) discipline is a benchmark fluid-flow (i.e., with resources being infinitely divisible) based single resource allocation model for integrated service networks in the traditional communication networks [34], [35]. Each service flow, say flow i, at a common GPS server (e.g., a network switch) is assigned a positive value, φ_i, indicating its priority in bandwidth allocation. The GPS server guarantees that the allocated transmission rate g_i for flow i satisfies

$$g_i \geq \frac{\varphi_i}{\sum\limits_{i\in I}\varphi_i}G \tag{1}$$

where G is the maximum service rate of the GPS server. Note that the inequality sign in (1) holds when some flows in I do not have packets to transmit, and thus more resources can be allocated among any backlogged flows. Therefore, GPS has the properties of achieving both service isolation and a high resource multiplexing gain among flows.

However, if GPS is directly applied in the bi-resource context (i.e., bi-resource GPS), it is difficult to simultaneously maintain fair allocation for both CPU time and link bandwidth and achieve high system performance. Consider flow i and flow j, with time profiles $[\tau_{i,1}, \tau_{i,2}]$ and $[\tau_{j,1}, \tau_{j,2}]$, respectively, traversing NFV node N_1, with the same service priority for fair resource sharing. Assume $\tau_{i,1} > \tau_{i,2}$ and $\tau_{j,1} < \tau_{j,2}$. If we apply the bi-resource GPS, both the maximum processing and transmission rates are equally divided for the two flows. Consequently, the performance of both flows traversing N_1 is not maximized. For flow i, due to unbalanced time profiles, the allocated link transmission rate $c_{i,2}$ is larger than the processing rate $c_{i,1}$, leading to resource wastage on link transmission; the situation reverses for flow j, where packets are accumulated for transmissions, causing an increase of queueing delay. Therefore, to improve the system performance, a basic principle [28] is that the fractions, $h_{i,1}$ and $h_{i,2}$, of CPU and bandwidth resources allocated to any flow $i\,(\in I)$ should be in the same proportion as its time profile, i.e., $(h_{i,1}/h_{i,2}) = (\tau_{i,1}/\tau_{i,2})$, to guarantee the allocated processing rate be equalized with the transmission rate, i.e., $c_{i,1} = c_{i,2}$. In such a way, queueing delay before packet transmissions of each flow can be eliminated. However, with this basic principle, if we apply GPS on one of the two resources (i.e., single-resource GPS with equalized processing and transmission rates), the allocation of the other type of resources among traffic flows continues to be unbalanced due to the discrepancy of time profiles among different flows.

To maximize the tradeoff between high performance and fair resource allocation for each traffic flow, we employ a DR-GPS scheme [28] for the bi-resource allocation. The DR-GPS combines the concepts of DRF [29] and GPS, in which the fractions of allocated dominant resources among different backlogged flows are equalized based on service priority, and the other type of resources are allocated to ensure the processing rate equal transmission rate for each flow (the basic principle applies). When some flows do not have backlogged packets for processing, their allocated resources are redistributed among other backlogged flows if any. Since there are a finite number of flow combinations to form a backlogged flow set out of I, we denote $B\,(\in I)$ as one of the flow combinations for a backlogged flow set. Each backlogged flow has a dominant resource consumption in either packet processing or packet transmission. We mathematically formulate DR-GPS in (P1) when the set, $I\,(|I| \geq 1)$, of traffic flows traverse N_1, where $|\cdot|$ is the set cardinality, as follows:

$$(\text{P1}):\ \max\left\{h_{1,1}, \ldots, h_{i,1}, \ldots, h_{j,1}, \ldots, h_{|B|,1}\right\}$$

$$\text{s.t.}\ \begin{cases} \sum_{i \in B} h_{i,1} \leq 1 & \text{(2a)} \\ \sum_{i \in B} h_{i,2} \leq 1 & \text{(2b)} \\ h_{i,1} = \dfrac{\tau_{i,1}}{\tau_{i,2}} h_{i,2} & \text{(2c)} \\ \dfrac{h_{i,d}}{w_i} = \dfrac{h_{j,d}}{w_j} \qquad \forall i; j \in B & \text{(2d)} \\ h_{i,1}, h_{i,2}, w_i, w_j \in [0, 1]. & \text{(2e)} \end{cases}$$

In (P1), w_i and w_j are weights of resource allocation to represent service priority for flow i and flow j, respectively, and $h_{i,d}$ is the fraction of occupied dominant resources of flow i, which is either $h_{i,1}$ or $h_{i,2}$. Constraint (2c) guarantees $c_{i,1} = c_{i,2}$; Constraint (2d) equalizes the fractions of allocated dominant resources among the backlogged flows. Problem (P1) is a linear programming problem and can be solved efficiently to obtain the optimal solutions of $h_{i,1}$ and $h_{i,2}$ for any flow i.

The DR-GPS has properties of 1) service isolation by guaranteeing a service rate in (1) to each flow and 2) work conservation by fully utilizing at least one of the two types of resources in serving the backlogged flows [28], [36], [37]. Although the queueing delay for packet transmissions is reduced by employing the DR-GPS scheme, the total packet delay[2] for each flow traversing the same NFV node should be evaluated. With GPS [34], [35], the process of multiple traffic flows passing through common NFV node N_1 is extracted as a tandem queueing model, shown in Fig. 2. The total packet delay is the summation of packet queueing delay before processing, packet processing delay and packet transmission delay. In the following, we develop an analytical model to evaluate the total packet delay for each traffic flow traversing N_1. Based on the delay modeling for flows traversing the first NFV node, the E2E packet delay for traffic flows passing through the embedded VNF chains can be analyzed.

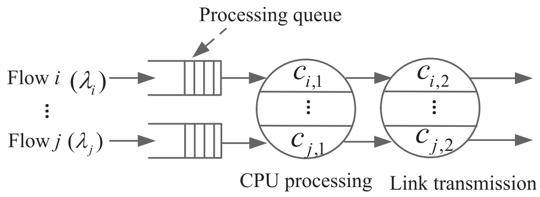

Fig. 2. Queueing model for multiple traffic flows traversing N_1.

V. End-to-End Delay Analysis

In this section, we first analyze the total packet delay for each traffic flow traversing the first NFV node, and then extend the delay model to evaluate the E2E delay for traffic lows passing through a sequence of NFV nodes of an embedded VNF chain.

A. Rate Decoupling

The main difficulty in analyzing the packet delay for each flow is that both the processing and transmission rates of each flow depend on the backlog status of other flows at the same NFV node. For the case of two flows, when one of the flows has an empty processing queue, its processing and transmission resources are given to the other backlogged flow to exploit the traffic multiplexing gain. Therefore, the processing (transmission) rate of each flow switches between two deterministic rate values, depending on the status of the other flow. For a general case where we have the set, I, of multiplexing flows which includes a set, B, of backlogged flows excluding the tagged flow i, the processing rate $c_{i,1}$ of flow i can be determined by solving (P1), based on set B of backlogged flows. We further denote B as B_r where $r = 1, 2, \ldots, \binom{|I|}{|B|}$, representing one of $\binom{|I|}{|B|}$ combinations of $|B|$ backlogged flows. Thus, $c_{i,1}$ changes with B_r. This correlation of queue status among flows leads to the processing rate of each flow jumping over discrete deterministic values, making the total packet delay analysis complex.

