IEEE ComSoc IEEE Communications Society | IEEE Open Journal of the Communications Society

Received 28 January 2024; accepted 18 February 2024. Date of publication 18 March 2024; date of current version 19 April 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3378088

CASMaT: Characteristic-Aware SFC Mapping for Telesurgery Systems in Cloud-Edge Continuum

SEYEDREZA TAGHIZADEH[1] (Member, IEEE), HALIMA ELBIAZE[1] (Senior Member, IEEE), ROCH H. GLITHO[2] (Senior Member, IEEE), AND WESSAM AJIB[1] (Senior Member, IEEE)

[1]Computer Science Department, Université du Québec à Montréal, Montreal, QC H3C 3P8, Canada

[2]Concordia Institute for Information Systems Engineering, Montreal, QC H3G 2W1, Canada

CORRESPONDING AUTHOR: S. TAGHIZADEH (e-mail: taghizadeh.seyedreza@courrier.uqam.ca)

This work was supported by the Fonds de recherche du Québec – Nature et technologies (FRQNT) Team Research Project under Grant 299847.

ABSTRACT Network Function Virtualization (NFV) empowers Internet Service Providers (ISPs) to place Virtual Network Functions (VNFs) efficiently in order to enhance the network performance without incurring a high cost. In this environment, Service Function Chains (SFCs) always need to steer the traffic through a sequence of VNF instances. Therefore, ISPs must adopt a suitable SFC embedding strategy to bolster their revenue. However, existing VNF placement and chaining methodologies harbour unrealistic assumptions, as they tackle the mapping problem from a generic standpoint, overlooking the distinctive characteristics of the constituent VNFs within a chain. Hence, they are not efficient when the strict requirements of life-critical applications, such as telesurgery, need to be satisfied. In this paper, taking into account the strict requirements of telesurgery-like systems, we present a cost-efficient characteristic-aware SFC mapping method for telesurgery Systems in the Cloud-Edge continuum. We formulate this problem as a Binary Linear Programming (BLP) model to embed SFC requests at minimal cost. Also, we propose an innovative heuristic algorithm that allocates each VNF based on its distinctive characteristics. Simulation results demonstrate that taking the characteristics of the VNFs into account when addressing the placement problem improves the system performance notably.

INDEX TERMS SFC mapping, network function virtualization, characteristic-aware NFV placement, telesurgery, cloud-edge continuum.

I. INTRODUCTION

CLOUD service providers use network functions like address translation, packet inspection, and firewall to enhance performance. NFV offers cost-effective alternatives by decoupling functions from physical devices, allowing VMs to run VNFs, improving performance and efficiency [1], [2], [3], [4]. However, concerns about reliability arise, particularly for critical applications like telesurgery.

Different VNF types have specific attributes. For instance, stateful VNFs like NAT require ongoing user connections, necessitating backup NATs to prevent disconnections. Transcoders' placement and traffic changes impact network performance. Considering network function characteristics in active, backup, and update paths can lower provider network costs. The study starts with an illustrative use case and requirements (Section II). It then presents research contributions and objectives (Section III). Relevant literature is reviewed (Section IV). The problem formulation and proposed heuristic algorithm are detailed in Section V. Evaluation includes a comparison with another method in Section IX. The paper concludes in Section X.

II. ILLUSTRATIVE USE CASE AND REQUIREMENTS

A. ILLUSTRATIVE USE CASE

Fig. 1 depicts a telesurgery scenario with four geographically distant hospitals denoted as *A*, *D*, *SC*, and *SR*. The blue line represents the *Active* path for endoscopic video streaming from *D* to *A*. The red and green paths indicate the *Backup* and State Update paths, respectively. The SFC includes H.264 codec and encryption VNFs. The cloud-edge continuum consists of nodes with varying resources and capabilities. Nodes and VNF failures are possible, and during placement,

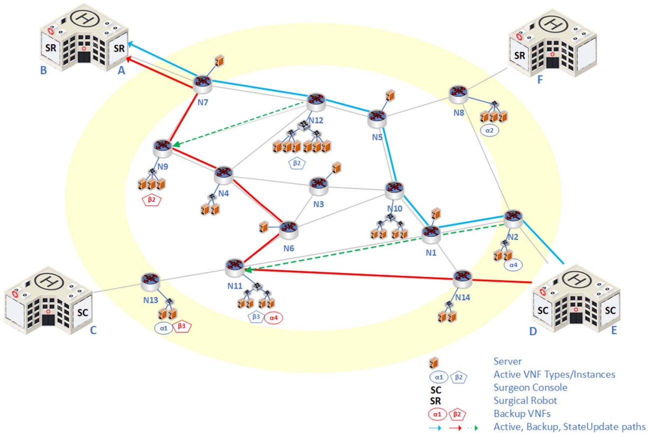


FIGURE 1. Telesurgery Use Case.

TABLE 1. Model setting.

Parameter	Value
Number of Nodes	20
Number of Requests	3,6,9,12,15,18,21,24
Number of VNFs per chain	4
Service Delay Threshold(ms) (d_r in BLP)	<100
Bandwidth Cost (Dollar/Gb)	5 - 20
Memory Cost (Dollar/Gb)	10 - 25
VNF License Cost (Dollar/vCPU)	100
End-user's Load (Gbps)	0.5 - 3
CPU demand (per VNF)	2 - 8
Memory Demand (Gb per VNF)	4 - 16
Traffic change ratio	0.5 - 4

backup instances avoid being on the same node as active instances. We focus on VNF placement and chaining, excluding failure recovery scenarios. Table 1 provides more information about the setting of the scenario.

B. REQUIREMENTS

provisioning telesurgery services in the cloud-edge continuum requires achieving optimal VNF placement to address key goals. Firstly, high reliability is critical due to the life-critical nature of telesurgery [5], [6]. Ensuring redundancy with backup VNFs allows for a swift takeover if the active chain fails [7]. Secondly, cost-efficiency is crucial in minimizing placement costs for both active and backup VNFs. It is essential to consider the unique characteristics of VNFs in each SFC request [8]. Lastly, delivering services with ultra-low latency is of utmost importance to maintain the effectiveness of telesurgery.

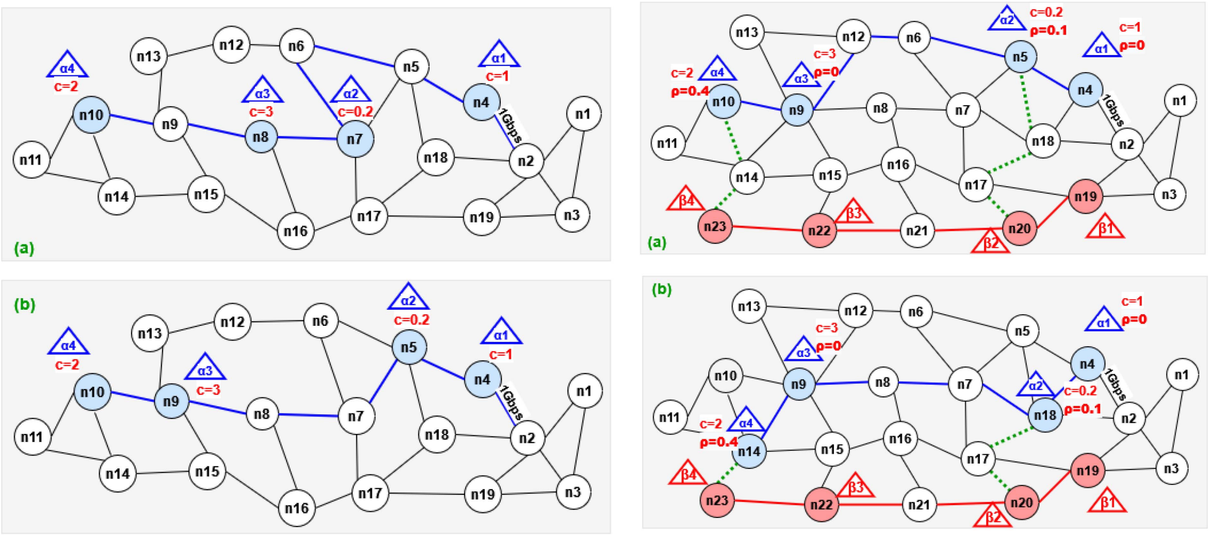
The accepted delay in telesurgery depends on the specific surgery being performed. Nonetheless, there are general guidelines concerning delays in telesurgery. Researches show that delays of 200 ms are considered ideal for telesurgery, while 300 milliseconds is also suitable [9]. Latencies of 400-500 ms may be acceptable but can be tedious, and latencies of 600-700 ms are difficult to deal with and only acceptable for low-risk and simple surgeries. Surgery becomes very difficult with a delay of 800 to 1000 ms, and in such cases, telementoring is a better choice [9]. In another study [10], the possibility of performing telesurgery over long communication links has been studied. This study has indications of successful telesurgery procedures over a satellite link with approximately 600 ms. Overall, while the accepted delay in telesurgery varies based on the procedure and other factors, latency between 200 ms to 300 ms is generally considered ideal, while higher latency may still be acceptable for certain procedures [10].

