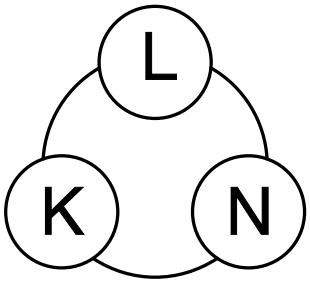

Technische Universität München
Lehrstuhl für Kommunikationsnetze
Prof. Dr.-Ing. Wolfgang Kellerer

Master‘s Thesis

Processing Prioritization of Application Functions in Future 6G Medical RANs

Author:	Gattulli, Giuseppe
Address:	St. G. Mameli 25b
	70037 Ruvo di Puglia
	Italy
Matriculation Number:	03770646
Supervisors:	M.Sc. Nicolai Kröger, Dr. Ph.D. Fidan Mehmeti
Begin:	09. October 2023
End:	01. March 2024

With my signature below, I assert that the work in this thesis has been composed by myself independently and no source materials or aids other than those mentioned in the thesis have been used.

München, 01.03.2024	
Place, Date	Signature

München, 01.03.2024	
Place, Date	Signature

Kurzfassung

Die Einführung von 6G in Kommunikationsnetzwerken markiert einen entscheidenden Fortschritt hin zu beispielloser Konnektivität und technologischem Fortschritt, mit der In-Netzwerk-Berechnung als Kernstück. Dieser innovative Ansatz zur Berechnung wird voraussichtlich die Latenz drastisch reduzieren und die Gesamteffizienz des Systems erhöhen. Digitale Zwillinge sind der Schlüssel zu dieser Evolution und bieten virtuelle Darstellungen physischer Systeme zur Optimierung der Netzwerkfunktionalität. Doch die Integration von In-Netzwerk-Berechnung und digitalen Zwillingen bringt Hindernisse mit sich, insbesondere bei der Verwaltung von Rechenressourcen über verschiedene Anwendungen hinweg; eine Komplexität, die in kritischen Sektoren wie Telemedizin und intelligenten Operationen zunimmt.
6G steht kurz davor, die Medizin durch Innovationen wie Fernuntersuchungswerkzeuge und tragbare Patientenüberwachungssysteme zu transformieren. Das Problem der Ressourcenkonkurrenz steht jedoch im Vordergrund und könnte die zuverlässige Übertragung kritischer Gesundheitsdaten beeinträchtigen. Eine intelligente Strategie zur Ressourcenzuweisung und Priorisierung von Rechenressourcen ist unerlässlich, um die Zuverlässigkeit lebenswichtiger Dienste zu erhalten, ohne die Netzwerkqualität zu beeinträchtigen.
Die Netzwerkfunktionsvirtualisierung (NFV) wird als eine Schlüsselstrategie zur Überwindung dieser Herausforderungen identifiziert und ermöglicht eine flexible Ressourcenzuweisung und effiziente Verwaltung von Anwendungsfunktionen (AFs), um die Belastung geteilter Ressourcen zu verringern. Dieser Ansatz zielt darauf ab, die Netzwerkresilienz und Effizienz zu steigern und fördert ein neues Paradigma für den Umgang mit unterschiedlichen Netzwerkanforderungen, die Priorisierung kritischer Dienste und das umsichtige Management der Leistung nicht wesentlicher Dienste.
Diese Arbeit schlägt eine neuartige Methode zur Netzwerkressourcenzuweisung vor, mit Schwerpunkt auf dem Gesundheitskontext. Sie erkennt die Notwendigkeit an, bestimmte AFs aufgrund ihrer kritischen Natur in das Netzwerk zu integrieren. Diese Methode unterscheidet sich dadurch, dass sie verschiedene Serviceniveaus basierend auf verfügbaren Ressourcen erlaubt, neben einer klaren Priorisierung für die Netzwerkeinbindung. Zur Validierung des vorgeschlagenen Konzepts wurden zwei Modelle entwickelt: ein mathematisches Modell, das eine optimale Lösung bietet, und ein heuristisches Lokalsuchmodell, das für die praktische Anwendung ohne die umfangreiche Verarbeitungszeit von Optimierungsproblemen konzipiert ist. Diese Modelle werden mit einem ähnlichen Ansatz in der bestehenden Literatur verglichen, um ihre Fähigkeit hervorzuheben, die Nuancen der Dienstleistungsvielfalt und die inhärenten Eigenschaften bestimmter Anwendungen (wie die Beendigungseigenschaft) zu erfassen.
Wie die Ergebnisse zeigen, erreicht unsere Methode eine Steigerung der Akzeptanzrate um bis zu 40% gegenüber der zuvor etablierten Lösung.

Schlüsselwörter: 6G, RAN, NFV, Telemedizin, Ressourcenzuweisung

Abstract

The deployment of 6G in communication networks signifies a pivotal advancement towards unparalleled connectivity and technological advancement, with in-network computing at its core. This innovative approach to computing is anticipated to drastically reduce latency and enhance overall system efficiency. Digital twins are key to this evolution, offering virtual representations of physical systems to optimize network functionality. Yet, the integration of in-network computing and digital twins introduces obstacles, especially in managing computing resources across diverse applications; a complexity that escalates in critical sectors like telemedicine and intelligent operations.

6G stands to transform medicine through innovations like remote examination tools and portable patient monitoring systems. However, the issue of resource contention looms large, potentially impeding the dependable transmission of crucial health data. An intelligent strategy for resource allocation and prioritization of computational resources is essential to maintain the reliability of vital services without degrading network quality.

Network Function Virtualization (NFV) is identified as a key strategy for overcoming these challenges, enabling flexible resource allocation and efficient management of application functions (AFs) to alleviate the strain of shared resources. This approach aims to boost network resilience and efficiency, fostering a new paradigm for handling varying network demands, prioritizing critical services, and judiciously managing the performance of non-essential services.

This thesis proposes a novel method for network resource allocation, with a focus on the healthcare context. It acknowledges the need for certain AFs to be integrated into the network due to their critical nature. This method sets itself apart by allowing for different levels of service based on available resources, alongside clear prioritization for network inclusion. To validate the proposed concept, two models have been developed: a mathematical model that provides an optimal solution, and a heuristic local search model designed for practical application without the extensive processing time of optimization problems. These models are compared against a similar approach in the existing literature to highlight their ability to capture the nuances of service diversity and the inherent characteristics of certain applications (such as termination property).

As demonstrated in the results, our method achieves a boost in acceptance ratio by up to 40% over the previously established solution.

Keywords: 6G, RAN, NFV, Telemedicine, Resource Allocation

Contents

Chapter 1

Introduction

In the ever-evolving landscape of communication networks, the imminent arrival of 6G foreshadows a new era of connectivity and technological advancement. At the heart of this progression lies the pivotal concept of in-network computing, a paradigm that promises to significantly reduce latency and enhance overall system performance [1]. In this dynamic environment, the integration of digital twins emerges as a key catalyst, particularly in applications proximate to end-users. They are virtual replicas that mirror physical entities or processes, introducing a transformative dimension to the functionality of the network [2].

The evolution of Open Radio Access Network (O-RAN) architecture, especially with the transition from 5G to the anticipated 6G, brings a significant structural transformation with the disaggregation of functions into the Radio Unit (RU), the Distributed Unit (DU), and the Centralized Unit (CU) [1]. This division is designed to streamline network operations and enhance data processing efficiency, which is crucial for in-network computing.

While the promise of in-network computing and digital twins holds tremendous potential, it also introduces a set of challenges, chief among them being the efficient sharing of processing resources. These resources, essential for the seamless operation of diverse applications, are commonly shared among numerous functions within the network. This shared resource model, while fostering versatility and adaptability, simultaneously poses a problem [3]. A challenge that becomes particularly pronounced in critical domains such as telemedicine and smart operations.

The rise of telemedicine, boosted by 5/6G technology, has the potential to significantly alter the landscape of healthcare delivery through the deployment of Private 5/6G networks. These networks offer more than just an upgrade to telecommunications; they redefine healthcare delivery and management through high reliability, fast data processing, and advanced security.

Private 5/6G networks lead this technological shift by providing exceptional bandwidth and low-latency connections, crucial for the deployment of the Internet of Medical Things

(IoMT), AI diagnostics, and personalized care adopting a more dynamic approach to patient care [4].
However, the very processing resources that make these advancements possible are subject to contention among myriad applications, potentially leading to performance degradation. In the realm of telemedicine, where timely and accurate data transmission is paramount, the stakes are exceptionally high.
This dilemma brings to the forefront an important question: How can we harness the benefits of in-network computing and digital twins without compromising the performance of mission-critical applications, especially in these sensitive fields? The answer lies in a nuanced understanding of resource allocation strategies and judicious prioritization of processing power.
Among these challenges, the importance of Network Function Virtualization (NFV) becomes crucial. NFV enables the decoupling of network functions from dedicated hardware, allowing for dynamic and flexible allocation of resources [5]. This virtualization paradigm offers a strategic solution to the problem of resource contention by facilitating the creation and management of Application Functions (AFs).
Our study explores new strategies in resource allocation, diverging from past research that viewed performance drops as penalties [6]. We see reducing quality in less critical services as a strategic choice to optimize network-wide application support, in collaboration with *MITI* reaserch group (an interdisciplinary team at *Klinikum rechts der Isar* of the Technical University of Munich) [7]. Their insights into the medical sector highlighted the essential requirement for certain applications to remain continuously operational on the network, given the critical processes they facilitate. We focus on ensuring essential applications remain operational, adjusting service levels based on resource availability. This approach aims to maximize support for high-priority applications while meeting the specific needs of all processes, enhancing network efficiency.

In particular, our main contributions are:

- Present a mathematical model designed to address the resource allocation challenge, aiming to enhance service quality for high-priority requests and ensure the fulfillment of critical demands.
- Identify a heuristic algorithm capable of solving the resource allocation problem defined by the mathematical model but with reduced computational time, making it suitable for real-world application scenarios.
- Assess and compare the effectiveness of the suggested methods against the model developed by *Hentati et al.*, focusing on performance outcomes.

The structure of this thesis evolves in the following order:

- Chapter 2 introduces some background on future communication networks, higlighting the importance of NFV and resource sharing, and E-health, in particular on how Tactite Internet represents the future of telemedicine. Additionally, it uncovers

some existing solutions to provide the necessary knowledge to understand the next chapters.

- Chapter 3 is structured into four main sections. The initial section delves into the description of the optimization problem being proposed. The subsequent section is dedicated to adapting the optimization problem outlined in [8]. The third section details the implementation of the heuristic algorithm that has been developed. Finally, the concluding section presents a comparative analysis of these different solutions.
- Chapter 4 concludes this work, summarizing the results and suggesting possible future directions.

Chapter 2

Background

This section provides a background on the evolution of future communication networks, emphasizing the critical roles of Network Functions Virtualization (NFV) and resource sharing. It specifically highlights the significance of these technologies in the context of E-health, focusing on how the Tactile Internet is poised to revolutionize telemedicine.
Moreover, it delves into current solutions and advancements in this field, offering essential insights and knowledge foundational for comprehending the subsequent chapters. This exploration not only showcases the importance of innovative network technologies and their application in healthcare but also sets the stage for a deeper understanding of the emerging trends and challenges in telemedicine facilitated by the Tactile Internet.

2.1 Future Communication Networks

The massive use of Internet of Things (IoT) applications will be key to the future of smart cities. This has significantly influenced the global development of fifth generation (5G) communication systems. Indeed, 5G must address challenges arising from the growing demand for bandwidth-intensive applications and the increase of smart devices within urban networks. The next generation of communications systems Beyond 5G (B5G) promises to overcome these obstacles by offering better network customization to meet specific quality-of-service (QoS) and quality-of-experience (QoE) needs. B5G Network Slicing (NS) further improves its 5G predecessor by incorporating features such as enhanced scalability, lower latency, and increased NS efficiency [9]. B5G excels in network management, especially for simultaneous multi-service traffic, satisfying diverse application requests and optimizing resource allocation. Additionally, B5G NS aims to enhance its capabilities with better inter-slice isolation and support for a larger user base with stringent QoS expectations. This includes the ability to more easily deploy applications such as holographic communications, telesurgery or augmented/virtual reality without compromising QoS and QoE, demonstrating B5G's ability to efficiently handle the demands of emerging applications.
Looking ahead, the sixth generation (6G) network has even more ambitious goals, with

the aim of further revolutionizing the communication landscape [10]. 6G is expected to introduce breakthroughs in speed, latency, and connectivity, pushing the boundaries of digital communication and enabling technologies such as Five-Sense Communications and the Tactile Internet (TI), which integrates human senses into communication systems. 6G networks will likely leverage artificial intelligence (AI) to a greater extent, enabling more autonomous, intelligent network operations and management. This could lead to more personalized and efficient services, with networks capable of adapting in real-time to user needs and environmental conditions. This multi-layered approach aims to ensure seamless global coverage and support for new types of services, including ultra-reliable low-latency communications (URLLC) and massive machine-type communications (mMTC). The main Key Performance Indicators (KPIs) that will be the focus of 6G are depicted in Figure 2.1.

