Joint Virtual Network Topology Design and Embedding for Cybertwin-Enabled 6G Core Networks

Junling Li, *Member, IEEE*, Weisen Shi, *Graduate Student Member, IEEE*, Qiang Ye, *Member, IEEE*, Shan Zhang, *Member, IEEE*, Weihua Zhuang, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—To efficiently allocate heterogeneous resources for customized services, in this article, we propose a network virtualization (NV)-based network architecture in cybertwin-enabled 6G core networks. In particular, we investigate how to optimize the virtual network (VN) topology (which consists of several virtual nodes and a set of intermediate virtual links) and determine the resultant VN embedding in a joint way over a cybertwin-enabled substrate network. To this end, we formulate an optimization problem whose objective is to minimize the embedding cost, while ensuring that the end-to-end (E2E) packet delay requirements are satisfied. The queueing network theory is utilized to evaluate each service's E2E packet delay, which is a function of the resources assigned to the virtual nodes and virtual links for the embedded VN. We reveal that the problem under consideration is formally a mixed-integer nonlinear program (MINLP) and propose an improved brute-force search algorithm to find its optimal solutions. To enhance the algorithm's scalability and reduce the computational complexity, we further propose an adaptively weighted heuristic algorithm to obtain near-optimal solutions to the problem for large-scale networks. Simulations are conducted to show that the proposed algorithms can effectively improve network performance compared to other benchmark algorithms.**

***Index Terms*—6G, Cybertwin, end-to-end (E2E) packet delay, network virtualization (NV), resource allocation, topology, virtual network embedding (VNE).**

Manuscript received February 4, 2021; revised May 20, 2021; accepted July 3, 2021. Date of publication July 14, 2021; date of current version November 5, 2021. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada; in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS); in part by the National Natural Science Foundation of China under Grant 61801011; and in part by the Beijing Municipal Natural Science Foundation under Grant L192028. *(Corresponding author: Qiang Ye.)*

Junling Li is with the Shenzhen Institute of Artificial Intelligence and Robotics for Society and the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: lijunling@cuhk.edu.cn).

Weisen Shi, Weihua Zhuang, and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: w46shi@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).

Shan Zhang is with the School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: zhangshan18@buaa.edu.cn).

Digital Object Identifier 10.1109/JIOT.2021.3097053

I. Introduction

THE 6G networks are envisioned to accommodate a proliferation of Internet-of-Things (IoT) devices and diversified services [1]. The traffic volume is expected to increase largely and the Quality-of-Service (QoS) demands are foreseen to be differentiated [2]–[4]. Consequently, effective and efficient network configurations are desired to achieve quick service provisioning for these on-demand and rapid-growing services. To support various IoT applications and realize service-oriented networking, three essential challenges arise on the currently employed networking architecture and the associated resource management. First, network capacity needs to be boosted to accommodate the data-hungry services with enormous traffic volume and trillions of end devices. Densely deployed communication infrastructures and network servers inevitably increase capital expenditure (CAPEX) and operational expenditure (OPEX) [1]. Second, strict service isolation is indispensable when network dynamics are considered during customized service provisioning. Finally, to support heterogeneous IoT services cost effectively, a unified framework is needed to integrate computing, storage, and networking resources and allocate global resources efficiently [5], [6].

Network virtualization (NV) and software-defined networking (SDN) are anticipated to provide the technical routes to realize wireless and wired service-oriented networking [7]–[9]. Particularly, SDN separates the data plane from the control plane and realizes centralized network control, making virtual and physical network resource provisioning more agile, thereby facilitating service delivery [10]. Through NV, each network provider is decomposed into the virtual network operator (VNO) and the infrastructure provider (InP) in core networks. The VNOs partition the computing and bandwidth resources on network servers and transmission links to create different virtual networks (VNs) [11]. Conceptually, each VN consists of virtual nodes and a set of virtual links, which interconnect the virtual nodes.

However, solely relying on the SDN/NV integrated networking architecture may not be sufficient to provide fine-grained resource orchestration for diversified service customization, as VNOs need to obtain service-level quantitative QoS mapped from end-user experience for better VN

creation. Cybertwin emerges as a promising architecture to enhance QoS provisioning for sliced VNs [12], [13], in which more advanced functionalities are augmented at the edge of a core network to assist VN establishment. Specifically, in a cybertwin-enabled core network, service requests sent from end users in RAN are properly categorized and aggregated as different service groups based on additional user-level information (e.g., user identity and addresses). Such information is retrieved at edge nodes (also called communication anchors) connecting RAN to the core network, where the user-level Quality-of-Experience (QoE) is interpreted into service-level QoS requirements for each of the service groups [13]. The grouped service requests with QoS demands are then forwarded from edge nodes to VNOs, which creates VNs by constructing VN topologies and reserving transmission and processing resources on physical links and network servers/switches, customized for service deliveries with differentiated QoS satisfaction. Instead of sending service requests directly from users to VNOs, the cybertwin-enabled SDN/NV core network architecture promotes more fine-grained VN creation, by using edge nodes as intermediate agents to establish a quantitative mapping from application-level user QoE to service-level QoS demands that can be used for VN resource orchestration.

One of the fundamental research issues under a cybertwin-enabled core network architecture is how to perform *VN topology design* (i.e., to optimize the resource allocation for constructing each VN), and VN embedding (VNE) (i.e., to embed the VN with optimized resource allocation onto a substrate network) [14]. In particular, while guaranteeing that the differentiated service requirements are satisfied, the physical resource allocation on the virtual nodes and virtual links is optimized to realize cost-effective VNE. Many research efforts have been dedicated to investigating the VNE problem [15]–[22], but most of the existing studies make an assumption that the VN topology's resource allocation is predefined. However, to achieve cost-effective service provisioning with guaranteed service quality, it is desired to study the two problems, i.e., the VN topology design and VNE, in a joint manner. On the other hand, accurate and comprehensive end-to-end (E2E) delay modeling for each VN embedded onto the substrate network is of great importance to support time-critical services in cybertwin-enabled core networks. Existing studies usually consider link propagation delay only, while ignoring the queueing delay and packet processing/transmission delay on each virtual node/link [23]–[26]. To perform resource allocation optimization for VNE, a more accurate and comprehensive E2E delay model is needed.

In this article, a QoS-guaranteed VNE problem is investigated where the packet-level E2E delay is taken as the QoS metric. Resource allocation for each VN and traffic routing configuration are optimized in a joint way. Under the resource constraints of the physical network, our purpose is to minimize the cost for VNE while ensuring that the services' differentiated E2E delay requirements are satisfied. To this end, a cybertwin-enabled SDN/NV architecture is first presented to enable E2E service deliveries in a core network. Our main contributions are summarized as follows.

1) We formulate the joint problem of the VN topology design and VNE as a mixed-integer nonlinear program (MINLP), which enables the optimization of VN resource allocation and VNE together. Different from the existing VNE formulations without the optimization of resource allocation on each VN, our proposed formulation can increase the network-wide resource utilization.
2) We propose a cost function of utilizing physical resources to balance the traffic load over the whole substrate network. Given the cost function, substrate nodes and links with more residual resources have higher chances of being chosen during VNE.
3) We apply the queueing theory to analyze the E2E packet delay, instead of propagation-only delays, resulting in more accurate E2E delay models.
4) We propose an improved brute-force search algorithm for the optimal solution of the MINLP. Further for a large-scale network, we propose an adaptively weighted heuristic algorithm to obtain near-optimal solutions with much lower computational complexity.

The remainder of this article is organized as follows. In Section II, an overview of related works is provided. In Section III, the system model is presented in detail. In Section IV, VN topology design and VNE are formulated as an MINLP in a joint way based on NV-based network architecture. In Section V, we transform the MINLP problem and propose an improved brute-force search algorithm to find the optimal solution. In Section VI, we propose an adaptively weighted heuristic algorithm to obtain optimal/near-optimal solutions to the MINLP. In Section VII, simulation results are presented to demonstrate the effectiveness of the proposed algorithms. Concluding remarks are drawn in Section VIII.

II. Related Works

Being one of the essential parts of NV, VNE deals with physical resource assignment to both virtual nodes and virtual links, which leads to two subproblems, i.e., virtual node mapping and virtual link mapping. The former addresses how to choose physical nodes to support the virtual nodes, while the latter solves how to embed the virtual links interconnecting the virtual nodes onto physical paths. Greedy methods are usually used to find the node mappings, while the k-shortest paths or multicommodity flow algorithms are typically used to find link mappings [14].