To remove the coupling effect of instantaneous processing rate of one flow fluctuating with the backlog status of other flows, we first determine the average processing rate $\mu_{i,1}$ for flow i, with the consideration of resource multiplexing among different flows, i.e., nonempty probabilities of processing queues from all other flows, through a set of high-order nonlinear equations given by

$$\begin{cases} \mu_{i,1} = \sum\limits_{|B|=0}^{|I|-1} \sum\limits_{r=1}^{M} \prod_{l \in B_r} \varrho_{l,1} \prod_{k \in \overline{B_r}} (1 - \varrho_{k,1}) c_{i,1} \\ \varrho_{i,1} = \frac{\lambda_i}{\mu_{i,1}} \qquad \forall i \in I. \end{cases} \tag{3}$$

[2] Total packet delay refers to the duration from the instant that a packet of one traffic flow reaches to the processing queue of the NFV node to the instant when it is transmitted out of the NFV node over a physical link.

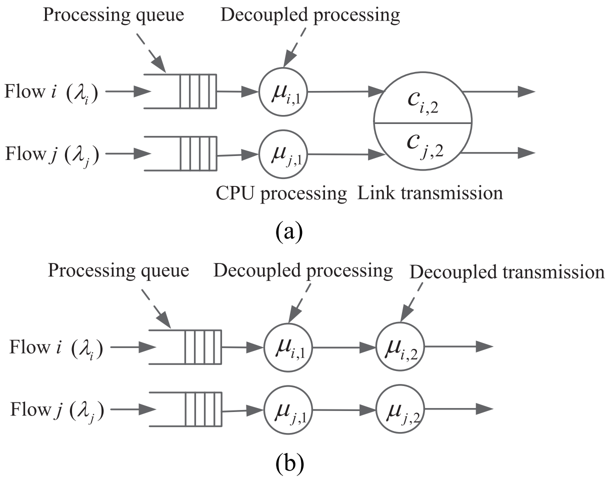

Fig. 3. Queueing model for (a) decoupled packet processing and (b) decoupled packet processing and transmission.

In (3), $M = \binom{|I|-1}{|B|}$, $\overline{B_r} = I\backslash\{i \bigcup B_r\}$, and $\varrho_{i,1}$ is the nonempty probability of processing queue for flow i at N_1. Given packet arrival rate for any flow in I, (3) has $2|I|$ equations with $2|I|$ variables and can be solved numerically for the set of average processing rates of each flow. For analysis tractability, we use the average processing rate, $\mu_{i,1}$, as an approximation for the instantaneous processing rate $c_{i,1}$ to transform the original correlated processing system to a number of $|I|$ uncorrelated processing queues. A case of two traffic flows at an NFV node is demonstrated in Fig. 3. With the decoupled deterministic processing rates, packet processing for each flow can be modeled as an M/D/1 queueing process, upon which we can further calculate both the decoupled packet processing delay and the total packet delay for processing. The accuracy of processing rate decoupling is verified through extensive simulations presented in Section VI.

In Fig. 3(a), the instantaneous link transmission rate for each flow is also correlated with the backlog status of other flows. To remove the transmission rate correlation for packet delay analysis at the first NFV node, we analyze the packet arrival process of each flow at link transmission (i.e., the departure process from the preceding packet processing) in the following.

B. Queue Modeling at the First NFV Node

We study the packet departure process from each decoupled processing at N_1. Taking flow i in Fig. 3 as an example, we focus on the interdeparture time between two successively departed packets from the decoupled processing. As indicated in [38] and [39], for an M/D/1 queueing system, the queue occupancy distribution in a steady state seen by a departing packet is the same as that seen by an arriving packet due to the Poisson characteristic of the arriving process. Therefore, a departing packet from the decoupled processing sees the same empty probability of the processing queue as an arriving packet. Let random variable Y_i be the interdeparture time of successive packets of flow i departing from the decoupled processing at N_1. If the lth departing packet sees a nonempty queue, then $Y_i = T_i$, where $T_i = (1/\mu_{i,1})$ is the decoupled processing time for a packet of flow i; if the departing packet sees an empty queue upon its departure, $Y_i = X_i + T_i$, where random variable X_i denotes the duration from the time of the lth packet departure of flow i to the time of $(l+1)$th packet arrival. Because of the memoryless property of a Poisson arrival process, X_i has the same exponential distribution as packet interarrival time with parameter λ_i. Therefore, the probability density function (PDF) of Y_i, $\xi_{Y_i}(t)$, can be calculated as

$$\xi_{Y_i}(t) = \big(1 - \varrho_{i,1}\big)\xi_{(X_i+T_i)}(t) + \varrho_{i,1}\xi_{T_i}(t) \tag{4}$$

where $\varrho_{i,1} = (\lambda_i/\mu_{i,1})$. Since X_i and T_i are independent random variables, the PDF of $X_i + T_i$ is the convolution of the PDFs of X_i and T_i. Thus, (4) is further derived as

$$\begin{aligned}\xi_{Y_i}(t) &= \left(1 - \frac{\lambda_i}{\mu_{i,1}}\right)\big[\xi_{X_i}(t) \circledast \xi_{T_i}(t)\big] + \frac{\lambda_i}{\mu_{i,1}}\xi_{T_i}(t)\\ &= \left(1 - \frac{\lambda_i}{\mu_{i,1}}\right)\big[\lambda_i e^{-\lambda_i t}u(t) \circledast \delta(t - T_i)\big]\\ &\quad + \frac{\lambda_i}{\mu_{i,1}}\delta(t - T_i)\\ &= \frac{\lambda_i\big(\mu_{i,1} - \lambda_i\big)}{\mu_{i,1}}e^{-\lambda_i(t-T_i)}u(t - T_i) + \frac{\lambda_i}{\mu_{i,1}}\delta(t - T_i) \end{aligned} \tag{5}$$

where $u(t)$ is the unit step function, $\delta(t)$ is the Dirac delta function, and $\circledast$ is the convolution operator. From (5), the cumulative distribution function (CDF) of Y_i is given by

$$F_{Y_i}(t) = \left[1 - \left(1 - \frac{\lambda_i}{\mu_{i,1}}\right)e^{-\lambda_i(t-T_i)}\right]u(t - T_i). \tag{6}$$

Based on (5) and (6), both mean and variance of Y_i can be calculated as

$$E[Y_i] = \frac{1}{\lambda_i} \quad \text{and} \quad D[Y_i] = \frac{1}{\lambda_i^2} - \frac{1}{{\mu_{i,1}}^2}. \tag{7}$$

From (6) and (7), we observe that, when λ_i is small, the departure process, delayed by the service time T_i, approaches the Poisson arrival process with parameter λ_i; when λ_i is increased to approach $\mu_{i,1}$, the departure process approaches the deterministic process with rate $\mu_{i,1}$.

