

Processing Prioritization of Modular Medical Applications in Future 6G Radio Access Networks

Nicolai Kröger¹, Giuseppe Gattulli², Franziska Jurosch³, Sven Kolb³, Dirk Wilhelm^{3,4,5}, Wolfgang Kellerer¹, Fidan Mehmeti¹

¹*Technical University of Munich, TUM School of Computation, Information and Technology, Chair of Communication Networks, firstname.lastname@tum.de*

²*Politecnico di Milano, firstname.lastname@mail.polimi.it*

³*Technical University of Munich, TUM School of Medicine and Health, TUM University Hospital, Research Group MITI*

⁴*Technical University of Munich, TUM School of Medicine and Health, TUM University Hospital, Department of Surgery*

⁵*Technical University of Munich, Munich Institute of Robotics and Machine Intelligence, firstname.lastname@tum.de*

Abstract—Medical applications, such as telemedicine or smart operation rooms, place stringent requirements on the underlying network architecture. 6G as the next-generation communication standard currently in research promises to satisfy the needs of such applications by utilizing advances in technology and networking concepts. One crucial concept for medical applications is the capability of using computing resources within the network. By placing the applications on such processing nodes in different locations within the Radio Access Network (RAN), the performance metrics of a medical application, such as latency, throughput, and availability can be optimized. However, problems arise when the available processing capabilities are not sufficient for all requested medical applications. In this paper, we formulate an Integer Linear Program (ILP) to address the problem of processing medical applications within the network when the processing capabilities are not sufficient. We consider the priority and different service levels of application functions and aim to place as many applications as possible with the best possible service quality. Additionally, we take into account that some applications must run in the network even if their priority is low. Furthermore, we propose a heuristic in order to obtain a good solution quickly. The evaluation of our solution and comparison to existing approaches shows an increase of accepted demands in the network by up to 35%.

Index Terms—6G, Prioritization, Heuristic, In-Network Computing, Medical Technology.

I. INTRODUCTION

The currently researched next generation of communication networks, 6G, addresses the networking requirements of emerging applications such as virtual reality and connected autonomous systems. In particular, 6G is envisioned to tremendously increase data rates and availability and to decrease the latency. This is achieved not only by technological advances such as higher frequency ranges, but also through a holistic design of applications, the underlying communication network, and in-network computing capabilities [1], [2].

One area which will especially benefit from the new features of 6G is the medical sector [3], [4]. Medical applications

place a number of different performance demands on the communication network. In-network computing plays a crucial role in satisfying these demands [2]. In particular, modular parts of a medical application, in this paper referred to as Modular Application Functions (MAFs) as introduced in [5], can be dynamically executed on various processing resources within the network. Note that the concept of placing MAFs extends the similar concept of VNFs by not only considering network-related functions, but also the applications. This allows to take application-specific requirements into account, enabling closer interaction between network and application. The main challenge is the optimal placement of such MAFs on processing nodes in a communication network. Existing work [6], [7], and [8] already covers a large area of aspects for placing VNFs. However, the existing literature does not take into account the special requirements of medical applications. In a first step, the authors in [9] combine the VNF placement approach and the requirements of medical applications, optimizing the placement costs.

In contrast, in this paper we consider a scenario where the available networking and processing capabilities are not sufficient to fulfill the demands of all requested medical applications. For this purpose, we formulate an Integer Linear Program (ILP) with the focus on admitting as many applications as possible to the network. Furthermore, for each MAF we consider the priority and different levels of service regarding the performance in terms of latency and throughput. The strategy is to execute higher prioritized MAFs with higher levels of service while lower prioritized MAFs experience lower levels of service or are terminated completely. Additionally, we also consider that some MAFs are *non-terminable*, i.e., they must be admitted to the network even if their priority is low and once placed, their execution cannot be terminated until their task is completed. That means that higher prioritized MAFs may experience lower service levels or are even terminated as the non-terminable MAF must be placed with at least the lowest possible service level. Examples of such non-terminable applications are logistics,

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the program of “Souverän. Digital. Vernetzt.”. Joint project 6G-life, project identification number: 16KISK002.

documentation or administrative tasks, etc. Since the time to find an optimal placement solution is considerably large, we furthermore propose a heuristic with reduced execution time. Finally, we evaluate our approach and compare it with the approach in [9] as a baseline. The main message of this paper is that the co-design of MAFs and the network in terms of available and required performance constraints and the priority can significantly increase the overall number of admitted applications within a network, especially in scenarios with insufficient resources. Furthermore, the main contributions of this paper are:

- We formulate an ILP to optimize the admittance ratio of MAFs in the network considering different levels of service, priorities and non-terminability of certain MAFs.
- We introduce a heuristic in order to obtain a near-optimal solution in a more reasonable time.
- We evaluate our results and compare them to existing solutions showing the capabilities of our proposed method.