Existing VNE studies can be mainly divided into two groups, i.e., the coordinated VNE which solves the two subproblems as a whole [19]–[22] and the uncoordinated ones which solve the two subproblems in a separate way [15]–[18]. Yu *et al.* [15] have studied the VNE problem incorporating path splitting and path migration with the objective of increasing the service acceptance ratio. To make the problem tractable, the authors have divided the formulated VNE problem into two subproblems and solved them independently. Later on, several topological attributes-based node ranking approaches have been presented to achieve improved solution quality of node mapping [16]–[18]. Such a group of approaches lacks the correlation between node embedding and

link embedding. This may lead to a low VN acceptance ratio and an increased embedding cost. To address this concern, the dependency of link embedding on node embedding has to be considered [19]–[22]. For example, Gong *et al.* [20] have investigated coordinated node and link mapping for the location-constrained problem (LC-VNE). The compatibility graph (CG) concept has been used to develop two heuristic algorithms, facilitating coordinated node and link embedding. Song *et al.* [22] have presented a coordinated VNE algorithm based on particle swarm optimization (PSO). The PSO-based algorithm allows a combination of node and link mapping by utilizing a step-by-step rule to update the particle positions. All the aforementioned VNE formulations assume a predefined set of physical resources allocated to a VN. Furthermore, they assume that each service's QoS can be ensured with the predefined physical resource allocation. This study differs from them in that the set of physical resources for each VN allocation is optimized. Moreover, we establish an analytical relationship between the amount of resources allocated to each VN and the QoS metric. The analytical relationship is incorporated into our VNE formulation to achieve QoS-guaranteed VNE.

On the other hand, some existing VNE formulations take each service's E2E packet delay as a constraint [23]–[26]. For example, Zhang *et al.* [23] have investigated VNE in a multicast service-oriented network and incorporated the delay and jitter constraints into the VNE formulation. In [25], the VNE problem has been formulated as an integer program where delay constraints are incorporated. The program was then solved by using a greedy algorithm, which allows migration of virtual nodes when the delay constraints are violated. Chochlidakis and Friderikos [26] have established a VNE optimization framework with the objective of minimizing the E2E packet delay in mobile networks, where user mobility has been taken into account. However, in terms of the analysis of E2E packet delay, most of the existing approaches consider the link propagation delay only, which does not appear to be comprehensive. In this study, we also involve the packet processing and transmission delays into E2E packet delay analysis to achieve a more comprehensive delay modeling.

In a word, existing VNE studies either assume that the set of physical resources assigned to an embedded VN are predefined or neglect the packet processing and transmission delays when performing the E2E packet delay analysis for a VN. How to deal with VN topology design and VNE in a joint way to realize E2E delay-guaranteed service provisioning is still an open subject in the literature.

III. System Model

A. Network Architecture

We consider a cybertwin-enabled SDN/NV-integrated networking architecture to support differentiated E2E service provisioning, as shown in Fig. 1. Service customization is achieved by slicing network resources for different VNs and inter-VN resource sharing is enabled to improve physical resource utilization. The networking architecture mainly consists of four components: 1) cybertwin-enabled edge clouds,

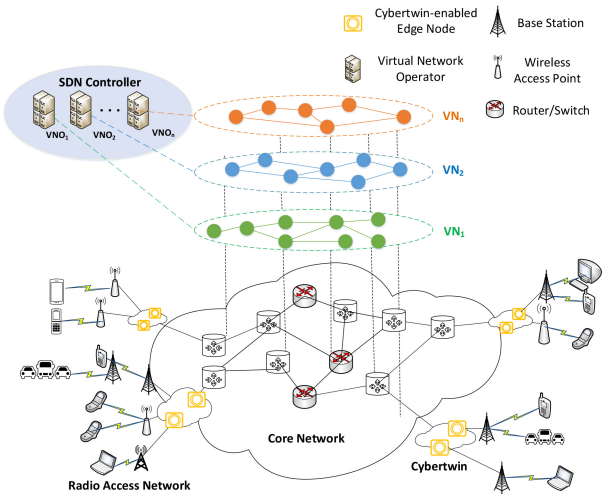

Fig. 1. Cybertwin-enabled SDN/NV networking architecture.

a set of network servers at the edge of the core network to aggregate end user traffic from RAN into different service groups by retrieving user-level information (e.g., user identity and addresses) and send service requests to VNOs with quantitative QoS requirements mapped from user QoE [13]; 2) core network, an interconnection of network servers through routers/switches and physical transmission links, providing specific functionalities for users in RAN (e.g., mobility management, access control, and processing); 3) VNOs, which are separate control entities taking charge of slicing the physical network resources for customized services to create different VNs; and 4) SDN controller, which helps optimization of the resources allocation on VNs. In the core network, each VN is managed by a VNO to ensure that one VN's dynamics does not affect other VNs. In this manner, the flexibility of each VN's resource management is increased since VNs have conceptual isolation for customized QoS guarantees.

B. Service Request

Once traffic aggregation and traffic grouping are done, different VNOs support multiple service requests in the form of multiple VNs. A service request is defined by a pair of source–destination (S-D) nodes in the substrate network, the E2E packet delay requirement of the request, and the arrival data rate of the request. We consider that processing (transmitting) the data packets require a certain amount of CPU processing (link bandwidth) resources to be allocated to each virtual node (link). We further consider that the optimal physical resource allocation on the virtual nodes and virtual links can be found by minimizing the embedding cost due to leasing InPs' physical resources and at the same time guaranteeing each service's E2E packet delay requirement. Two illustrative service requests are presented at the top of Fig. 2. As shown in the figure, the delay requirement and arrival data rate of the first service request are 10 ms and 150 packet/s, respectively. The second service request has a delay requirement of 20 ms and an arrival data rate of 250 packet/s. We set each packet size as 4000 bits in this study [27].

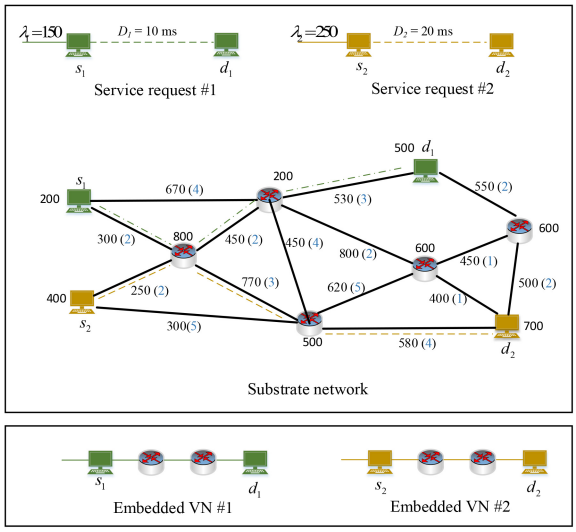

Fig. 2. Illustration of the joint VN topology design and embedding problem with one substrate network and two E2E VN requests. The problem includes the optimization of VN resource allocation and the embedding of the resultant VN onto the substrate network.

C. Substrate Network

The substrate network (i.e., the core network) is represented by an abstracted undirected graph, i.e., $G = (N, L)$. Here, N is a set containing all the physical nodes (i.e., servers/switches in the substrate network), and L is a set that contains all the undirected physical transmission links interconnecting the nodes in the substrate network. All the physical nodes are able to process and forward data packets. Denote the available CPU resources on node $n \in N$ by C_n. Denote the transmission rate available on link $l \in L$ by B_l. Denote the constant propagation delay on link $l \in L$ by δ_l. Let the set containing the neighboring nodes of node $v \in N$ be $\mathcal{N}_v$. The set $\mathcal{N}_v$ contains physical node u if there exists a direct link $l = (u, v)$ that interconnects nodes u and v. An example of a substrate network is presented in Fig. 2, where the available link bandwidths are represented by the numbers over each link; the constant propagation delay is represented by the numbers in brackets; and the available CPU processing rates are denoted by the numbers beside the physical nodes.

D. Embedded VN Topology

Once a service request is embedded onto a substrate network, we obtain an embedded VN topology. Let the set that contains all the service requests be denoted by $\mathcal{R} = \{1, 2, \ldots, i, \ldots, R\}$, where R is the total number of service requests. Let the embedded VN corresponding to service request i be denoted by $G'_i = (N'_i, L'_i)$. Here, $N'_i \subset N$ is a set that contains all the physical nodes chosen to operate the ith VN's virtual nodes, and $L'_i \subset L$ is the set that contains all the physical links that interconnect the embedded virtual nodes. For simplicity, for each VN, we consider a single-path traffic routing between each pair of S-D nodes. The S-D node pair of a VN are the two physical nodes where a traffic flow enters into and ends in the substrate network. Let the source node and destination node corresponding to the ith VN be denoted by $s_i \in N'_i$ and $d_i \in N'_i \setminus \{s_i\}$, respectively. Let the ith VN's E2E delay requirement and arrival data rate be represented by D_i and λ_i, respectively. Two illustrative VNs are presented at the bottom of Fig. 2.