Since the packet departure rate from each decoupled processing is the same as packet arrival rate at each processing queue, the decoupled transmission rate $\mu_{i,2}$ for flow i is the same as $\mu_{i,1}$, by solving the set of (3). At this point, we have a complete decoupled queueing model of both packet processing and packet transmission for flow i traversing the first NFV node N_1, shown in Fig. 3(b). The average total packet delay for flow i is determined by

$$D_{i,1} = \frac{1}{\mu_{i,1}} + \frac{\lambda_i}{2{\mu_{i,1}}^2(1 - \varrho_{i,1})} + \frac{1}{\mu_{i,2}}. \tag{8}$$

Before modeling the delay for flows going through the second NFV node, N_2, we analyze the departure process for packet transmissions of each flow at N_1. Similar to the analysis of packet departure process from the decoupled processing for flow i at N_1, we set time 0 as the instant when the lth packet departs from the processing queue and reaches the transmitting queue for immediate packet transmission. Since we have

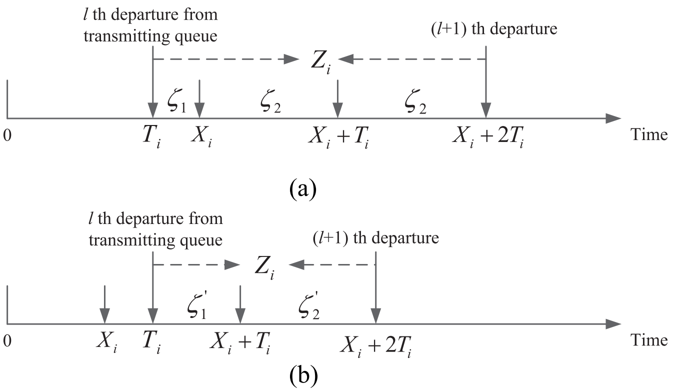

Fig. 4. Composition of Z_i under different cases. (a) $X_i > T_i$. (b) $X_i \le T_i$.

$\mu_{i,1} = \mu_{i,2}$, T_i also indicates packet transmission time, i.e., $T_i = (1/\mu_{i,2})$. Let Z_i denote packet interdeparture time for flow i passing though link transmission. If the lth departing packet from the processing queue sees a nonempty queue, we have $Z_i = T_i$; otherwise, the following two cases apply, as illustrated in Fig. 4.

Case 1: If the $(l+1)$th packet's arrival time at the processing queue is greater than the lth packet's transmission time at the transmitting queue, i.e., $X_i > T_i$, we have

$$Z_i = \zeta_1 + 2\zeta_2 = (X_i - T_i) + 2T_i = X_i + T_i \tag{9}$$

where ζ_1 indicates the duration from the instant that the lth packet departs from the transmission queue till the instant that the $(l+1)$th packet arrives at the processing queue, and $\zeta_2 = T_i$.

Case 2: If the $(l+1)$th packet arrives at the processing queue while the lth packet is still at the transmission queue, i.e., $X_i \le T_i$, we have

$$Z_i = \zeta_1' + \zeta_2' = [T_i - (T_i - X_i)] + T_i = X_i + T_i \tag{10}$$

where ζ_1' denotes the remaining processing time on the $(l+1)$th packet at the processing queue after the lth packet departs from transmission queue, and $\zeta_2' = T_i$.

As a result, the PDF of Z_i is given by

$$\begin{aligned}\xi_{Z_i}(t) &= \left(1-\varrho_{i,1}\right)\big[P\{X_i \le T_i\}\xi_{(X_i+T_i)}(t) \\ &\qquad + P\{X_i > T_i\}\xi_{(X_i+T_i)}(t)\big] + \varrho_{i,1}\xi_{T_i}(t) \\ &= \left(1-\varrho_{i,1}\right)\xi_{(X_i+T_i)}(t) + \varrho_{i,1}\xi_{T_i}(t).\end{aligned} \tag{11}$$

Comparing with (4) and (11), we conclude that Z_i and Y_i have exactly the same probability distribution, and thus any order of statistics (e.g., expectation and variance).

C. Delay Over the Virtual Link Between NFV Nodes N_1 and N_2

So far, we derive the packet departure process of each flow from the link transmission at the first NFV node N_1. Before reaching to the second NFV node N_2, the flows may go through a sequence of network switches and physical links forwarding the traffic. The transmission rates allocated to flow i from these switches and links are the same as the transmission rates $\mu_{i,2}$ from N_1 to maximize the bandwidth utilization [40]. Therefore, queueing delays on switches and links are not considered for each flow. The total packet transmission delay for flow i traversing n_1 switches and n_1 links before reaching N_2 is given by

$$D_{i,1}^{(f)} = \frac{2n_1}{\mu_{i,2}}. \tag{12}$$

D. Delay at the Second NFV Node

Proposition 1: A Poisson packet flow traverses a tandem of $k\,(k = 1, 2, 3, \ldots)$ servers, each with deterministic service capacity $Y^{(k)}$. If we have $Y^{(q)} \ge Y^{(q-1)}$, $\forall q \in [2, k]$, the departure process of the traffic flow coming out of the kth$\,(k \ge 2)$ server remains the same as the departure process from the first server.

The proof of Proposition 1 is given in Appendix A.

According to Proposition 1, the arrival process at the second NFV node N_2 is the same as the traffic departure process from N_1. Based on (7) and (11), the arrival rate for flow i at N_2 is λ_i. Thus, the same method can be used as in (3) to get a set of decoupled processing and transmission rates $\mu_{i,1}'$ and $\mu_{i,2}'$ for flow i at N_2, as shown in Fig. 5, by taking into consideration the time profiles of traffic flows going through the new VNF(s) at N_2 and instantaneous processing and transmission rates $c_{i,1}'$ and $c_{i,2}'$ allocated to flow i. The main difference in packet delay modeling for flow i at N_2 from that at N_1 is that the packet arrival process for flow i is a general process with average arrival rate λ_i. The process has the interarrival time Z_i with the same CDF, expectation and variance as those of Y_i in (6) and (7). Thus, we can model packet processing at N_2 as a G/D/1 queueing process,[3] where the average packet queueing delay before processing for flow i at N_2 is given by [38]

$$W_{i,2} = \frac{\lambda_i\left(\frac{1}{\lambda_i^2} - \frac{1}{{\mu_{i,1}}^2} - \sigma_i^2\right)}{2\left(1-\varrho_{i,1}'\right)} \le \frac{\lambda_i\left(\frac{1}{\lambda_i^2} - \frac{1}{{\mu_{i,1}}^2}\right)}{2\left(1-\varrho_{i,1}'\right)}. \tag{13}$$

In (13), $\varrho_{i,1}' = (\lambda_i/\mu_{i,1}')$, T_e is the idle duration within interarrival time of successive packets of flow i at N_2, with variance σ_i^2.