III. PROPOSED SOLUTION

In telesurgery systems, certain VNFs have distinct requirements that demand special consideration. For example, real-time endoscopic image streaming requires maintaining the codec state throughout the VNF chain. Similarly, VNFs responsible for patient privacy and security should be placed close to the network edge. Each VNF type exhibits unique characteristics, setting it apart in terms of requirements and capabilities. Current SFC mapping methods often overlook these specific VNF characteristics with some unrealistic assumptions. They assume the outgoing traffic rate of each VNF is the same as the incoming traffic rate. Besides, a great portion of the works suppose that the VNF is either stateless or it is compatible with all kinds of hardware. No research work considers all the attributes that we have considered, namely traffic change, geo-location dependency, statefulness, and hardware appropriateness. Our proposed mapping method considers these attributes, aiming to optimize SFC placement comprehensively. In addition to CPU, memory, and bandwidth consumption, we also consider the following relevant attributes for each VNF:

In/Out Traffic Ratio: Our model considers the traffic variation resulting from VNF processing, which can cause substantial differences between incoming and outgoing traffic. This accurate representation of bandwidth fluctuations allows for more realistic and efficient resource allocation for SFC requests.

Figure 2(a)(a) illustrates an SFC with four VNFs, and their active instances are denoted as $\alpha 1, \alpha 2, \alpha 3$, and $\alpha 4$. The traffic change ratio is represented by c. Assuming the traffic demand for this SFC request is 1Gbps, the total bandwidth cost of the chain is calculated as: $((1 \times 1 \times (1+1+1)) + (1 \times 0.2 \times 1) \ + (0.2 \times 3 \times (1+1)) = 4.4$. However, if



(a) In/Out traffic change

(b) Statefulness

FIGURE 2. VNF characteristics.

traffic change is considered during VNF mapping, as shown in Fig 2(a)(b), the total bandwidth cost of the chain reduces to: $(1\times 1\times 1)$ $+(1\times 0.2\times(1{+}1{+}1{+}1)){+}(0.2\times 3\times 1)$ $=2.4$.

Geo-Location dependency: Some VNFs are location-dependent, such as those handling security and patient privacy, which are better placed close to end-users. In contrast, other VNFs like load balancers are not location-restricted and can be placed anywhere in the cloud-edge continuum. Our model considers the geo-location dependency of VNFs for appropriate placement decisions.

Statefulness: In our model, we distinguish between stateful and stateless VNFs. Stateful VNFs retain interaction status, requiring constant state updates for standby instances. Stateless VNFs, on the other hand, don't need state updates during mapping. Considering statefulness is crucial for optimal VNF placement and chaining in telesurgery scenarios.

To illustrate the statefulness characteristic, we refer to Fig. 2(b) where $\alpha 2$ and $\alpha 4$ are stateful, while $\alpha 1$ and $\alpha 3$ are stateless. considering updates between the active and backup paths, ρ denotes the state update ratio, and c represents the traffic change ratio. Fig. 2(b)(a) shows the mapping without considering statefulness, resulting in a total update cost of: $(1\times 0.1\times(1+1+1+1))+(0.6\times 0.4\times(1+1))=0.88$. However, in Fig. 2(b)(b), a better placement considering stateful characteristics reduces the update cost to: $(1\times 0.1\times(1+1))+(0.6\times 0.4\times 1)=0.44$. This emphasizes the need to consider VNF characteristics during mapping for effective placement.

Hardware appropriateness: In VNF placement, specific hardware requirements for each VNF are crucial. For example, DPI may require hardware-accelerated encryption, and IDS functions might have specific hardware demands. Considering the suitability of server hardware for each VNF is essential for realistic VNF placement. AWS offers various volume types with different IOPS capacities, such as io2 volumes, making them ideal for performance-intensive and critical applications. Leveraging such specialized volume types optimizes VNF performance effectively.

In a nutshell, the contributions of this paper can be summarized as follows:

- Considering memory, CPU, bandwidth, and geo-locations in the substrate network on the one hand, and the requirements of the telesurgery systems and characteristic-based requirements of the VNFs in each SFC on the other hand, we formulate the VNF placement problem as a BLP model, aiming to minimize the embedding cost for each SFC request.
- We propose a novel algorithm to place and chain VNF instances while respecting the requirements of the telesurgery systems.

Our research provides novel insights for VNF placement optimization, particularly within the context of telesurgery. To the best of our knowledge, this is the first paper to discuss the characteristic aware SFC mapping for reliable telesurgery systems in the cloud-edge continuum.

IV. RELATED WORKS

Our characteristic-based SFC mapping perspective is novel compared to existing works. Some papers have partially focused on VNF characteristics, but none of them considered all the specific characteristics in the mapping process. For example, in [11], the authors proposed a method based on network resource utilization but neglected specific VNF characteristics. In other works, such as [2], the focus was on fault-tolerant placement of stateful VNFs, but they did not address the dynamic mapping of VNFs in a cloud-edge continuum. Similarly, works in [1], [12], and [13] had different objectives and did not involve VNF placement. While some papers like [14] and [15] addressed specific aspects of VNF optimization, they did not provide complete VNF placement algorithms. We also compared our model to SFC-EP [16], which dealt with geo-distributed cloud systems, but our approach is distinct in considering all VNF characteristics for optimal mapping in telesurgery systems.

V. SFC MAPPING PROBLEM

A. PROBLEM DESCRIPTION

In geo-distributed servers, ISPs should make an optimal plan to embed SFC requests. Since VNFs can be flexibly placed at various network locations for various SFC requests, ISPS need to decide on the optimal placement of each request. The traffic of SFC requests should traverse a series of VNF instances, and the SFC mapping can influence the cost of CPU, memory, and links. Therefore, ISPs should focus on placing and chaining VNF instances for SFC requests to reduce costs and ultimately improve their revenue.

B. BLP FORMULATION

The VNF placement problem inherently involves binary decisions, where each VNF may either be placed on a server or not. The binary decision variable framework of BLP aligns well with the nature of VNF placement, making it a suitable choice for capturing these discrete placement choices. Besides, as we need to efficiently allocate resources, the linear programming approach provides an effective means to optimize these allocations. Moreover, BLP has been effectively utilized in diverse network optimization problems, yielding positive outcomes. The track record of BLP's success in related fields bolsters its credibility when applied to VNF placement. Considering these points, here we use BLP to model our problem.

Consider a network G, in which the nodes are represented by the set F, and links by the set E, where $\mathrm{f} \in \mathrm{F}$ and $\mathrm{e} \in \mathrm{E}$. A service request r is denoted as $r = \langle G_\mathrm{r}, \varsigma_\mathrm{r}, d_\mathrm{r} \rangle$, where ς_r is the demanded bandwidth, and $G_\mathrm{r} = (\mathrm{Q_r}, V_\mathrm{r})$ represents the service function graph for request r. $\mathrm{Q_r}$ is the set of requested VNFs, and V_r is the set of virtual edges for request r. To be realistic, we avoid fixed capacities for each type of VNF. Instead, instances of each type are dynamically instantiated based on the user's demand. This approach allows for more flexibility in resource allocation according to specific SFC request requirements.