TABLE I
SUMMARY OF IMPORTANT SYMBOLS

Notation	Description
$\mathcal{R}$	Set of service requests
C_n	CPU processing rate available on physical node n
B_l	Bandwidth resource available on link l
δ_l	Constant propagation delay on link l
s_i	Source node of the ith service request
d_i	Destination node of the ith service request
$\mathcal{N}_v$	Set of neighboring nodes for physical node $v \in N$
λ_i	Arrival data rate corresponding to service request i
D_i	E2E packet delay requirement corresponding to service request i
p_n	Unit cost of CPU resource utilization on node n
p_{uv}	Unit cost of bandwidth resource utilization on physical link $l = (u, v)$
$T_{q,i}$	Average queueing delay corresponding to the ith service request
T_i	Total E2E delay corresponding to the ith service request

TABLE II
DECISION VARIABLES

Decision variables	Definition
$C'_{n,i}$	CPU resource allocated to node $n \in N'_i$ for the ith service request
B'_{s_i,d_i}	Bandwidth resource allocated to each link $l \in L'_i$ for the ith service request
x^i_n	Binary variable which indicates if node n is chosen to forward the data traffic for the ith service request
y^i_{uv}	Binary variable which indicates if link (u, v) is chosen to forward the data traffic for the ith service request i

IV. Problem Formulation

Given the traffic statistics and QoS requirement of each VN, how to optimize resource allocation on virtual nodes and virtual links, and subsequently embed the resultant VN onto the substrate network is what we call the QoS-guaranteed VNE problem in this article. Tables I and II summarize the important symbols and a list of decision variables are used in the formulation in this section.

The constraints of the QoS-guaranteed VNE problem include physical resource constraints, E2E delay constraints, and traffic routing constraints. For the link mapping, it is required that the amount of bandwidth resources allocated to link $l \in L'_i$ of the ith VN (i.e., B'_l) is upper bounded by B_l. In other words, we have $B'_l \leq B_l \ \forall l \in L'_i$. Since we do not consider traffic splitting and all the nodes are assumed to have no traffic inflation and deflation, we have the same link bandwidth demands for any link $l \in L'_i$. Let this common bandwidth requirement for the ith service request be denoted as B'_{s_i,d_i}. Hence, we have the preceding bandwidth capacity constraint expressed as $B'_{s_i,d_i} \leq B_l \ \forall l \in L'_i$. Denote the CPU processing rate allocated to the virtual node operated (embedded) on physical node n as $C'_{n,i}$ for the ith service request. Similarly, we have $C'_{n,i}$ upper bounded by C_n, expressed as

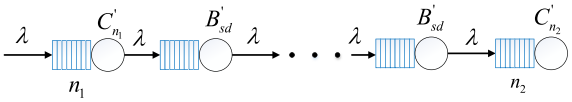

Fig. 3. Network of queues to optimize the physical resource allocation on virtual nodes and links for each E2E service request.

$C'_{n,i} \leq C_n \ \forall n \in N'_i$. Note that the CPU processing rate is converted to packet processing rate (in the unit of packet per second), considering the virtual node computational capacity and the number of cycles needed to process 1 bit of data.

A network of queues is established to analyze the E2E packet delay of a traffic flow passing through an embedded VN. We show an example of such a network of queues in Fig. 3. In the figure, n_1 and n_2 are two network servers that operate the source node and destination node, respectively. Several intermediate nodes and a set of transmission links are also contained in the network to perform traffic forwarding. A customized service flow with arrival data rate λ enters the network at n_1 and leaves the network at n_2. One service flow's traffic arrivals are assumed to follow a Poisson process. In the VNE process, in situations where multiple service flows are embedded onto a common physical path in the network, both the transmission rate over each common link along the path and the CPU processing rate on each common network server have to be shared in a proper way among all the concerned service flows. Such resource sharing is typically conducted following a certain multiresource sharing policy, such as the dominant-resource generalized processor sharing [28]. Subsequently, we can establish a comprehensive E2E delay model for each service flow embedded onto the substrate [29]. This, nevertheless, makes the formulation of the QoS-guaranteed VNE problem more complicated. To make the resource sharing problem tractable, it is assumed in this article that the amount of link transmission rates and CPU processing rates allocated to each service request are independent random variables and exponentially distributed [30]. Based on this assumption, we model packet transmission at a virtual link and packet processing on a virtual node as an M/M/1 queue system [30]. As a result, we model the queueing process of a service flow passing through an entire embedded VN as an open Jackson queueing network [31]. Let the arrival rate of the traffic flow for the ith VN be denoted by λ_i. For each node selected to process the data in the VN, we have the following capacity constraint:

$$\lambda_i / C'_{n,i} < 1 \quad \forall n \in N'_i \quad \forall i \in \mathcal{R}. \tag{1}$$

For each transmission link selected to transmit the data in the VN, we have the following capacity constraint:

$$\lambda_i / B'_{s_i,d_i} < 1 \quad \forall l \in L'_i \quad \forall i \in \mathcal{R}. \tag{2}$$

For the ith service request, we denote the packet waiting time at the processing queue of $n \in N'_i$ and that at the transmission queue of link $l \in L'_i$ as $T_{n,i}$ and $T_{l,i}$, respectively. The expectation of $T_{n,i}$ and $T_{l,i}$ when the M/M/1 queues are at equilibrium are found from [31]

$$E[T_{n,i}] = \frac{1}{C'_{n,i} - \lambda_i} \quad \forall n \in N'_i \quad \forall i \in \mathcal{R} \tag{3}$$

and

$$E[T_{l,i}] = \frac{1}{B'_{s_i,d_i} - \lambda_i} \quad \forall n \in N'_i \quad \forall i \in \mathcal{R} \tag{4}$$

Hence, for the ith service request, we can find the average packet queueing delay corresponding to the entire network of queues from [31]

$$\begin{aligned} T_{q,i} &= \sum_{n \in N'_i} E[T_{n,i}] + \sum_{l \in L'_i} E[T_{l,i}] \\ &= \sum_{n \in N'_i} \frac{1}{C'_{n,i} - \lambda_i} + \sum_{l \in L'_i} \frac{1}{B'_{s_i,d_i} - \lambda_i} \quad \forall i \in \mathcal{R}. \end{aligned} \tag{5}$$

The E2E delay requirement ($\forall i \in \mathcal{R}$), with the consideration of the propagation delay δ_l, is formulated as [31]

$$\sum_{n \in N'_i} \frac{1}{C'_{n,i} - \lambda_i} + \sum_{l \in L'_i} \frac{1}{B'_{s_i,d_i} - \lambda_i} + \sum_{l \in L'_i} \delta_l \leq D_i. \tag{6}$$

Now, we define binary variables x_n^i ($n \in N$, $i \in \mathcal{R}$) and y_{uv}^i ($u, v \in N$, $i \in \mathcal{R}$) for the purpose of VNE problem formulation. The binary variable $x_n^i = 1$ if physical node n operates a certain virtual node for service request i, and $x_n^i = 0$ otherwise. The binary variable $y_{uv}^i = 1$ if the link $l = (u, v)$ is selected to forward the traffic from the source node to the destination node for the ith service request and $y_{uv}^i = 0$ otherwise. Considering the capacity constraints of each substrate node and substrate link, we have

$$\sum_{i \in \mathcal{R}} y_{uv}^i B'_{s_i,d_i} \leq B_{uv} \quad \forall (u, v) \in L \tag{7}$$

$$\sum_{i \in \mathcal{R}} x_n^i C'_{n,i} \leq C_n \quad \forall n \in N. \tag{8}$$

By introducing x_n^i and y_{uv}^i, (6) can be transformed into the following form ($\forall i \in \mathcal{R}$):

$$\sum_{n \in N} x_n^i \frac{1}{C'_{n,i} - \lambda_i} + \sum_{(u,v) \in L} y_{uv}^i \left(\frac{1}{B'_{s_i,d_i} - \lambda_i} + \delta_{uv} \right) \leq D_i. \tag{9}$$

The traffic routing constraints at the source node s_i, destination node d_i, and each embedded node in between are given, respectively, by [32] ($\forall i \in \mathcal{R}$)

$$\sum_{v \in \mathcal{N}_{s_i}} y_{s_i v} - \sum_{v \in \mathcal{N}_{s_i}} y_{v s_i} = 1 \tag{10}$$

$$\sum_{u \in \mathcal{N}_{d_i}} y_{u d_i} - \sum_{u \in \mathcal{N}_{d_i}} y_{d_i u} = 1 \tag{11}$$

$$\sum_{v \in \mathcal{N}_u} y_{vu} = \sum_{v \in \mathcal{N}_u} y_{uv} \quad \forall u \in N \setminus \{s_i, d_i\} \tag{12}$$

where constraint (10) guarantees that at least one node v from the neighboring nodes of s_i (i.e., the set $\mathcal{N}_{s_i}$) has to be chosen with $y_{s_i v} = 1$; Constraint (11) implies that at least one node u from the neighboring nodes of d_i (i.e., the set $\mathcal{N}_{d_i}$) has to be chosen with $y_{u d_i} = 1$; Constraint (12) is for every other node

$u \in N$ no matter if physical node u is selected to be on the traffic routing path from s_i to d_i.