Since the arrival process at N_2 for each flow correlates with the preceding decoupled processing rates at N_1, as indicated in (6), the G/D/1 queueing model is not accurate especially when λ_i becomes large [38]. Also, it is difficult to obtain the distribution of T_e to calculate σ_i^2 in (13). Using the upper bound in (13) to approximate $W_{i,2}$ is not accurate when λ_i is small (the queueing system is lightly loaded), since the probability of an arriving packet at the processing queue of N_2 seeing an empty queue increases, and σ_i^2 becomes large.

From (6) and (7), in the case of $\mu_{i,1}' < \mu_{i,2}$, Z_i is more likely to approach an exponentially distributed random variable than a deterministic value with varying λ_i under the condition of $\varrho_{i,1}' < 1$. Therefore, to make the arrival process of each flow at the processing queue of N_2 independent of processing and

[3] Note that we consider the case where $\mu_{i,1}' < \mu_{i,2}$, $\forall i \in I$; for the case of $\mu_{i,1}' \ge \mu_{i,2}$, there is no queueing delay for processing at N_2.

TABLE II
SIMULATION PARAMETERS

Traffic flows Parameters	Flow i	Flow j
Rate vector (Firewall)	[1000, 2000] packet/s	[750, 500] packet/s
Rate vector (DNS)	[1000, 1250] packet/s	—
Rate vector (IDS)	—	[800, 312.5] packet/s
n_1	20	20
n_2	25	25
Simulation time	1000 s	1000 s

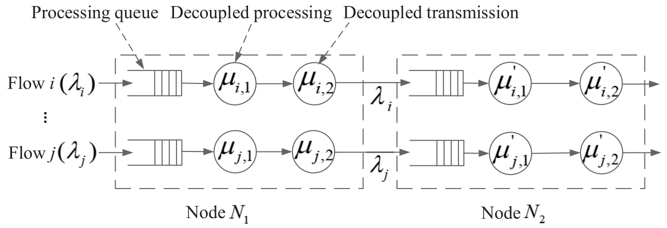

Fig. 5. Decoupled queueing model for traffic flows traversing the first and second NFV nodes in sequence.

transmission rates at N_1, we approximate packet arrival process of flow i at N_2 as a Poisson process with rate parameter λ_i, and establish an M/D/1 queueing model to represent the packet processing for flow i. Proposition 2 indicates that the average packet queueing delay $Q_{i,2}$ in the M/D/1 queueing model is an improved upper bound over that in the G/D/1 system in (13), especially when the input traffic is lightly loaded.

Proposition 2: Given $\mu'_{i,1} < \mu_{i,2}$, $Q_{i,2}$ is an upper bound of $W_{i,2}$ when the processing queue for flow i at N_2 is both lightly loaded and heavily loaded.

The proof of Proposition 2 is given in Appendix B.

According to the approximation on packet arrival process for flow i at N_2, average total packet delay at N_2 is calculated, independently of the processing and transmission rates at N_1, given by

$$D_{i,2} = \begin{cases} \frac{1}{\mu'_{i,1}} + \frac{\lambda_i}{2{\mu'_{i,1}}^2\left(1-\varrho'_{i,1}\right)} + \frac{1}{\mu'_{i,2}}, & \mu'_{i,1} < \mu_{i,2} \\ \frac{1}{\mu'_{i,1}} + \frac{1}{\mu'_{i,2}}, & \mu'_{i,1} \geq \mu_{i,2}. \end{cases} \tag{14}$$

E. Average E2E Delay

Based on the same methodology of delay modeling for packets traversing N_2, the average total packet delay for flow i traversing each subsequent NFV node (if any) can be derived independently. Under the condition that the decoupled packet processing rate of flow i at one subsequent NFV node $N_z\,(z > 2)$ is smaller than the decoupled packet transmission rate at its preceding NFV node N_{z-1}, using an approximated M/D/1 queueing model to represent packet processing at N_z is valid since packet arrival process of flow i at N_z is more likely to approach a Poisson process with varying λ_i. In general, the average E2E delay for a packet of flow i passing through an embedded VNF chain, consisting of m NFV nodes, is the summation of average total delay for a packet passing through all NFV nodes and the total transmission delay on switches and links along the path forwarding the packet, given by

$$D_i = \sum_{z=1}^{m} D_{i,z} + \sum_{z=1}^{m} D^{(f)}_{i,z}. \tag{15}$$

In (15), $D_{i,z}$ is the average packet delay for flow i passing through zth NFV node of the embedded VNF chain, $D^{(f)}_{i,z}$ is the total packet transmission delay for flow i traversing n_z switches and n_z links before reaching NFV node N_{z+1}, and is determined in the same way as in (12).

VI. SIMULATION RESULTS

In this section, simulation results are presented to verify the accuracy of the proposed packet delay modeling of each flow passing through an embedded VNF chain. All simulations are carried out using the network simulator OMNeT++ [41]. We consider the network scenario where two equally weighted flows, i and j, traversing logic VNF chains firewall (f_1) → DNS (f_3) and firewall (f_1) → IDS (f_2), respectively, are embedded on a common physical path and share a set of processing and transmission resources, as shown in Fig. 1. Flow i represents DNS request traffic, whereas flow j indicates a video-conferencing data streaming. The packet arrival rate λ_i for flow i is set to 150 packet/s with packet size of 4000 bits [42]. The packet size for flow j is set to 16 000 bits, and we vary its arrival rate, λ_j, from 75 packet/s to 350 packet/s to reflect different traffic load conditions. The rate vector for each flow traversing an NFV node is tested over OpenStack [43], which is a resource virtualization platform installed on each NFV node. By injecting traffic flows with different packet sizes into different VNFs programmed on the OpenStack, we test maximum available packet processing and transmission rates for different flows. With DR-GPS, each flow is allocated a fraction of the maximum available processing and transmission rates, upon which packet-level simulation is conducted to evaluate packet delay of each flow traversing each NFV node. Table II summarizes the rate vectors for flows i and j traversing different VNFs. Other important simulation settings are also included.

A. Packet Delay at the First NFV Node (N_1)

We first compare packet processing delay and packet queueing delay for each flow traversing N_1. In Fig. 6, it is demonstrated that both packet processing delay and packet queueing delay derived using rate decoupling between flows i and j are close to the simulation results with rate coupling. We can see from Fig. 6(a) that the decoupled processing