The objective is to minimize the cost of active and backup paths for each request, including the cost of SFC mapping and state updates.

$$\mathbb{A} = \sum_{\mathrm{r}=1}^{\mathrm{R}} \sum_{\mathrm{q}=1}^{\mathrm{Q_r}} \left(\sum_{\mathrm{v}=1}^{\mathrm{V_r}} \sum_{\mathrm{e}=1}^{\mathrm{E}} \mathrm{B_e b_{r,q} z_{r,q,v,e}} + \sum_{\mathrm{f}=1}^{\mathrm{F}} \frac{(\mathrm{C_f} \pi_{\mathrm{r,q}} + \mathrm{M_f} \Psi_{\mathrm{r,q}}) \mathrm{u_{r,q,f}}}{1 - w_{\mathrm{r,q}} |\varrho_{\mathrm{r,q}} - \zeta_{\mathrm{f}}|} \right) \quad (1)$$

Eq. (1) represents the bandwidth, memory, and CPU costs of the active path in request r. Here, r, q, v, and e represent the requests, VNFs, virtual links, and physical links, respectively. $\mathrm{B_e}$ denotes the bandwidth cost of a unit load (1Gbps) on link $\mathrm{e} \in \mathrm{E}$, and $\mathrm{b_{r,q}}$ shows the bandwidth demand for VNF q in request r, which is equal to the outgoing traffic of VNF q in request r:

$$\mathrm{b_{r,q}} = \mathrm{c_{r,q}} \times \mathrm{a_{r,q}} \quad (2)$$

in which $\mathrm{a_{r,q}}$ is defined as:

$$\mathrm{a_{r,q}} = \begin{cases} \varsigma_\mathrm{r} & \text{if } \mathrm{q} = 1 \\ \mathrm{b_{r,q-1}} & \text{otherwise} \end{cases} \quad (3)$$

Constant bandwidth demand for the function chain in each request was commonly assumed in literature [16], [17], [18], [19], [20]. However, to address the varying incoming and outgoing traffic of VNFs, it is necessary to recompute the outgoing traffic based on the incoming traffic $\mathrm{a_{r,q}}$ and traffic change ratio $\mathrm{c_{r,q}}$. We use the binary variable $\mathrm{z_{r,q,v,e}}$ to indicate whether link $\mathrm{e} \in \mathrm{E}$ is used to host virtual link $v \in V_\mathrm{r}$ and forward traffic of VNF q in request r when embedding the active SFC. Variables $\mathrm{M_f}$ and $\mathrm{C_f}$ are used to calculate the path's memory and CPU costs, respectively. The binary variable $\mathrm{u_{r,q,f}}$ indicates whether node f is used to place an instance of VNF q in request r in the active path. Considering the geo-location dependency of each VNF, we calculate the absolute difference between ζ_f (proximity of node $\mathrm{f} \in \mathrm{F}$ to the edge) and $\varrho_{\mathrm{r,q}}$ (preference level for placing VNF q in request r close to the edge). Smaller absolute differences indicate more suitable nodes for hosting VNF q in request r. For location-independent VNFs, the variable $w_{\mathrm{r,q}}$ nullifies the geo-location dependency.

$$\mathbb{B} = \sum_{\mathrm{r}=1}^{\mathrm{R}} \sum_{\mathrm{q}=1}^{\mathrm{Q_r}} \left(\sum_{\mathrm{v}=1}^{\mathrm{V_r}} \sum_{\mathrm{e}=1}^{\mathrm{E}} \mathrm{B_e b_{r,q}} x_{\mathrm{r,q,v,e}} + \sum_{\mathrm{f}=1}^{\mathrm{F}} \frac{(\mathrm{C_f} \pi_{\mathrm{r,q}} + \mathrm{M_f} \Psi_{\mathrm{r,q}}) \mathrm{o_{r,q,f}}}{1 - w_{\mathrm{r,q}} |\varrho_{\mathrm{r,q}} - \zeta_{\mathrm{f}}|} \right) \quad (4)$$

The backup path's bandwidth, memory, and CPU costs for SFC request r are calculated using Eq. (4). The binary variable $x_{\mathrm{r,q,v,e}}$ indicates link usage for forwarding VNF traffic in the backup SFC. Additionally, the binary variable $\mathrm{o_{r,q,f}}$ shows whether node f is used to place a backup instance of VNF q in request r.

$$\mathbb{C} = \sum_{\mathrm{r}=1}^{\mathrm{R}} \sum_{\mathrm{q}=1}^{\mathrm{Q_r}} \sum_{\mathrm{e}=1}^{\mathrm{E}} \mathrm{B_e k_{r,q}} y_{\mathrm{r,q,e}} \quad (5)$$

Eq. (5) calculates the cost of continuous state updates from each active instance to its corresponding backup instance in request r. The stateful characteristic of VNFs is considered during the SFC mapping process. The variable $\mathrm{k_{r,q}}$ represents the state update rate of VNF q in request r:

$$\mathrm{k_{r,q}} = \rho_{\mathrm{r,q}} \times \mathrm{a_{r,q}} \quad (6)$$

in which $\mathrm{a_{r,q}}$ is determined by Eq. (3). We set $\rho_{\mathrm{r,q}} = 0$ for stateless VNFs to differentiate them from stateful ones. The number of update paths equals the number of stateful VNFs in the active path. There are no CPU and memory costs in the update paths as no VNF placement is needed in these paths.

$$\mathbb{D} = \sum_{f=1}^{F}\left(\phi_f g_f + \sum_{r=1}^{R}\sum_{q=1}^{Q_r}\big(u_{r,q,f} + o_{r,q,f}\big)h_{r,q}(1 - \upsilon_{r,q,f})\right) \tag{7}$$

Eq. (7) represents the licensing and site costs. It includes the site license cost of node $f \in F$, denoted as g_f, and the license cost of VNF q in request r, denoted as $h_{r,q}$. The binary variable ϕ_f indicates whether a request uses node f. Additionally, $\upsilon_{r,q,f}$ is used to avoid redundant license costs by reusing already deployed instances, thereby reducing overall licensing expenses. We minimize the costs incurred by the bandwidth, memory, CPU, and licensing in the active, backup, and update paths of all requests while considering the characteristics of each VNF type.

$$\underset{Z,U,X,O,Y,\Phi,\Upsilon}{\text{minimize}} \quad \mathbb{A} + \mathbb{B} + \mathbb{C} + \mathbb{D} \tag{8}$$

subject to:

$$\sum_{r=1}^{R}\sum_{q=1}^{Q_r} u_{r,q,f}o_{r,q,f} = 0, \forall f \in F \tag{9}$$

Constraint (9) guarantees that if node $f \in F$ is used to place VNF q in request r in an active path for request r, it cannot be used to place the backup instance of the same VNF for the same request.

$$\sum_{r=1}^{R}\sum_{q=1}^{Q_r}\left(k_{r,q}y_{r,q,e} + \sum_{v=1}^{V_r} b_{r,q}\big(z_{r,q,v,e} + x_{r,q,v,e}\big)\right) \leq BW_e, \forall e \in E \tag{10}$$

$$\sum_{r=1}^{R}\sum_{q=1}^{Q_r}\Psi_{r,q}\big(u_{r,q,f} + o_{r,q,f}\big) \leq \mu_f, \forall f \in F \tag{11}$$

$$\sum_{r=1}^{R}\sum_{q=1}^{Q_r}\pi_{r,q}\big(u_{r,q,f} + o_{r,q,f}\big) \leq \xi_f, \forall f \in F \tag{12}$$

Constraints (10), (11), and (12) guarantee that the consumption of bandwidth, memory and CPU cannot exceed the available resources respectively.

$$\sum_{q=1}^{Q_r}\left(\sum_{v=1}^{V_r}\sum_{e=1}^{E}\triangle_e b_{r,q}z_{r,q,v,e} + \sum_{f=1}^{F}u_{r,q,f}\Lambda_{r,q}a_{r,q}\right) + \beth_{r,a} \leq d_r, \forall r \in R \tag{13}$$

$$\sum_{q=1}^{Q_r}\left(\sum_{v=1}^{V_r}\sum_{e=1}^{E}\triangle_e b_{r,q}x_{r,q,v,e} + \sum_{f=1}^{F}o_{r,q,f}\Lambda_{r,q}a_{r,q}\right) + \beth_{r,b} \leq d_r, \forall r \in R \tag{14}$$

Constraints (13) and (14) guarantee respectively that the delay of the active path and backup path respect the maximum tolerated delay of each request, in which $\beth_{r,a}$ and $\beth_{r,b}$ are delay from the source to the first VNF of SFC_r in the active and backup path respectively.

$$\sum_{f=1}^{F}u_{r,q,f} = 1, \forall r \in R, \forall q \in Q_r \tag{15}$$

$$\sum_{f=1}^{F}o_{r,q,f} = 1, \forall r \in R, \forall q \in Q_r \tag{16}$$

Constraints (15) and (16) guarantee the presence of at least one node to host instances of each VNF in the active and backup paths of request r, respectively. For a concise list of symbols and variables, refer to Table 2.