The overall objective of the QoS-guaranteed VNE problem is to minimize the VNE cost denoted by $\mathcal{C}$. The cost is defined as the sum of the cost of resource utilization on substrate links and that on substrate nodes for all the service requests

$$\mathcal{C} = \sum_{i \in \mathcal{R}} \left(w_1 \sum_{n \in N} x_n^i p_n C'_{n,i} + w_2 \sum_{(u,v) \in L} y_{uv}^i p_{uv} B'_{s_i,d_i} \right). \quad (13)$$

Here, B'_{s_i,d_i} and $C'_{n,i}$ represent the bandwidth resources allocated to each link used for forwarding the traffic from the source to the destination and the CPU resources allocated to node n, respectively. The symbol p_n represents the unit cost for utilizing CPU resources on physical node n; w_1 and w_2 are two weighting parameters used to reflect the importance of node resource and link resource in calculating the total embedding cost; and the symbol p_{uv} denotes the unit cost for utilizing the bandwidth resources of physical link (u, v). The purpose of minimizing the VNE cost is to better balance the resource utilization among all the physical nodes and links. In this way, we aim to reduce the congestion level at the bottleneck nodes and increase the number of accommodated services, ultimately increasing the InPs' total profit in the long run [15], [19], [22].

The following three properties are taken into account in designing the expression of the cost functions p_n and p_{uv}.

1) We design p_n and p_{uv} as the functions of the physical resources available at physical node n and physical link (u, v).
2) p_n and p_{uv} are desired to be monotonically decreasing functions with respect to the available physical resources.
3) For physical nodes (links) with sufficient physical resources, the unit cost is expected to remain steady, for the nodes (links) with less available physical resources, p_n (p_{uv}) is expected to change more largely as C_n (B_{uv}) changes.

According to the considerations above, we design p_n and p_{uv} as follows:

$$p_n = e^{-\frac{C_n - C_{\min}}{K}} \quad (14)$$

$$p_{uv} = e^{-\frac{B_l - B_{\min}}{K}} \quad (15)$$

where $B_{\min}$ and $C_{\min}$ are the minimum available bandwidth resources at a link and the minimum available CPU resources on a node in the whole network, respectively. Parameter K in the proposed heuristic algorithm is a user-defined constant, which reflects the cost changing rate of utilizing node/link resources with respect to the available resources. Our investigation shows that the cost increase rate becomes higher as the value of K decreases. Therefore, the value of K should be small when we have a resource-limited substrate network, while K can be large when the network-wide physical resources are abundant.

Now, the QoS-guaranteed VNE problem can be formulated as an MINLP as follows:

$$\begin{aligned} (\text{P1}) \quad & \min_{x_n^i, y_{uv}^i, C'_{n,i}, B'_{s_i,d_i}} \quad \mathcal{C} \\ \text{s.t.} \quad & (1),\ (2),\ (7)-(12) \\ & x_n^i \in \{0, 1\} \quad \forall n \in N \\ & y_{uv}^i \in \{0, 1\} \quad \forall (u, v) \in L. \end{aligned} \quad (16)$$

Through the minimization of the total embedding cost $\mathcal{C}$ in (P1), we can find an embedded VN topology together with the optimal resource allocation imposed on each virtual node and virtual link in the VN.

Remark 1: We stress that problem (P1) contains binary variables x_n^i and y_{uv}^i, and continuous variables B'_{s_i,d_i} and $C'_{n,i}$. It also involves a nonlinear objective and several nonlinear constraints in fractional form. Hence, (P1) is an MINLP which is NP-hard [33]. Handling nonlinearities in (16) is quite challenging since it involves a large number of combinatorial variables. Therefore, finding the optimal solution to the problem is computationally complicated, especially for large-scale networks.

V. Problem Transformation and Optimal Solutions

The service requests are considered to be accommodated in a first-come-first-serve (FCFS) manner.[1] Hence, the preceding formulation for a set of service requests can be decomposed into single-service format which remains in the same form for different services. To make (P1) tractable, we transform it into a nonlinear program (NLP). Since x_n^i and y_{uv}^i are variables that indicate whether each substrate node or link is chosen to forward the data traffic, we can transform (P1) into an NLP if we can find the values of x_n^i and y_{uv}^i in advance. According to the above intuitions, an improved brute-force search algorithm is presented in this section that computes the optimal solution to (P1) with relatively high computational overhead. In the next section, we further develop an adaptively weighted heuristic algorithm that can find the near-optimal solution to (P1) with much better scalability.

A. Improved Brute-Force Search Algorithm With Network Pruning

The original MINLP presented in (P1) is simplified to be an NLP by finding alternative routing paths for the embedding of a VN to obtain fixed x_n^i and y_{uv}^i. Next, the optimal resource allocation for the embedded virtual nodes and virtual links are found by finding the solution to the NLP. Following this idea, an improved brute-force search algorithm that can find the optimal solution to the MINLP is presented here. The details of the algorithm are given by Algorithm 1. We first prune the whole substrate network based on (1) and (2) to reduce the search space of finding the optimal path. Denote the substrate network after pruning by $\tilde{G}_i = (\tilde{N}_i, \tilde{L}_i)$. Here, $\tilde{N}_i = \{n | n \in N, C_n > \lambda_i\}$ contains all the substrate node that has enough residual resource to forward the data and $\tilde{L}_i = \{l | l \in L, B_l > \lambda_i\}$ contains all the substrate links that

[1] In this study, we consider that service request arrivals follow a Poisson process and are accommodated in a FCFS manner. For the case of multiple service requests arriving simultaneously, an additional admission control mechanism is required to accommodate the service requests based on their differentiated QoS requirements.

Algorithm 1: Improved Brute-Force Search Algorithm

Input : $G = (N, L)$, s_i, d_i, λ_i, D_i

Prune the whole substrate network $\tilde{G}$ based on (1) and (2) to reduce the search space of finding the optimal path;

Perform a brute-force search process in the whole pruned substrate network to search for all the path candidates $\mathcal{P}_i = \{P_1, P_2, \cdots, P_M\}$ in $\tilde{G}_i$ based on (17);

for $P_m \in \mathcal{P}_i$ **do**

  Fix the values of binary variables x_n^i and y_{uv}^i according to P_m;

  Formulate and solve the NLP (P2) to obtain the optimal resource allocation and the minimum embedding cost for the mth path candidate $\mathcal{C}(P_m)$;

The embedding costs are compared among all the path candidates $\mathcal{C}(P_m)$;

Choose the path candidate with the minimum embedding cost as the optimal substrate path, i.e., $P_m^* = \arg\min_{P_m} \mathcal{C}(P_m)$;

The corresponding optimal resource allocation is imposed onto each virtual node and virtual link along the chosen optimal routing path P_m^*;

Output: Optimal resource allocation $C'_{n,i}$ and B'_{s_i,d_i}
The chosen routing path from s_i to d_i

can be chosen to forward the data. Then, a brute-force search process is performed in the whole pruned substrate network to search for all the substrate paths from s_i to d_i, along which the aggregate propagation link delay is no larger than D_i. Denote the set containing all the E2E substrate paths found by the search process for the ith service request as $\mathcal{P}_i$, i.e.,

$$\mathcal{P}_i = \left\{ P_m \Big| \sum_{l \in \bar{L}(P_m)} \delta_l < D_i \right\}, \quad m = 1, \ldots, M \tag{17}$$

where P_m represents the mth path candidate; M denotes the total number of substrate path candidates; and $\bar{L}(P_m)$ is the set containing all the substrate links exist in the mth substrate path P_m. Among all the routing path candidates $P_1, P_2, \ldots, P_M$, we are about to choose the one with the minimum embedding cost to support the embedded VN for the ith service request.