The remainder of the paper is structured as follows. In Section II, existing related work is described. Then, we introduce the proposed model in Section III. In Section IV, we formulate an ILP for the problem, followed by a corresponding heuristic algorithm in Section V. The results are presented in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK

Recent work covers various aspects of the placement of VNFs. The authors in [6] introduce an ILP for the joint VNF chain placement and resource allocation, minimizing costs and VNF migration frequency. The idea of VNF migration is also used in [10], where the objective is to minimize the overall power consumption. Liu *et al.* [11] investigate the VNF reconfiguration problem with the focus on optimized reconfiguration costs and resource consumption in IoT networks. Jointly optimizing radio and VNF resource allocation, the authors in [12] propose an approach to minimize the overall deployment costs while guaranteeing end-to-end delay requirements. In [13], the authors aim to minimize the slice performance degradation by optimizing VNF migration based on traffic prediction. The authors in [14] and [15] further optimize the resource usage while considering stringent time constraints by leveraging parallel and shared VNF processing. Akahoshi *et al.* [16] leverage dynamic VNF computation resource usage to optimize the overall resource usage. The authors in [17] introduce with *Holu* a fast heuristic framework for solving the joint VNF placement and routing problem considering power consumption and resource constraints in the network in reasonable time.

Recently, some works such as [18], [19] started to leverage Machine Learning (ML) and Artificial Intelligence (AI) in their framework for the VNF placement. A further framework for VNF placement and traffic prediction in a 5G O-RAN architecture is introduced in [20]. Focusing more on the reliability and availability, the authors in [7] formulate an optimization problem for the placement of VNF chains in 5G networks. The authors in [21] target the dynamic VNF placement,

resource allocation, and traffic routing within 5G networks considering various real-world parameters. Promwongsa *et al.* [8] already target the next generation of networks, 6G, and introduce a joint VNF placement and scheduling problem for latency sensitive VNFs focusing on the optimal determination of whether to reuse an existing VNF or to place a new one.

Very important for critical applications is prioritization. This is especially the case for scenarios with insufficient processing and networking resources. The placement of VNF chains with priority has been investigated in [22]. In their work, the authors formulate an optimization problem to minimize the total deployment costs considering VNF sharing and two types of service: *priority* and *best-effort*. In contrast to their work, in this paper we propose an approach tailored to the specific requirements of medical application scenarios.

Hentati *et al.* [9] take a first step to combine medical applications and VNF placement for the scenario of a one-to-one remote robotic surgery. Based on the requirements of such a scenario, they formulate a joint placement and scheduling optimization problem as ILP, considering haptic and video traffic. In particular, they aim to minimize the deployment costs constrained by end-to-end latency, reliability, and throughput aspects. In contrast to their work, in this paper we consider multiple medical use cases and therefore cover additional aspects. Moreover, we focus on the scenario when the available networking and in-network processing capabilities are not sufficient to serve all demands of different medical applications. In such a scenario, we aim to maximize the number of served demands. Hereby, we consider the priority of MAFs and different levels of service. Additionally, we take into account that some applications must run in the network even though their priority is low. We then compare our approach to the one in [9] and show the improvements our approach offers.

III. PROBLEM FORMULATION

In [5], we introduce the concept of MAFs as modular parts of medical applications, which are executed on processing units (PU) within the network. Additionally, we demonstrate the MAF concept for several medical applications in [23], [3], and [4]. In our model, this envisioned RAN architecture is structured as a (logical) fully meshed graph $G = (N, L)$, where N consists of physical nodes, i.e., APs, PUs, and switches, which are all linked together through connections L . Note that real networks may not provide direct physical connections between two nodes. In order to achieve a fully-meshed graph in such a scenario, virtual links need to be added to abstract the path between two nodes into a direct connection. In the considered graph, only PUs can process MAFs ($N' \subset N$). Each processing unit n' has a computational resource capacity $c_{cpu}^{n'}$ and an availability $\phi_{n'}$. Switches form another subset of nodes, i.e., $N'' \subset N$. The connection (u, v) between node u and node v can transmit a limited data rate, $c_{\pi}^{(u,v)}$, and experience propagation delay, $d_{(u,v)}$. Each MAF, $a \in A$, in the network has a terminability characteristic k_a , indicating whether an MAF is non-terminable, i.e., $k_a = 1$,