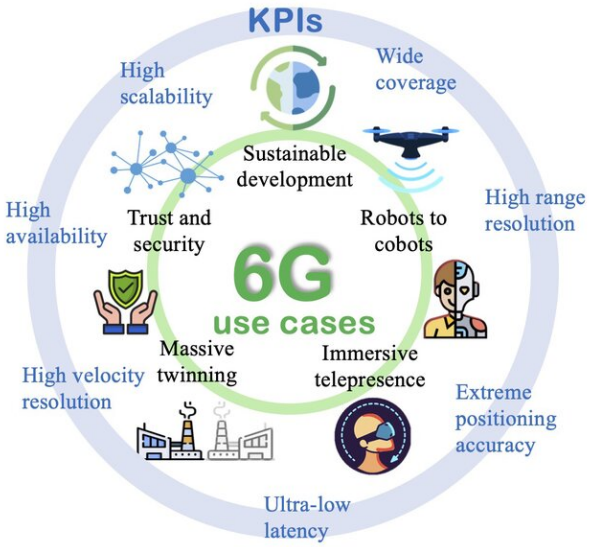

Figure 2.1: Use case families for 6G [11].

2.1.1 Network Function Virtualization

In the context of 5G and the forthcoming 6G Radio Access Networks (RAN), Network Function Virtualization (NFV) plays a pivotal role by introducing a level of flexibility, scalability, and efficiency previously unattainable with traditional RAN architectures. By enabling the virtualization of network functions, which can then be deployed on general-purpose hardware, NFV marks a significant shift in how network services are provided (as referenced in [9] and illustrated in Figure 2.2). This transformation is crucial for 5G/B5G/6G RANs as it supports the ultra-low latency and high reliability required by next-generation applications such as autonomous driving, remote healthcare, and immersive augmented and virtual reality experiences.
Using NFV within the RAN, network operators can dynamically allocate and manage

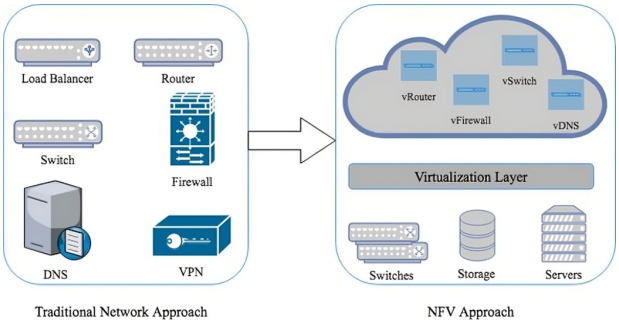

Figure 2.2: Difference between traditional network approach and NFV approach [12].

resources based on real-time demands, ensuring that critical applications receive the bandwidth, processing power, and low latency they require. This adaptability is essential to meet the diverse and stringent performance criteria of 5G and 6G networks, where the ability to quickly deploy and scale network functions can significantly enhance service quality and user experience. Moreover, the role of NFV in virtualizing RAN functions facilitates a more cost-effective and agile network infrastructure, enabling operators to respond more swiftly to market changes and new service requirements [13].

2.1.2 Resource Sharing

Today's wireless technologies are evolving to offer IP connectivity that enables quicker Internet access, a range of multimedia applications, and various services on-demand for the end-user. With the rollout of 5G networks, there has been a significant surge in consumer expectations and a 1000-fold increase in data demand. This surge necessitates sophisticated management, monitoring, and a degree of programmability and flexibility to achieve the desired performance levels and high data rates at reduced costs. By 2023, the Internet was estimated to have 5.3 billion users, representing 66% of the global population, along with 29.3 billion networked devices. Furthermore, global 5G subscriptions are expected to exceed 5.3 billion by the year 2029 [14, 15]. In the domain of cellular networks, the fluctuating wireless conditions and the limited availability of shared spectrum necessitate equitable resource allocation and scheduling by the base station. These resources, managed by the base station (BS), which oversees the flow queue of incoming requests and schedules them at the time of deployment, have traditionally been allocated in a static manner, leading to suboptimal use of resources. To improve efficiency, resources are now virtually allocated into "slices" to meet specific service demands, requiring virtualization of resources at the BS to be managed frequently due to the dynamic nature of cellular network conditions [16].
As we look beyond 5G towards the next generation of wireless technology, 6G is poised to revolutionize the landscape even further. 6G aims to build on the foundation laid by 5G, pushing the boundaries of connectivity with potentially terahertz (THz) frequencies, enabling even faster data speeds, lower latency, and higher reliability. To achieve these

ambitious goals, 6G will require innovations in network infrastructure, including intelligent resource management and advanced AI algorithms for network optimization. This next-generation technology aims to overcome the challenges encountered by 5G, including achieving higher data rates, reducing latency, and developing more adaptable, scalable, and intelligent network architectures, including addressing RAN resource allocation issues. With 6G, the vision is to create a more integrated, intelligent, and user-centric network that can support the ever-growing demands of the digital era [17].

2.2 E-health

E-health, or electronic health, represents a broad range of healthcare practices supported by electronic processes and communication. It emerged as a means to enhance the efficiency, quality, and delivery of healthcare services through the use of information and communication technologies (ICT). This concept encompasses a variety of applications, including electronic health records (EHRs), telemedicine, health information systems, mobile health (mHealth) applications, and electronic prescribing [18]. E-health initiatives are designed to enhance healthcare accessibility, particularly in remote areas, and to boost patient engagement by offering tools for health monitoring and management. Additionally, they aim to streamline the exchange of health information among healthcare providers, aiding in decision-making and enhancing health outcomes. Telemedicine stands out as a key application of IoT in healthcare, and it must prioritize ease of use, accessibility, and flexibility. The hardware design should appeal to patients and professionals alike, and integration with existing healthcare systems should be straightforward. Healthcare organizations should use IoT to create effective, affordable telemedicine solutions [19].
The development of e-health has been driven by advancements in technology, the increasing demand for convenient and efficient healthcare services, and the need for healthcare systems to reduce costs while maintaining or improving the quality of care. With the potential to reach underserved populations and streamline healthcare processes, e-health has become a key component in the transformation of healthcare delivery worldwide. It supports public health measures, clinical care, healthcare administration, and education, while adhering to standards and regulations to ensure privacy and security of patient information. As the digital health landscape continues to evolve, e-health is at the forefront of innovations that aim to address global health challenges, improving health literacy, and empowering individuals with the tools they need to lead healthier lives [20].

Tactile Internet

As e-health initiatives harness the power of ICT to revolutionize healthcare delivery, they pave the way for the integration of more immersive and interactive technologies like those offered by the Tactile Internet (TI). The convergence of these domains promises to enrich the healthcare ecosystem with unparalleled capabilities for remote diagnosis, treatment, and patient care. Imagine a scenario where telemedicine evolves to incorporate the real-

time haptic feedback of the TI. This fusion not only extends the reach of healthcare services but also deepens the level of engagement and effectiveness of medical interventions.
The IEEE P1918.1 standard working group defines the TI as "A network, or a network of networks, for remotely accessing, perceiving, manipulating, or controlling real and virtual objects or processes in perceived real-time" [21]. This definition marks a shift in the evolution of Internet from a platform primarily for exchanging text, audio, and video data to one that facilitates the real-time exchange of haptic control messages. This evolution signifies a leap from the early text-centric Internet to a multimedia and mobile Internet, then to the IoT, and now towards an era emphasizing haptic communication. The TI introduces groundbreaking applications, such as remote surgeries, allowing doctors to not only see and hear but also feel the environment of a distant patient in real-time, thereby greatly expanding the possibilities for interaction and communication over the Internet [22].
The term "Tactile Internet" was first coined in 2014 by Fettweis, presenting it as a technology that permits the control and manipulation of real and virtual objects over the Internet with minimal latency [23]. Subsequently, in March 2016, the IEEE P1918.1 standards and working group were set up to define the architectural framework of the TI. This framework characterizes the TI as a network tailored for real-time remote interactions with objects and processes. It highlights applications such as remote robotic surgery, autonomous driving, and virtual reality, as discussed in [21]. An illustration of this concept is provided in Figure 2.3. These applications require exceptionally low latency, alongside high reliability and security, to function effectively and safely. while the latency requirements of TI applications may vary from ¡ 10 ms up to tens of milliseconds, the TI targets an ultra-low end-to-end round-trip latency of 1 ms, which necessitates the physical closeness of applications to their endpoints to fulfill these rigorous demands [24]. Under ideal conditions, the maximum distance between endpoints to achieve this latency is restricted to 150 km, considering the speed of light limitation (300 km/ms) on propagation delay. However, practical constraints often diminish this distance due to various delays like queueing, channel access, and transmission delays. The reliability requirements for the TI also vary by application, starting from a failure rate of 10^{-3}, with the most critical applications, such as telesurgery, demanding a reliability up to 10^{-7} [25].

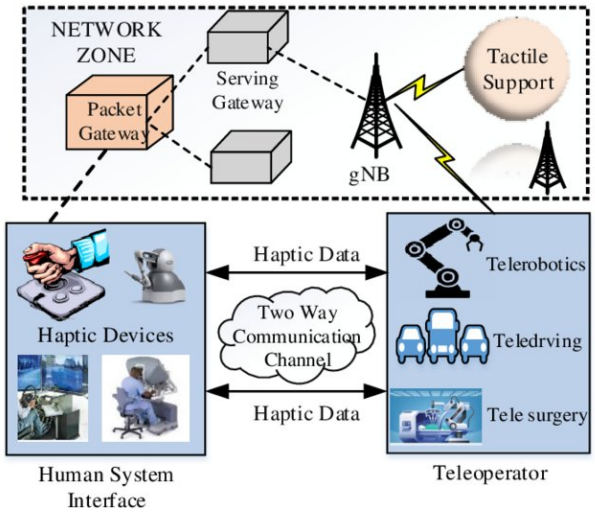

Figure 2.3: Example of Tactile Internet architecture [26].

2.3 Related works

Numerous studies in the literature propose and examine methods to determine whether a specific number of requests can be satisfied given constraints on network resources. A compilation of potential solutions to this issue is introduced and elaborated on in the following:

VNF Migration One notable study is presented in [27], where the focus is on the joint service function placement and resource allocation problem. This work examines strategies to meet end-to-end (E2E) delay requirements by migrating applications installed on network nodes. Applications with more critical demands are positioned nearer to the end user (at edge nodes), while less critical applications are located in the core network. However, this can lead to a service disruption that can be problematic in the medical context.
The same idea of virtual network function (VNF) migration is also proposed in [28], here the objective is to minimize the total energy consumption, which includes both consolidation and migration energies in response to variations in the intensity of the service function chain (SFC) request.
The study in [6] showcases the optimization of 5G RAN slicing within wavelength division multiplexing (WDM) metro-aggregation networks through a strategy based on traffic prediction. This method seeks to minimize service degradation penalties and lower the volume of migrated traffic by dynamically adjusting and migrating RAN slices in response to anticipated traffic changes.
In [29], the VNF reconfiguration problem is addressed by determining the migration of VNF and rerouting service paths between IoT networks and clouds that optimize the reconfiguration cost and the resource consumption, while satisfying the resource and QoS constraints.

Shared VNFs In [30], a different approach is presented, the concept of parallel VNF processing is introduced, allowing different requests that require the same type of VNF to share a single virtual machine (VM) hosting that VNF, aiming to meet stringent time constraints. The study proposes a stochastic model where the execution of the VNF chain is represented as a queuing network. The primary objective is to maximize the profit for the network provider, considering both the acceptance rate of requests and the power consumption.