For service request i, denote the set containing all the substrate nodes and the set containing all the substrate links in the mth path candidate as $N'_i(P_m)$ and $L'_i(P_m)$, respectively. The following NLP is formulated to obtain the optimal resource allocation and the minimal embedding cost for the mth path candidate:

$$\begin{aligned} \text{(P2)} \quad & \min_{C'_{n,i}, B'_{s_i,d_i}} \ \mathcal{C}(P_m) \\ \text{s.t.} \quad & C'_{n,i} \le C_n \quad \forall n \in N'_i(P_m) \\ & B'_{s_i,d_i} \le B_l \quad \forall l \in L'_i(P_m) \\ & \sum_{n \in N'_i(P_m)} \frac{1}{C'_{n,i} - \lambda_i} + \sum_{l \in L'_i(P_m)} \frac{1}{B'_{s_i,d_i} - \lambda_i} + \sum_{l \in L'(P_m)} \delta_l \le D_i \end{aligned} \tag{18}$$

where the objective $\mathcal{C}(P_m)$ is given by

$$\mathcal{C}(P_m) = w_1 \sum_{n \in N'_i(P_m)} p_n C'_{n,i} + w_2 \sum_{l \in L'_i(P_m)} p_l B'_{s_i,d_i}. \tag{19}$$

Note that the objective of such a constrained optimization a linear function, in which the constraints contain a nonlinear inequality constraint and several linear constraints. We can apply the active-set algorithm or the interior-point algorithm to solve the NLP to find the optimal resource allocation with the minimum embedding cost for P_m.

Finally, the embedding costs are compared among all the path candidates and we choose the path candidate with the minimal embedding cost as the optimal substrate path, denoted by P_m^*, found from

$$P_m^* = \arg\min_{P_m} \ \mathcal{C}(P_m), \quad m \in \{1, 2, \ldots, M\}. \tag{20}$$

At the same time, the corresponding optimal resource allocation is imposed onto each virtual node and virtual link along the optimal routing path P_m^*.

VI. Adaptively Weighted Heuristic Algorithm With Low Complexity

Although the improved brute-force search algorithm is able to find the optimal solution to the MINLP, it involves high computational complexity and is not scalable to large-scale networks. Take a 50-node and 123-link substrate network as an example, the number of path candidates found by the brute-force search process to embed an E2E VN with delay requirement 40 ms is 445. As a result, the time it takes to find the optimal routing path is about 14.8 min on a computer with an Intel i7-9750H CPU @2.6 GHz. In practice, VNE has to be done very quickly, even faster than the service request's arrival. Therefore, in what follows, we propose an adaptively weighted heuristic algorithm with low computational complexity to obtain near-optimal solutions to (16) efficiently.

A. Handling Single Service Request

The main advantage of the adaptively weighted heuristic algorithm lies in that it chooses one routing path without performing a brute-force search process while obtaining near-optimal solutions. The algorithm is a two-step algorithm. From the source to the destination of each service request i, we first find a substrate path with relatively low embedding cost. Then, the optimal resource allocation on the virtual nodes and virtual links are computed for the routing path found previously.

Algorithm 2 shows the steps of the proposed adaptively weighted heuristic algorithm. First, we choose the substrate path such that the chosen path has great potential to support the ith VN with a relatively low embedding cost. To this end, an adaptive weight is defined and assigned to each physical link in the whole substrate network. Next, the k-shortest path algorithm [34] is utilized to find the shortest path from source to the destination of the ith service request. This shortest path is then selected as the routing path. Subsequently, the optimal resource allocation on the virtual nodes and virtual links are

Algorithm 2: Adaptively Weighted Heuristic Algorithm With Low Complexity

Input : $G = (N, L)$, s_i, d_i, λ_i, D_i

1: Prune the substrate network based on (1) and (2) to obtain $\tilde{G}_i$;
2: For each link (u, v) in $\tilde{G}_i$, find its new weight w_{uv} based on (21);
3: With the new weights, use shortest path algorithms to find the shortest routing path from s_i to d_i;
4: Fix the values of binary variables x_n^i and y_{uv}^i in (16) according the shortest path found previously;
5: Formulate and solve (16) where binary variables are fixed to get the optimal resource allocation and the minimum embedding cost;

Output: Optimal resource allocation B'_{s_i,d_i} and $C'_{n,i}$
The selected routing path from s_i to d_i

found for the selected routing path. We also prune the original substrate network based on (1) and (2). Then, weight w_{uv} is defined for each link adaptively in the pruned substrate network, shown as follows:

$$w_{uv} = \eta_d \delta_{uv} + \eta_c \left(e^{-\frac{B_{uv}-B_{\min}}{K}} + e^{-\frac{C_u-C_{\min}}{K}} + e^{-\frac{C_v-C_{\min}}{K}} \right) \quad (21)$$

where B_{uv} and δ_{uv} are the available link resource and the link propagation delay on link $(u, v) \in \tilde{L}_i$, respectively, η_d and η_c are weighting coefficients, with $\eta_d + \eta_c = 1$, and C_v and C_u are the CPU resources available on node $v \in \tilde{N}_i$ and node $u \in \tilde{N}_i$, respectively. The intuition behind (21) is that we prefer the physical links with a larger amount of available bandwidth resources and smaller propagation delay to be chosen, and prefer physical nodes with a larger amount of CPU resources to be chosen to form the routing path for the current service request. This gives us a higher chance to obtain a low embedding cost after solving the NLP. Once finding the routing path using the shortest path algorithm is done, the optimal resource allocation imposed on the nodes and links in the VN is found by solving the NLP (P2).

B. Extension to Online Scenarios

The proposed adaptively weighted heuristic algorithm can be used for efficient VN topology design and embedding. Hence, we intend to extend it to handle different service requests in online scenarios.[2] In online scenarios, we are faced with consistent arrivals and leavings of VN requests over time. Fig. 4 illustrates extending the adaptively weighted heuristic algorithm to online scenarios. Each time a new VN request arrives at the substrate network, the proposed adaptively weighted heuristic algorithm is triggered to perform QoS-guaranteed VNE. If the embedding process is a success, the corresponding service request will be accommodated by the substrate network. The physical resource status of substrate nodes and links is simultaneously updated. Otherwise,

[2]In online scenarios, if the improved brute-force search algorithm is applied, the computational complexity would be extremely high, as illustrated in the next section (i.e., Section VII).

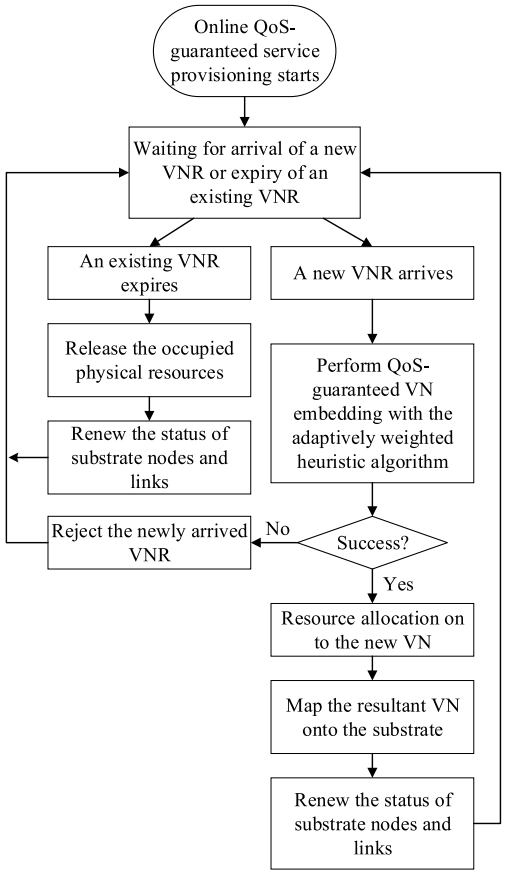

Fig. 4. Illustration of extending the adaptively weighted heuristic algorithm to online scenarios.

the substrate network will reject the service request. When an existing service request expires based on its lifetime, the physical resource status of the substrate nodes and links is renewed the corresponding resources occupation for the VN.