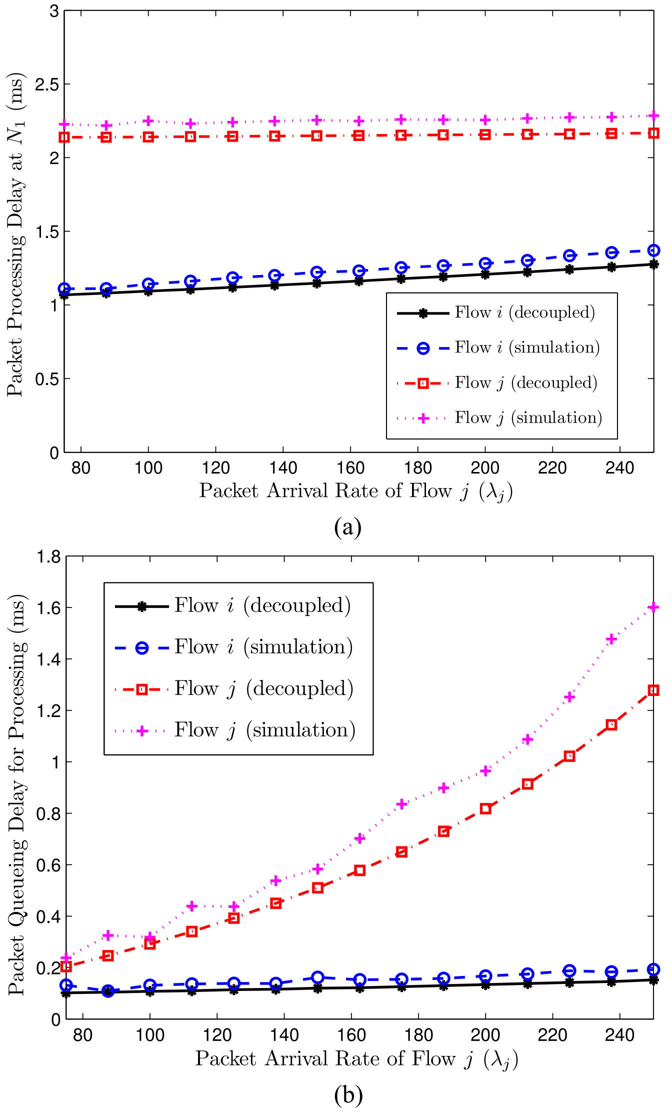

Fig. 6. Packet delay for processing at N_1. (a) Average packet processing delay. (b) Average packet queueing delay.

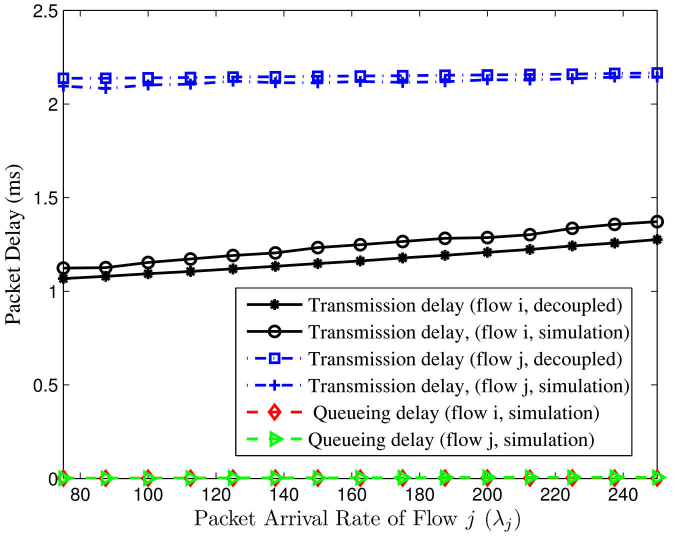

Fig. 7. Delay for packet transmissions at N_1.

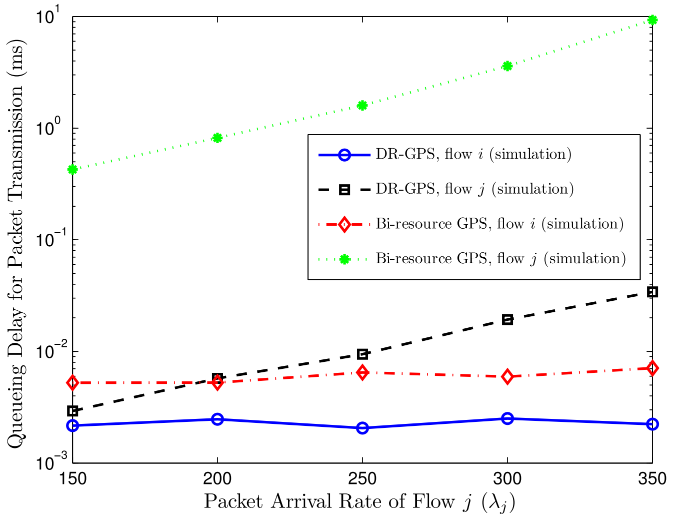

Fig. 8. Queueing delay for packet transmissions under different resource allocation disciplines.

rate for flow i decreases with λ_j, since the processing queue nonempty probability for flow j increases statistically, shrinking the decoupled processing rates for the other flow. Packet transmission delay and queueing delay before transmission are evaluated for both flows at N_1 in Fig. 7. For packet transmission delay, the analytical results match the simulation results well, and there is almost no queueing delay before packet transmissions, which indicates the accuracy of the proposed transmission rate decoupling. In Fig. 8, we compare queueing delays before packet transmissions for flows i and j at N_1 by employing DR-GPS and bi-resource GPS schemes. Although the bi-resource GPS achieves fair allocation on both CPU and bandwidth resources between the two flows, the amount of allocated bandwidth resources is overly provisioned for flow i and is underestimated for flow j, due to the discrepancy of time profiles for different flows. Thus, packet queueing delay before link transmission for flow j becomes much larger than that with the DR-GPS where packet queueing delays for both flows at the transmission link of N_1 are minimized.

B. Packet Delay at the Second NFV Node (N_2)

After traversing the first NFV node, N_1, packet processing delay for flow i and flow j at the second NFV node, N_2, is evaluated in Fig. 9. With a close match between analytical and simulation results, it is verified that the processing rate decoupling at N_2 is accurate. Packet queueing delay before processing for both flows at N_2 are demonstrated in Figs. 10 and 11. As λ_j increases, the packet queueing delay for flow i increases slightly since the resource multiplexing gain obtained by flow i from flow j becomes small. We can see from Fig. 10 that the M/D/1 queueing model for packet processing at N_2 provides a tighter upper bound than the G/D/1 queueing delay upper bound for flow i. In comparison with the results for flow j in Fig. 11, we observe that the G/D/1 upper bound of packet queueing delay is looser for flow i, since the processing queue

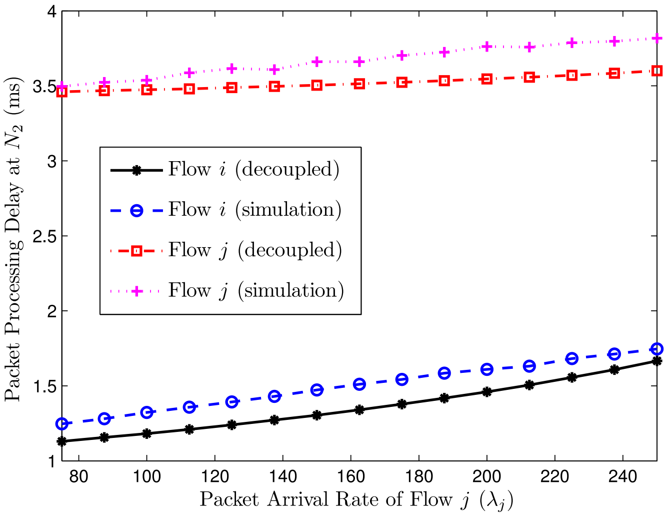

Fig. 9. Packet processing delay at N_2.