or not, i.e., $k_a = 0$. Additionally, the priority p_a ranks the importance on an ascending scale. Furthermore, each MAF has a specific demand for computational resources γ_a to run on a processing node, and an availability ϕ_{I_a} . The required data rate of an MAF is denoted as $\lambda_{h,a}$ and represents the maximum achieved traffic peak. The data rate and the latency $\tau_{h,a}$ of an MAF correspond to a service level h of different service levels $h \in H$. The different service levels represent an upper and a lower bound of the performance of an MAF. The level of service is given to an MAF by a network controlling entity responsible for placing all MAFs. Varying the level of service for each MAF based on its priority allows to accept more demands within the network. An example for varying service is the change of video encoding schemes to lower resolution in order to reduce the required throughput. Note that for the determination of MAF attributes different aspects, such as ethical ones, need to be considered.

The MAFs are now placed on nodes within a *demand*. Each demand $d \in D$ consists of a source node s_d , a destination node t_d and a required MAF r_d . It has specific performance requirements such as a minimum data rate f_{π_d} , a maximum end-to-end delay f_{to_d} , and a required availability level ϕ_{R_d} , for reliable service access. Note that even though PUs may differ in their characteristics, in this paper we do not consider it in the first step. In the medical context, the overall acceptance ratio of demands is of interest. Thus, in this paper we aim to maximize the acceptance ratio of demands within the network considering the importance of each individual demand. In our model, we make the following assumptions:

- In order to simplify the data flow modeling, we assume fully meshed networks, where each node is directly connected to all other nodes. Thus, if there is no physical link between two nodes, the direct connection between both is virtually added, abstracting the path between the two nodes.
- There are enough computing resources available to at least execute all non-terminable MAFs.
- The analysis method considers one MAF at a time, excluding interactions between MAFs, i.e., MAF chains are not considered and are deferred to future work.
- Multiple instances of the same MAF can be deployed across the network for multiple demands.
- Each demand uses one instance of an MAF; sharing MAFs is part of the future work.
- MAFs do not change the throughput, i.e., the incoming throughput to each MAF is the same as the outgoing.
- The optimization model ignores task rescheduling or resource reallocation times, concentrating on static resource allocation and immediate MAF performance. This implies that MAFs are not terminated after the processing since they potentially need to process data in the future.
- All MAFs comply to the performance limit given to them by the placement controller. This ensures no unwanted behavior within the network.

The optimization problem at hand is categorized as an ILP problem, given that the decision variables must be integers

due to the discrete nature of resource allocation. Fractional variables are impractical, emphasizing the requirement for whole units in allocation decisions, ensuring that resources are fully and effectively utilized.

IV. OPTIMIZATION PROBLEM

Based on the described model and assumptions, in the following the optimization problem for medical applications is formulated. The decision variables are as follows:

- $x_{a,d}^{n'} \in \{0, 1\}$: Indicates whether MAF a is deployed for demand d on processing node n' , with 1 meaning deployed and 0 otherwise.
- $y_d^{(u,v)} \in \{0, 1\}$: Denotes whether demand d utilizes the link between nodes u and v in the network, with 1 for usage and 0 otherwise.
- $z_d \in \{0, 1\}$: Indicates whether the traffic for the demand d is admitted to the network, with 1 indicating admission and 0 otherwise.
- $m_{h,a,d}^{n'} \in \{0, 1\}$: Shows whether a specific service level h is selected for MAF a in relation to demand d on processing node n' , with 1 if selected and 0 otherwise.

The objective function, shown in (1), is designed to maximize the value of accepted demands, focusing on high-priority healthcare services to ensure the most effective service amidst an increased number of network demands. It incorporates a penalty W for not accepted demands, to incentivize the maximization of request acceptances in the network, and is

$$\max \sum_{d \in D} \left(z_d \cdot p_{r_d} \cdot \sum_{n' \in N'} \sum_{h \in H} m_{h,r_d,d}^{n'} \cdot \lambda_{h,r_d} - W \cdot (1 - z_d) \right). \quad (1)$$

This objective function is subject to various constraints, ensuring that performance, placement, and routing requirements are fulfilled. Next, we will formulate all of them.