VNFs Resizing This method, as introduced in [31], allows multiple VNF instances on a single host to share computing resources, unlike traditional models that allocate fixed resources to each instance. Computing capacity is dynamically allocated to the VNF currently processing flows, ensuring optimal resource utilization without simultaneous processing by multiple VNFs. The model assumes instant resizing of VNFs, enabling flexible and efficient management of computing capacity among different VNFs as required, without any delay.
Ghaznavi et al. [32] proposes an horizontal scaling of VNFs, operating under the assumption of a single VNF instance-type. The goal is minimizing operational costs in providing VNF services. This approach involves a balance, considering both bandwidth and host resource consumption together to achieve an optimal trade-off.
An elastic VNF orchestration framework for Open RAN (O-RAN) is introduced in [33] work to support the demanding latency, reliability, and bandwidth needs of 5G services. The framework employs machine learning (ML) for dynamic scaling based on traffic forecasting and reinforcement learning (RL) for optimizing VNF placement. These approaches aim to enhance flexibility, reduce operational costs, and improve resource utilization while adhering to Service Level Agreements (SLAs).

Power Aware VNF placement Holu, as introduced in [34], is a fast heuristic framework developed to solve the power-aware and delay-constrained joint VNF placement and routing problem within SFCs. SFCs frequently encounter challenges related to the under-utilization and over-provisioning of physical machines (PMs) and network switches, resulting in inefficient power consumption. Holu's objective is to enhance power efficiency in deploying SFCs by reducing the number of necessary PMs and network devices. It addresses this challenge by breaking it down into two sequential sub-problems: firstly, a centrality-based VNF placement on PMs, and secondly, a routing problem that assigns delay budgets and finds the least-cost path that meets delay constraints.

VNF placement with AI The proposed framework in [35] introduces an advanced resource allocation strategy for virtualized network environments by leveraging AI, specifically integrating a Convolutional/Long Short Term Memory neural network, to predict and allocate processing capacities efficiently. Central to this approach are two key processes: a monitoring procedure that periodically assesses the required processing capacities of virtual instances and a subsequent integrated allocation/prediction procedure

that evaluates future capacity needs based on the monitored data. This method aims to minimize the costs associated with resource allocation and QoS degradation, ensuring optimal resource utilization.
The article [36] introduces a NFV framework that leverages reinforcement learning for efficient SFC embedding in dynamic wireless networks, aiming to reduce end-to-end delays and improve SFC acceptance ratios. By modeling the problem as a Markov decision process and implementing a Q-learning strategy, this approach reduces costs of the network infrastructure while enhancing service quality.

VNF placement in Tactile Internet context *Gholipoor et al.* [37] presents a novel joint strategy for managing both radio and NFV resource allocation in heterogeneous networks, tackling all sources of delay within the framework of the TI, which seeks to achieve ultra-low E2E delays. It develops an online heuristic algorithm specifically for NFV resource allocation, designed to minimize costs while meeting stringent E2E delay criteria, and demonstrates considerable savings in network expenses.
The proposed solution in [8] investigates the allocation of resources in a 5G-supported TI framework for remote robotic surgery, utilizing NFV. It focuses on the optimal placement and scheduling of application components for haptic and video data in such systems. The research goal is to optimize infrastructure cost savings while increasing the rate of demand acceptance. Moreover, it demonstrates that segmenting application traffic among several VNF-forwarding graphs (VNF-FGs), each designed for different QoS needs, substantially improves cost savings and the capacity of the system over using a single, rigid QoS approach.
In general, most of these studies overlook the possibility of providing variable service levels to these requests, adapting the service to meet the needs of the demand and adapting to the available network resources. Furthermore, a still unexplored aspect is the need to allocate some AFs within the network due to the specific services they offer.

The research presented by [8] stands out as a basis within the context of this thesis, primarily because of its similarity to the current study. This similarity is not merely superficial; it extends to the core methodologies and objectives pursued by both studies, making *Hentati et al.*'s work an ideal benchmark against which the present research can be measured. Their research focuses on optimizing the placement and scheduling of application components in an NFV-based system for remote robotic surgery, with the goal of ensuring the round-trip latency of haptic data stays within acceptable limits. Additionally, it aims at maintaining system reliability to prevent any discrepancies between the surgeon and patient that could arise from data loss. By adopting and slightly modifying this proposed model, this thesis aims to facilitate a direct comparison between the two approaches. Beyond the contributions outlined by [8], our work explores the possibility of offering a more diverse range of services tailored to the dynamic availability of network resources, thereby improving the ability of the network to accommodate various service requests.
Furthermore, a key element of the model proposed by *Hentati et al.* is its lack of a mecha-

nism for managing requests that cannot be terminated. Within their system, this omission could lead to such requests being omitted from integration into the network. This approach to prioritization reveals a fundamental divergence in the treatment of service requests between the two studies. It highlights a critical aspect of network resource management, suggesting that the efficiency and effectiveness of utilizing network resources can be significantly influenced by how service requests, especially non-terminable ones, are managed. Such differences in approach could have profound implications on the overall performance and resource optimization strategies of network service providers, underscoring the importance of the comparative analysis undertaken in this thesis.

Chapter 3

Implementation/Results

This chapter delves into the intricacies of the optimization problem we are addressing. It starts by establishing a solid base, providing an in-depth explanation of the optimization problem in question. Following this initial discussion, the narrative moves on to scrutinize how the optimization problem, as detailed in [8], integrates into our present situation. The narrative then progresses to a thorough exploration of the development and deployment of a heuristic algorithm, detailing the rationale and process behind its formulation. The chapter concludes with a comprehensive comparative study, contrasting the different solutions discussed throughout. This final part not only emphasizes the unique features and advantages of each method but also sheds light on avenues for achieving more sophisticated and effective solutions in optimization research.

3.1 Optimization Problem

3.1.1 Problem Modeling

Overview

The network in the context of Radio Access Networks (RANs) is depicted as a complex system, designed as a substrate mesh network symbolized as $G = (N, L)$. N represents various physical entities that include Access Points (APs), Distributed Units (DUs), Control Units (CUs), Switches, and Core components, as illustrated in Figure 3.1. These entities are interconnected through a series of physical links L, forming a robust infrastructure for data transmission and communication.

Nodes Characteristics

At the heart of this network, certain nodes, specifically DUs and CUs, are equipped with the capability to accommodate and deploy Application Functions (AFs), thereby marking them as a specialized subset, denoted as $N^{'} \subset N$. This distinction is not merely structural

but functional, as these nodes come with significant resource capacity, particularly in terms of computational resources, indicated by $c^{n'}_{cpu}$ for a given node $n' \in N'$.
Additionally, the operational effectiveness of each node within the RAN is indicated by its availability, av_n, which quantifies the likelihood of the node performing effectively upon demand.

Links Characteristics

A connection between any two nodes, u and v, is represented as (u, v). One of the primary constraints on a link is its data rate capacity, symbolized as $c^{(u,v)}_{\pi}$, which determines the maximum amount of data that can be transmitted between the two nodes. This capacity is crucial for ensuring that the network can handle the volume of data traffic required by various applications and services without congestion.
In addition, each link also has an associated propagation delay, denoted as $d_{(u,v)}$, which represents the time it takes for data to travel from one node to the other across the link. This delay is an important factor in the overall performance of the network, as it affects the latency experienced by end-users and can impact the effectiveness of real-time applications and services.

Appication Functions Characteristics

Within the network, AFs, denoted as $a \in A$. The notation $|A|$ signifies both the size of the set and the total functions provided by the medical site. Each AF is defined by its terminability k_a and priority p_a, with k_a indicating whether an AF is critical (1) or non-essential (0), and p_a ranking its importance on an ascending scale. These attributes, terminability and priority, independently influence the network resource management, ensuring critical operations are maintained and efficiently prioritized.
Furthermore, each AF has a specific demand for computational resources (γ_a) from the node on which it is installed, and its availability (av_{I_a}) reflects its operational readiness.
Key performance indicators, data rate ($\lambda_{h,a}$) and latency ($\tau_{h,a}$), are adjusted according to the service levels set (H), facilitating a wide range of application needs. This proactive QoS regulation allows the network to strategically allocate resources, optimizing service delivery for various applications based on urgency and performance requirements, thereby transforming the network into an intelligent system that manages demands efficiently, enhancing user experience and network performance.

Demands Characteristics

The network supports a wide array of application needs, each with specific performance requirements such as a minimum data rate (f_{π_d}), to ensure data transfers at the needed speed; a maximum delay limit (f_{to_d}), to cap the data travel time without affecting performance; and a required availability level (av_{R_d}), for reliable access to the services of the network. Demands are uniquely identified by their source node (s_d), destination node (t_d),

and the specific AF needed (r_d), facilitating precise resource and service provisioning. This level of detail in request characterization allows for a highly tailored approach to resource and service allocation, ensuring that each application receives the most efficient and effective distribution of the network capabilities.

Optimization Approach

The optimization problem under consideration is categorized as an Integer Linear Programming (ILP) problem. This classification indicates that the problem is structured around linear relationships among decision variables that are constrained to take integer values. The necessity for decision variables to be integers is intrinsically connected to the problem's relevance to resource allocation processes. In fact, the notion of fractional variables has no meaning in this context, since resource allocation requires integer units. Allocating only portions of the demand is unfeasible, so all variables must represent complete quantities.

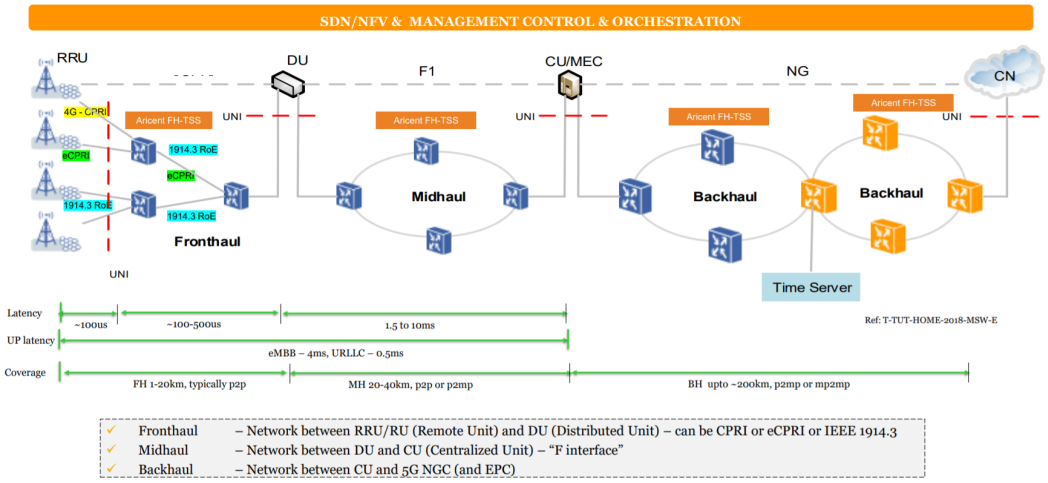

Figure 3.1: Example of Open RAN F1 interface between the DU and CU [38].

Sets Definitions

Table 3.1 gives a comprehensive summary of the different collections that play a crucial role in the optimization problem. These sets are essential for understanding the parameters, variables, and constraints that need to be considered when finding the best solution to the problem at hand.

Set	Explanation
$G = (N, L)$	RAN substrate network with physical nodes (N) and physical links (L)
N	Set of physical nodes in the RAN
$N^{'} \subset N$	Set of physical nodes in the RAN that includes only DUs and CUs
$N^{''} \subset N$	Set of physical nodes in the RAN that includes only switches
L	Set of physical links connecting nodes in the RAN
A	Collection of AFs offered by the medical center in the RAN
D	Set of demands
H	Set of level of service offered

Table 3.1: Sets definition for the optimization problem

Parameters Definitions

Table 3.2 presented here summarize the various parameters that are fundamental for the optimization problem.