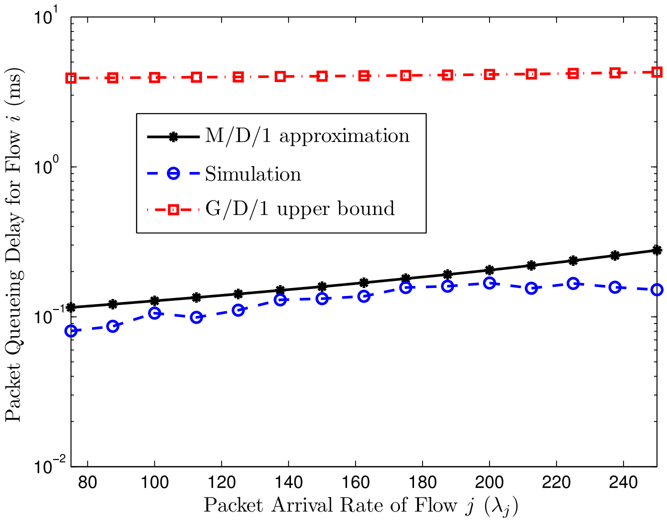

Fig. 10. Packet queueing delay for flow i at N_2.

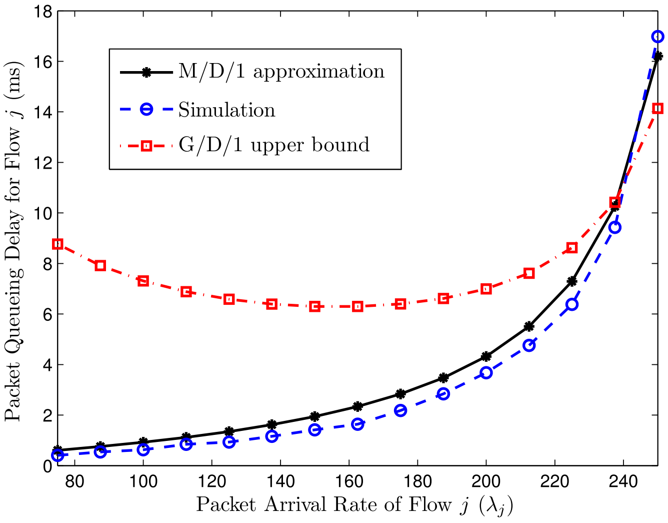

Fig. 11. Packet queueing delay for flow j at N_2.

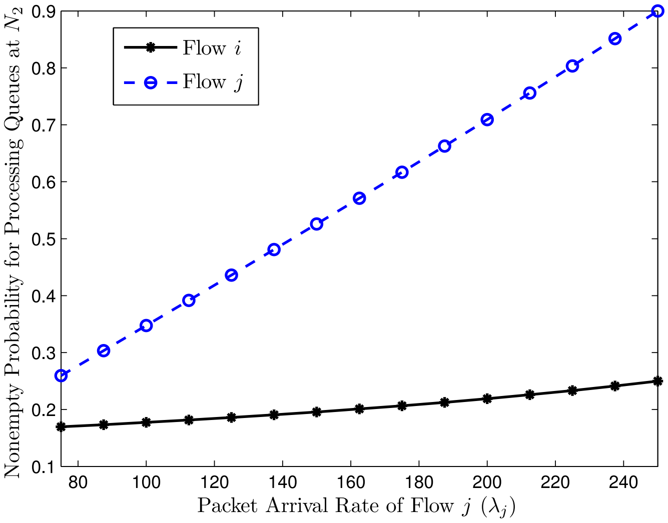

Fig. 12. Nonempty probability for processing queues at N_2.

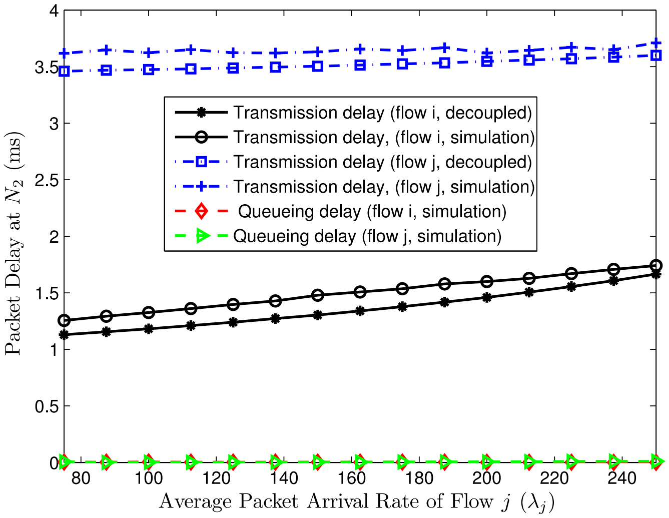

Fig. 13. Delay for packet transmissions at N_2.

for flow i is lightly loaded with low queue nonempty probability shown in Fig. 12, whereas the G/D/1 queueing delay upper bound becomes tighter for flow j with the increase of λ_j but less accurate in a heavy traffic load condition. For both flows, the proposed M/D/1 queueing model is an improved upper bound to approximate the packet queueing delay before processing at N_2.

Packet transmission delay for both flows at N_2 is demonstrated in Fig. 13. Similar to Fig. 7, we can see the decoupled transmission rates for each flow are close to the simulation results, and queueing delay before packet transmissions is negligible. Lastly, we evaluate in Fig. 14 the E2E delay for packets of each flow going through the whole embedded physical network path, which is a summation of the total packet delay for traversing all NFV nodes and packet transmission delay over all physical links and network switches. It is demonstrated that the proposed analytical modeling provides an accurate delay evaluation on E2E packet processing and transmission for multiple flows traversing VNF chains embedded on a common physical network path. The proposed analytical modeling can help to support E2E delay-aware VNF chain embedding.

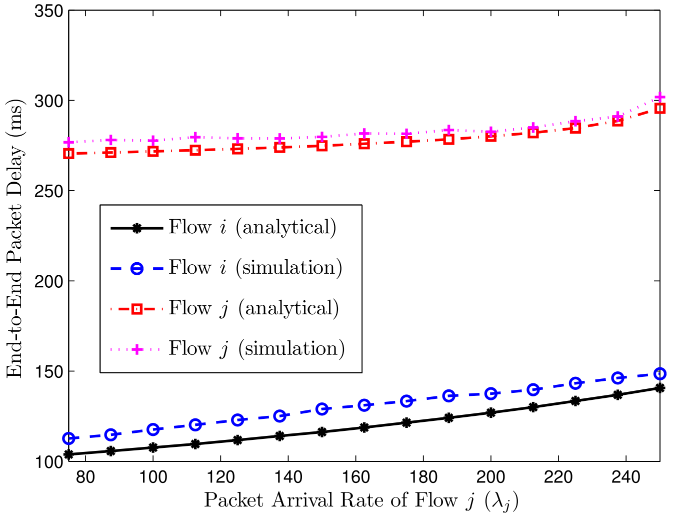

Fig. 14. E2E packet delay for both flows.