Starting with the performance related constraints, constraint (2) ensures that demands utilizing non-terminable MAFs are always integrated into the network:

$$k_{r_d} \leq z_d, \quad \forall d \in D. \quad (2)$$

Constraint (3) guarantees that the deployed MAFs meet the minimum data rate requirements of the demands:

$$x_{r_d,d}^{n'} \cdot f_{\pi_d} \leq \sum_{h \in H} m_{h,r_d,d}^{n'} \cdot \lambda_{h,r_d}, \quad \forall n' \in N', d \in D. \quad (3)$$

Constraint (4) caps the cumulative delay experienced by demands, incorporating both processing and propagation delays:

$$z_d \cdot f_{to_d} \geq \sum_{n' \in N'} \sum_{h \in H} m_{h,r_d,d}^{n'} \cdot \tau_{h,r_d} + \sum_{u \in N} \sum_{v \in N, (u,v) \in L} (y_d^{(u,v)} + y_d^{(v,u)}) \cdot d_{(u,v)}, \quad \forall d \in D. \quad (4)$$

Constraint (5) is an approximation based on [24] and [25]. It aims at ensuring that the network availability aligns with the stringent availability requirements of the demand:

$$\phi_{R_d} \cdot z_d \leq 1 - \left((1 - \phi_{s_d}) + \sum_{m \in N} x_{r_d,d}^m \cdot (1 - \phi_{I_{r_d}} \phi_m) + \sum_{n \in N} \sum_{v \neq n \in N} (1 - x_{r_d,d}^n) y_d^{(v,n)} (1 - \phi_n) \right), \quad \forall d \in D. \quad (5)$$

More on the placement site, constraint (6) guarantees that the required MAF for a demand is installed on a single processing node within the network, provided the demand is accepted:

$$z_d \leq \sum_{n' \in N'} x_{r_d,d}^{n'} \leq 1, \quad \forall d \in D. \quad (6)$$

Constraint (7) mandates that exactly one MAF is activated for each accepted demand, preventing any redundant activation that could otherwise strain network resources:

$$\sum_{n' \in N'} \sum_{a \in A} x_{a,d}^{n'} = z_d, \quad \forall d \in D. \quad (7)$$

Constraint (8) limits CPU allocation per processing node to prevent overloads:

$$\sum_{d \in D} \sum_{a \in A} x_{a,d}^{n'} \cdot \gamma_a \leq c_{cpu}^{n'}, \quad \forall n' \in N'. \quad (8)$$

Constraint (9) enforces a strict one-to-one correspondence between an MAF and its service type for each demand:

$$\sum_{h \in H} m_{h,r_d,d}^{n'} = x_{r_d,d}^{n'}, \quad \forall n' \in N', d \in D. \quad (9)$$

Constraint (10) guarantees that the total data rate of demands on link (u, v) must not exceed the capacity of the link:

$$c_{\pi}^{(u,v)} \geq \sum_{n' \in N'} \sum_{d \in D} \sum_{h \in H} z_d \cdot (y_d^{(u,v)} + y_d^{(v,u)}) \cdot m_{h,r_d,d}^{n'} \cdot \lambda_{h,r_d}, \quad \forall u, v \in N, u \neq v, (u, v) \in L. \quad (10)$$

Constraint (11) controls the node activation by setting nodes that cannot host MAFs, i.e., non-processing nodes, for the demand to 0:

$$\sum_{n \notin N'} x_{r_d,d}^n = 0, \quad \forall d \in D. \quad (11)$$

In order to correctly route traffic flows, constraints (12)-(14) ensure the flow conservation from source to destination node for each demand:

$$\sum_{u \in N, u \neq s_d} y_d^{(s_d,u)} = z_d, \quad \forall d \in D. \quad (12)$$

$$\sum_{v \in N, v \neq u} y_d^{(u,v)} - \sum_{v \in N, v \neq u} y_d^{(v,u)} = 0, \quad \forall d \in D, u \in N \setminus \{s_d, t_d\}. \quad (13)$$

$$\sum_{u \in N, u \neq t_d} y_d^{(u,t_d)} = z_d, \quad \forall d \in D. \quad (14)$$

Constraint (15) mandates that for any intermediate node incoming flows must match outgoing flows, ensuring network flow conservation:

$$y_d^{(v,u)} \leq \sum_{b \in N, b \neq u, v} y_d^{(u,b)}, \quad \forall d \in D, u \in N \setminus \{s_d, t_d\}, v \neq u \in N. \quad (15)$$

Constraint (16) requires that for any accepted demand with MAF r_d on node n' , the path must include n' , activated by at least one incoming link:

$$z_d \cdot x_{r_d,d}^{n'} \leq \sum_{u \in N, u \neq n'} y_d^{(u,n')}, \quad \forall d \in D, \forall n' \in N'. \quad (16)$$

