Remote Robotic Surgery: Joint Placement and Scheduling of VNF-FGs

Amina Hentati†, Amin Ebrahimzadeh†, Roch H. Glitho†‡, *Senior Member, IEEE*
Fatna Belqasmi*, and Rabeb Mizouni**
†CIISE, Concordia University, Montréal, QC, Canada
‡University of Western Cape, Bellville, South Africa
* Zayed University, Dubai, United Arab Emirates
** Khalifa University, Abu Dhabi, United Arab Emirates

***Abstract*—Remote robotic surgery is one of the most interesting Tactile Internet (TI) applications. It has a huge potential to deliver healthcare services to remote locations. Moreover, it provides better precision and accuracy to diagnose and operate on patients. Remote robotic surgery requires ultra-low latency and ultra-high reliability. The aforementioned stringent requirements do not apply for all the multimodal data traffic (i.e., audio, video, and haptic) triggered during a surgery session. Hence, customizing resource allocation policies according to the different quality-of-service (QoS) requirements is crucial in order to achieve a cost-effective deployment of such system. In this paper, we focus on resource allocation in a softwarized 5G-enabled TI remote robotic surgery system through the use of Network Functions Virtualization (NFV). Specifically, this work is devoted to the joint placement and scheduling of application components in an NFV-based remote robotic surgery system, while considering haptic and video data. The problem is formulated as an integer linear program (ILP). Due to its complexity, we propose a greedy algorithm to solve the developed ILP in a computationally efficient manner. The simulation results show that our proposed algorithm is close to optimal and outperforms the benchmark solutions in terms of cost and admission rate. Furthermore, our results demonstrate that splitting application traffic to multiple VNF-forwarding graphs (VNF-FGs) with different QoS requirements achieves a significant gain in terms of cost and admission rate compared to modeling the whole application traffic with one VNF-FG having the most stringent requirements.**

***Index Terms*—Tactile Internet, Remote Robotic Surgery, Network Function Virtualization, Latency, Virtual Network Function, VNF Forwarding Graph, Joint Placement and Scheduling.**

I. Introduction

Nowadays, more clinical operations are being performed via teleoperation using surgical robots such as the da Vinci surgical robot. In 5G-enabled Tactile Internet era, remote control of the real and/or virtual objects via haptic communications in addition to the conventional audio, video, and text is enabled [1]. This could reshape our society and drive significant changes in our lives [2].

Remote robotic surgery application is a human-machine interaction, where the human is the surgeon and the machine is the surgical robot. Contrary to in-person surgery and conventional robotic surgery such as Da Vinci Surgical system, a surgeon performs the surgery at a remotely located patient site via the transmission of commands and feedbacks through a communication network [3]. Remote robotic surgery offers a wide range of benefits including increased dexterity, filtering of hand tremors, and high-quality 3D visualization [4]. It also incurs less blood loss and pain, and hence, can result in reduced trauma and shorter recuperation time [5].

In the legacy networks, dependency on network hardware–also known as middleboxes–is one of the important obstacles to provide cost-effective services [6]. To cope with this, Network Function Virtualization (NFV) have emerged. In NFV networks, application components can be implemented as Virtualized Network Functions (VNFs). To provide a service, VNFs are chained in a specific order to form a structured graph, which is commonly referred to as VNF-Forwarding Graph (VNF-FG) [7]. In order to fulfill the strict delay and reliability requirements of Tactile Internet applications, how to choose the proper placement of VNFs and how to schedule them to realize the given service are two important problems, especially for latency-sensitive applications [6]. The VNF placement problem consists of placing the VNFs into NFV-enabled nodes and map the virtual links between them onto substrate links. On the other hand, the scheduling problem focuses on determining the execution schemes of the VNFs required to process the traffic of a given VNF-FG.

Recently, many studies have been carried out to propose efficient placement and scheduling algorithms that affect both quality of service (QoS) and provider cost. Specifically, VNF placement and scheduling have a significant impact on delay [6]. Therefore, for delay-sensitive applications, it is necessary to consider these problems jointly. In reality, applications involve different types of traffic with different QoS requirements. Remote robotic surgery is one interesting example that involves the exchange of haptic data in addition to the multimedia data, which have different QoS requirements. Hence, it is important to consider the coexistence of different types of data traffic and customize the placement and scheduling of VNFs according to their different QoS requirements in order to fulfill cost-effective deployment. For this purpose, this paper focuses on joint placement and scheduling of VNFs in NFV-based remote robotic surgery use-case, where both haptic and video data are considered.

Video feedbacks helps the surgeon track the surgery with

978-3-903176-51-5

a high-quality resolution in real time. The designed solution addresses latency, reliability, and throughput aspects over the aforementioned use-case. To the best of our knowledge, joint VNF placement and scheduling, that focus on both haptic and video data, has not been considered in the literature.

The remainder of this paper is organized as follows. Section II presents the illustrative use-case, requirements and the related work. Section III describes the considered system model while Section IV formulates the problem. The proposed solution is described in Section V. In Section VI, numerical results are presented. Finally, Section VII concludes the paper.

II. Use-case description, Requirements, and Related Work

A. Use-case Description

In this work, we focus on a one-to-one remote robotic surgery use-case. Contrary to in-person surgery, in remote robotic surgery the doctor performs a surgery on a patient while being away from the operating table and even at the other end of the world. In the following, we focus on the traffic exchanged during the remote robotic surgery process, which is mainly based on video and haptic data. The other types of traffic (i.e., audio and data) are kept for future work. Once generated, the video and haptic data, each modeled by a separate VNF-FG, are transmitted through basic VNFs, before reaching the destination.