Parameter	Explanation
$c_{cpu}^{n'}$	Computational capabilities of node n' in terms of CPU
av_n	Availability of node n
$c_{\pi}^{(u,v)}$	Offered data rate of link (u, v)
$d_{(u,v)}$	Propagation delay of link (u, v)
k_a	Binary variable that indicates if AF a is terminable
p_a	Priority level of AF a, where 1 is the lowest
γ_a	Capacity required by AF instance a
av_{I_a}	Availability of AF instance a
$\lambda_{h,a}$	Data rate of type h offered by AF instance a
$\tau_{h,a}$	Latency of type h offered by AF instance a
$s_d \in N$	Source node of demand d
$t_d \in N$	Destination node of demand d
$r_d \in A$	AF requested by demand d
f_{π_d}	Minimum data rate supported by demand d
f_{to_d}	Maximum end-to-end delay requested by demand d
av_{R_d}	Minimum availability required by demand d

Table 3.2: Parameters definition for the optimization problem

3.1.2 Assumption

- In a fully mesh network, every node (or device) is interconnected directly with every other node. This structure enhances redundancy and reliability, as data can be routed through various paths. From a logical standpoint, assuming a completely interconnected network simplifies the modeling of data flows and interactions between nodes, as any node can communicate with another without relying on a specific path.
- It is assumed that each node in the network can allocate a specific portion of its CPU resources exclusively for critical application functions that cannot be terminated. This ensures that essential services maintain their functionality even under conditions where the network is under heavy load or part of the resources are dedicated to other tasks.
- This approach focuses on analyzing one AF at a time, ignoring any potential interactions or dependencies between different AFs. This simplification allows for a more straightforward analysis of each AF's performance and requirements without the complexity of considering AF chains.
- Each AF is linked to a single demand or a virtual flow that aggregates multiple flows. This assumption implies that the analysis does not consider scenarios where an AF must handle multiple distinct demands simultaneously, simplifying the mapping between AFs and the network load they are responsible for.
- The model allows for multiple instances of the same AF to be deployed across different nodes in the network, as long as those nodes have sufficient resources to support the AFs' requirements. This flexibility aids in distributing the load and optimizing the network's overall performance by leveraging available resources across the network.
- It is assumed that AFs do not alter the volume of data entering or exiting a network node. This assumption simplifies the analysis of data flows, as the focus can remain on routing and resource allocation without needing to account for changes in data volume caused by the AFs themselves.
- The optimization problem does not consider the time or disruptions associated with rescheduling tasks or reallocating resources (reschedule time). By excluding these factors, the model focuses on the static allocation of resources and the immediate performance of AFs without the additional complexity of dynamic adjustments.

3.1.3 Decision Variables

The decision points within the problem are represented by the variables listed in Table 3.3, and the goal of the optimization is to determine their optimal values.

Variable	Explanation
$x_{a,d}^{n'} \in \{0,1\}$	1 if the AF $a \in A$ is deployed for demand d on node $n' \in N'$; and 0, otherwise
$y_d^{(u,v)} \in \{0,1\}$	1 if demand $d \in D$ use the link $(u,v) \in N\text{x}N$; and 0, otherwise
$z_d \in \{0,1\}$	1 if the traffic of demand $d \in D$ is admitted to the network; and 0, otherwise
$m_{h,a,d}^{n'} \in \{0,1\}$	1 if a specific type of service $h \in H$ is selected for AF $a \in A$ of demand d on node $n' \in N'$; and 0, otherwise

Table 3.3: Variables definition for the optimization problem

3.1.4 Objective Function

In the medical sector, the overarching objective revolves around the capacity of the network infrastructure to seamlessly integrate and manage an ever-increasing volume of requests. This challenge is not just about scaling up to accommodate more data; it's about doing so in a way that prioritizes critical medical applications and services according to their importance and urgency. The strategy presented in 3.1 focuses on optimizing the performance of the network by carefully assessing and prioritizing the demands placed upon it. Each request or demand that comes through the network is evaluated based on several key factors, including its priority level and the data rate it requires. Priority levels are determined by the critical nature of the service or application making the request. For instance, real-time patient monitoring and telemedicine sessions could have higher priority over less critical data transfers, such as routine administrative tasks.
Moreover, consideration is given to the different service levels needed across various nodes within the network. This involves ensuring that each node is equipped to handle its specific load effectively, with adjustments made for higher priority services to guarantee they have the necessary bandwidth and resources.
By focusing on maximizing the overall value of admitted demands, the approach is not merely about using the network's capacity efficiently. It ensures that the most critical applications, those that directly impact patient care and outcomes, are given the precedence they require. This approach allows for the creation of a healthcare network that is not only robust and capable of handling a large volume of requests but also intelligent and responsive to the varying needs of medical services. A penalty given by W is added to

ensure that the network accepts the maximum number of demands.

$$\max \sum_{d \in D} \Bigg((z_d \cdot p_{r_d} \cdot \sum_{n' \in N'} \sum_{h \in H} m^{n'}_{h,r_d,d} \cdot \lambda_{h,r_d}) - W \cdot (1 - z_d) \Bigg) \tag{3.1}$$

3.1.5 Problem Constraints

Non-Terminability Constraint (3.2): when a demand d selects a non-terminable AF $a \in A$ ($k_{r_d} = 1$), it is crucial to guarantee its integration into the network

$$k_{r_d} \leq z_d \quad \forall d \in D \tag{3.2}$$

AF Installation Constraint (3.3): ensures that the AF a required by demand $d \in D$ (r_d) is installed on at most one node $n' \in N'$ with sufficient CPU space, if the demand d is accepted

$$z_d \leq \sum_{n' \in N'} x^{n'}_{r_d,d} \leq 1 \quad \forall d \in D \tag{3.3}$$

One AF per Demand Constraint (3.4): enforces the requirement that only one AF a is activated for each demand d accepted in the network

$$\sum_{n' \in N'} \sum_{a \in A} x^{n'}_{a,d} = z_d \quad \forall d \in D \tag{3.4}$$

Resource Constraint (3.5): CPU resources on a node $n' \in N'$ are limited, and therefore, these resources cannot be exceeded by the hosted AFs

$$\sum_{d \in D} \sum_{a \in A} x^{n'}_{a,d} \cdot \gamma_a \leq c^{n'}_{cpu} \quad \forall n' \in N' \tag{3.5}$$

Service Selection Constraint (3.6): ensures that for each AF $a \in A$ installed on a node n' of the network for demand d, only one specific type of service $h \in H$ is selected, enforcing a strict one-to-one relationship between AFs and selected services

$$\sum_{h \in H} m^{n'}_{h,r_d,d} = x^{n'}_{r_d,d} \quad \forall n' \in N',\ d \in D \tag{3.6}$$

AF capacity Constraint (3.7): ensures that, for each node n' and demand d, if the AF is deployed on that node ($x^{n'}_{r_d,d} = 1$), then the selected service type h must provide a data rate (λ_{h,r_d}) sufficient to meet or exceed the minimum data rate requirement (f_{π_d}) of the demand d

$$x^{n'}_{r_d,d} \cdot f_{\pi_d} \leq \sum_{h \in H} m^{n'}_{h,r_d,d} \cdot \lambda_{h,r_d} \quad \forall n' \in N',\ d \in D \tag{3.7}$$

Link Constraint (3.8): the sum of the data rate required by all demands $d \in D$ served by link (u, v) should not be larger than the capacity of the link (u, v)

$$\sum_{n' \in N'} \sum_{d \in D} \sum_{h \in H} z_d \cdot (y^{(u,v)}_d + y^{(v,u)}_d) \cdot m^{n'}_{h,r_d,d} \cdot \lambda_{h,r_d} \leq c^{(u,v)}_{\pi} \quad \forall u, v \in N,\ u \neq v,\ (u, v) \in L \tag{3.8}$$

Flow Conservation Constraints (3.9 - 3.11): guarantee that, for every node (excluding source s_d and destination t_d) in the network $n \in N$, the difference between all outgoing and incoming flows for each demand d remains consistent, if active

$$\sum_{u \in N, u \neq s_d} y_d^{(s_d,u)} = z_d \quad \forall d \in D \tag{3.9}$$

$$\sum_{v \in N, v \neq u} y_d^{(u,v)} - \sum_{v \in N, v \neq u} y_d^{(v,u)} = 0 \quad \forall d \in D,\ u \in N \backslash \{s_d, t_d\} \tag{3.10}$$

$$\sum_{u \in N, u \neq t_d} y_d^{(u,t_d)} = z_d \quad \forall d \in D \tag{3.11}$$

Intermediate Node Flow Continuity Constraint (3.12): ensure that for any intermediate node in the network (neither the source nor the destination of a demand), if there is a flow directed towards an intermediate node ($y_d^{(v,u)}$), there must be a corresponding outgoing flow from node u to some other node b in the network

$$y_d^{(v,u)} \leq \sum_{b \in N, b \neq u,v} y_d^{(u,b)} \quad \forall d \in D, u \in N \backslash \{s_d, t_d\}, v \neq u \in N \tag{3.12}$$

Node Inclusion Constraint (3.13): ensures that when demand d is admitted (z_d) and AF r_d is deployed on node n' ($x_{r_d,d}^{n'}$), the demand's path includes node n' by ensuring that at least one incoming link is active

$$z_d \cdot x_{r_d,d}^{n'} \leq \sum_{u \in N, u \neq n'} y_d^{(u,n')} \quad \forall d \in D, \forall n' \in N' \tag{3.13}$$

Delay Constraint (3.14): the cumulative delay for a demand d, is expressed as the sum of the processing time of the associated AF $a \in A$, denoted by $\tau_{h,a}$, and the propagation time $d_{(u,v)}$ of the utilized links. This summation is constrained by the specified end-to-end delay f_{to_d} for each demand $d \in D$

$$\sum_{u \in N} \sum_{v \in N, u \neq v, (u,v) \in L} (y_d^{(u,v)} + y_d^{(v,u)}) \cdot d_{(u,v)} + \sum_{n' \in N'} \sum_{h \in H} m_{h,r_d,d}^{n'} \cdot \tau_{h,r_d} \leq z_d \cdot f_{to_d} \quad \forall d \in D \tag{3.14}$$

Service Control Constraint (3.15): ensure that service levels are not activated on nodes that are not capable of hosting AFs

$$\sum_{h \in H} m_{h,r_d,d}^{n} = 0 \quad \forall n \notin N',\ d \in D \tag{3.15}$$

Node Activation Control Constraint (3.16): ensure that if a node is not capable of hosting AFs for the demand, then is set to 0

$$\sum_{n \notin N'} x_{r_d,d}^{n} = 0 \quad \forall d \in D \tag{3.16}$$

Connection Constraint (3.17, 3.18): ensures that if an AF is deployed on a node for a specific demand, that node can't receive (send) traffic for the same demand from (to) any other node, except its source (destination) or switch nodes

$$x^n_{r_d,d} \cdot y^{(u,n)}_d = 0 \quad \forall d \in D,\ n \in N,\ u \in N \text{ if } (u \neq s_d \text{ and } u \notin N'') \text{ or } u = t_d \tag{3.17}$$

$$x^n_{r_d,d} \cdot y^{(n,u)}_d = 0 \quad \forall d \in D,\ n \in N,\ u \in N \text{ if } (u \neq t_d \text{ and } u \notin N'') \text{ or } u = s_d \tag{3.18}$$

Path Continuity Constraint (3.19): ensure that if the destination node is not the selected node for the AF, then the link between source and destination is deactivated

$$y^{(s_d,t_d)}_d \leq x^{t_d}_{r_d,d} \quad \forall d \in D \tag{3.19}$$

Routing Logic Constraints (3.20 - 3.22): ensure to go from source to destination only passing through switches and the activated node for the desired AF

$$y^{(u,n)}_d + y^{(n,u)}_d \leq x^n_{r_d,d} \quad \forall d \in D,\ n \notin N'' \text{ and } n \neq s_d, t_d,\ u \in N' \text{ and } u \neq n, t_d \tag{3.20}$$

$$\sum_{u \in N} y^{(u,s_d)}_d = 0 \quad \forall d \in D \tag{3.21}$$

$$\sum_{u \in N} y^{(t_d,u)}_d = 0 \quad \forall d \in D \tag{3.22}$$

Availability Constraint (3.23): linear approximation of the availability constraint, guarantees that the mapping's availability meets the specified demand d's availability requirement. This applies to both the selected AF and the path nodes, ensuring their availability [39]

$$av_{R_d} \cdot z_d \leq av_{s_d} \cdot \prod_{n,v \neq n \in N} y^{(v,n)}_d \cdot av_n \cdot av_{I_{r_d}}^{x^n_{r_d,d}} \quad \forall d \in D \tag{3.23}$$

Availability constraint (3.23) is not linear. This can be approximated by the following linear constraint 3.24 [40]

$$\begin{aligned} av_{R_d} \cdot z_d \leq 1 - \Bigg(& (1 - av_{s_d}) + \sum_{m \in N} x^m_{r_d,d} \cdot (1 - av_{I_{r_d}} av_m) \\ & + \sum_{n \in N} \sum_{v \neq n \in N} (1 - x^n_{r_d,d}) \cdot y^{(v,n)}_d \cdot (1 - av_n) \Bigg) \quad \forall d \in D \end{aligned} \tag{3.24}$$

3.2 Hentati Adaption

This section presents a detailed comparative analysis between the Integer Linear Programming (ILP) models developed by [8] and our proposed model. The comparison focuses on the methodological differences in handling Virtual Network Function Forwarding Graphs

(VNF-FGs) and outlines the necessary modifications to ensure a fair and valid comparison. The foundational work by *Hentati et al.* serves as a benchmark for our analysis. To facilitate a meaningful comparison, it is imperative to introduce modifications to their models, aligning them with our unique approach to VNF-FG definition and optimization. In this way, both solutions can process the same set of requests.