Constraint (17) prevents activating service levels on nodes unable to host MAFs:

$$\sum_{h \in H} m_{h,r_d,d}^n = 0, \quad \forall n \notin N', \forall d \in D. \quad (17)$$

Constraints (18) and (19) guarantee that a node with a deployed MAF for a demand only exchanges traffic with its source or destination node, while blocking other nodes:

$$x_{r_d,d}^n \cdot y_d^{(u,n)} = 0, \quad \forall d \in D, n \in N, u \neq s_d \notin N''. \quad (18)$$

$$x_{r_d,d}^n \cdot y_d^{(n,u)} = 0, \quad \forall d \in D, n \in N, u \neq t_d \notin N''. \quad (19)$$

Constraint (20) deactivates the source-destination link if the destination is not the selected MAF node:

$$y_d^{(s_d,t_d)} \leq x_{r_d,d}^{t_d}, \quad \forall d \in D. \quad (20)$$

Constraints (21) to (23) mandate that traffic from the source to the destination passes only through switches and the node activated for the required MAF:

$$y_d^{(u,n)} + y_d^{(n,u)} \leq x_{r_d,d}^n, \quad \forall d \in D, n \neq s_d, t_d \notin N'', u \neq n, t_d \in N'. \quad (21)$$

$$\sum_{u \in N} y_d^{(u,s_d)} = 0, \quad \forall d \in D. \quad (22)$$

$$\sum_{u \in N} y_d^{(t_d,u)} = 0, \quad \forall d \in D. \quad (23)$$

In summary, the optimization problem can be written as

$$\max \sum_{d \in D} \left(z_d \cdot p_{r_d} \cdot \sum_{n' \in N'} \sum_{h \in H} m_{h,r_d,d}^{n'} \cdot \lambda_{h,r_d} - W \cdot (1 - z_d) \right), \quad (P1a)$$

$$\text{s. t.} \quad (2) - (23). \quad (P1b)$$

V. PROPOSED HEURISTIC

The placement problem is classified as NP-hard, rendering the brute-force method ineffective. This holds especially in extensive scenarios. To overcome the issues of scalability, we introduce a heuristic method, named *Modular Application Function Allocation Prioritization (MAFAP)*, that is efficient, has a low complexity, and is therefore quick in providing solutions close to the optimum. It unfolds in two stages: *i)* Sorted by terminability and priority, it assigns the demands within the network at the lowest possible service level as outlined in Algorithm 1; *ii)* It enhances the service provided to the demand as outlined in Algorithm 2. Note that by varying the number of iterations in the second step, the execution time and the acceptance ratio achieved by the heuristic can be adjusted for

each individual scenario. The overall time complexity of the Local Search algorithm is given by $O(D \log D + DNS \cdot p \log p)$ where D is the number of demands, N is the number of demands nodes, S is the number services and p is the number of paths. This complexity suggests that the algorithm operates in polynomial time for common scenarios.

Algorithm 1 Place network demands

Require: G , demand_info, app_func, app_serv, link_info

Ensure: $select_node, select_path, select_serv$ if possible, otherwise error

```

1:  $select\_node, select\_path, select\_serv \leftarrow None$ 
2: for all nodes in  $G$  considering capacity do
3:   if node does not have enough capacity then
4:     Go to the next node
5:   end if
6:   for all services meeting demand do
7:     for all paths through node do
8:       if path meets delay and capacity constraints then
9:         Calculate path availability
10:        if path availability meets demand then
11:          Update selection variables
12:          if suitable service found then
13:            Break loop
14:          end if
15:        end if
16:      end if
17:    end for
18:  end for
19: end for
20: if no node selected then
21:   return error
22: else
23:   Deduct resources from  $G$ 
24: end if
25: return  $select\_node, select\_path, select\_serv$ 

```

VI. EVALUATION

In this section, we show results related to Section IV and the corresponding heuristic presented in Section V. Additionally, they are compared to state-of-the-art approaches.

TABLE I: Simulation parameters

Parameter	Value
c_{cpu}^n	<i>uniform</i> (3, 6)
$c_{\pi}^{(u,v)}$	<i>randint</i> (0.5, 1.5) Gbps
$d_{(u,v)}$	<i>uniform</i> (1, 8) ms
f_{π_d}	<i>uniform</i> (0.5, 400) Mbps
f_{to_d}	<i>uniform</i> (20, 60) ms
av_{R_d}	one 9 to five 9s

A. Setup Description

For the evaluation, we investigate the performance and the impact of various parameters on it for the optimization problem (*Optimal*), introduced in Section IV, and the corresponding heuristic *MAFAP*, presented in Section V. The results are compared to those of two other approaches:

Algorithm 2 Upgrade level of service for demand

Require: G , demand_info, path, serv, app_serv, link_info

Ensure: Upgraded service identifier or *None*

```

1: Extract  $req\_af$  from demand_info
2:  $select\_serv \leftarrow None$ 
3: Restore capacity for current service along  $path$ 
4: for all services matching  $req\_af$  do
5:   if all links in  $path$  have enough capacity then
6:      $select\_serv \leftarrow serv-1$ 
7:   end if
8: end for
9: if  $select\_serv \neq serv$  then
10:  Deduct capacity for new service along  $path$ 
11:  return  $select\_serv$ 
12: else
13:  Restore original capacity if no upgrade is possible
14:  return None
15: end if

```

TABLE II: MAF parameters

MAF Type	k_a	p_a	γ_a	ϕ_{I_a}	λ [Mbps]	τ [ms]
<i>MAF1</i>	1	2	0.3	0.99999	[5, 600]	[6, 55]
<i>MAF2</i>	0	3	0.5	0.9999	[10, 800]	[3, 50]
<i>MAF3</i>	1	4	0.6	1	[15, 650]	[9, 48]
<i>MAF4</i>	0	5	0.45	0.99995	[3, 700]	[7, 45]
<i>MAF5</i>	1	1	0.28	0.9999	[1, 300]	[6, 52]
<i>MAF6</i>	1	3	0.37	0.99998	[10, 580]	[2, 48]
<i>MAF7</i>	0	2	0.33	1	[6, 550]	[6, 52]

- *Optimal Joint Placement and Scheduling Algorithm (OJPSA)*: This approach is adapted from the optimization problem presented in [9]. In particular, only one MAF and not a chain is considered. In order to align their approach with our optimization problem, we relax the assumptions of time slots and instead add the routing constraints of the optimization problem in this paper. Additionally, a limited capacity for each link is added since the authors in [9] assume unlimited possible traffic on each link in contrast to the approach in this paper.
- *Random*: This simple strategy employs a method that randomly places demands, adhering to constraints such as minimum data rate, availability, and maximum end-to-end delay. However, the terminability property or other service levels than the optimal one are not considered.

All tests were conducted on a common KVM processor. The CPU configuration includes 8 physical cores and 8 logical processors, with the capability to utilize up to 8 threads. For solving the optimization problem, the Gurobi Optimizer [26] is used. In our optimization formulation, the weight W associated with rejecting a demand is set to 10. This weight shows a balanced distribution of acceptance ratio and level of service with respect to the data rates $\lambda_{h,a}$ used in our scenario. Higher values of W result in potentially more accepted MAFs on the cost of lower level of service. Vice versa, lower values of W lead to fewer accepted MAFs but with higher level of service. The used parameters for the demands, links and nodes

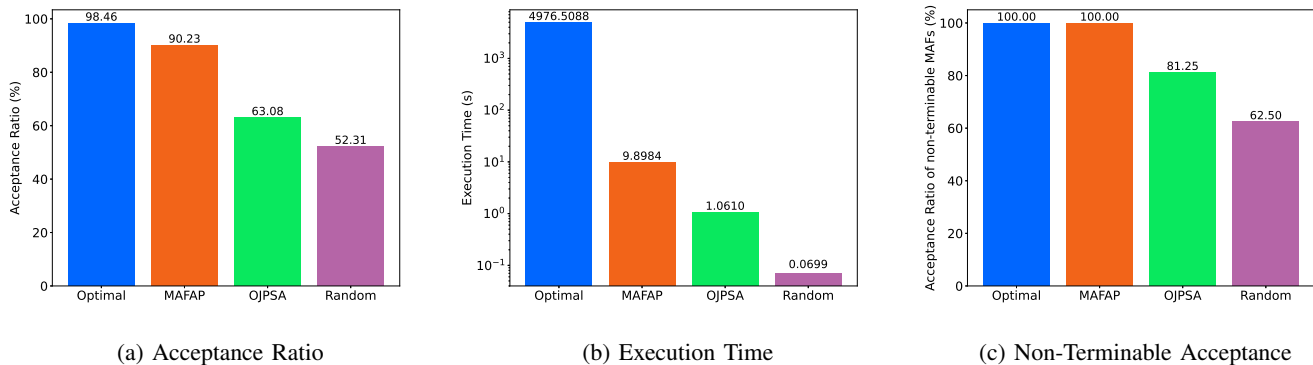


Fig. 1: Comparison of the results of the different approaches.

are summarized in Table I. We evaluate a varying number of demands to be placed based on the considered topology. Hereby, each demand uses one out of seven MAFs with different characteristics (see Table II). We selected these values deliberately to cover many possible medical applications with various different requirements, as described in Section III.