Theorem 1: Characteristic Aware SFC mapping problem is NP-hard.

Proof: Let A be the characteristic aware SFC mapping problem and B be the bin packing problem, proven to be NP-hard [21]. Problem B involves packing items with weights w into bins with capacities c, aiming to minimize cost while not exceeding the bins' maximum capacity. To show the NP-hardness of problem A, we demonstrate that an instance of problem B can be reduced to an instance of problem A.

An instance of problem B can be transformed into an instance of problem A in the following way: i) consider each item in the bin packing problem as a VNF in the SFC placement problem, ii) set the integer weight of each item to be equal to the characteristic-based resource requirement of each VNF, iii) consider the total number of available bins as the total number of substrate nodes, iv) set the capacity of each bin to be equal to the available resource in each substrate node, and v) consider that each item is placed in only one bin, as a VNF is placed in only one of the substrate nodes. In this way, we can reduce problem B to problem A. Considering this, if A is not NP-hard, then B is also not NP-hard (since B is reducible to A), which is a contradiction. Therefore, it can be concluded that A is also an NP-hard problem. ■

VI. CHARACTERISTIC AWARE SFC MAPPING FOR TELESURGERY SYSTEMS (CASMAT)

Our characteristic-aware SFC mapping heuristic algorithm aims to optimize VNF placement based on their specific properties, such as outgoing traffic and state update rate. The algorithm prioritizes user requests with more stringent requirements and ensures that VNFs with higher outgoing traffic are placed closer to succeeding VNFs. Additionally, backup instances are located near active instances to minimize network bandwidth consumption. By considering these aspects, the algorithm provides efficient and effective SFC mapping for telesurgery systems in the cloud-edge continuum.

TABLE 2. Symbols and definitions.

Notation	Definition
Network Inputs	
F	Set of nodes
E	Set of links
G	Infrastructure network graph $\mathrm{G} = (\mathrm{F}, \mathrm{E})$
$\mathrm{BW_e}$	Bandwidth of link $\mathrm{e} \in \mathrm{E}$
μ_f	Memory of node $\mathrm{f} \in \mathrm{F}$
ξ_f	CPU capacity of node $\mathrm{f} \in \mathrm{F}$
$\triangle_\mathrm{e}$	Delay of unit load (1Gb) on link $\mathrm{e} \in \mathrm{E}$
$\mathrm{g_f}$	Site license cost of node $\mathrm{f} \in \mathrm{F}$
$\mathrm{B_e}$	Bandwidth cost of unit load (1Gbps) on link $\mathrm{e} \in \mathrm{E}$
$\mathrm{M_f}$	Memory cost for unit resource (1GB) on node $\mathrm{f} \in \mathrm{F}$
$\mathrm{C_f}$	CPU cost for unit resource (vCPU) on node $\mathrm{f} \in \mathrm{F}$
Service Inputs	
R	Set of all service requests $r \in \mathrm{R}$
$\mathrm{Q_r}$	Set of VNFs in request r
$\mathrm{V_r}$	Set of virtual edges in request r
$\mathrm{G_r}$	Service function graph $\mathrm{G_r} = (\mathrm{Q_r}, \mathrm{V_r})$ for request r
$\Lambda_\mathrm{r,q}$	Delay of VNF q in request r for processing unit load
$\mathrm{W_r}$	Set of location dependencies for VNFs in SFC request r, $\mathrm{w} \in \mathrm{W_r}$, $\mathrm{w} \in \{0, 1\}$
$\mathrm{K_r}$	Set of statefulness characteristics for VNFs in request r, $\mathrm{k} \in \mathrm{K_r}, \mathrm{k} \in \{0, 1\}$
Θ_r	Set of preference levels of choosing a node close to the edge for request r, $\varrho \in \Theta_\mathrm{r}, 0 \leq \varrho \leq 1$
$\mathrm{c_{r,q}}$	Traffic change ratio of VNF q in request r
$\mathrm{w_{r,q}}$	Location dependency of VNF q in request r, 1 if VNF q is location dependent, and 0 otherwise
$\rho_\mathrm{r,q}$	State update ratio of VNF q in request r
$\mathrm{h_{r,q}}$	License cost of VNF q in request r
$\mathrm{Q_r}$	Set of VNFs requested by SFC request r, $\mathrm{q} \in \mathrm{Q_r}$
ς_r	Bandwidth demand in request r
Ψ_r	Set of memory demand in SFC request r, $\psi \in \Psi_\mathrm{r}$
Π_r	Set of CPU demand in SFC request r, $\pi \in \Pi_\mathrm{r}$
$\mathrm{d_r}$	The maximum tolerable delay in SFC request r
Variables	
$\mathrm{u_{r,q,f}}$	1 if an instance of VNF q in request r is placed on node $\mathrm{f} \in \mathrm{F}$ on the active path, 0 otherwise
$\mathrm{o_{r,q,f}}$	1 if an instance of VNF q in request r is placed on node $\mathrm{f} \in \mathrm{F}$ on the backup path, 0 otherwise
$\mathrm{z_{r,q,v,e}}$	1 if link $\mathrm{e} \in \mathrm{E}$ hosts virtual link $\mathrm{v} \in \mathrm{V_r}$ for forwarding the traffic of VNF q in request r to the next VNF when embedding the Active SFC for request r, 0 otherwise
$\mathrm{x_{r,q,v,e}}$	1 if link $\mathrm{e} \in \mathrm{E}$ is used to host the virtual link $\mathrm{v} \in \mathrm{V_r}$ for forwarding the traffic of VNF q in request r to the next VNF when embedding the Backup SFC for request r, 0 otherwise
$\mathrm{y_{r,q,e}}$	1 if link $\mathrm{e} \in \mathrm{E}$ is used in the update path of VNF q in request r, 0 otherwise
$\upsilon_\mathrm{r,q,f}$	1 if an instance with the same type of requested VNF q in request i is already deployed on f
ϕ_f	1 if node f is used by a request, 0 otherwise
Other symbols	
ζ_f	Proximity level of the node $\mathrm{f} \in \mathrm{F}$ to the edge, $0 \leq \zeta_\mathrm{f} \leq 1$
$\varrho_\mathrm{r,q}$	Preference level for placing VNF q in request r close to the edge, $0 \leq \varrho_\mathrm{r,q} \leq 1$
$\mathrm{a_{r,q}}$	Incoming traffic to VNF q in request r
$\mathrm{b_{r,q}}$	Bandwidth demand for VNF q in request r
$\mathrm{k_{r,q}}$	State update rate of VNF q in request r
Φ	A vector that holds the value of ϕ_f for all nodes, $\phi_\mathrm{f} \in \Phi$
$\gimel_\mathrm{r,q,f}$	Characteristic-aware rank value of node f for location-dependent VNF q in request r
$\gimel'_\mathrm{r,q,f}$	Characteristic-aware rank value of node f for location-independent VNF q in request r

In the next step, we compute κ_f as the general rank of each node according to the Eq. (17).

$$\kappa_\mathrm{f} = \alpha \times \tau_\mathrm{f} + \beta \times \mathrm{m_f} + (1 - \alpha - \beta) \times \sum_{\mathrm{e}=1}^{\mathrm{E_f}} \chi_\mathrm{e} \tag{17}$$

where τ_f and $\mathrm{m_f}$ denote the remaining CPU and memory of node f respectively. Besides, χ_e represents the remaining bandwidth capacity of edge e. The set $\mathrm{E_f}$ denotes the set of all connected links to the node f. The weights α and β are used to bias node selection by focusing on either node resources or aggregate bandwidth resources. Indeed, considering a situation where most nodes have high capacities, the placement should focus more on the quality of outgoing links towards other nodes and vice-versa. Hence, these parameters can be considered for intensification or diversification during the search process for the most suitable candidate node.