B. Requirements

Remote robotic surgery raises several challenges. First, the infrastructure should allow the exchange of a unidirectional visual data and bidirectional haptic data. Moreover, in accordance with the Tactile Internet requirements, the round trip latency for the haptic data should be in the order of a few milliseconds [5]. This is mainly because a remote robotic surgery procedure requires an ultra-responsive connectivity to make it feasible to exchange real-time sensation data. This also avoids the cyber-sickness phenomenon, which occurs when multiple senses are involved in the same interaction, but there is a mis-synchronization between the feedback of different senses. Moreover, the system should be reliable since data loss results in inconsistency between the surgeon and patient. The system requirements should also be customized according to the traffic type. For instance, the latency requirement set for the haptic data is smaller than that of the video traffic [8] in order to have a cost-efficient component placement. Finally, the cost of deploying services and delivering them to end users must be minimized for cost-effectiveness reasons.

C. Related Work

In this subsection, we review the relevant literature on the VNF-FG placement and scheduling. First, we review the existing solutions to date on joint VNF placement and scheduling. Then, we review the existing solutions to date on VNF placement for video data.

To the best of our knowledge, our work is the only one that studies the placement and scheduling of application components as VNFs in an NFV-based system while considering both haptic and video data.

1) Joint VNF Placement and Scheduling: VNF placement and scheduling were investigated jointly in [9] by considering the buffer capacity of nodes and the processing time of VNFs. Joint optimization of three stages of NFV resource allocation (i.e., VNF-FG composition, placement, and scheduling) was studied in [10] by considering network cost and service performance. In [11], radio and NFV resource allocation were jointly considered. The objective was to minimize the total cost function, while guaranteeing E2E delay of each connection. In [12], an analytical model was presented to evaluate E2E packet delay for multiple traffic flows traversing a common embedded VNF chain. Although the above mentioned studies focus on the joint placement and scheduling and some of them consider cost and latency requirements, none of them tackles the aforementioned problem while considering different latency and reliability requirements for the different types of traffic flows triggered by the application.

2) VNF Placement for Video Data: In [13], the authors designed a VNF to monitor the QoE at the client machine for online video service in the network. In [2], a 5G network slice framework was proposed along with an adaptive VNF placement approach to automatically accommodate to service-specific requirements. Although interesting, these works do not consider the coexistence of different traffic types which makes the VNF placement problem more challenging.

III. System Model

This section describes the system model. In this paper, we consider that multiple VNF-FGs are used for implementing the considered use-case, each representing a specific type of traffic. More specifically, a VNF-FG is dedicated for the haptic input command sent by the surgeon to the robot while two other VNF-FGs represent the haptic and video feedbacks sent back from the robot to the surgeon. The traffic of each VNF-FG is processed by a sequence of VNFs in a predefined order and transmitted from one function to another at a guaranteed data rate. We assume that the order of VNFs of each VNF-FG is known and fixed. With these considerations in mind, we focus on joint problem of placement and scheduling of VNF-FGs.

Each VNF-FG r can be modeled by a directed graph $G_r(\mathcal{V}_r, \mathcal{E}_r)$, where $\mathcal{V}_r$ is the set of ordered VNFs to be placed into the physical infrastructure and $\mathcal{E}_r$ is the set of virtual links. Specifically, $\mathcal{V}_r$ is composed of the set of ordered VNF indices (i.e., $\mathcal{V}_r = \{1, \cdots, |\mathcal{V}_r|\}$). It should be noted that we add two VNFs to represent virtual source and destination of VNF-FG r denoted by indices 0 and $|\mathcal{V}_r| + 1$, respectively. Therefore, for VNF-FG r, VNFs v_0^r and $v_{|\mathcal{V}_r|+1}^r$ must be mapped onto the source node s_r and the destination d_r, respectively.

The substrate network is represented by a graph $G = (\mathcal{J}, \mathcal{L})$, where $\mathcal{J}$ is the set of nodes and $\mathcal{L}$ denotes the set of physical links. A VNF v_i of VNF-FG r, denoted as

v_i^r, has resource requirement $f_{v_i^r}$ and a processing capacity $p_{v_i^r}$. Each physical node n_j has a limited capacity c_{n_j}. The deployment of VNF v_i^r in the network incurs a cost $\alpha_{v_i^r}$ while hosting software on node n_j incurs a cost β_{n_j}. We note that costs are different among VNFs (resp. nodes) since they require different software and site licences (resp. they have different resources capabilities) [14]. Each virtual link is denoted as $e_i^r = (v_i, v_{i+1})$, while each physical link is denoted as $q_{jk} = (n_j, n_k), \forall j \neq k$ and has a fixed propagation delay $d_{q^{j,k}}$. We assume that the nodes are fully connected (i.e., there is a direct link between all pairs of nodes) and the bandwidth of each link is adequate to forward the assigned traffic [15]. To avoid excessive coordination overhead, each VNF is assumed to have sufficient capacity to process the traffic [16].

We consider a time-slotted model, where $\mathcal{T}$ denotes the set of time slots, each with duration τ. The system remains unchanged during each time slot [17]. The main goal is to find the best placement and scheduling of VNFs such that all the traffic requirements in terms of latency and reliability are satisfied and the total cost is minimized. The considered cost function includes (i) hosting cost (i.e., cost of hosting VNFs on physical nodes), (ii) communication cost (i.e., cost of forwarding traffic through physical links), (iii) licence cost (i.e., total software license costs for the VNFs), and (iv) penalty of not serving the traffic of a given VNF-FG (which may affect the characteristics of the optimal solution [17]).