VII. Conclusion

In this paper, we present an analytical model to evaluate E2E packet delay for multiple traffic flows traversing a common embedded VNF chain. With DR-GPS, both CPU and bandwidth resources are allocated among different flows at each NFV node to achieve dominant-resource fair allocation with high resource utilization. A tandem queueing model is established to describe packets of each flow passing through an NFV node and its outgoing link. By removing the coupling effect on instantaneous packet processing rates among multiple flows, an M/D/1 queueing model is used to determine average packet delay at each decoupled processing queue of the first NFV node. The correlation of packet transmission rates is also removed in the modeling based on the analysis of packet departure process from each decoupled processing queue. We further analyze the packet arrival process of each flow at the subsequent NFV node, and establish an approximated M/D/1 queueing model to determine the average packet delay of a flow at a decoupled processing queue of the NFV node, which is proved to be a more accurate upper bound than that using a G/D/1 queueing model in both lightly loaded and heavily loaded traffic conditions. Packet transmission delay over each embedded virtual link between consecutive NFV nodes is also derived for E2E delay calculation. Simulation results demonstrate the accuracy and effectiveness of the proposed analytical E2E packet delay modeling for achieving delay-aware VNF chain embedding.

Appendix A
Proof of Proposition 1

Define $Z^{(k)}$ as packet interdeparture time at the kth server, and define $\rho^{(l)}$ as the probability that the lth departing packet sees a nonempty queue at the first server. If the lth departing packet sees an empty queue at the first server, we let $X^{(1)}$ be the duration from time 0 till the instant that $(l+1)$th packet arrives at the server. With $Y^{(q)} \geq Y^{(q-1)}$, there is no queueing delay at each of the servers following the first server. Similar to the description in Fig. 4, if the lth departing packet sees a nonempty queue, we have $Z^{(k)} = Y^{(1)}$; Otherwise, two cases are considered.

Case 1: If $X^{(1)} > \sum_{q=2}^{k} Y^{(q)}$

$$Z^{(k)} = \left(X^{(1)} - \sum_{q=2}^{k} Y^{(q)}\right) + \sum_{q=1}^{k} Y^{(q)} = X^{(1)} + Y^{(1)}. \tag{16}$$

Case 2: If $X^{(1)} \leq \sum_{q=2}^{k} Y^{(q)}$

$$Z^{(k)} = \sum_{q=1}^{k} Y^{(q)} - \left(\sum_{q=2}^{k} Y^{(q)} - X^{(1)}\right) = Y^{(1)} + X^{(1)}. \tag{17}$$

Hence, $Z^{(k)}$ has the same PDF as Y_i derived in (4), which ends the proof. ■

Appendix B
Proof of Proposition 2

When λ_i is small, σ_i^2 in (13) is close to that for an M/D/1 queueing system, given by [38]

$$\sigma_i^2 \approx \frac{1}{\lambda_i^2} - \frac{1}{{\mu'_{i,1}}^2}. \tag{18}$$

Hence, $W_{i,2}$ is further derived as

$$\begin{aligned} W_{i,2} &\approx \frac{\lambda_i\left[\frac{1}{\lambda_i^2} - \frac{1}{{\mu_{i,1}}^2} - \left(\frac{1}{\lambda_i^2} - \frac{1}{{\mu'_{i,1}}^2}\right)\right]}{2\left(1-\varrho'_{i,1}\right)} \\ &= \frac{\lambda_i\left(\frac{1}{{\mu'_{i,1}}^2} - \frac{1}{{\mu_{i,1}}^2}\right)}{2\left(1-\varrho'_{i,1}\right)}. \end{aligned} \tag{19}$$

When λ_i becomes large, the idle duration within interarrival time of successive packets of flow i at N_2 is small, making σ_i^2 negligible. Thus, we have

$$W_{i,2} \approx \frac{\lambda_i\left(\frac{1}{\lambda_i^2} - \frac{1}{{\mu_{i,1}}^2}\right)}{2\left(1-\varrho'_{i,1}\right)} \approx \frac{\lambda_i\left(\frac{1}{{\mu'_{i,1}}^2} - \frac{1}{{\mu_{i,1}}^2}\right)}{2\left(1-\varrho'_{i,1}\right)}. \tag{20}$$

On the other hand, under both traffic load cases, $Q_{i,2}$ in the approximated M/D/1 queueing system is derived as

$$Q_{i,2} = \frac{\lambda_i}{2{\mu'_{i,1}}^2\left(1-\varrho'_{i,1}\right)} \geq W_{i,2}. \tag{21}$$

Thus, we prove that $Q_{i,2}$ is an upper bound of $W_{i,2}$ in both lightly loaded and heavily loaded traffic conditions, and becomes a tighter upper bound than that in the G/D/1 system in (13) when λ_i is small. ■

Acknowledgment

The authors would like to thank Y. Cao (an undergraduate research assistant) for his help in testing over OpenStack the time profiles for traffic flows traversing an NFV node.

References

[1] Q. Ye *et al.*, "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.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

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

[4] Q. Ye and W. Zhuang, "Token-based adaptive MAC for a two-hop Internet-of-Things enabled MANET," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1739–1753, Oct. 2017.

[5] H. Li, M. Dong, and K. Ota, "Radio access network virtualization for the social Internet of Things," *IEEE Cloud Comput.*, vol. 2, no. 6, pp. 42–50, Nov./Dec. 2015.

[6] W. Wu *et al.*, "Enhance the edge with beamforming: Performance analysis of beamforming-enabled WLAN," in *Proc. IEEE WiOpt*, May 2018, pp. 1–6.

[7] X. Duan, C. Zhao, S. He, P. Cheng, and J. Zhang, "Distributed algorithms to compute Walrasian equilibrium in mobile crowdsensing," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4048–4057, May 2017.

[8] G. Yang, S. He, and Z. Shi, "Leveraging crowdsourcing for efficient malicious users detection in large-scale social networks," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 330–339, Apr. 2017.

[9] L. Lyu, C. Chen, S. Zhu, and X. Guan, "5G enabled codesign of energy-efficient transmission and estimation for industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2690–2704, Jun. 2018, doi: 10.1109/TII.2018.2799685.

[10] F. Lyu *et al.*, "SS-MAC: A novel time slot-sharing MAC for safety messages broadcasting in VANETs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3586–3597, Apr. 2018, doi: 10.1109/TVT.2017.2780829.

[11] W. Xu, H. A. Omar, W. Zhuang, and X. S. Shen, "Delay analysis of in-vehicle Internet access via on-road WiFi access points," *IEEE Access*, vol. 5, pp. 2736–2746, 2017.

[12] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

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

[14] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

[15] N. Zhang *et al.*, "Software defined networking enabled wireless network virtualization: Challenges and solutions," *IEEE Netw.*, vol. 31, no. 5, pp. 42–49, May 2017.

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

[17] J. Ordonez-Lucena *et al.*, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.