For each request, we calculate the preferred place of each VNF as:

$$\mho_\mathrm{r,q} = \sum_{\mathrm{a}=1}^{\mathrm{q}} \eth_\mathrm{r,a} \times \theta_\mathrm{r} \tag{18}$$

in which, θ_r is the geographical distance between the patient side and surgeon side in request r, and $\eth_\mathrm{r,q}$ is calculated as:

$$\eth_\mathrm{r,q} = \frac{\aleph_\mathrm{r,q}}{\sum_{\mathrm{z}=1}^{\mathrm{l}+1} \aleph_\mathrm{r,z}} \tag{19}$$

where:

$$\aleph_\mathrm{r,q} = \frac{\sum_{\mathrm{x}=1}^{\mathrm{l}+1} \mathrm{a_{r,x}}}{\mathrm{a_{r,q}}} \tag{20}$$

In this equation, $\mathrm{a_{r,q}}$ is calculated according to Eq. (3). Here, *l+1* refers to the node after the last VNF, which is naturally the destination node.

In the next step, for each VNF, we search for candidate nodes according to $\mho_\mathrm{r,q}$ and place the VNF on the candidate node with the highest characteristic-aware rank value. In the case of dealing with a location-dependent VNF, we calculate the characteristic-aware rank of each node as follows by taking edge proximity preference and license cost into account.

$$\gimel_\mathrm{r,q,f} = \frac{\gamma \kappa_\mathrm{f} + (1 - \gamma) \upsilon_\mathrm{r,q,f}}{|\varrho_\mathrm{r,q} - \zeta_\mathrm{f}|} \tag{21}$$

In the case of a location-independent VNF, the characteristic-aware rank is simply calculated as:

$$\gimel'_\mathrm{r,q,f} = \gamma \kappa_\mathrm{f} + (1 - \gamma) \upsilon_\mathrm{r,q,f} \tag{22}$$

Afterward, we place backup instances for each active instance, considering the proximity between them, which is more beneficial for stateful VNFs with higher state update rates. For VNFs with higher update rates, we search within a radius of $\frac{\wp}{\mathrm{k_{r,q}}}$, while for those with lower rates, we use a radius of $\wp$. The k-shortest path with the lowest delay is chosen between the backup instances of VNF q and VNF *q-1* within the same request. If the total delay does not meet the constraint, the placement process is repeated with increased freedom based on $\mho_\mathrm{r,q}$. If no placement meets the delay constraint, the request is rejected.

Let's break down the cost components, considering both capital expenditure (CapEx) and operational expenditure (OpEx) aspects:

- Active Path Cost (Eq. (1)):
  - Bandwidth Cost:
    - CapEx: $\mathrm{B_e}$ represents the bandwidth cost of a unit load (1 Gbps) on link $\mathrm{e} \in \mathrm{E}$. This is a one-time capital cost.
    - OpEx: $\mathrm{b_{r,q}}$ is the bandwidth demand for VNF q in request r, determined by $\mathrm{c_{r,q}}$ (traffic change ratio) and $\mathrm{a_{r,q}}$ (incoming traffic). The bandwidth cost is incurred continuously during the operation.
  - CPU and Memory Costs:
    - CapEx: $\mathrm{C_f}$ and $\mathrm{M_f}$ are constants representing CPU and memory costs, respectively, for node $\mathrm{f} \in \mathrm{F}$. These are one-time capital costs.
    - OpEx: The costs are related to the usage of nodes for placing VNF instances in the active path ($\mathrm{u_{r,q,f}}$). The proximity of nodes to the edge (ζ_f) is considered, impacting the OpEx.
- Backup Path Cost (Eq. (4)):
  - Similar breakdown as Active Path Cost: Bandwidth, CPU, and Memory costs, with similar considerations for OpEx and CapEx.
- Update Path Cost (Eq. (5)):
  - Bandwidth Cost: Similar to the Active Path, but specifically for continuous state updates from active to backup instances.
- License Cost (Eq. (7)):
  - CapEx: $\mathrm{g_f}$ represents the site license cost for node $\mathrm{f} \in \mathrm{F}$, and $\mathrm{h_{r,q}}$ represents the license cost of VNF q in request r. These are one-time capital costs.
  - OpEx: Incurred for the usage of nodes and VNF instances, with considerations for license reuse ($\upsilon_\mathrm{r,q,f}$) to reduce overall licensing expenses.

It is worth mentioning that CapEx is primarily associated with constants like bandwidth cost, CPU and memory costs, and licensing costs, while OpEx is mainly driven by the continuous usage of resources, including bandwidth, CPU, memory, and licensing, as well as the state update costs.

Algorithm 1 encapsulates the essence of the SFC mapping procedure.

Algorithm 1: CASMaT

```
Data: $G$: network graph, $R$: user requests, $\Im$: search
      limit
Result: VNF mapping
Sort $R$ based on the aggregated requirements
for $f \in F$ do
    Compute $\kappa_f$ using Eq. (17)
for $r \in R$ do
    for $q \in Q_r$ do
        Compute $\mho_\mathrm{r,q}$ using Eq. (18)
        if $w_\mathrm{r,q} = 1$ then
            calculate $\daleth_\mathrm{r,q,f}$ based on Eq. (21)
        else
            calculate $\daleth'_\mathrm{r,q,f}$ based on Eq. (22)
        Sort candidate nodes in radius $\wp$ of $\mho_\mathrm{r,q}$, based
        on the characteristic-aware rank Place active
        instance of $VNF_\mathrm{r,q}$ on the node with the highest
        rank
        if $q \neq l$ then
            find k-shortest paths between active
            instances of $VNF_\mathrm{r,q}$ and $VNF_\mathrm{r,q-1}$
            select path with the lowest delay
        Sort candidate nodes in radius $\frac{\wp}{\mathrm{k_{r,q}}}$ of $q_\mathrm{r,q}$,
        based on the characteristic-aware rank
        Place backup instance of $q_\mathrm{r,q}$ on the node with
        the highest rank
if constraints respected then
    restore $\wp$ to the default value
    Confirm removal of the nodes and links capacities
    Recompute $\kappa_f$ for the hosting nodes
else
    if $stop \leq \Im$ then
        Decrease $\wp$
        Remove currently hosting nodes from the list of
        candidate nodes
        Redo the for loop in 1
    else
        Reject the request
```

VII. COMPUTATIONAL COMPLEXITY

To analyze the complexity of the algorithm, we need to examine the key operations and loops. For the initialization section, there is no significant complexity. Sorting the user requests based on aggregated requirements has a time complexity of *O(R log R)*, where *R* is the number of user requests. To iterate over requests, we have an outer loop and an inner loop. the computational complexity for iterating over the user requests is $O(R)$. Sorting the candidate nodes based on the characteristic-aware rank has a time complexity of $O(\mathrm{C_{max}} log \mathrm{C_{max}})$. Placing an active instance has the complexity of $O(1)$. Finding k-shortest path has the complexity of $O(K_\mathrm{r,q}.(|E| + |V|Log|V|))$. Sorting the candidate nodes for the backup instance has a computational complexity of $O(\mathrm{C_{max}} log \mathrm{C_{max}})$, and placing a backup instance has the complexity of $O(1)$. Constraint checking and parameter adjusting have the complexity of $O(1)$. If we integrate these complexities into the overall complexity analysis, the complexity of the algorithm can be approximated as:

$$O(R \log R + F \cdot \mathrm{E_f} + R \cdot (q + \mathrm{C_{max}} \log \mathrm{C_{max}} + \mathrm{k_{r,q}} \cdot (|E| + |V| \log |V|)))$$

Therefore, in the worst case, the total computational complexity is polynomial in terms of input parameters.