The traffic of VNF-FG r with size w_r needs to be transmitted from the source node s_r to destination node d_r with a guaranteed data rate b_r. It has both latency and reliability requirements denoted by l_r and γ_r, respectively. It is worth noting that w_r and b_r represent the peak of traffic size and data rate, respectively. Given that traffic demand and rate are time-variant in practice, a prudent approach would be to take their peak values while mapping and scheduling VNFs [16]. Similar to [18], we assume that those peak values are fixed over time. This approach minimizes the probability of violating the required latency [18]. We assume that VNFs are not shared between different VNF-FGs and that VNF instances are terminated once the processing is completed.

IV. Problem Formulation

Recall from Section III that we consider the interaction between the surgeon and robot as multiple VNF-FGs, each composed of series of interconnected VNFs. In the following, the joint placement and scheduling of VNFs is formulated as an integer linear program (ILP). To do so, we need to define the following parameters and optimization variables:

- $x^t_{v_i^r,n_j}$ is a binary variable (b.v.) that is equal to 1 if VNF i of VNF-FG r is hosted at node n_j and starts processing traffic at time slot t; otherwise, it is 0.
- $s^t_{v_i^r,n_j}$ is a b.v. that is equal to 1 if VNF i of VNF-FG r is hosted at node n_j and being processing traffic at time slot t; otherwise, it is 0.
- $w_{v_i^r,n_j}$ is a b.v. that is equal to 1 if VNF i of VNF-FG r is hosted at node n_j; otherwise, it is 0.
- $y^t_{l_i^r,q_j}$ is a b.v. that is equal to 1 if virtual link i in VNF-FG r is mapped onto physical link q_j and starts transmitting traffic at time slot t; otherwise it is 0.
- $h^t_{l_i^r,q_j}$ is a b.v. that is equal to 1 if virtual link i in VNF-FG r is mapped onto physical link q_j and being transmitting traffic at time slot t; otherwise it is 0.
- $z_{l_i^r,q_j}$ is a b.v. that is equal to 1 if virtual link i in VNF-FG r is mapped onto physical link q_j and starts transmitting traffic at time slot t; otherwise it is 0.
- Γ_r is a b.v. that is equal to one if the traffic of VNF-FG r is admitted to the network; otherwise, it is 0.
- $\Delta^r_{n_j}$ (resp. $\Theta^r_{n_j}$) is a binary parameter that is equal to 1 if $n_j = s_r$ (resp. $n_j = d_r$); otherwise, it is 0 (resp. 0).

In the following, we explain the constraints that need to be considered for (i) a successful mapping and (ii) to make a schedule feasible. Mapping and scheduling variables are linked in Constraints (1) and (2). Specifically, Constraints (2) ensure that if a VNF is placed at a node n_j, it must start processing traffic at the same node, while Constraints (1) ensure that if a traffic w_r on virtual link e_i^r starts to be transmitted over a physical link q_{jk}, it must be mapped to that link.

$$\sum_{t\in\mathcal{T}} x^t_{v_i^r,n_j} = w_{v_i^r,n_j}, r \in \mathcal{R}, i \in \mathcal{V}_r, j \in \mathcal{J}, \tag{1}$$

$$\sum_{t\in\mathcal{T}} y^t_{e_i^r,q_{jk}} = h_{e_i^r,q_{jk}}, r \in \mathcal{R}, i \in \mathcal{V}_r \cup \{0\}, k \neq j \in \mathcal{J}. \tag{2}$$

Constraints (3) ensure that each VNF i of VNF-FG r cannot be deployed on multiple nodes:

$$\sum_{j\in\mathcal{J}} w_{v_i^r,n_j} \leq 1, r \in \mathcal{R}, i \in \mathcal{V}_r. \tag{3}$$

Constraints (4) ensure that the requirements of the mapped VNFs onto a node cannot exceed the capacity of that node:

$$\sum_{j\in\mathcal{J}} w_{v_i^r,n_j} f_{v_i^r} \leq c_{n_j}, j \in \mathcal{J}. \tag{4}$$

Constraints (5) ensure that VNFs (resp. virtual links) are mapped to physical nodes (resp. physical links) only if the traffic of the VNF-FG is admitted into the network.

$$\sum_{i\in\mathcal{V}_r}\sum_{j\in\mathcal{J}} w_{v_i^r,n_j} = |\mathcal{V}_r|\Gamma_r, r \in \mathcal{R}. \tag{5}$$

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{J},k\neq j} z_{e_i^r,q_{jk}} = \Gamma_r, i \in \mathcal{V}_r, r \in \mathcal{R}. \tag{6}$$

Constraints (7) and (8) define the time-slots during which processing of traffic is performed by VNF v_i^r at node n_j and transmitted over physical link q_{jk}, respectively.

$$\sum_{t\leq t'<t+d_{jk}+w_r/b_r} h^{t'}_{e_i^r,q_{jk}} \geq (d_{jk} + \frac{w_r}{b_r}) y^t_{e_i^r,q_{jk}}, r \in \mathcal{R}, i \in \mathcal{V}_r, i, j \neq k \in \mathcal{J}, t \in \mathcal{T}. \tag{7}$$

$$\sum_{\substack{t'\geq t \\ t'<t+w_r/p_{v_i^r}}} s^{t'}_{v_i^r,n_j} \geq w_r x^t_{v_i^r,n_j}/p_{v_i^r}, r \in \mathcal{R}, i \in \mathcal{V}_r, j \in \mathcal{J}, t \in \mathcal{T}. \tag{8}$$