VII. Performance Evaluation

To evaluate the performance of the proposed algorithms, a discrete-event network simulator has been developed by C++. The GT-ITM tool was used for the generation of mesh substrate networks and service requests [35]. This enables efficient customization of both service requests and the substrate network. The link propagation delay is set to be proportional to the distance between two end nodes. All the substrate nodes and links have a capacity which is a random variable following a uniform distribution. The weighting parameters in (13) are set as $w_1 = 0.1$ and $w_2 = 0.1$. The value of constant K is set as 60.0. The two parameters for adaptive weights are set as $\eta_d = 0.4$ and $\eta_c = 0.6$.

Two benchmark algorithms are chosen for comparison.

1) Equal-delay resource allocation and resource greedy link embedding. We choose the substrate path with the maximum residual resources as the routing path to support the targeted VN. The resources allocation on the nodes and links along this routing path are obtained from forcing the packet queueing delay over each node and link to be equal.

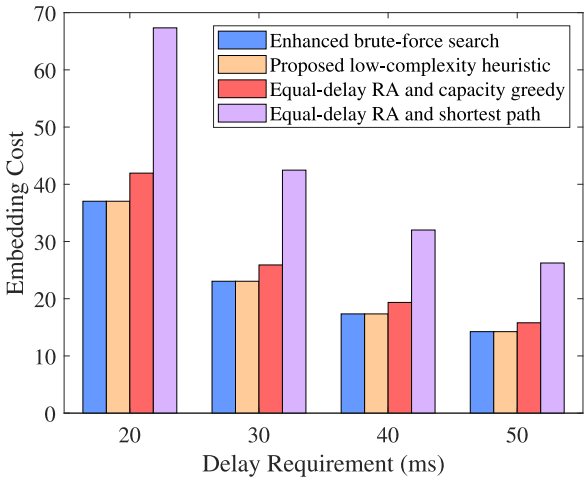

Fig. 5. Performance comparison among the four algorithms with varying E2E delay requirements (50-node substrate network, and the traffic arrival rate is 150 packet/s).

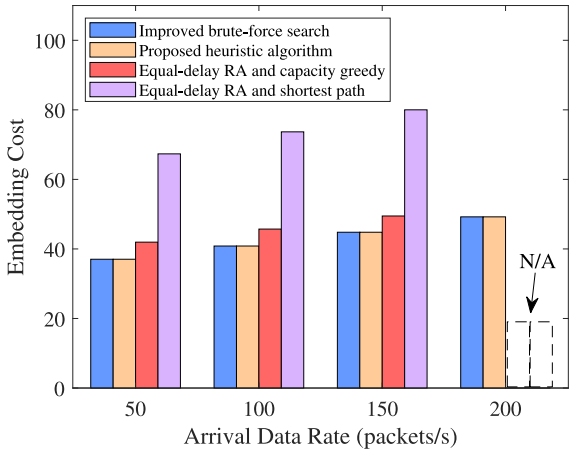

Fig. 6. Performance comparison among the four algorithms (50-node substrate network, and the E2E packet delay requirement is 35 ms).

2) Equal-delay resource allocation and shortest path link embedding. We choose the routing path using the shortest path algorithm [15], and use equal-delay resource allocation to obtain the resources allocated to the nodes and links. The performance of the proposed two algorithms is first evaluated in a single-service scenario. Then, the advantages of the proposed adaptively weighted heuristic algorithm are demonstrated in different online scenarios.

A. Single-Service Scenario of QoS-Guaranteed VNE

1) Embedding Cost: Consider a substrate network with 50 nodes and 130 links, where the capacities of all the substrate nodes and substrate links have uniform distribution inside [600 packet/s, 800 packet/s]. For the service request, we randomly choose the source and destination nodes from all the substrate nodes. We set the traffic arrival rate as 150 packet/s, and the E2E packet delay requirement is varied from 20 ms to 50 ms. The comparison of the embedding costs among the four algorithms is shown in Fig. 5. It can be seen from the figure that the embedding cost enlarges as the E2E delay requirement becomes more strict. The reason for this is that a more stringent delay requirement typically requires more CPU processing and bandwidth resources, which leads to a higher embedding cost. Moreover, we stress that the improved brute-force search method and the proposed heuristic algorithm find the same routing path. As a result, the embedding costs obtained from the two algorithms are the same for all cases. We also observe that the performance gap between the proposed algorithms and the benchmark algorithms enlarges as the delay requirement gets more stringent, which shows the effectiveness of the proposed heuristic algorithm in terms of optimizing the allocated resources.

Next, QoS-guaranteed VNE is performed for the same service request, where the delay requirement is fixed and the traffic arrival rate is varying. The performance comparison among the four algorithms is shown in Fig. 6. We can see that the cost of the proposed heuristic algorithm and that of the improved brute-force search algorithm are the same for all the feasible cases (all the cases except for $\lambda_i = 200$). Their cost is lower than those obtained from the two benchmark algorithms. This is, again, due to that the same routing path is found by the proposed heuristic algorithm and the improved brute-force search method. In comparison, the packet forwarding and routing paths found by the other two benchmarks are suboptimal with a higher embedding cost. For the infeasible case, the embedding costs obtained from the two benchmarking algorithms are not available due to that the service request is rejected when $\lambda = 200$. What can also be concluded is that the embedding cost goes higher as the traffic arrival rate increases for all the algorithms in comparison.

TABLE III
SCALE OF THE THREE SUBSTRATE NETWORKS USED FOR SIMULATION

Parameter	Substrate I	Substrate II	Substrate III
Number of substrate nodes	50	75	100
Number of substrate links	130	335	552

Finally, the CPU resource allocation result at various substrate nodes for the considered VN is presented in Fig. 7. From the figure, we can see that there are two substrate nodes chosen to be the intermediate nodes. The relations between CPU processing rates and the arrival data rates of different nodes are diversified due to their different unit costs. It can be seen that the optimal CPU resource allocation at each substrate node on the routing path has an approximately linear relation with the arrival data rate, but is upper bounded by the maximum node capacity. Fig. 8 shows a similar trend, in which the optimal link resource allocation on the two consecutive substrate links (i.e., Physical link 2 and Physical link 3) shows a linear relationship with respect to the arrival data rate. The optimal link resource allocation over the other substrate links (i.e., Physical link 1) also increases and is upper bounded by the maximum link capacity.

2) Time Complexity: Three substrate networks of different scales are used to analyze the execution time of the two proposed algorithms. The parameters showing the scales of the three networks are given by Table III. Three different service

TABLE IV
EXECUTION TIME COMPARISON OVER THREE SUBSTRATE NETWORKS OF DIFFERENT SCALES

Execution time / Delay requirement	Substrate I		Substrate II		Substrate III	
	Improved brute-force search	Adaptively weighted heuristic	Improved brute-force search	Adaptively weighted heuristic	Improved brute-force search	Adaptively weighted heuristic
30 ms	2.95 s	1.41 s	13 s	1.07 s	85 s	1.17 s
35 ms	5.89 s	1.45 s	103 s	1.14 s	408 s	1.21 s
40 ms	7.37 s	1.45 s	346 s	2.04 s	1358 s	1.25 s
45 ms	10.31 s	2.08 s	720 s	2.13 s	2306 s	1.58 s

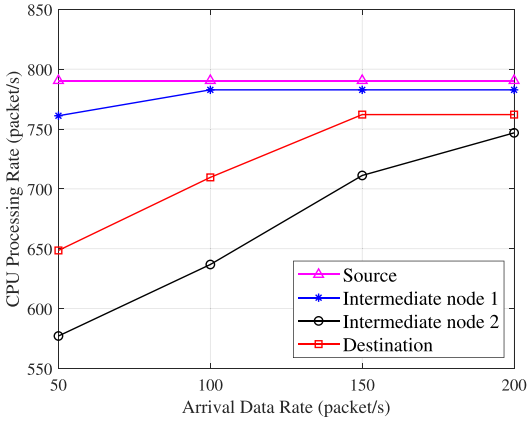

Fig. 7. Optimal CPU resource allocation on various intermediate substrate nodes with different arrival data rates (we set the packet delay requirement as 35 ms).