A key distinction lies in the conceptualization of VNF-FGs:

- *Hentati et al.*'s methodology involves a sequence of AFs, a VNF-FG, for each demand.
- Our model, in contrast, associates each demand with a single AF, eliminating the need for a sequence and simplifying the graph structure.

Key Modifications

Significant adjustments are necessary to align *Hentati et al.*'s model with our approach:

1. **Dependency Among AFs**
   - *Hentati et al.*'s framework assumes dependencies among AFs, requiring sequential processing and specific constraints $(12, 14)$ for sequence maintenance.
   - Our simplified model negates the need for multiple AFs per request, rendering sequential processing constraints unnecessary.
2. **Time Slot Definition**
   - The original emphasis on time slots aims at facilitating sequential flow among AFs, involving complex variables for time management (x, s, y, h).
   - We preserve the concept of end-to-end delay as a critical constraint but simplify its implementation, adjusting the model to exclude unnecessary variables.
3. **Imposition of Routing Rules**

   The same routing rules as defined by our constraints (specifically constraints 3.12, 3.13, and 3.20-3.22) are imposed. These constraints enforce that the flow must proceed from the source to the destination via the designated node where the AF is allocated for that demand, utilizing only switches to navigate through the network. This ensures a consistent routing logic across both models.
4. **Data Rate Specification**

   Hentati et al.'s model does not offer service diversification, we set the best data rate provided by the AF as the sole service option. This adjustment addresses the limitation in *Hentati et al.*'s model regarding service diversity, ensuring that our comparison takes into account the best service offering available for each AF.

3.3 Heuristic Algorithm

3.3.1 Approach Overview

This code presents an advanced algorithm for dynamic demand placement in network graphs, focusing on optimizing the allocation of network resources based on application functions, services, and user demands. It integrates a heuristic approach to efficiently map network demands to the most suitable nodes, paths, and services within a given network topology, considering constraints such as service requirements, data rates, and end-to-end delay. The algorithm employs a series of functions designed to validate demand data, process the network graph for available resources, and place or upgrade demands based on the calculated optimal paths and service levels.
The provided code comprises a suite of functions designed for the strategic placement of demands within a network graph, focusing on optimizing network resource allocation while adhering to specific application functions and service requirements, offering a scalable solution to meet the evolving needs of modern network infrastructures. Each function plays a crucial role in the overall algorithm, contributing to its ability to make informed decisions regarding demand placement and service upgrades.
Below is a detailed description of each function and its role within the algorithm:

heuristic_placement() This is the main function orchestrating the placement of demands within a network graph (‘G‘), figure 3.2. It takes four arguments: the network graph itself, two dictionaries (‘app_func‘ and ‘app_serv‘) detailing the properties of application functions and services available in the network, and a dictionary of demands (‘demands‘) that need to be placed. The function processes these inputs to determine the optimal mapping of demands to network nodes, paths, and services, considering factors such as terminability, priority, and resource constraints. It returns a tuple containing mappings of demands to nodes, paths, services, any errors encountered during placement, and the count of successfully placed demands.

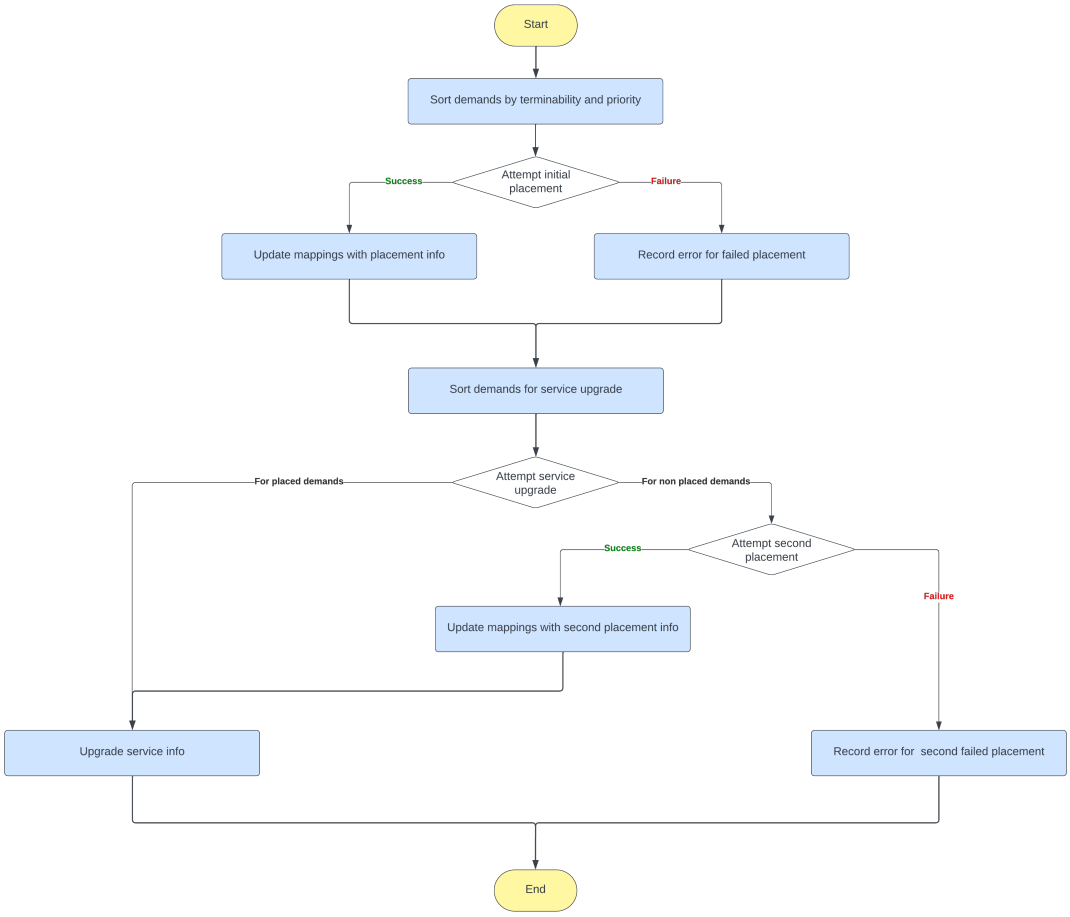

Figure 3.2: Flow Chart of the Heuristic Algorithm.

validate_demand_data() This function validates the input data for each network demand against the specified application functions. It checks if all necessary keys (e.g., 'source', 'destination', 'af', 'e2e_delay') are present in the demand information and verifies that the specified application function exists within the network's application function dictionary.

dfs() 'dfs' stands for Depth-First Search, a fundamental graph traversal method [41]. This function is designed to explore possible paths from a start node to a goal node within the network graph, prioritizing the exploration of switch nodes and avoiding loops by keeping track of visited nodes. It is instrumental in finding all feasible paths through the network that could potentially host the network demands, taking into account the specific requirements of source, intermediate, and destination nodes.

calculate_path_util_perc() This utility function is designed to calculate the minimum percentage of utilization across all links within a specific route. Its significance cannot be overstated when it comes to the critical task of assessing the overall quality and reliability

of a route within a network infrastructure. By taking into account the current usage patterns of the network, this function aids in ensuring that requests are intelligently directed towards routes that boast sufficient capacity. This strategic routing is instrumental in avoiding potential bottlenecks on network segments that are already operating near or at their maximum load.
Additionally, the function is crucial for accurately deploying the requested AF on a strategically chosen node along the path from the source to the destination, performing access control. This placement ensures the selection of the route with the most bandwidth resources available across its links. In doing so, it not only optimizes the performance and efficiency of the network but also enhances the QoS for users by leveraging the most resource-abundant path available. Such approach to network management ensures a more balanced distribution of traffic, paving the way for potential service improvements post-initial setup. This maximizes the utilization of available bandwidth and minimizes the risk of congestion and performance degradation.

find_paths_through_node_ordered() Building upon the ‘find_paths_through_node‘ function, this enhanced version leverages a more complex depth-first search (dfs) strategy to identify every possible route from a source to a destination via a specified intermediate node. This is particularly beneficial in scenarios requiring demands to pass through predetermined areas or nodes of the network, as it enables the algorithm to account for these specific constraints while identifying viable paths for demand allocation.
Besides, this improved function also ranks the discovered paths based on their utilization percentages, determined by the ‘calculate_path_util_perc‘ function. By doing so, it gives precedence to the paths with lower utilization, potentially fostering a more efficient allocation of network resources.

place_demand() The ‘place_demand‘ function attempts to place a single demand within the network, considering the available application functions, services, and the current state of the network graph. It evaluates potential nodes and paths to find a fit that meets the demand’s requirements while adhering to constraints like service requirements, data rates, and end-to-end delay. The function returns a tuple containing the selected node, path, and service for the placed demand or an error code if placement is not possible. The ‘second_try‘ input parameter is a boolean that indicates whether the ‘place_demand‘ function is being invoked for the first time. This parameter allows the algorithm to adjust its approach based on the invocation context. If it’s the initial call, the algorithm aims to allocate the demand at the minimum service level required to meet the demand’s minimum rate requirement. On the other hand, if the function is called a second time (‘second_try‘ is true), the algorithm attempts to place the demand in the network at the lowest possible service level. An example is shown in Algorithm 1.

Algorithm 1 Place Network Demands

Require: G, demand_info, app_func, app_serv, second_try, link_info
Ensure: *select_node*, *select_path*, *select_serv* if possible, otherwise error

```
Initialize variables for node, path, and service selection to None
for all nodes in G considering capacity do
    for all services meeting demand do
        for all paths through node do
            if path meets delay and capacity constraints then
                Calculate path availability
                if path availability meets demand then
                    Update selection variables
                    if suitable service found and second try then
                        Break loop
                    end if
                end if
            end if
        end for
        if service selected then
            Break service loop
        end if
    end for
    if node selected then
        Break node loop
    end if
end for
if no node selected then
    return error
else
    Deduct resources from G
end if
return select_node, select_path, select_serv
```

upgrade_demands() This function aims to upgrade the service level of already placed demands, if possible, based on the current network state and available services. It iterates over the paths of placed demands, checking if a higher service level can be provided without violating any constraints. If an upgrade is feasible, it updates the demand's service level, potentially improving the quality of service for the demand without requiring re-placement. An example is shown in Algorithm 2.

Algorithm 2 Place demand in the network

Require: G, demand_info, path, serv, app_serv, link_info
Ensure: Upgraded service identifier or $None$

```
Extract req_af from demand_info
select_serv ← None
Restore capacity for current service along path
for all services matching req_af do
    if all links in path have enough capacity then
        select_serv ← service identifier
    end if
end for
if select_serv ≠ serv then
    Deduct capacity for new service along path
    return select_serv
else
    if serv ≠ None then
        Restore original capacity if no upgrade is possible
    end if
    return None
end if
```

Each function within this codebase is intricately designed to work together, forming a comprehensive solution for demand placement and optimization in network graphs. The algorithm carefully considers multiple aspects of network management, from validating demand data and exploring feasible paths to optimizing resource allocation and service levels, showcasing a robust approach to handling complex network demands.

3.3.2 Greedy

Following the general overview of the heuristic demand placement algorithm, it's pertinent to delve into why this approach is fundamentally considered greedy [42]. The algorithm's essence lies in its sequential decision-making process, where decisions on demand placement are made based on current network conditions and the immediate optimality of choices without deliberation on future consequences or possibilities of revisiting and altering these choices later. This characteristic aligns with the core principle of greedy algorithms, which prioritize local optimality in the hope that these local decisions collectively lead toward a globally optimal or near-optimal solution. The reliance of the algorithm on placing demands based on the current availability of network resources underscores a typical greedy strategy—making the best immediate decision from the current standpoint. Furthermore, the absence of backtracking or reevaluation of past decisions in light of new information cements its classification as a greedy approach. The application of this methodology to network demand placement showcases its efficacy in providing efficient, scalable solutions

that, while not guaranteeing global optimality, achieve significant improvements in network resource utilization and service delivery within the constraints of real-time or near-real-time operational requirements.