Constraints (9) ensure that transmitting over physical link $q_{kk'}$ does not occur until the transmissions over previous links q_{jk} are completed.

$$y^{t'}_{v^r_i,q_{jk}} \leq 1-y^{t}_{v^r_i,q_{kk'}}, r \in \mathcal{R}, i \in \mathcal{V}_r, j,k,k' \in \mathcal{J}, \\ t,t' \in \mathcal{T} | t' \leq t + w_r/b_r + d^{kk'}. \quad (9)$$

Constraints (10) and (11) ensure that for each VNF-FG r, first and last VNFs are mapped to source and destination nodes.

$$w_{v^r_0,n_j} = \Gamma_r \Delta^r_{n_j}, r \in \mathcal{R}, j \in \mathcal{J}, \quad (10)$$

$$w_{v^r_{|\mathcal{V}_r|+1},n_j} = \Gamma_r \Theta^r_{n_j}, r \in \mathcal{R}, j \in \mathcal{J}. \quad (11)$$

Constraints (12) ensure that for each VNF-FG, processing of traffic w_r is performed according to its chaining order of VNFs. In other words, VNF v^r_{i+1} cannot start processing traffic before VNF v^r_i completes its processing.

$$1 - \sum_{j \in \mathcal{J}} x^{t}_{v^r_i,n_j} \geq \sum_{j \in \mathcal{J}} x^{t'}_{v^r_{i+1},n_j}, r \in \mathcal{R}, i \in \mathcal{V}_r, \\ t,t' \in \mathcal{T} | t' \leq t + w_r/p_{v^r_i}. \quad (12)$$

Constraints (13) ensure that transmitting traffic w_r over virtual link e^r_i cannot start before the traffic has completely been processed at VNF v^r_i.

$$y^{t'}_{e^r_i,q_{mn}} \leq 1 - \sum_{j \in \mathcal{J}} x^{t}_{v^r_i,n_j}, \forall r \in \mathcal{R}, i \in \mathcal{V}_r, m,n \in \mathcal{J} \quad (13) \\ i \in \mathcal{V}_r, t,t' \in \mathcal{T} | t' \leq t + \frac{w_r}{p_{v^r_i}}$$

Constraints (14) ensure that processing of traffic at VNF v^r_{i+1} will not start before the transmission on virtual link e^r_i has completed.

$$1 - y^{t}_{e^r_i,q_{mn}} \geq 1 - \sum_{j \in \mathcal{J}} x^{t'}_{v^r_{i+1},n_j}, r \in \mathcal{R}, m \neq n \in \mathcal{J}, i \in \mathcal{V}_r, \\ t,t' \in \mathcal{T} | t' \leq t + d^{mn} + w_r/b_r. \quad (14)$$

Constraints (15) represent the flow conservation.

$$\sum_{j \in \mathcal{K} \mathcal{E} \mathcal{J}, k \neq j} \sum z_{e^r_i,q_{jk}} - \sum_{j \in \mathcal{K} \mathcal{E} \mathcal{J}, k \neq j} \sum z_{e^r_i,q_{kj}} = w_{v^r_i,n_j} - w_{v^r_{i+1},n_j}, \\ r \in \mathcal{R}, i \in \mathcal{V}_r, j \in \mathcal{J}. \quad (15)$$

Constraints (16) ensure that the processing and transmission delay does not exceed the required latency of VNF-FG r.

$$\sum_{j \in \mathcal{K} \mathcal{E} \mathcal{J}, k \neq j} \sum y^{t}_{e^r_{|\mathcal{V}_r|},q_{jk}} (t + \frac{w_r}{b_r} + d^{jk}) \leq l_r, r \in \mathcal{R}, j,k \in \mathcal{J}, t \in \mathcal{T}. \quad (16)$$

Constraints (17) ensure that the reliability of the mapping satisfies the reliability requirement of VNF-FG r. According to [19], in sequential case, the VNF-FG is available when both VNFs and nodes (where VNFs are running) are all reliable. Hence, the overall reliability expression is given as follows:

$$\prod_{i \in \mathcal{V}_r} \gamma_{v^r_i} \sum_{j \in \mathcal{J}} w_{v^r_i,n_j} u_{n_j} \geq \gamma_r, \forall r \in \mathcal{R}. \quad (17)$$

Reliability constraints (17) are not linear. These can be approximated by the following linear constraints [20]:

$$1 - \sum_{i \in \mathcal{V}_r} \big(1 - \gamma_{v^r_i} \sum_{j \in \mathcal{J}} w_{v^r_i,n_j} u_{n_j}\big) \geq \gamma_r \Gamma_r, \forall r \in \mathcal{R}. \quad (18)$$

The cost function can be expressed as follows:

$$\sum_{r \in \mathcal{R}} \big(\sum_{i \in \mathcal{V}_r} \sum_{j \in \mathcal{J}} \big(w_{v^r_i,n_j} \alpha_{v^r_i} \beta_{n_j} + \sum_{k \in \mathcal{J}} z_{e^r_i,q_{jk}} \delta_{q_{jk}} \big) + W(1-\Gamma_r) \big), \quad (19)$$

where $W(1-\Gamma_r)$ is a penalty incurred by non allocating resources to serve the traffic of VNF-FG r. The joint VNF placement and scheduling can be written as follows:

$$\text{Minimize } (19) \quad \text{(P1a)}$$

$$\text{subject to } (1)-(16), (18) \quad \text{(P1b)}$$

V. Proposed algorithm

The joint placement and scheduling is NP-hard since the scheduling-only or the placement-only sub-problem is NP-hard [9], [11]. This makes the brute-force approach inefficient to solve it especially for large scale scenarios. To cope with the scalability of the problem, we propose a greedy algorithm. Greedy approaches are widely used to solve NP-hard problems to provide fast, low complexity and efficient solutions.