[18] X. Li, J. Rao, H. Zhang, and A. Callard, "Network slicing with elastic SFC," in *Proc. IEEE VTC*, Toronto, ON, Canada, Sep. 2017, pp. 1–5.

[19] J. Li *et al.*, "Joint resource allocation and online virtual network embedding for 5G networks," in *Proc. IEEE GLOBECOM*, Singapore, Dec. 2017, pp. 1–6.

[20] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 533–546, Sep. 2016.

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

[22] H. Li, K. Ota, M. Dong, and M. Guo, "Mobile crowdsensing in software defined opportunistic networks," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 140–145, Jun. 2017.

[23] H. Li, M. Dong, and K. Ota, "Control plane optimization in software-defined vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7895–7904, Oct. 2016.

[24] C. Clark *et al.*, "Live migration of virtual machines," in *Proc. ACM USENIX*, May 2005, pp. 273–286.

[25] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Predicting the performance of virtual machine migration," in *Proc. IEEE MASCOTS*, Aug. 2010, pp. 37–46.

[26] S. Ayoubi, C. Assi, K. Shaban, and L. Narayanan, "MINTED: Multicast virtual network embedding in cloud data centers with delay constraints," *IEEE Trans. Commun.*, vol. 63, no. 4, pp. 1291–1305, Apr. 2015.

[27] A. Ghodsi, V. Sekar, M. Zaharia, and I. Stoica, "Multi-resource fair queueing for packet processing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 1–12, Oct. 2012.

[28] W. Wang, B. Liang, and B. Li, "Multi-resource generalized processor sharing for packet processing," in *Proc. ACM IWQoS*, Montreal, QC, Canada, Jun. 2013, pp. 1–10.

[29] A. Ghodsi *et al.*, "Dominant resource fairness: Fair allocation of multiple resource types," in *Proc. ACM NSDI*, Apr. 2011, pp. 323–336.

[30] F. C. Chua, J. Ward, Y. Zhang, P. Sharma, and B. A. Huberman, "Stringer: Balancing latency and resource usage in service function chain provisioning," *IEEE Internet Comput.*, vol. 20, no. 6, pp. 22–31, Nov./Dec. 2016.

[31] B. Xiong, K. Yang, J. Zhao, W. Li, and K. Li, "Performance evaluation of OpenFlow-based software-defined networks based on queueing model," *Comput. Netw.*, vol. 102, pp. 172–185, Jun. 2016.

[32] M. Jarschel *et al.*, "Modeling and performance evaluation of an OpenFlow architecture," in *Proc. 23rd ITC*, San Francisco, CA, USA, Sep. 2011, pp. 1–7.

[33] N. Egi *et al.*, "Understanding the packet processing capability of multi-core servers," Intel, Santa Clara, CA, USA, Rep. LABOS-REPORT-2009-001, 2009.

[34] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[35] Z.-L. Zhang, D. Towsley, and J. Kurose, "Statistical analysis of the generalized processor sharing scheduling discipline," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 6, pp. 1071–1080, Aug. 1995.

[36] D. C. Parkes, A. D. Procaccia, and N. Shah, "Beyond dominant resource fairness: Extensions, limitations, and indivisibilities," *ACM Trans. Econ. Comput.*, vol. 3, no. 1, p. 3, Mar. 2015.

[37] A. Gutman and N. Nisan, "Fair allocation without trade," in *Proc. ACM AAMAS*, Jun. 2012, pp. 719–728.

[38] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data Networks*, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall, 1987.

[39] L. Kleinrock, *Queueing Systems, Volume 1: Theory*. New York, NY, USA: Wiley, 1976.

[40] S. Larsen, P. Sarangam, R. Huggahalli, and S. Kulkarni, "Architectural breakdown of end-to-end latency in a TCP/IP network," *Int. J. Parallel Program.*, vol. 37, no. 6, pp. 556–571, Dec. 2009.

[41] *OMNeT++ 5.0*. Accessed: Dec. 2017. [Online]. Available: http://www.omnetpp.org/omnetpp

[42] A. Liska and G. Stowe, *DNS Security: Defending the Domain Name System*, Syngress, Maryland Heights, MO, USA, 2016.

[43] *Openstack (Release Pike)*. Accessed: Dec. 2017. [Online]. Available: https://www.openstack.org

Qiang Ye (S'16–M'17) received the B.S. degree in network engineering and M.S. degree in communication and information system from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2009 and 2012, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2016.

He has been a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, since 2016. His current research interests include SDN and NFV, network slicing for 5G networks, VNF chain embedding and end-to-end performance analysis, medium access control and performance optimization for mobile ad hoc networks, and Internet of Things.

Weihua Zhuang (M'93–SM'01–F'08) has been with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, since 1993, where she is a Professor and a Tier I Canada Research Chair in wireless communication networks.

Dr. Zhuang was a recipient of the 2017 Technical Recognition Award from the IEEE Communications Society Ad Hoc & Sensor Networks Technical Committee, one of 2017 ten N2Women (Stars in Computer Networking and Communications) and a co-recipient of several Best Paper Awards from IEEE conferences. She was the Editor-in-Chief of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2013, the Technical Program Chair/Co-Chair of IEEE VTC Fall 2017 to Fall 2016, and the Technical Program Symposia Chair of IEEE GLOBECOM 2011. 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 an IEEE Communications Society Distinguished Lecturer from 2008 to 2011.

Xu Li received the B.Sc. degree in computer science from Jilin University, Changchun, China, in 1998, the M.Sc. degree in computer science from the University of Ottawa, Ottawa, ON, Canada, in 2005, and the Ph.D. degree in computer science from Carleton University, Ottawa, ON, in 2008.

He was a Research Scientist (Tenure) with Inria, France. He is a Staff Researcher with Huawei Technologies Inc., Ottawa, ON. His current research interests include 5G system design and standardization, along with over 90 refereed scientific publications, over 40 3GPP standard proposals, and over 50 patents and patent filings.

Dr. Li was a recipient of NSERC PDF Awards, the IEEE ICNC 2015 Best Paper Award, and a number of other awards. He was on the Editorial Boards of the *IEEE Communications Magazine* and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He was a TPC Co-Chair of IEEE VTC 2017 (Fall) LTE, 5G and Wireless Networks Track, and the IEEE Globecom 2013 Ad Hoc and Sensor Networking Symposium.

Jaya Rao (M'14) received the B.S. and M.S. degrees in electrical engineering from the University of Buffalo, Buffalo, NY, USA, in 2001 and 2004, respectively, and the Ph.D. degree from the University of Calgary, Calgary, AB, Canada, in 2014.

In 2014, he joined Huawei Technologies Canada, Ottawa, ON, Canada, where he has been involved with research and design on CIoT, URLLC, and V2X-based solutions in 5G new radio, and is currently a Senior Research Engineer. He has contributed with Huawei at 3GPP RAN WG2, RAN WG3, and SA2 meetings on topics related to URLLC, network slicing, mobility management, and session management. From 2004 to 2010, he was a Research Engineer with Motorola Inc., Chicago, IL, USA.

Dr. Rao was a recipient of the Best Paper Award of IEEE WCNC 2014.