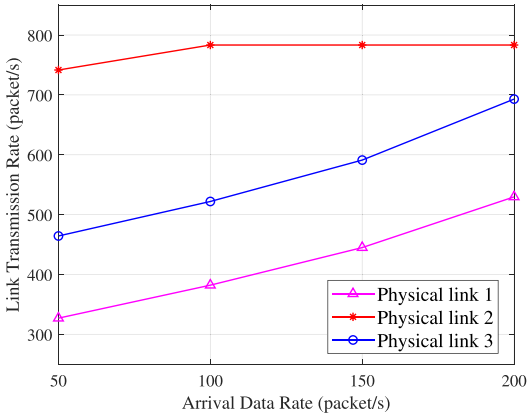

Fig. 8. Optimal link resource allocation with different arrival data rates (we set the packet delay requirement as 35 ms).

requests are used for the three different substrate networks. The E2E delay requirement and the traffic arrival rate for each service request are set to be random numbers in the range of [20 ms, 50 ms] and [150, 250], respectively. Both algorithms are executed on a Legion Y540 Laptop with an Intel i7-9750H CPU @2.6 GHz. The proposed algorithms are executed in a centralized way.

The execution time of the two algorithms for the three substrates is shown in Table IV. For the first substrate network (i.e., Substrate I), the execution time of the improved brute-force search algorithm varies from 2.95 to 10.31 s for different delay requirements. This is 2–5 times that of the low-complexity adaptively weighted heuristic algorithm. For the second (third) substrate network, the execution time of the improved brute-force search algorithm is about 10–300 times (60–1400 times) that of the low-complexity adaptively weighted heuristic algorithm. This comparison implies that the proposed heuristic algorithm scales significantly better than the improved brute-force search algorithm. We want to point out that the main reason for this is that the adaptively weighted heuristic algorithm defines a new weight for each link in the network and finds the weighted shortest path based on the new weights. Instead, the improved brute-force search algorithm finds all the path candidates and thus its complexity highly relies on the total number of routing path candidates.

B. Online Scenario of QoS-Guaranteed VNE

In online scenarios, we generate the service requests and let them arrive to the substrate network following a Poisson process. We consider that each VN request's lifetime has an exponential distribution with an average lifetime 10 s. We set the service's average arrival rate as four VNs per second and set the total simulation time to be 45 s. The VN acceptance ratio is used as the performance metric, i.e., the ratio of the number of accepted service requests over the total number of service requests arrived to the network.

1) Impact of Arriving Traffic Data Rate: Consider online VN topology design and embedding with different arriving data rates (λ_i) of service requests on a 50-node substrate network. The E2E delay requirement has a uniform distribution in the range of 30 and 45 ms. We consider three cases, where the range of arrival data rate for the service requests is [150, 250], [200, 400], and [400, 500]. The service acceptance ratio comparison of the three algorithms is shown in Fig. 9 (a). We can see that for all the cases, the proposed low-complexity heuristic algorithm achieves the highest service acceptance ratio. Another conclusion is that for all the three algorithms in comparison, the service acceptance ratio becomes smaller as the average arrival data rate enlarges. This is because a larger amount of node/link resources are required to ensure a more stringent E2E packet queueing delay when λ increases. This will lead to a higher chance of the rejection of newly arrived services. The last conclusion that can be drawn is that, as λ increases, the performance gap enlarges. This illustrates the superiority of the proposed algorithm in scenarios where the traffic is dense.

2) Impact of E2E Delay Requirement: Consider online VN topology design and embedding with different delay

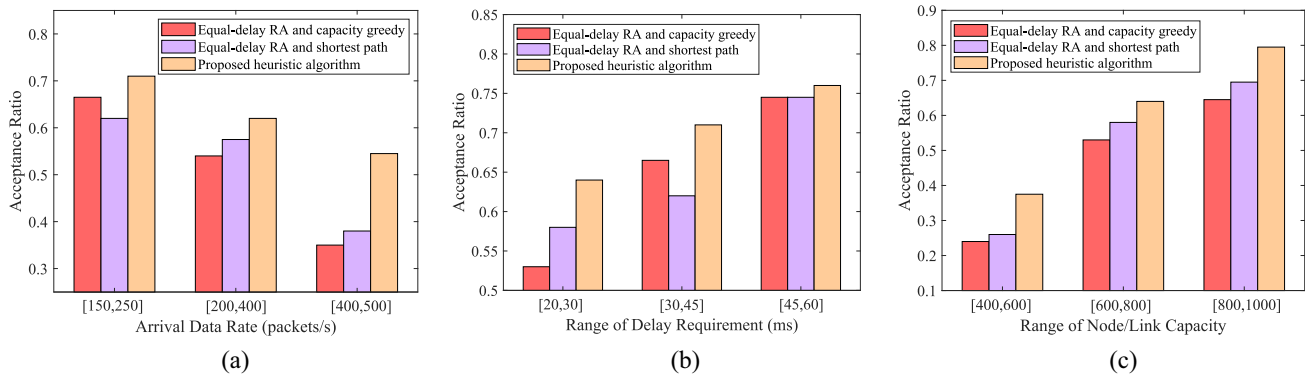

Fig. 9. Comparison of performance among the three algorithms from different aspects. (a) Varying arrival data rate of VN requests. (b) Varying delay requirement of VN requests. (c) Varying node/link capacities of the substrate network.

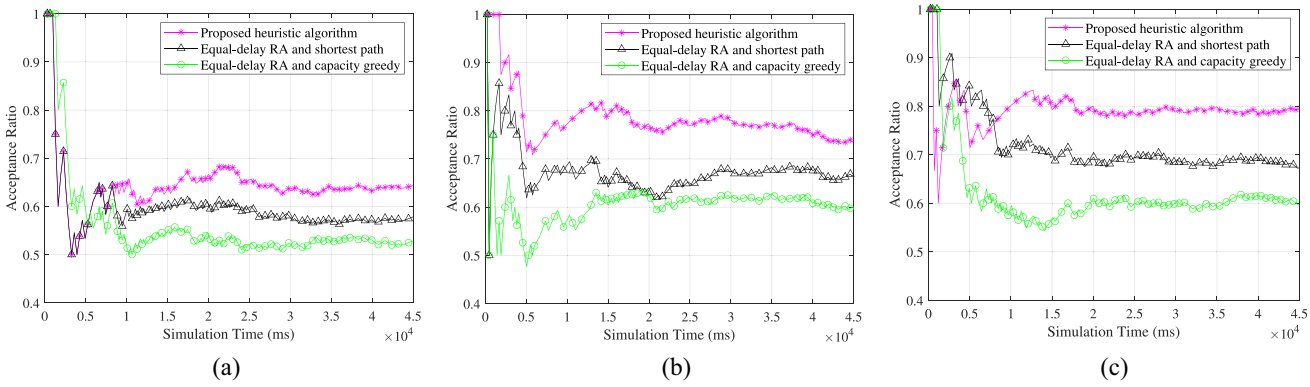

Fig. 10. Comparison of acceptance ratio among the three algorithms over three substrate networks of different scales. (a) Substrate I. (b) Substrate II. (c) Substrate III.

requirements on the same 50-node substrate network. The delay requirement for the service requests has a uniform distribution in the range of [20, 30], [30, 45], and [45, 60] (all in ms), respectively. The arrival data rates are set to have uniform distribution inside 150 and 250 (packets/s). Fig. 9(b) shows the performance comparison of the three algorithms. We can see that the performance of the proposed algorithm is the best in all cases under consideration. On average, using the proposed algorithm, the substrate network can admit about 10% more service requests than that using the two benchmarking algorithms. Moreover, the service acceptance ratio enlarges as the average E2E delay requirement increases (i.e., becomes looser). This is reasonable because with a loose E2E delay requirement a longer packet queueing delay can be tolerated by the service request. This means less resource consumption and, therefore, more service requests can be accommodated by the substrate network.

3) Impact of Node and Link Capacity: Now, we consider online VN topology design and embedding with different average node/link capacities over the 50-node substrate network. We consider three cases, where the ranges of the node/link capacities are set as [400, 600], [600, 800], and [800, 1000], respectively. The service requests' arrival data rates are uniformly distributed inside [150, 250] packets/s. The VN requests' E2E packet delay requirement is uniformly distributed inside [20, 30] ms. The performance comparison between the three different algorithms is shown in Fig. 9(c). From the figure, we can see that the proposed low-complexity heuristic algorithm achieves a higher acceptance ratio over the other two algorithms in all three cases.