Our proposed algorithm is divided into three phases: (i) admission control phase, (ii) mapping phase, and (iii) scheduling phase. The admission phase aims to pre-determine whether the traffic requirements of each VNF-FG in terms of completion time can be satisfied based on the best case (i.e., where there is no waiting time). Specifically, we sort VNF-FGs according to their cost penalties. Then, we compute the completion time based on processing times and communication delays (i.e., including propagation and transmission delays) along the shortest path from the source to the destination. If the latency requirements are not satisfied, then, the VNF-FG traffic is rejected. This phase ensure (i) efficient resource allocation by avoiding to allocate resources partially to a VNF-FG while its VNFs cannot be fully served, and (ii) reduces computational burden. This phase corresponds to lines 1-5 in Algorithm 1. Next, we greedily and sequentially place the VNFs with the objective of minimizing the communication time while verifying whether reliability and latency requirements are satisfied. This phase is described in the mapping function given in Algorithm 2. More precisely, we sort nodes according to their distance to the current node, Then, we sequentially and greedily place VNFs on adjacent nodes that can support the VNF requirements. Next, if reliability and completion times satisfy the requirements, we move to the next VNF. If at one point of time, requirements are violated, the node where the first VNF is placed is removed from the set of candidate nodes and VNF remapping is performed. This phase is described in lines 6-14 of Algorithm 1. In the last phase, VNFs are scheduled sequentially on the nodes where they were placed considering processing, communication and waiting times as well as dependency between VNFs. Finally, latency requirements are checked. If they are satisfied, the VNF-FG is admitted and served, otherwise, the VNF-FG traffic is rejected.

This phase is described in lines 15-22 of Algorithm 1. After performing the scheduling phase, the algorithm checks if the latency requirement of each VNF-FG is satisfied or not. If not, a cost equal to the penalty weight is assigned to this VNF-FG, otherwise, the cost is calculated base on the mapping output.

Algorithm 1: Joint Placement and Scheduling Algorithm (JPSA).

Input: $\mathcal{R}, \mathcal{V}_r, \mathcal{J}, l_r, a_r, s_r, d_r, \gamma_{v_i^r}, u_j \forall i \in \mathcal{V}_r, j$

1 $\mathcal{S}_c \leftarrow$ Sorted VNF-FGs $\mathcal{R}$ according to cost penalty.
2 **for** $r \in \mathcal{S}_c$ **do**
3 $P_r \leftarrow$ Shortest path from s_r to d_r
4 $ctime(r) \leftarrow$ Completion time along path P_r.
5 **if** $(ctime(r) > l_r)$ **then** $\Gamma_r = 0$; $\mathcal{S}_c = \mathcal{S}_c \setminus r$;
6 $\mathcal{N} = \mathcal{J}$
7 **for** $r \in \mathcal{S}_c$ **do**
8 $n_{ref} = s_r$
9 $Y \leftarrow$ Check if P_r is feasible for VNF-FG r
10 **while** *($Y = 0$) and ($|\mathcal{N}| \neq 0$)* **do**
11 $[Y, P_r, ctime(r), w_{v_i^r,n_j}] =$
12 Map$(\mathcal{V}_r, \mathcal{N}, n_{ref}, l_r, s_r, d_r, w_r, \delta_r, p_{v_i^r}, f_{v_{i,r}}, c_{n_j}, d_{q^{j,k}})$
13 **if** $Y = 0$ **then** $\mathcal{N} = \mathcal{N} \setminus P_r\{1\}$;
14 **if** *($Y = 0$) or ($Y = 2$)* **then** $\mathcal{S}_c = \mathcal{S}_c \setminus r$; $cost(r) = penalty(r)$;
15 **for** $r \in \mathcal{S}_c$ **do**
16 Schedule VNF-FG on mapped nodes and links based on the path P_r considering the time needed for traffic processing, transmission and propagation.
17 Compute $ctime(r)$ and $rel(r)$.
18 **if** *($ctime(r) \leq l_r$) and ($rel(r) \geq \gamma_r$)* **then** Update scheduling variables and node capacities;
19 **else** $\mathcal{S}_c = \mathcal{S}_c \setminus r$; Set mapping and scheduling variables to 0;
20 **for** $r \in \mathcal{S}_c$ **do**
21 **if** $ctime(r) > l_r$ **then** $cost(r) = penalty(r)$;
22 **else** $cost(r)$= cost of all the placed VNFs and used physical nodes and links.;

Output: Total cost, mapping and scheduling variables

Algorithm 2: Map function.