3.3.3 Local Search

In light of the inherent limitations of greedy algorithms, a refined strategy to improve the quality of outcomes involves conducting multiple iterations of the heuristic demand placement process, with each cycle rearranging the order of nodes in the network. This technique, which adopts a local search approach, methodically navigates through various potential solutions by modifying the order in which nodes are evaluated for accommodating demands. Such adjustments expand the search space of the algorithm while maintaining its core attributes of speed and efficiency. This process of iterative optimization is designed to enhance the number of successfully placed requests, with a particular focus on ensuring the highest quality of service for high-priority demands. Shifting the sequence in which nodes are considered from one iteration to the next enables the discovery of more effective paths and node assignments that might not be apparent during an initial evaluation, gradually moving toward a solution that is either optimal or very close to optimal. This strategy leverages the greedy nature of the algorithm for initial placements, while the repeated iterations and exploration of different node sequences introduce a methodical variability, enhancing the overall efficacy of demand placements within the network. The objective of this repetitive process is to strike a delicate balance between the computational expediency of greedy algorithms and the comprehensive scrutiny typical of exhaustive searches, aiming for an enhanced network operation and resource distribution. This method reveals a trade-off between the runtime of the algorithm and the quality of solutions achieved, which can be adjusted based on the scale of the network and the complexity of the demands being addressed.

3.3.4 Time Complexity

To analyze the time complexity of the proposed heuristic algorithm (`heuristic_placement`), the algorithm is dissected into its fundamental components. This approach facilitates a comprehensive understanding of how each part contributes to the overall performance and efficiency of the algorithm. By examining the operations carried out by each function within these components, it becomes possible to identify and quantify the computational resources required.

1. **Initial Sorting of Demands**: Demands are sorted twice on the basis of different criteria (terminability and priority, then priority alone). The sorting operations are $O(D \log D)$, where D is the number of demands.
2. **Demand Placement** (`place_demand`):
    - Iterating through each demand and potentially through each node and service

in the graph for placement. If we have D demands, N nodes, and S services, the worst-case scenario without considering internal calculations would be $O(DNS)$.

- Internal operations like finding paths (`find_paths_through_node_ordered`) also calculates the utility percentage for each path and then sort them. The complexity for finding paths is $O(N + E)$, where E is the number of edges in the graph G. Calculating the utility percentage for each path adds an additional $O(n \cdot p)$ complexity, where p is the number of paths and n is the average path length. Sorting these paths based on utility percentage can be $O(p \log p)$. Therefore, the overall complexity might be dominated by the sorting operation, depending on the number of paths found.

3. **Service Upgrade** (`upgrade_demands`): Iterating through each service and every link that composes the path for the upgrade. If S is the number of services in 'app_serv' and n is the average path length minus one (approximated to the average path length), the function complexity is $O(S \cdot n)$.

Overall Time Complexity: Considering these elements, the primary contributors to the algorithm's time complexity are the demand placement and path finding processes. The overall time complexity can be summarized as $O(D \log D) + O(DNS \cdot p \log p)$, indicating a polynomial time complexity in practical scenarios.

3.4 Performance Evaluation

In this section of the thesis, we aim to provide a concrete visualization of the differences between three distinct methods of problem-solving presented before. The primary objective of this comparative analysis is to demonstrate the superior performance of our proposed solution, particularly in its capability to adapt the services offered based on the availability of network resources. This adaptability feature stands in contrast to the benchmarks established by the study conducted by *Hentati et al.*, which has been selected for comparative purposes. Our analysis encompasses both the optimization problem and the heuristic local search algorithm. The findings reveal that, when utilizing our solution, the network is capable of accommodating a larger number of requests. This result underscores the effectiveness of our approach in enhancing network service adaptability and performance, thereby offering a significant improvement over the methodologies reviewed, including the one proposed by [8].

3.4.1 Methodology

Hardware

All the tests have been run on an Intel(R) Core(TM) i5-8250U CPU (1.60GHz base frequency, up to 8 threads), which supports instruction sets [SSE2—AVX—AVX2], and consists of 4 physical cores and 8 logical processors.

Optimization Solver

Gurobi Optimizer is used to solve optimization problems, known for its exceptional performance, reliability, and user-friendliness [43]. It supports a broad range of optimization tasks and is scalable, making it suitable for both small and large-scale problems. Its integration with Python allows for straightforward problem definition and solution, enhancing its utility in academic and research contexts. An academic license is used to access the full functionality of Gurobi.

3.4.2 Results

When considering the deployment of a fully connected network architecture, a systematic approach is essential to facilitate the transition from theoretical design to practical application within a live network environment. This process encompasses several critical stages that must be meticulously executed to ensure the operational efficiency and reliability of the network. The stages are delineated as follows:

1. Computation of propagation delay and throughput for Virtual Links: This initial phase involves the precise calculation of two fundamental parameters—propagation delay and throughput—for each virtual link within the network. Propagation delay refers to the time it takes for a signal to travel from the source to the destination across the network. Throughput, on the other hand, measures the rate at which data is successfully transmitted over these virtual links. This step is pivotal as it provides the necessary data inputs for the subsequent optimization of the network, ensuring that the network design can meet the required performance criteria under varying conditions.

2. Optimization of the network on a fully meshed graph: The second phase is centered on resolving an optimization problem formulated on the basis of the fully connected (or fully meshed) network topology. This optimization process is critical to improve the overall performance and efficiency of the network.

3. Mapping of Virtual Links to Physical Links and activation: The final stage involves the practical implementation of the optimized network design by mapping the virtual links, as determined by the optimization process, onto the physical network infrastructure. This mapping process involves assigning the virtual links to specific physical connections and network devices that facilitate the actual data transmission. Following this mapping, the virtual links are activated within the physical network, thereby operationalizing the optimized network configuration.

In our specific context, the focus is predominantly on the second phase—optimization of the network on a fully meshed graph. This prioritization is grounded in the understanding that the calculations of propagation delay and throughput for virtual links (phase 1) serve merely as preparatory inputs for the optimization challenge. Similarly, the mapping of virtual links to physical elements and their activation (phase 3) is viewed as a direct

outcome of the optimization process. Consequently, the optimization phase is deemed pivotal, as it directly influences the efficiency and efficacy of the operational capabilities of the network, dictating the overall success of the network deployment strategy.

Performed Tests

This section describes the settings used in our simulations.

Network characteristics

- Node computational power ($c_{cpu}^{n'}$): The processing capability of each node is randomly chosen between 3 and 6 units.
- Offered data rate ($c_{\pi}^{(u,v)}$): The amount of data offered to the network varies from 0.5 to 1.5 Gigabits per second (Gbps).
- Propagation delay ($d_{(u,v)}$): Each physical link experiences a delay between 1 and 8 milliseconds (ms), as shown in Figure 3.1.

Application Function properties

- Computational demand (γ_a): Each AF requires a certain amount of processing power, varying randomly between 0.1 and 0.6 units, as referenced in [44].
- Availability: The probability of an AF or a node being available for use is set between 0.9999 and 1 (γ_{I_a}, u_n), based on [39].
- Throughput ($\lambda_{h,a}$): The capacity of each AF to handle data ranges from 1 to 800 Mbps. This variation depends on the specific service level offered and the priority level of the AF (5 levels).
- Processing delay ($\tau_{h,a}$): The time required for an AF to process incoming data varies between 2 and 55 ms, also determined by the service level and the priority of the AF.

Demand characteristics

- Minimum offered data rate (f_{π_d}): The minimum amount of data requested to the AF varies from 0.5 to 400 Mbps.
- Maximum delay (f_{to_d}): The maximum e2e delay admitted by the request ranges from 20 to 60 ms.
- Minimum availability (γ_{R_d}): The minimum requested availability so that the request can be satisfied is set between 0.9 and 0.99999.

Optimization parameters

- Admission penalty weight (W): In our optimization formulation (Objective function 3.1), the weight W associated with rejecting a data request is set to 10. This prioritizes admitting as many requests as possible.
- Traffic penalty weight (W): The weight W assigned to each data traffic instance in Hentati et al.'s formulation is fixed at 1000. This high value, as explained in [8], ensures the scheduler prioritizes admitting data traffic over rejecting it (which has a lower penalty with a smaller weight).
- Link and node cost: The cost of using a link, a node, or hosting an AF on a node is randomly set between 0 and 11, as described in [44].

These parameters are summarized in Table 3.4.

Parameter	**Value**
$c_{cpu}^{n'}$	$uniform(3, 6)$
u_n	four 9s to 1
$c_{\pi}^{(u,v)}$	$randint(0.5, 1.5)$ Gbps
$d_{(u,v)}$	$uniform(1, 8)$ ms
k_a	$randint(0, 1)$
p_a	$randint(1, 5)$
γ_a	$uniform(0.1, 0.6)$
γ_{I_a}	four 9s to 1
$\lambda_{h,a}$	$uniform(1, 800)$ Mbps
$\tau_{h,a}$	$uniform(2, 55)$ ms
f_{π_d}	$uniform(0.5, 400)$ Mbps
f_{to_d}	$uniform(20, 60)$ ms
γ_{R_d}	one 9 to five 9s

Table 3.4: Simulation parameters

By performing measurements on our system we want to achieve four results:

- Compare the four approaches mentioned (3.1, 3.2, 3.3) regarding their acceptance ratio and computation time, under the same network and demand conditions.
- Illustrate the capacity of the network to handle non-terminable requests for each approach.
- Assess the impact of the number of available service levels on the acceptance ratio, given constant network and demand parameters.
- Demonstrate the benefit of offering multiple service levels on enhancing the acceptance ratio in the context of rising demands, while maintaining consistent network

parameters.

The network configuration under discussion is structured around a total of 10 individual nodes. These nodes are categorized into several distinct types to facilitate various functionalities within the network. Specifically, the configuration includes 3 access points (APs), 3 distributed units (DUs), 2 switches, and 2 centralized units (CUs). Among the various node types, only the DUs and CUs are compatible with hosting application functions, with a total of 45 links in a mesh configuration. This limitation means that out of the total 10 nodes, application functions can be installed on just 5 of them. This selective installation capability is visually represented in Figure 3.3, where nodes capable of supporting application functions are marked in red.

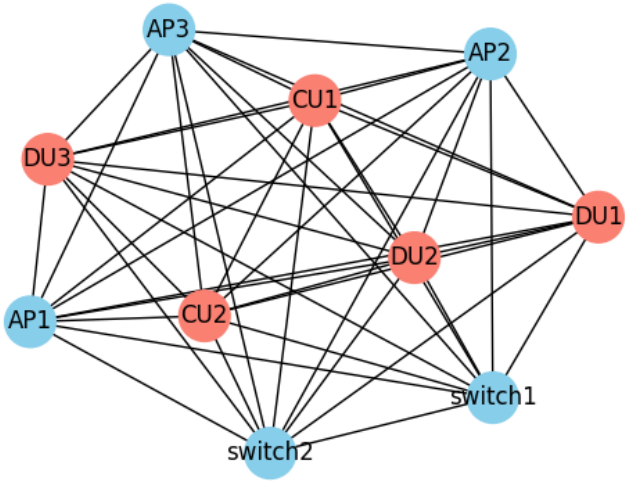

Figure 3.3: Simulation model network structure.

Concerning the diversity of application functions, it's noteworthy that there exist seven distinct types, each characterized by unique parameters. These distinctions and characteristics are concisely collected in the table referenced as Table 3.5, providing a comprehensive overview of the various application functions and their specific attributes.

AF Type	k_a	p_a	γ_a	γ_{I_a}	λ **[Mbps]**	τ **[ms]**
$AF1$	1	2	0.3	0.99999	$(5, 600)$	$(6, 55)$
$AF2$	0	3	0.5	0.9999	$(10, 800)$	$(3, 50)$
$AF3$	1	4	0.6	1	$(15, 650)$	$(9, 48)$
$AF4$	0	5	0.45	0.99995	$(3, 700)$	$(7, 45)$
$AF5$	1	1	0.28	0.9999	$(1, 300)$	$(6, 52)$
$AF6$	1	3	0.37	0.99998	$(10, 580)$	$(2, 48)$
$AF7$	0	2	0.33	1	$(6, 550)$	$(6, 52)$

Table 3.5: Application Function parameters

Acceptance ratio & computation time

For this analysis, a particular scenario has been chosen to evaluate the various methods previously outlined. Specifically, within the established network configuration, there are 45 distinct requests seeking allocation within the network, with up to 20% being for non-terminable AFs. Furthermore, 6 levels of data rate and processing delay are provided for each AF.
In terms of the local search method, the algorithm examines 40 different network configurations in an effort to achieve the highest acceptance ratio.
The outcomes of this comparison are presented as follows:

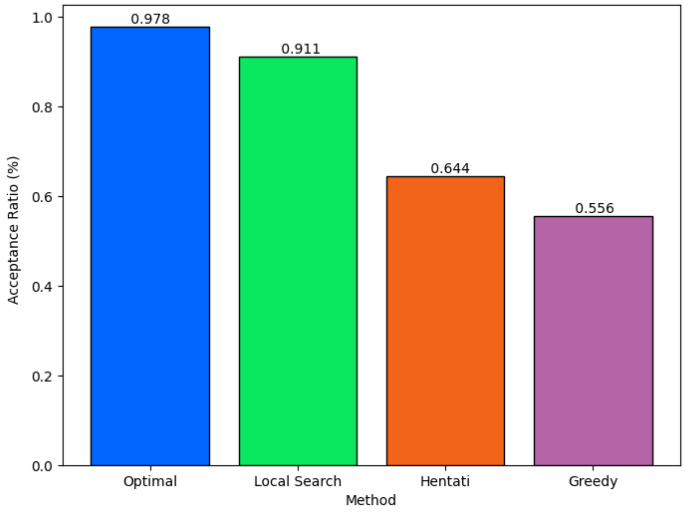

Figure 3.4: Simulation acceptance ratios comparison.