VIII. COMPLIANCE WITH EXISTING TELESURGERY PLATFORMS

Several prominent telesurgery platforms, such as the da Vinci Surgical System [22], Telelap ALF-X [23], and Raven II [24], exhibit a lack of presumptions concerning the technical implementation of the requisite networking platform. These platforms prioritize the fulfillment of reliability, latency, and bandwidth requirements, showcasing an agnostic stance toward the underlying networking technology. As for networking platforms, the prevalent adoption of 5G technology presents another noteworthy networking platform. Given its complete compliance with Network Function Virtualization (NFV), 5G emerges as a key technological enabler for the realization of advanced networks [25]. Additionally, the Anvari Telesurgery System has successfully linked a rural and an urban hospital in Canada by leveraging commercially available networks, such as Virtual Private Networks (VPN), specifically tailored for telesurgery applications [26]. Its utilization of standard networking protocols and infrastructures positions it as amenable to integration with a Virtual Network Function (VNF)-based platform.

IX. PERFORMANCE EVALUATION

This section presents the results of our performance evaluation for CASMaT, our proposed SFC mapping method in the Cloud-Edge Continuum. As CASMaT is the first characteristic-aware SFC mapping research, there are no existing state-of-the-art algorithms to directly compare it with. The only related paper, [11], also lacks consideration of VNF-specific characteristics. However, we compared CASMaT with [16], which does not consider the specific characteristics of each VNF type but focuses on the utilized and remaining network resources for VNF placement. This comparison allowed us to assess the effectiveness of our proposed algorithm.

A. EXPERIMENTAL SETUP

We used a customized version of "topology-generator" [27] to create various network typologies for evaluating our proposed SFC mapping method. The base scenario includes 20 servers randomly connected to routers through 2 to 4 links each. We considered 2 end-users with different requirements and locations. The delay threshold for each request was set to 100 ms, meeting the ideal latency requirement for telesurgery. To simulate the varying network conditions, the available bandwidth for each link varies between 1Gbps and 10Gbps during the simulation. Details of all simulation settings are shown in Table 1.

B. PERFORMANCE METRICS

To evaluate the effectiveness of the proposed algorithm, the following performance metrics are considered:

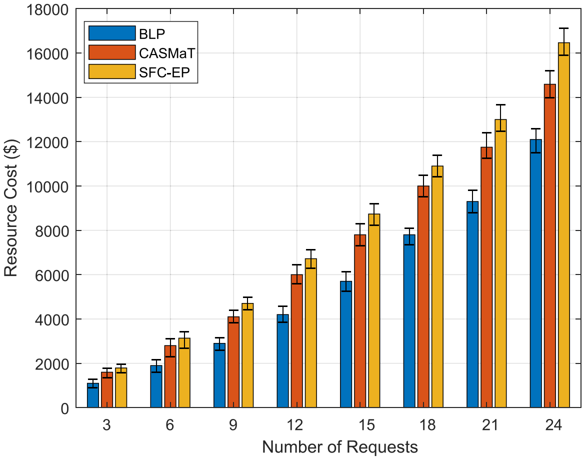


FIGURE 3. **Average Total Cost**

1) *Total cost (Dollar):* The sum of all costs, including operational cost, communication cost, and licensing cost. The reduced total cost is a measure of the cost-efficiency of a VNF placement algorithm.
2) *Latency:* The average duration for fulfilling a request, which includes both computation and communication latency between the two end-users.
3) *Latency Gap:* The gap between the average requested latency and the average achieved latency. A reduced gap indicates the capability of the method to avoid over-provisioning.
4) *Proximity Conformity:* The gap between the requested conformity level and the achieved conformity, and is indicative of how successful the placement method has been in respecting the preferred conformity level of the VNFs.
5) *Number of Used Nodes:* The total number of nodes utilized to map a requested SFC. this includes the nodes for both Active and Backup paths.
6) *Resource Utilization:* It is defined as the total amount of the utilized resources on a server to serve a request. As the resources on each server encompass CPU and memory resources, the resource utilization results are depicted in form of two figures.

C. RESULTS AND DISCUSSION

Performance metrics are used to compare the simulation results of our method with that of SFC-EP. In each of the figures, the BLP shows the result of the presented Binary Linear Programming, and CASMaT shows the result of the presented heuristic algorithm.

In Fig. 3, our proposed method improves the total cost compared to SFC-EP. However, BLP outperforms CASMaT since it considers all possible placements for each VNF, while CASMaT focuses on characteristic-aware locations. The cost variations of each method are displayed on top of the bars.

Fig. 4 shows the average delay for each method as a function of the number of requests, ranging from 3 to 24.

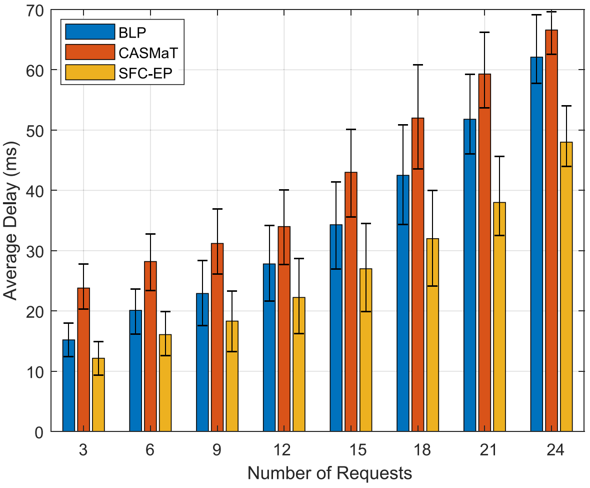


FIGURE 4. Average Latency.

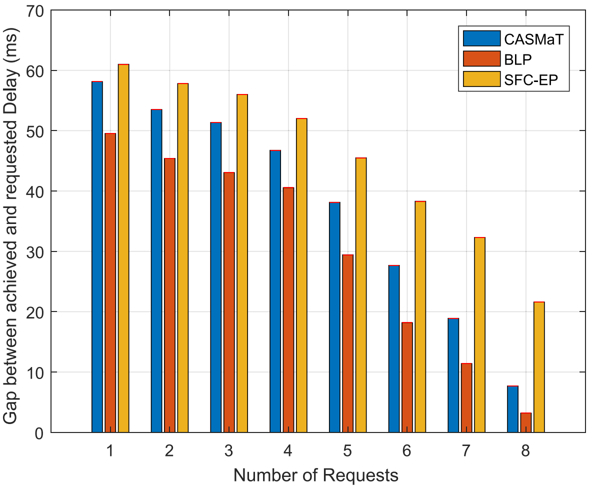


FIGURE 5. Average Latency Gap.

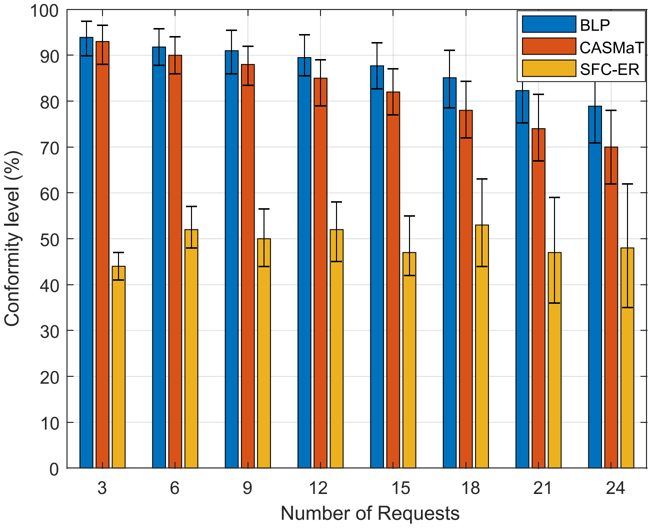


FIGURE 6. Average Conformity Level.