Input: $\mathcal{V}_r, \mathcal{N}, n_{ref}, l_r, s_r, d_r, w_r, \delta_r, p_{v_i^r}, f_{v_{i,r}}, c_{n_j}, d_{q^{j,k}}$

for $i \in \mathcal{V}_r$ **do**
  $\mathcal{S}_n \leftarrow$ Sorted nodes according to distances to n_{ref}.
  for $j \in S_n$ **do**
  if $c_j \geq l_{v_i^r}$ **then**
  $ctime \leftarrow$ Current completion time.
  $rel \leftarrow$ Current reliability.
  if *($ctime \leq l_r$) and ($rel \geq \delta_r$)* **then**
  $w_{v_i^r,n_j} = 1$; $n_{ref} = j$
  $c_{n_j} = c_{n_j} - f_{v_i^r}$; $P_r = P_r \cup \{j\}$
  break
  if $\sum_{j \in \mathcal{J}} x_{v_i^r,n_j} = 0$ **then** $Y = 2$; **break** ;
if $\sum_{i \in \mathcal{V}_r} \sum_{j \in \mathcal{J}} w_{v_i^r,n_j} == |\mathcal{V}_r|$ **then**
  $P \leftarrow$ Shortest path from n_{ref} to d_r
  $d \leftarrow distance(n_{ref}, n_j, P)$
  $ctime(r) = ctime(r) + d$ along path P
  if $ctime(r) \leq l_r$ **then** $Y = 1$;
  else $Y = 0$;

Output: Y, P_r, $ctime(r)$ and mapping variables.

A. Time Complexity Analysis

In this subsection, we present the complexity analysis of the proposed JPSA algorithm. First, sorting the given set of VNF-FGs $\mathcal{R}$ (line 1 in Algorithm 1) using quick-sort algorithm returns a solution with worst-case complexity of $\mathcal{O}(|\mathcal{R}| \lg |\mathcal{R}|)$. Second, assuming that the Dijkstra's algorithm is used to obtain the shortest path, then the worst-case time complexity of the loop (lines 2-5) is equal to $\mathcal{O}(|\mathcal{R}|(|\mathcal{L}| + |\mathcal{J}| \lg |\mathcal{J}|)$. The time complexity of the mapping function is equal to $\mathcal{O}(\max_r |\mathcal{V}_r|(|\mathcal{J}| \lg |\mathcal{J}| + |\mathcal{J}|) + |\mathcal{L}| + |\mathcal{J}| \lg |\mathcal{J}| + |\mathcal{J}| \max_r |\mathcal{V}_r|) = \mathcal{O}(\max_r |\mathcal{V}_r||\mathcal{J}| \lg |\mathcal{J}| + |\mathcal{L}|)$. Hence, the worst-case time complexity of the mapping phase given in lines 7-14 is equal to $\mathcal{O}(|\mathcal{R}||\mathcal{J}|(\max_r |\mathcal{V}_r||\mathcal{J}| \lg |\mathcal{J}| + |\mathcal{L}|)) = \mathcal{O}(\max_r |\mathcal{V}_r||\mathcal{R}||\mathcal{J}|^2 \lg |\mathcal{J}| + |\mathcal{R}||\mathcal{J}||\mathcal{L}|))$. The worst-case time complexity of the scheduling in lines 15-19 is $\mathcal{O}(|\mathcal{T}||\mathcal{R}| \max_r |\mathcal{V}_r|)$. The complexity of the last loop given in lines 20-22 is equal to $|\mathcal{R}|$. Finally, the overall worst-case time complexity of the algorithm is then given by $\mathcal{O}(|\mathcal{R}| \lg |\mathcal{R}| + |\mathcal{R}||\mathcal{J}||\mathcal{L}| + \max_r |\mathcal{V}_r||\mathcal{R}|(T + |\mathcal{J}|^2 \lg |\mathcal{J}|))$.

VI. Numerical Results

In this section, the performance of the proposed algorithm JPSA is evaluated and compared to baseline algorithms and the optimal solution obtained by solving the ILP model.

A. Simulation Setup

In our simulations, we consider a scenario with $|\mathcal{J}| = 10$ NFV-enabled nodes and $|\mathcal{L}| = \sum_{k=1}^{|\mathcal{J}|-1} k = 45$ links. The penalty weight W for each data traffic is set to a value between 200 and 2000, which is chosen large enough to allow the scheduler to put more priority in admitting data traffic since never admitting them can result in the least cost using small penalty weight. The effect of the penalty weight will be shown later in Fig. 1. The number of VNFs is set to a random value between 4 and 5 for the haptic command and feedback while it is set to a random values between 5 and 6 for video traffic. The processing capacity of each VNF is randomly selected between 2000 and 3000 Mbits per second. Each node has an available capacity selected randomly between 0 and 2 units while the requirement of each VNF is selected randomly between 0.1 and 2 units [16]. The reliability of a VNF or a node is randomly selected between 0.9999 and 1 [19]. The cost of using a link, a node or to host a VNF on a physical node is randomly set to a random value between 0 and 11 [16]. We use CPLEX solver to carry out the optimal results while other simulations are obtained using MATLAB. We run all the simulations multiple number of times and compare their average for evaluation. For ease of reference, the remaining simulation parameters are summarized in TABLE I.

TABLE I: Simulation parameters.