The bar chart in Figure 3.4 illustrates how the four different methods for allocating requests impact acceptance ratios within the network environment. The optimal method establishes a benchmark by fully meeting demands, achieving an acceptance ratio of 97.8%. The local search method follows with an impressive acceptance ratio of 91.1%, indicating its efficiency in accommodating demands, despite not matching the performance of the optimal method. The Hentati method shows a moderate performance with a 64.4% acceptance ratio, highlighting a significant drop in the ability to fulfill demands due to limitations in service selection. Lastly, the greedy method, at a 55.6% acceptance ratio, shows it can satisfy just over half of the demand requests, placing it at the bottom in terms of performance.

In the subsequent Figure 3.5 the precise percentage of AFs accommodated within the network corresponding to each priority level is depicted.

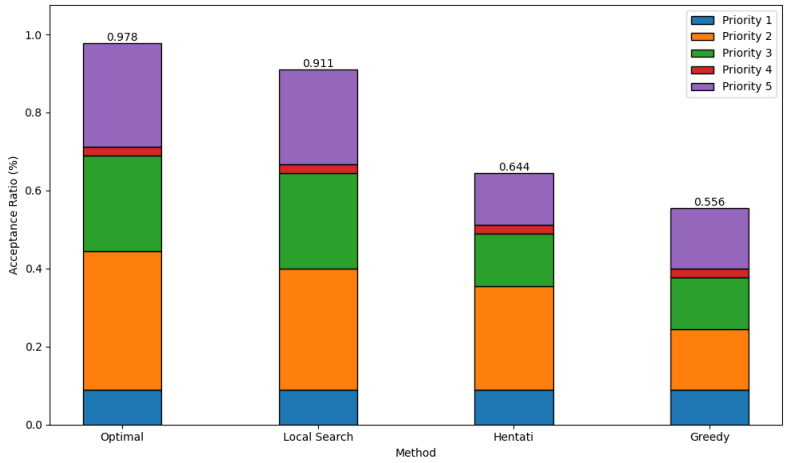

Figure 3.5: Comparison of the acceptance ratio for demands of each priority level across different methods.

Figure 3.6 presents the count of AFs associated with a particular priority level that are accepted in the network by both the optimization and heuristic models, serving as a basis for evaluating the effectiveness of the local search in achieving optimal results.

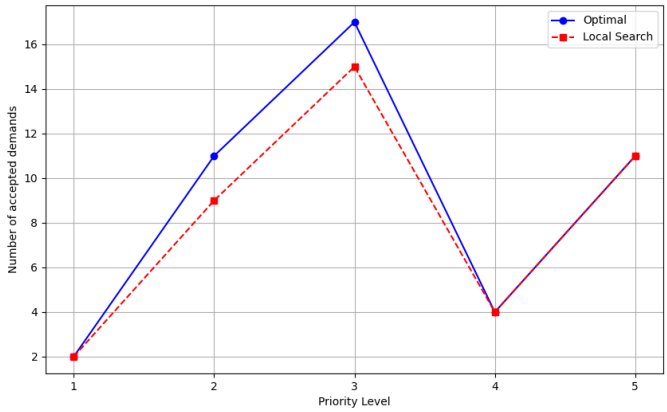

Figure 3.6: Number of demands accepted in the network by priority level.

Specifically, the analysis can focus on the average data rate provided to demands based on their priority level. As illustrated in Figure 3.7, the optimization problem is notably effective in achieving superior request acceptance outcomes by efficiently allocating average data rates per request. In contrast, the local search method results in a lower number of accepted applications, yet it offers a higher data rate to those it does accept. This, however, leads to link saturation, which in turn prevents the acceptance of new demands.

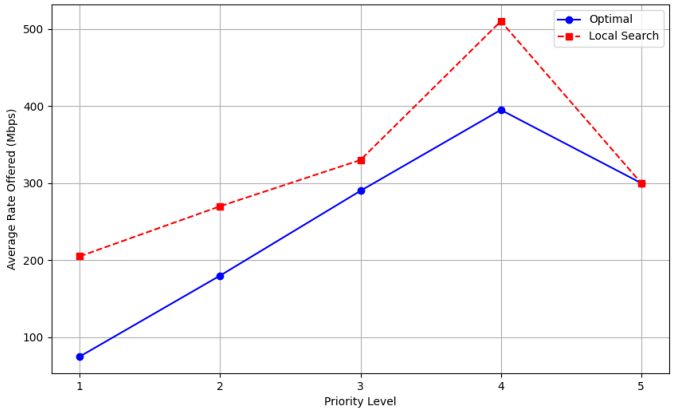

Figure 3.7: Comparison of the average data rate for each priority level.

Despite the optimization method achieving the highest benchmarks in terms of acceptance ratio, it demands significantly more resources and requires longer computation time, as detailed in Figure 3.8. These factors are crucial when evaluating the overall efficiency and practicality of the various methods. Given these considerations, the local search method

emerges as the most balanced option, effectively standing out when all factors are taken into account. The local search method offers a compelling compromise between performance and resource efficiency. While it slightly trails the optimization method in achieving the absolute best in acceptance ratios and service quality, it significantly reduces the resource consumption and computational time required. This balance is particularly important in real-world applications where resource availability and time constraints are often critical considerations.

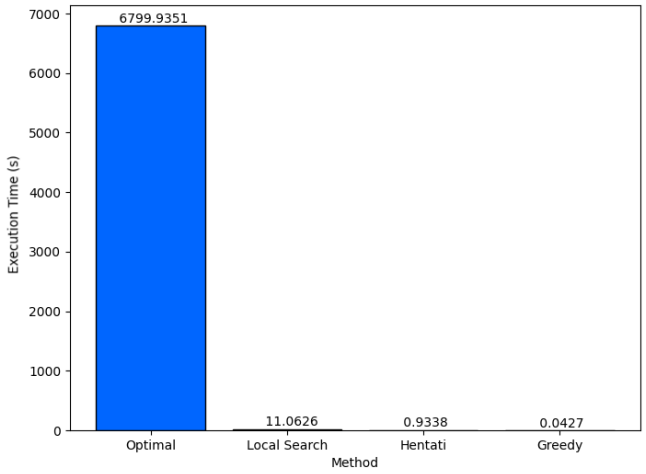

Figure 3.8: Simulation computation times comparison.

To summarize, although the optimization approach establishes a benchmark for potential acceptance rates, its demand on resources and time cannot be ignored. The local search strategy emerges as the preferable option by providing a solution that is both more efficient in terms of resources and time, without significantly sacrificing performance. This is further illustrated in Figure 3.9, which showcases the performance disparity between the two methods, making it clear that when all relevant aspects of network request allocation are considered, the local search method stands out.

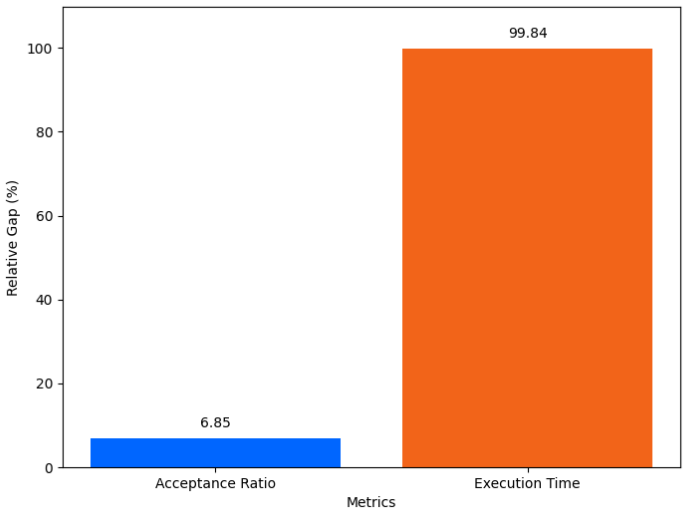

Figure 3.9: Relative gap between the optimal solution and the local search method.

Non-terminability property

This section delves into another critical key aspect of our research, underscoring the essential role of non-terminability for certain AFs within a healthcare setting, as previously discussed. The concept of non-terminability is paramount in ensuring that some application functions are integrated into the network.
The importance of this non-terminability feature is not considered by the Hentati and greedy approaches. These methods lack the essential attributes necessary for incorporating important AFs. The absence of non-terminability support renders these approaches unsuitable for scenarios where this characteristic is crucial, particularly in the medical sector.
Within the context of our study, it is clear that these methodologies fall short of addressing the distinct needs critical to medical decision-making processes, highlighting the efficacy of our proposed algorithms (both optimal and local search) in accommodating these demands and grasping the intricacies of this selection mechanism. This is illustrated in Figure 3.10, where it becomes evident that both Hentati and greedy approaches fail to fulfill all non-terminable AF requests in the network, each missing at least one request.

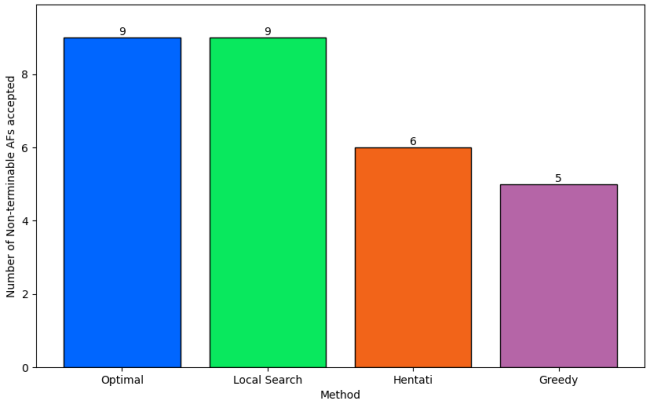

Figure 3.10: Comparison of accepted non-terminable demands by method.

Impact of varying levels of service

The study underscores the importance of providing varied service levels to boost the network's ability to handle an increased volume of requests. To explore this, tests were performed using different levels of service while keeping the demand constant (50, with up to 20% being non-terminable) in all experiments. The algorithms were assessed for all potential service level combinations, represented by h as introduced in Section 3.1.1. In this context, the first element of h indicates the variety of data rates available from the AF ($\lambda_{h,a}$), while the second component signifies the range of computational delays the AF can manage ($\tau_{h,a}$). Due to the complexity of the optimization problem, we simplified the network for comparative purposes. The configuration used in this and subsequent sections consists of 2 APs, 2 CUs, 1 DU, and 1 switch (Figure 3.11).

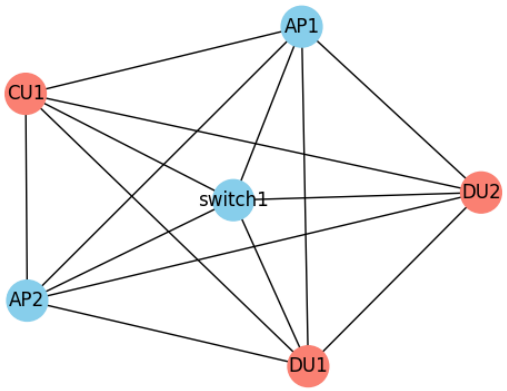

Figure 3.11: Simplified simulation model network structure.

Figure 3.12 illustrates the failure of both Hentati and greedy methods to consider adjusting the service level offered based on available resources, resulting in fewer demand acceptances. Moreover, the acceptance ratio remains relatively unchanged since these methods can only provide the maximum level of service.
Conversely, it is evident that our proposed solutions are capable of adapting to the network by offering different types of services. Specifically, the network appears to be more constrained by the data rate provided by the links than by the end-to-end delay. This is observed as the acceptance ratio remains roughly the same for 12 levels of service in terms of processing time as it does for 3 levels. However, the scenario changes when the number of levels offered for the data rate increases, this significantly improves the acceptance rate.