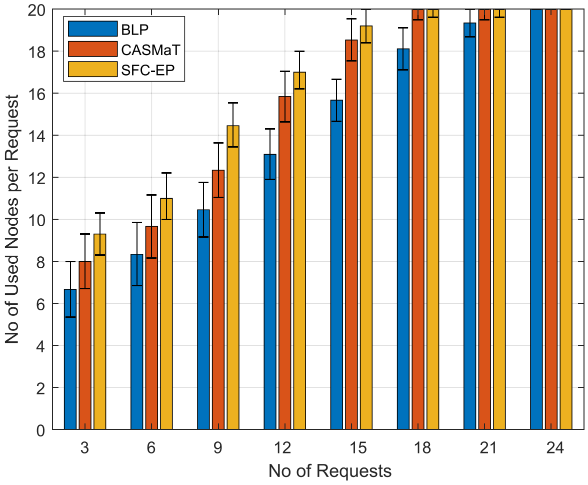


FIGURE 7. Average No. of Used Nodes

CASMaT displays a higher latency than the other methods, but this delay is not an issue for the service provider as long as the services are provided within the user's requested delay constraint. It is important to note that the placement of VNFs is from a service provider's perspective, and the SFCs' delay is not reduced at the expense of consuming more network resources.

Fig. 5 shows the delay gap between requested and achieved latency. A higher gap indicates better performance, ensuring services are provided with lower latency, even if it consumes more network resources. However, we aim to avoid significant gaps, maintaining a balance between meeting user requirements and optimizing resource utilization for cost efficiency.

In Fig. 6 we have compared how conformant each method is to the required edge conformity level, according to the conformity attribute of each VNF. As we can see in this figure, SFC-EP has no sense of conformity, hence the average conformity level is always around 50%, whereas in CASMaT the conformity varies between 93% and 70%, and it always outperforms that of SFC-EP.

In Fig. 7, we compare the number of nodes used by each method. Each request requires two SFCs with four VNFs each, totaling eight VNF placements per request. However, multiple requests can share the same servers for their VNFs. CASMaT uses fewer nodes compared to SFC-EP, although it doesn't perform as well as BLP. This reduces resource usage and improves cost efficiency.

In Figs. 8 and 9, resource utilization is the same across all three methods. Each method allocates only the necessary resources that VNFs fully utilize. Any additional resources required when sharing VNFs are added at that time. Thus, the three methods use the same amount of resources both in theory and practice.

To evaluate the scalability of the proposed method and also compare it with a state-of-the-art algorithm, we scaled the number of the requests in the simulation to 200 requests and compared the results with SFC-EP and also the SFC

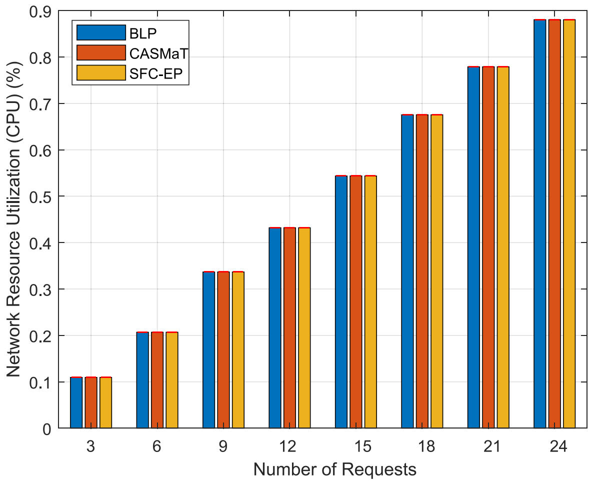


FIGURE 8. Average CPU Utilization

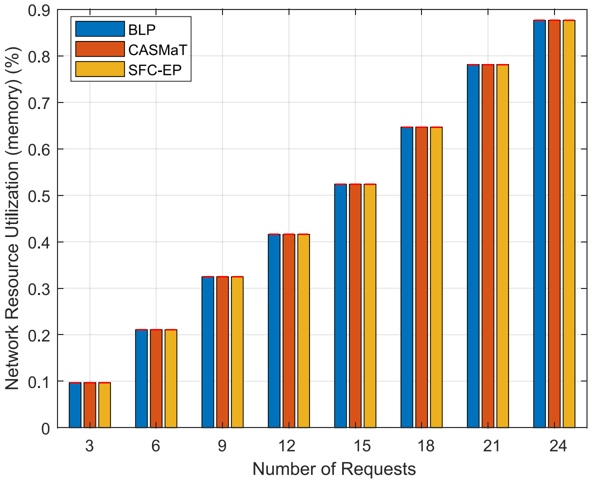


FIGURE 9. Average Memory Utilization

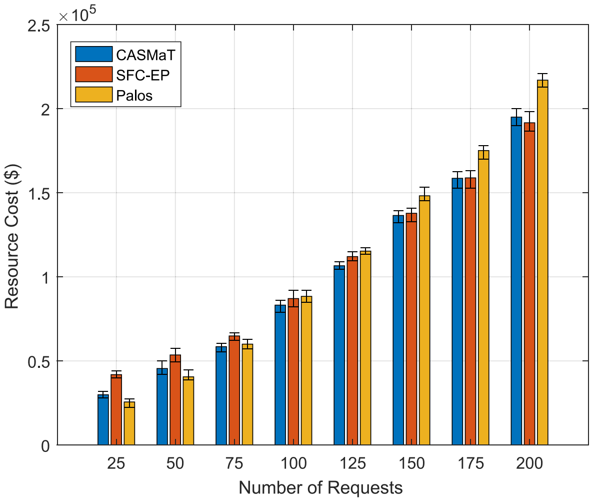


FIGURE 10. Average Total Cost with Scaled Number of Requests.

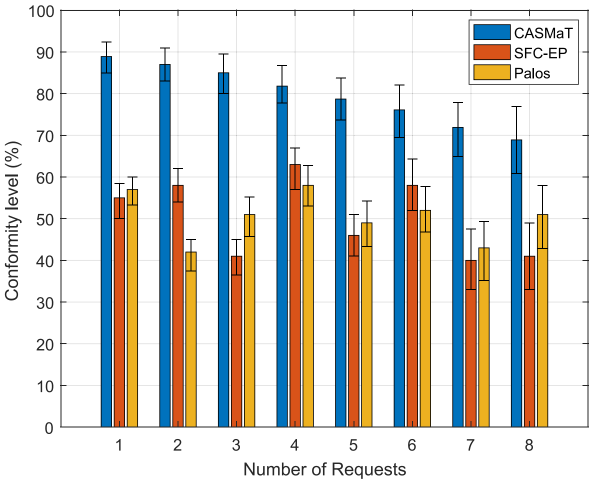


FIGURE 11. Average Conformity Level with Scaled Number of Requests.

deployment algorithm presented in [28]. As Fig. 10 shows, CASMaT outperforms SFC-EP in most cases, but in a high number of requests, SFC-EP is too close to CASMaT, and even in some points better than it. However, Palos does not perform as well as CASMaT as the number of requests scales.

We have also evaluated the conformity level of these three algorithms and presented the results in Fig. 11. The SFC-EP and Palos placement algorithms lack consideration for the specific requirements necessitating the proximity of certain Virtual Network Functions (VNFs) to the network edge. This absence of accommodation for edge-centric placement criteria is evident in the evaluation results, where an apparent randomness in conformity levels is observed.

X. CONCLUSION

This paper addresses characteristic-aware SFC mapping for telesurgery scenarios in the cloud-edge continuum. The goal is to optimize VNF placement cost while meeting the service delay threshold. A BLP is formulated to tackle the NP-hard problem, and a heuristic algorithm, CASMaT, is proposed.

CASMaT is evaluated against SFC-RP and BLP, showing superior performance in total cost, proximity conformity, number of used nodes, and avoiding excessive network resource utilization for lower latency. Besides, it has also been compared with Palos in a scaled network and exhibited a lower growth rate in the total cost. As the future work, it is recommended to concentrate on the failure recovery of the proposed method.

REFERENCES

[1] B. Yang et al., “Algorithms for fault-tolerant placement of stateful virtualized network functions,” in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–7.