Parameter	Notation	Value
Number of time slots	$\|\mathcal{T}\|$	1000
Number of VNFs	$\|\mathcal{V}_r\|$	$rand(4,5)$ for haptic, $rand(5,6)$ for video
Traffic size	w_r	$rand(400,600)$ Mbits
Data rate	b_r	$rand(2,3)$ Gbps
Reliability requirements	γ_r	one 9 to five 9s
VNF requirements	$f_{v_i^r}$	$rand(0.1,2)$
VNF processing capacities	$p_{v_i^r}$	$rand(2000,3000)$ Mbps
Node capacities	c_{n_j}	$rand(1,3)$
Propagation delays	$d^{i,j}$	$rand(0,1)$ time slots

B. Baseline Algorithms

Since there are no comparable works in the literature, we compare the performances of our proposed algorithm to the following two baseline algorithms:

- The randomized mapping with first-deadline first-served scheduling (RM-FDFS) algorithm: It is a purely randomized algorithm. It follows two main steps: mapping and scheduling phases. During the mapping phase, for each VNF, it randomly selects a node from the feasible set of NFV-enabled nodes that satisfy both latency and reliability requirements to host that VNF. Then, node capacities are updated accordingly. Mapping each virtual link onto one or multiple physical links is performed according to the shortest path algorithm between the two NFV-enabled nodes hosting two consecutive VNFs. Once the mapping phase is completed, the RM-FDFS algorithm schedules the VNFs according to the FDFS policy.
- The joint VNF placement and scheduling using a single VNF-FG with the most stringent QoS requirements algorithm, denoted as single VNF-FG based joint placement and scheduling algorithm (SFG-JPSA): It is similar to our proposed greedy JPSA. The main difference relies on modeling the VNFs required to process the different kind of traffic and setting up the QoS requirements. Specifically, all the VNFs belonging to the different VNF-FGs are grouped into a single VNF-FG. The most stringent latency and reliability requirements among the three VNF-FGs are assigned to that single VNF-FG.

Fig. 1 illustrates the admission rate of the optimal solution as a function of the penalty weights W. We can see that the admission rate increases as the penalty weights increase. This occurs since a low penalty weight value makes the scheduler behavior more conservative because discarding that traffic leads to a cheaper cost and resources can then be used to serve other traffic. This might suggest to carefully adjust these penalty weights to be larger than the processing and communication costs, which model the cost of resource requirements when the VNF-FG traffic is admitted. If, however, the penalty weight is smaller than the cost of resource requirements, then the policy of never admitting traffic will result in the least cost. For this reason, the penalty weights are set to random values between 200 and 2000 which are large enough to allow the algorithm to prioritize the maximization of the number of admitting traffic of the VNF-FGs over non admitting them.

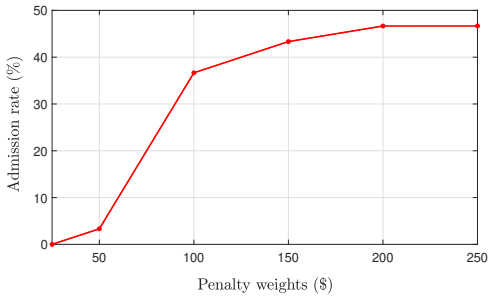
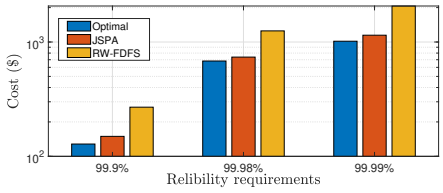
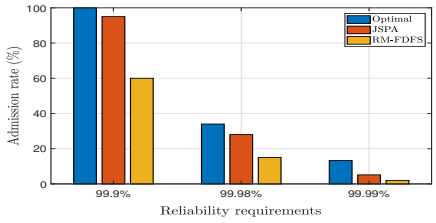


Fig. 1: Impact of penalty weights on admission rate.



(a) Impact of reliability on cost.



(b) Impact of reliability on admission rate.

Fig. 2: Impact of reliability on the performance.

Fig. 2 plots the cost as a function of reliability requirements. The cost increases as reliability requirements become more stringent. This can be explained by the fact that admitting and scheduling traffic with challenging requirements consumes more resources. Hence, much more money are needed.

Fig. 3 illustrates the total achieved cost and the admission rate as a function of latency requirements. As expected, the total cost and the node utilization decrease as the latency requirements become more stringent while the admission rate decreases. This occurs because less stringent latency requirements can tolerate longer transmission, communication and waiting times. Hence, the probability of admitting and scheduling traffic within the required latency requirements is increased. We also notice that our algorithm JSPA outperforms the baseline solution in terms of cost, admission rate and node utilization is close to the ones obtained by the optimal solution.

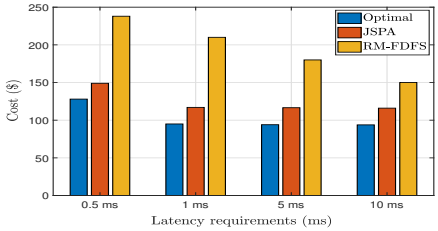
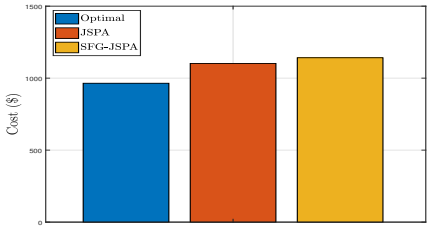
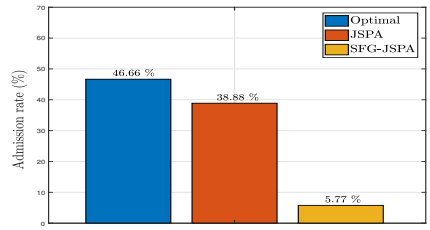


Fig. 3: Impact of latency on cost.



(a) Cost comparison.



(b) Admission rate comparison.