All results are obtained for a scenario where the topology consists of 12 nodes: 3 APs, 7 PUs, and 2 switches. 90 demand requests are to be placed within the network, where 18% include non-terminable MAFs, each offering 5 levels of data rate and processing delay. MAFAP evaluates 60 configurations, i.e., iterations, to enhance the acceptance ratio.

B. Evaluation

In the following, the obtained results for the described test setup are presented and discussed. Hereby, we focus on the overall performance and the impact of single parameters on it.

1) *Acceptance Ratio*: Fig. 1a displays the impact on the acceptance ratio for the four allocation methods. The *Optimal* method achieves 98.46%, closely followed by the *MAFAP* method with 90.23%. The *OJPSA* method reaches 63.08%, reflecting a decrease in demand fulfillment. The reason for that significant difference lies in the various possible service levels of our proposed approaches, whereas *OJPSA* always assume the best possible service. Finally, the *Random* method ranks lowest with an acceptance ratio of 52.31%.

2) *Computation Time*: Although the optimization method scores the highest in acceptance ratio, it also requires more resources and longer computation times, as shown in Fig. 1b. Considering practicality, *MAFAP* presents the best balance, offering a good compromise between performance and execution time. It lags slightly behind the optimization method regarding the acceptance ratio but considerably cuts down on resource use and processing time, making it preferable for real-world scenarios where resources and time are limited. Note that the execution time of *MAFAP* is almost $10\times$ higher than for *OJPSA* since more parameters are taken into account. However, to the best of our knowledge the relevance of this in medical use cases still needs to be investigated. In any case, the number of iterations in the heuristic can be adapted for faster execution time if needed.

3) *Number of Accepted Non-Terminable Demands*: Fig. 1c shows the number of accepted non-terminable demands in the network. It can be observed that *OJPSA* and *Random* do not place all crucial demands in the network. The reason lies in their design, which does not consider such a property. This renders them ineffective for scenarios where continuous operation is essential. In contrast, *Optimal* and *MAFAP* approaches place all of the non-terminable demands by design.

4) *Summary*: After evaluating the network performance, it is clear that *Optimal* and *MAFAP* significantly outperform *OJPSA*, improving request acceptance ratio by up to 35%. This improvement stems from their ability to adapt service levels to the available resources. Although *MAFAP* has a lower acceptance ratio than the *Optimal* solution, its reduced execution time renders it more practical for real-world applications. Additionally, our proposed methods adeptly accommodate the non-terminability feature, crucial in healthcare settings. Intuitively, the consumption of processing and networking resources of *Optimal* and *MAFAP* should be improved compared to *OJPSA* due to the more granular placement options provided with the different service levels. A thorough analysis of the resource consumption as well as the costs of the placement is deferred to future work.

VII. CONCLUSION

In this paper, we propose a new approach for placing MAFs within the network in medical scenarios, where the available computing and networking resources are potentially not sufficient to serve all requested demands. For this, we formulate an optimization problem as an ILP, considering the priority and different levels of service for each MAF. Additionally, we take the MAF terminability into account to ensure the placement of non-terminable MAFs regardless of their priority. Furthermore, we introduce a heuristic to solve such a problem faster. Our results show an increased acceptance ratio by up to 35% compared to baseline approaches. In the future, we will extend our approach by considering chains of MAFs with different priorities. Furthermore, we will evaluate our approach in a medical testbed and consider other scenarios with different comparison approaches.