[2] G. Yuan et al., “Fault tolerant placement of stateful VNFs and dynamic fault recovery in cloud networks,” *Comput. Netw.*, vol. 166, Jan. 2020, Art. no. 106953.

[3] R. Munoz et al., “Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks,” *J. Opt. Commun. Netw.*, vol. 7, no. 11, pp. B62–B70, 2015.

[4] Z. Xu, W. Liang, A. Galis, and Y. Ma, “Throughput maximization and resource optimization in NFV-enabled networks,” in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–7.

[5] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Tactile-Internet-based telesurgery system for healthcare 4.0: An architecture, research challenges, and future directions," *IEEE Netw.*, vol. 33, no. 6, pp. 22–29, Nov./Dec. 2019.

[6] V. Börner, D. Rabin, B. Benjamin, M. Dolores, B. Christiane, and H. Fuchs, "5G mobile communication applications for surgery: An overview of the latest literature," *Artif. Intell. Gastrointest. Endosc.*, vol. 2, no. 1, pp. 1–11, 2021.

[7] Y. Chen and J. Wu, "Latency-efficient VNF deployment and path routing for reliable service chain," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 651–661, Jan.–Mar. 2021.

[8] "Network functions virtualisation (NFV); Resiliency requirements," ETSI, Sophia Antipolis, France, documnet ETSI GS NFV-REL 001, 2015. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-REL/001_099/001/01.01.01_60/gs_nfv-rel001v010101p.pdf

[9] S. Xu, M. Perez, K. Yang, C. Perrenot, J. Felblinger, and J. Hubert, "Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dV-Trainer(®) simulator," *Surg. Endosc.*, vol. 28, no. 9, pp. 2569–2576, Sep. 2014.

[10] R. Rayman et al., "Effects of latency on telesurgery: An experimental study," *Med. Image Comput. Comput. Assist. Interv.*, vol. 8, no. 2, pp. 57–64, 2005.

[11] S. I. Kim and H. S. Kim, "A VNF placement method based on VNF characteristics," in *Proc. Int. Conf. Inf. Netw.*, 2021, pp. 864–869.

[12] W. Mao, L. Wang, J. Zhao, and Y. Xu, "Online fault-tolerant VNF chain placement: A deep reinforcement learning approach," in *Proc. IFIP Netw. Conf.*, 2020, pp. 163–171.

[13] D. Zeng, L. Gu, Y. Chen, S. Pan, and Z. Qian, "Cost efficient state-aware function placement and flow scheduling for NFV networks," in *Proc. IEEE SmartWorld*, 2018, pp. 1352–1357.

[14] Z. Alomari, M. F. Zhani, M. Aloqaily, and O. Bouachir, "On minimizing synchronization cost in NFV-based environments," in *Proc. Int. Conf. Netw. Service Manag.*, 2020, pp. 1–9.

[15] K. A. Noghani, A. Kassler, and J. Taheri, "On the cost-optimality trade-off for service function chain reconfiguration," in *Proc. Int. Conf. Cloud Netw.*, 2019, pp. 1–6.

[16] J. Pei, P. Hong, K. Xue, and D. Li, "Efficiently embedding service function chains with dynamic virtual network function placement in geo-distributed cloud system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 10, pp. 2179–2192, Oct. 2019.

[17] L. Liu, S. Guo, G. Liu, and Y. Yang, "Joint dynamical VNF placement and SFC routing in NFV-enabled SDNs," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4263–4276, Dec. 2021.

[18] L. Qu, C. Assi, M. J. Khabbaz, and Y. Ye, "Reliability-aware service function chaining with function decomposition and multipath routing," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 835–848, Jun. 2020.

[19] M. Chen, Y. Sun, H. Hu, L. Tang, and B. Fan, "Energy-saving and resource-efficient algorithm for virtual network function placement with network scaling," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 1, pp. 29–40, Mar. 2021.

[20] Y. Liu, J. Pei, P. Hong, and D. Li, "Cost-efficient virtual network function placement and traffic steering," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman, 1990.

[22] F. Cepolina and R. P. Razzoli, "An introductory review of robotically assisted surgical systems," *Int. J. Med. Robot.*, vol. 18, no. 4, p. e2409, Aug. 2022.

[23] E. Altobelli et al., "1405 Telelap Alf-X: A novel telesurgical system for the 21st century," *J. Urol.*, vol. 189, no. 4S, pp. e575–e576, Apr. 2013.

[24] B. Kehoe et al., "Autonomous multilateral debridement with the raven surgical robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 1432–1439.

[25] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN—Key technology enablers for 5G networks," 2018, *arXiv:1806.07316*.

[26] C. Cazac and G. Radu, "Telesurgery—An efficient interdisciplinary approach used to improve the health care system," *J. Med. Life*, vol. 7, no. 3, pp. 137–141, 2014.

[27] C. Ghali. "Random topology generator." github. 2018. [Online]. Available: https://github.com/cesarghali/topology-generator

[28] W. Liang, C. Li, L. Cui, and F. P. Tso, "Position-aware packet loss optimization on service function chain placement," *Digit. Commun. Netw.*, to be published.

SEYEDREZA TAGHIZADEH (Member, IEEE) received the bachelor's degree in computer engineering and the master's and Ph.D. degrees in information technology engineering. While focusing on IoT and similar concepts during his Ph.D. studies with the University of Tehran, he pursued his sabbatical research with UQAM University, carrying out research into IoT. He has been a Project Manager for developing security apps and IoT platforms while leading the Cloud Computing Team, Research Center of Iran's Ministry of Information and Communications Technology. He has been engaging actively as a leading member in various industrial IoT and cybersecurity projects, and working with global IoT-leader industrial companies. He is currently a Postdoctoral Fellow with UQAM University while holding the position of Assistant Professor with Shiraz University. His research interests are primarily in the areas of the Internet of Things, artificial intelligence, machine learning, cybersecurity, and cloud–edge computing.

HALIMA ELBIAZE (Senior Member, IEEE) received the B.S. degree in applied mathematics from the University of MV, Morocco, in 1996, the M.Sc. degree in telecommunication systems from Université de Versailles in 1998, and the Ph.D. degree in computer science from Télécom Sud-Paris, France, in 2002. Since 2003, she has been with the Department of Computer Science, Université du Québec Montréal, QC, Canada, where she is currently a Full Professor. She is the Head of TRIME, a Research Laboratory in Telecommunications, Networks, Mobile and Embedded Computing. She had been awarded many research grants from both public agencies and industry. Her research interests include performance evaluation, traffic engineering, cloud computing, and next generation IP networks.

ROCH H. GLITHO (Senior Member, IEEE) received the M.Sc. degree in business economics from the University of Grenoble, Grenoble, France, the first M.Sc. degree in pure mathematics and the second M.Sc. degree in computer science from the University of Geneva, Switzerland, and the Ph.D. degree in informatics from the Royal Institute of Technology, Stockholm, Sweden. He is currently a Full Professor with Concordia University, Montreal, QC, Canada, where he holds a Canada Research Chair. He also holds the Ericsson/ENCQOR-5G Senior Industrial Research Chair in cloud and edge computing for 5G and beyond. He was with industry and has held several senior technical positions with Ericsson, Sweden, and Canada. He was the Editor-in-Chief of the *IEEE Communications Magazine* and IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He was also an IEEE Distinguished Lecturer.

WESSAM AJIB (Senior Member, IEEE) received the Engineer Diploma degree in physical instruments from INPG, Grenoble, France, in 1996, and the master's and Ph.D. degrees in computer networks from the École Nationale Supérieure des Tèlècommunication, Paris, in 1997 and 2000, respectively. He had been an Architect and a Radio Network Designer with Nortel Networks, Ottawa, ON, Canada, from October 2000 to June 2004. He had conducted many outbreaking projects on the third generation of wireless cellular networks. He followed a Postdoctoral Fellowship with the École Polytechnique de Montréal, Montreal, QC, Canada, from 2004 to 2005. Since June 2005, he has been with the Department of Computer Sciences, Universite du Quebec, Montreal, where he is currently a Full Professor. His research interests include wireless communications and networks, multiple access design, traffic scheduling, machine learning algorithms for wireless networks, and resource allocation in 5G and 6G.