Fig. 4: Single VNF-FG vs multiple VNF-FGs where $l_1 = l_2 = 0.5$ ms and $l_3 = 3$ ms and $\gamma_1 = \gamma_2 = 0.9999$ and $\gamma_3 = 0.9$.

Fig. 4 compares both the cost and the admission rates for the optimal solution and the ones provided by the proposed algorithm JSPA and the baseline algorithm SFG-JSPA. Using a single VNF-FG with the most strict latency and reliability requirements (as in SFG-JSPA) results in very low admission rate compared to JSPA and the optimal solution. Although, the admission rate is low, the cost obtained by this admission rate is high compared to other solutions. This demonstrates that using multiple VNF-FGs and customizing the QoS accordingly is more cost-efficient. This validates the purpose of this work.

VII. Conclusions

In this paper, the joint virtual network function (VNF) placement and scheduling problem considering both latency and reliability requirements is investigated. The problem is formulated as an integer linear program (ILP). Since this problem is challenging and complex, a simple and efficient greedy algorithm is proposed. The simulation results demonstrated that our proposed algorithm achieves a small performance gap compared to the optimal solution. Moreover, it outperformed the baseline algorithms in terms of cost and admission rate. Furthermore, our results demonstrated that splitting application traffic to multiple VNF-forwarding graphs (VNF-FGs) with different quality-of-service (QoS) requirements achieves a significant gain in terms of cost and resource usage compared to modeling the whole application traffic with one VNF-FG having the most stringent requirements.

References

[1] N. Promwongsa, A. Ebrahimzadeh, D. Naboulsi, S. Kianpisheh, F. Belqasmi, R. Glitho, N. Crespi, and O. Alfandi, "A comprehensive survey of the tactile internet: State-of-the-art and research directions," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 1, pp. 472–523, 2020.

[2] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware vnf placement for service-customized 5g network slices," in *Proc. IEEE INFOCOM*, 2019, pp. 2449–2457.

[3] P. J. Choi, R. J. Oskouian, and R. S. Tubbs, "Telesurgery: past, present, and future," *Cureus*, vol. 10, no. 5, 2018.

[4] T. Leal Ghezzi and O. Campos Corleta, "30 years of robotic surgery," *Springer World journal of surgery*, vol. 40, no. 10, pp. 2550–2557, 2016.

[5] F. Boabang, A. Ebrahimzadeh, R. H. Glitho, H. Elbiaze, M. Maier, and F. Belqasmi, "A machine learning framework for handling delayed/lost packets in Tactile Internet remote robotic surgery," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 4, pp. 4829–4845, 2021.

[6] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Trans. Netw. Serv.*, vol. 13, no. 3, pp. 518–532, 2016.

[7] N. Herbaut, D. Negru, D. Dietrich, and P. Papadimitriou, "Service chain modeling and embedding for nfv-based content delivery," in *Proc. IEEE ICC*, 2017, pp. 1–7.

[8] C. Sarathchandra, S. Robitzsch, M. Ghassemian, and U. Olvera-Hernandez, "Enabling bi-directional haptic control in next generation communication systems: Research, standards, and vision," in *Proc. IEEE CSCN*, 2021, pp. 99–104.

[9] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. IEEE NetSoft)*, 2015, pp. 1–9.

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

[11] N. Gholipoor, H. Saeedi, N. Mokari, and E. A. Jorswieck, "E2E QoS guarantee for the Tactile Internet via joint NFV and radio resource allocation," *IEEE Trans. Netw. Serv.*, vol. 17, no. 3, pp. 1788–1804, 2020.

[12] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, 2018.

[13] L. Dinh-Xuan, M. Seufert, F. Wamser, and P. Tran-Gia, "Study on the accuracy of qoe monitoring for http adaptive video streaming using vnf," in *Proc. IFIP/IEEE INSM*, 2017, pp. 999–1004.

[14] M. Dieye, S. Ahvar, J. Sahoo, E. Ahvar, R. Glitho, H. Elbiaze, and N. Crespi, "CPVNF: Cost-efficient proactive VNF placement and chaining for value-added services in content delivery networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 2, pp. 774–786, 2018.

[15] J. Li, W. Shi, P. Yang, and X. Shen, "On dynamic mapping and scheduling of service function chains in SDN/NFV-enabled networks," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.

[16] N. Promwongsa, A. Ebrahimzadeh, R. Glitho, and N. Crespi, "Joint VNF placement and scheduling for latency-sensitive services," *IEEE Trans. Netw. Sci. Eng.*, 2022.

[17] P. T. A. Quang, A. Bradai, K. D. Singh, G. Picard, and R. Riggio, "Single and multi-domain adaptive allocation algorithms for VNF forwarding graph embedding," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 1, pp. 98–112, 2018.

[18] H. Alameddine, M. H. K. Tushar, and C. Assi, "Scheduling of low latency services in softwarized networks," *IEEE Transactions on Cloud Computing*, vol. 9, no. 3, pp. 1220–1235, 2021.

[19] M. Wang, B. Cheng, S. Wang, and J. Chen, "Availability-and traffic-aware placement of parallelized SFC in data center networks," *IEEE Trans. Netw. Serv.*, vol. 18, no. 1, pp. 182–194, 2021.

[20] R. Guerzoni, Z. Despotovic, R. Trivisonno, and I. Vaishnavi, "Modeling reliability requirements in coordinated node and link mapping," in *Proc. IEEE SRDS*, 2014, pp. 321–330.