4) Performance Comparison Over Different Network Scales: Finally, the performance comparison between the proposed low-complexity heuristic algorithm and the two benchmarks is compared over different substrate networks, i.e., the three substrate networks as given in Table III. The range of E2E packet delay requirement and that of arrival data rate for the VN requests are set as [20, 30] (ms) and [150, 250] (packets/s), respectively. The available CPU and bandwidth resources are both uniformly distributed inside [600, 800]. The service acceptance ratio comparison for the three algorithms for the three network scales is shown in Fig. 10. We can see that the proposed algorithm maintains the highest acceptance ratio in all the cases in comparison. The main rationality behind this is that the resource allocation on each VN found by the proposed algorithm is optimized to ensure the QoS requirements. Furthermore, by optimizing the resource allocation, resource redundancy for supporting each VN is avoided and, thus, the physical resource utilization in the whole network is improved.

VIII. Conclusion and Future Work

In this article, a joint optimization problem has been formulated for VN topology design and VNE with the objective of minimizing the VNE cost, while ensuring the differentiated

QoS requirements in cybertwin-enabled 6G core networks. Two algorithms have been developed to solve the formulated problem for substrate networks of small and large scales, respectively. Simulations results have demonstrated that the proposed approach can improve the overall resource utilization and facilitate QoS-guaranteed service provisioning for cybertwin-enabled 6G core networks. For future work, we will consider artificial intelligence-empowered cybertwin for QoS-guaranteed VNE in a dynamic network environment.

REFERENCES

[1] X. You *et al.*, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China Inf. Sci.*, vol. 64, no. 1, pp. 1–74, 2021.

[2] S. Verma, S. Kaur, M. A. Khanand, and P. S. Sehdev, "Toward green communication in 6G-enabled massive Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5408–5415, Apr. 2021.

[3] N. Zhang, P. Yang, J. Ren, D. Chen, L. Yu, and X. Shen, "Synergy of big data and 5G wireless networks: Opportunities, approaches, and challenges," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 12–18, Feb. 2018.

[4] G. Sun, Z. Xu, H. Yu, X. Chen, V. Chang, and A. V. Vasilakos, "Low-latency and resource-efficient service function chaining orchestration in network function virtualization," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5760–5772, Jul. 2020.

[5] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proc. IEEE*, vol. 108, no. 2, pp. 274–291, Feb. 2020.

[6] X. Tian, W. Huang, Z. Yu, and X. Wang, "Data driven resource allocation for NFV-based Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8310–8322, Oct. 2019.

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

[8] X. Shen *et al.*, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, no. 1, pp. 45–66, Feb. 2020.

[9] A. J. Gonzalez, G. Nencioni, A. Kamisinski, B. E. Helvik, and P. E. Heegaard, "Dependability of the NFV orchestrator: State of the art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3307–3329, 4th Quart., 2018.

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

[11] J. Li, W. Shi, N. Zhang, and X. Shen, "Delay-aware VNF scheduling: A reinforcement learning approach with variable action set," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 304–318, Mar. 2021, doi: 10.1109/TCCN.2020.2988908.

[12] Q. Yu, J. Ren, Y. Fu, Y. Li, and W. Zhang, "Cybertwin: An origin of next generation network architecture," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 111–117, Dec. 2019.

[13] Q. Yu *et al.*, "A fully-decoupled RAN architecture for 6G inspired by neurotransmission," *J. Commun. Inf. Netw.*, vol. 4, no. 4, pp. 15–23, Dec. 2019.

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

[15] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *Proc. ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Apr. 2008.

[16] X. Cheng *et al.*, "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 38–47, Apr. 2011.

[17] S. Zhang, Z. Qian, J. Wu, S. Lu, and L. Epstein, "Virtual network embedding with opportunistic resource sharing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 816–827, Mar. 2014.

[18] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Toronto, ON, Canada, May 2014, pp. 1–9.

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

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

[21] S. R. Chowdhury *et al.*, "Multi-layer virtual network embedding," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 3, pp. 1132–1145, Sep. 2018.

[22] A. Song, W.-N. Chen, T. Gu, H. Zhang, and J. Zhang, "A constructive particle swarm optimizer for virtual network embedding," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1406–1420, Jul.–Sep. 2020.

[23] M. Zhang, C. Wu, M. Jiang, and Q. Yang, "Mapping multicast service-oriented virtual networks with delay and delay variation constraints," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Miami, FL, USA, Dec. 2010, pp. 1–6.

[24] M. M. A. Khan, N. Shahriar, R. Ahmed, and R. Boutaba, "Multi-path link embedding for survivability in virtual networks," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 2, pp. 253–266, Jun. 2016.

[25] Z. Cai, F. Liu, N. Xiao, Q. Liu, and Z. Wang, "Virtual network embedding for evolving networks," in *Proc. IEEE Global Telecommun. Conf. (GlOBECOM)*, Miami, FL, USA, Dec. 2010, pp. 1–5.

[26] G. Chochlidakis and V. Friderikos, "Low latency virtual network embedding for mobile networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, 2016, pp. 1–6.

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

[28] W. Wang, B. Liang, and B. Li, "Multi-resource generalized processor sharing for packet processing," in *Proc. IEEE/ACM 21st Int. Symp. Qual. Serv. (IWQoS)*, Montreal, QC, Canada, Jun. 2013, pp. 1–10.

[29] 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.

[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] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1987.

[32] O. Alhussein *et al.*, "A virtual network customization framework for multicast services in NFV-enabled core Networks," *IEEE J. Sel. Areas of Commun.*, vol. 38, no. 6, pp. 1025–1039, Jun. 2020.

[33] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys Oper. Res. Manage. Sci.*, vol. 17, no. 2, pp. 97–106, 2012.

[34] D. Eppstein, "Finding the k shortest paths," *SIAM J. Comput.*, vol. 28, pp. 652–673, Feb. 1999.

[35] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, San Francisco, CA, USA, 1996, pp. 594–602.

Junling Li (Member, IEEE) received the B.S. degree from Tianjin University, Tianjin, China, in 2013, the M.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2020.

She is currently a Joint Postdoctoral Research Fellow with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, University of Waterloo, and the Chinese University of Hong Kong, Shenzhen. Her interests include game theory, machine learning, software-defined networking, network function virtualization, and vehicular networks.

Dr. Li received the Best Paper Award at the IEEE/CIC International Conference on Communications in China in 2019.

Weisen Shi (Graduate Student Member, IEEE) received the B.S. degree from Tianjin University, Tianjin, China, in 2013, the M.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2020.

His interests include spaceŰairŰground-integrated networks, UAV communication and networking, and RAN slicing.

Qiang Ye (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2016.

From December 2016 to September 2019, he was a Postdoctoral Fellow and a Research Associate with the Department of Electrical and Computer Engineering, University of Waterloo. Since September 2019, he has been an Assistant Professor with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN, USA. His research interests include network slicing for 5G networks, edge intelligence for autonomous vehicular networks, artificial intelligence for future networking, protocol design, and performance analysis for the Internet of Things.

Dr. Ye is the Editor of the *International Journal of Distributed Sensor Networks* (SAGE Publishing) and *Wireless Networks* (Springer Nature), and an Area Editor of the *Encyclopedia of Wireless Networks* (Springer Nature). He was a Technical Program Committee Member for several international conferences, including the IEEE GLOBECOM'20, VTC'17, VTC'20, and ICPADS'20.

Shan Zhang (Member, IEEE) received the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2016.

She is currently an Assistant Professor with the School of Computer Science and Engineering, Beihang University, Beijing. She was a Postdoctoral Fellow with the University of Waterloo, Waterloo, ON, Canada, from 2016 to 2017. Her research interests include mobile-edge computing, network virtualization, and intelligent management.

Dr. Zhang received the Best Paper Award at the Asia–Pacific Conference on Communication in 2013.

Weihua Zhuang (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Dalian Maritime University, Dalian, China, and the Ph.D. degree in electrical engineering from the University of New Brunswick, Fredericton, NB, Canada.

She has been with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, since 1993, where she has been a Professor and a Tier I Canada Research Chair of Wireless Communication Networks.

Prof. Zhuang was a recipient of the 2021 R. A. Fessenden Award from the IEEE Canada, the 2017 Technical Recognition Award in Ad Hoc and Sensor Networks from the IEEE Communications Society, 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, a Technical Program Chair/Co-Chair of IEEE VTC 2017/2016 Fall, and a Technical Program Symposia Chair of IEEE Globecom 2011. She is an elected member of the Board of Governors and a Vice President for Publications of the IEEE Vehicular Technology Society. She was an IEEE Communications Society Distinguished Lecturer from 2008 to 2011. He is a Fellow of the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular *ad hoc* and sensor networks.

Dr. Shen received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, 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 (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, and IEEE Globecom'07, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the President Elect of the IEEE ComSoc. He was the Vice President for Technical & Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*. 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 Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.