REFERENCES

- [1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.
- [2] M. H. et al., "A secure and resilient 6G architecture vision of the german flagship project 6G-ANNA," *IEEE Access*, vol. 11, pp. 102 643–102 660, 2023.
- [3] S. Kolb, F. Jurosch, N. Kröger, F. Mehmeti, L. Bernhard, S. Speidel, W. Kellerer, and D. Wilhelm, "6G in Clinical Applications: Integrating New Network Approaches in Healthcare," *Current Directions in Biomedical Engineering*, vol. 10, no. 2, 2024.
- [4] F. Jurosch, N. Kröger, S. Kolb, F. Mehmeti, E. Martens, S. Speidel, W. Kellerer, D. Wilhelm, and J. Fuchtmann, "6G networks for the operating room of the future," *Progress in Biomedical Engineering*, vol. 6, no. 4, 2024.
- [5] N. Kröger, S. Kolb, F. Jurosch, D. Wilhelm, and W. Kellerer, "Processing modular application functions in future medical 6G radio access networks," in *2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2024, pp. 1–6.
- [6] D. Harutyunyan, N. Shahriar, R. Boutaba, and R. Riggio, "Latency-aware service function chain placement in 5G mobile networks," in *Proc. of IEEE NetSoft*, 2019.
- [7] P. K. Thiruvassagam, A. Chakraborty, A. Mathew, and C. S. R. Murthy, "Reliable placement of service function chains and virtual monitoring functions with minimal cost in softwarized 5G networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, 2021.
- [8] N. Promwongsa, A. Ebrahimzadeh, R. H. Glitho, and N. Crespi, "Joint VNF placement and scheduling for latency-sensitive services," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, 2022.
- [9] A. Hentati, A. Ebrahimzadeh, R. H. Glitho, F. Belqasmi, and R. Mizouni, "Remote robotic surgery: Joint placement and scheduling of VNF-FGs," in *Proc. of IEEE CNSM*, 2022.
- [10] V. Eramo, M. Ammar, and F. G. Lavacca, "Migration energy aware reconfigurations of virtual network function instances in NFV architectures," *IEEE Access*, vol. 5, pp. 4927–4938, 2017.
- [11] Y. Liu, Y. Lu, X. Li, Z. Yao, and D. Zhao, "On dynamic service function chain reconfiguration in IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10969–10984, 2020.
- [12] N. Gholipour, H. Saeedi, N. Mokari, and E. A. Jorswieck, "E2E QoS guarantee for the tactile internet via joint NFV and radio resource allocation," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1788–1804, 2020.
- [13] H. Yu, F. Musumeci, J. Zhang, M. Tornatore, L. Bai, and Y. Ji, "Dynamic 5G RAN slice adjustment and migration based on traffic prediction in WDM metro-aggregation networks," *Journal of Optical Communications and Networking*, vol. 12, no. 12, pp. 403–413, 2020.
- [14] F. Malandrino, C. F. Chiasserini, G. Einziger, and G. Scalosub, "Reducing service deployment cost through VNF sharing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2363–2376, 2019.
- [15] S. Kianpisheh and R. H. Glitho, "Joint admission control and resource allocation with parallel VNF processing for time-constrained chains of virtual network functions," *IEEE Access*, vol. 9, 2021.
- [16] K. Akahoshi, F. He, and E. Oki, "Service deployment model with virtual network function resizing," *Proc. of IEEE GLOBECOM*, 2021.
- [17] A. Varasteh, B. Madiwalar, A. V. Bemten, W. Kellerer, and C. Mas-Machuca, "Holu: Power-aware and delay-constrained VNF placement and chaining," *IEEE Transactions on Network and Service Management*, vol. 18, pp. 1524–1539, 2021.
- [18] V. Eramo and T. Catena, "Application of an innovative convolutional/LSTM neural network for computing resource allocation in NFV network architectures," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2929–2943, 2022.
- [19] J. Chen, X. Cheng, J. Chen, and H. Zhang, "A lightweight SFC embedding framework in SDN/NFV-enabled wireless network based on reinforcement learning," *IEEE Systems Journal*, vol. 16, no. 3, 2022.
- [20] K. Ali and M. Jammal, "Proactive VNF scaling and placement in 5G O-RAN using ML," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 174–186, 2024.
- [21] M. Golkarifard, C. F. Chiasserini, F. Malandrino, and A. Movaghar, "Dynamic VNF placement, resource allocation and traffic routing in 5G," *Computer Networks*, vol. 188, 2021.
- [22] A. Mohamad and H. S. Hassanein, "PSVShare: A priority-based SFC placement with VNF sharing," in *Proc. of IEEE NFV-SDN*, 2020.
- [23] S. Kolb, A. Madden, N. Kröger, F. Mehmeti, F. Jurosch, L. Bernhard, W. Kellerer, and D. Wilhelm, "6G in medical robotics: Development of network allocation strategies for a telerobotic examination system," *International Journal of Computer Assisted Radiology and Surgery*, 2024.
- [24] M. Wang, B. Cheng, S. Wang, and J. Chen, "Availability- and traffic-aware placement of parallelized SFC in data center networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, 2021.
- [25] R. Guerzoni, Z. Despotovic, R. Trivisonno, and I. Vaishnavi, "Modeling reliability requirements in coordinated node and link mapping," in *Proc. of IEEE SRDS*, 2014.
- [26] "Gurobi optimizer," <https://www.gurobi.com/solutions/gurobi-optimizer/> [Accessed: (23/03/2024)].