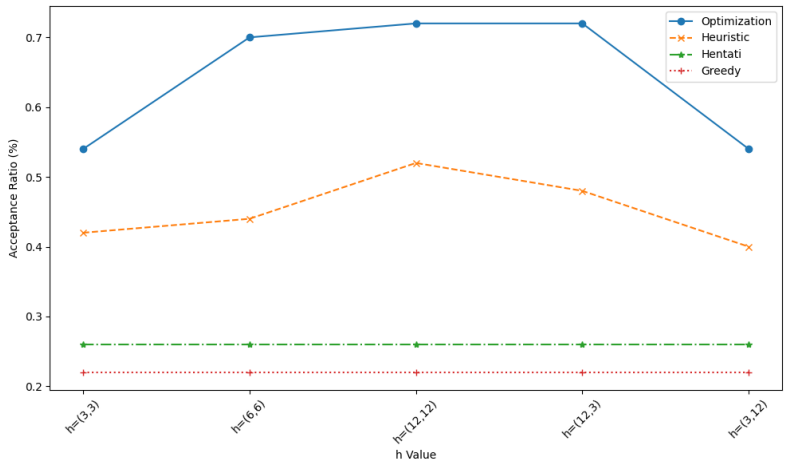

Figure 3.12: Analyzing the acceptance rates of various methods across multiple service levels.

Enhancing acceptance ratios with multi-level services while rising demand

The findings, as depicted in Figure 3.13, reveal a distinct trend: expanding the spectrum of service levels available through the local search algorithm leads to an uptick in the number of demands the network can fulfill. However, this pattern does not hold for the Hentati or greedy algorithms. In these instances, the number of service levels does not influence the strategy for node accommodation since they are designed to deliver only the maximum service level, as already said before.
The chart showcases the acceptance ratio outcomes for various methods and service levels, as the number of network requests gradually increases from 50, incrementing by 10, until reaching 100.
The observed increase in demand acceptance correlates inversely with the network's inherent resource limitations. Despite the rising volume of requests, the static nature of the network's configurations poses a growing challenge for resource allocation. This constraint underlines the delicate balance between expanding service offerings and the finite capabilities of the network infrastructure.

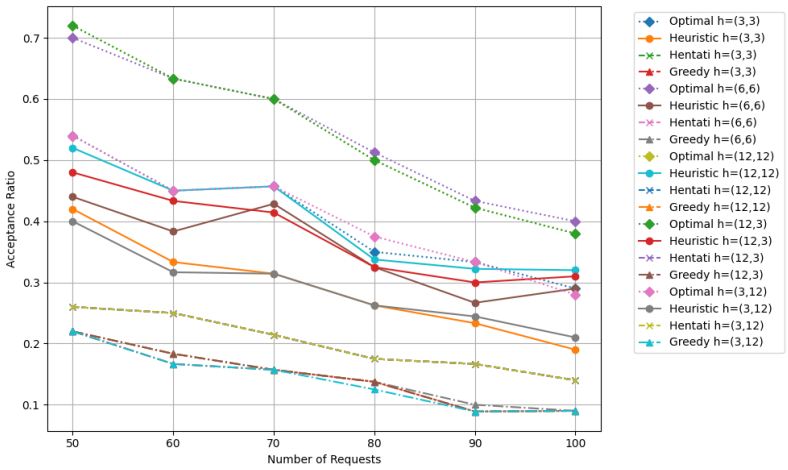

Figure 3.13: Comparison of acceptance ratios relative to the volume of requests across various levels of service for different algorithms.

To facilitate a more nuanced analysis of each value of h, Figure 3.14 offers a series of plots that examine the different values independently. By isolating each h value and presenting them in separate plots, the figure provides a clearer visualization of the relationship between the levels of service offered and the network's demand acceptance performance.

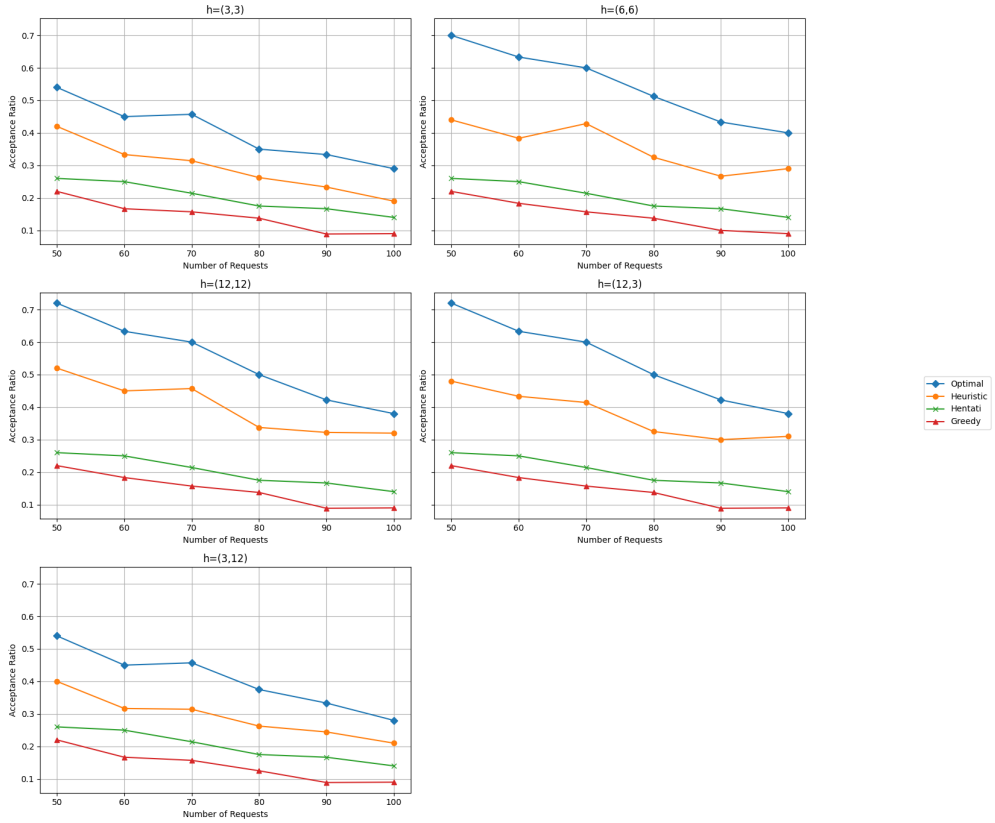

Figure 3.14: An in-depth examination and comparison of how acceptance ratios vary across different levels of service, focusing on the performance of various methods.

3.4.3 Results Evaluation

Following a comprehensive examination of network performance, the effectiveness of the suggested strategies is evident: both the optimal solution and local search methods deliver impressive outcomes. They can adjust to network resource constraints, providing a refined response to requests, thereby enhancing the network request acceptance rate compared to the current *Hentati et al.* model, with up to 40% better results.
As mentioned before, while the heuristic approach results in a lesser rate of request acceptance than the optimal solution, the significant reduction in execution time makes local search practical for real-world applications.

Chapter 4

Conclusions and Outlook

Throughout this work, it was presented a comprehensive exploration into the challenge of optimizing resource allocation within medical radio access networks (RANs), with a particular focus on supporting essential application functions (AFs) critical to healthcare operations. Recognizing the constraints posed by limited resources in RANs, our research aimed to develop a strategy that effectively maximizes the accommodation of diverse requests within a healthcare setting. This was initially approached through a mathematical model based on the 5G RAN framework, which, due to practical limitations in execution time, led us to pursue a heuristic solution.
Our refined methodology, evolving from a greedy method to a local search strategy, emphasizes the prioritization of non-terminable AFs. This prioritization is key to maintaining the continuity of critical healthcare applications, aligning our approach with the operational requirements of medical facilities. By benchmarking our findings against a baseline approach from the literature, we have demonstrated the effectiveness of our strategy in addressing the unique needs of medical RANs, particularly in prioritizing indispensable services while accommodating as many applications as possible through varied service levels.
The transition to heuristic methods reflects a pragmatic adaptation to the complexities of real-world scenarios, highlighting the importance of flexible and responsive strategies in the management of healthcare technologies. Our study underscores the critical balance between ensuring reliable healthcare services and optimizing network performance.
The results demonstrate that while optimizing, the approach accounts for the non-terminability of some AFs by ensuring their placement within the network, thus optimizing application acceptance rates. Although the local search method may yield lower values, it does so in considerably less time.

Future work In the pursuit of enhancing the efficiency and effectiveness of our network's application acceptance process, several key areas for future research and development have been identified. These areas not only offer the potential to refine the current methodologies but also to explore innovative strategies for overcoming the limitations inherent in the existing system. Below, we outline these critical areas of focus:

- The current heuristic method offers potential for further enhancements and optimizations to align acceptance rates more closely with those predicted by optimization models. Exploring alternative approaches, such as genetic algorithms or ant colony optimization, could allow for a more efficient exploration of solutions within a constrained timeframe.
- To more accurately assess the limitations of the system and identify effective strategies for increasing application acceptance, future simulations should model more complex and realistic scenarios.
- Additionally, investigating the system's behavior in an online scenario where demands arrive randomly will be crucial. This study would provide insights into the adaptability and efficiency of the system in handling unpredictable demand patterns, offering opportunities for real-time optimization and dynamic resource allocation.
- Future work could explore the implications of managing chains of AFs instead of singular ones, introducing complexity in problem definition by necessitating prioritization across the entire flow and addressing bottlenecks due to resource-limited AFs. It would also be important to ensure continuity in service by preventing interruptions in any AF in the chain, as any disruption could affect the entire flow.
- Investigating the impact of emerging technologies, such as quantum computing and machine learning algorithms, on optimizing network performance and application processing efficiency could provide groundbreaking improvements.

Appendix A

Notation und Abkürzungen

5G	Fifth generation
6G	Sixth generation
AI	Artificial Intelligence
AF	Application Function
AP	Access Point
B5G	Beyond fifth generation
BS	Base Station
CPU	Central Processing Unit
CU	Centralized Unit
DU	Distributed Unit
E2E	End to end
e-health	electronic health
EHR	Electronic Health Record
eNodeB	evolved Node B or E-UTRAN Node B
ICT	Information and Communication Technologies
ILP	Integer Linear Programming
IoT	Internet of Things
IoMT	Internet of Medical Things
IP	Internet Protocol
LTE	Long Term Evolution
mHealth	mobile health
ML	Machine Learning
mMTC	massive Machine-Type Communications
NFV	Network Function Virtualization
NS	Network Slicing
O-RAN	Open Radio Access Network
QoE	Quality of Experience
QoS	Quality of Service

PM	Physical Machine
RAN	Radio Access Network
RL	Reinforcement Learning
RU	Radio Unit
SDN	Software Defined Network
SFC	Service Function Chain
SLA	Service Level Agreement
TI	Tactile Internet
UE	User Equipment
URLLC	Ultra-Reliable Low-Latency Communications
VM	Virtual Machine
VNF	Virtual Network Function
VNF-FG	VNF-Forwarding Graph
WDM	Wavelength Division Multiplexing

List of Figures

List of Tables

Bibliography

[1] Solmaz Niknam, Abhishek Roy, Harpreet S. Dhillon, Sukhdeep Singh, Rahul Banerji, Jeffery H. Reed, Navrati Saxena, and Seungil Yoon. Intelligent o-ran for beyond 5g and 6g wireless networks. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 215–220, 2022.

[2] Stefan Mihai, Mahnoor Yaqoob, Dang V. Hung, William Davis, Praveer Towakel, Mohsin Raza, Mehmet Karamanoglu, Balbir Barn, Dattaprasad Shetve, Raja V. Prasad, Hrishikesh Venkataraman, Ramona Trestian, and Huan X. Nguyen. Digital twins: A survey on enabling technologies, challenges, trends and future prospects. *IEEE Communications Surveys and Tutorials*, 24(4):2255–2291, 2022.

[3] Mohammad Asif Habibi, Meysam Nasimi, Bin Han, and Hans D. Schotten. A comprehensive survey of ran architectures toward 5g mobile communication system. *IEEE Access*, 7:70371–70421, 2019.

[4] Chenxi Huang, Jian Wang, Shuihua Wang, and Yudong Zhang. Internet of medical things: A systematic review. *Neurocomputing*, 557:126719, 2023.

[5] Juliver Gil Herrera and Juan Felipe Botero. Resource allocation in nfv: A comprehensive survey. *IEEE Transactions on Network and Service Management*, 13(3):518–532, 2016.

[6] Hao Yu, Francesco Musumeci, Jiawei Zhang, Massimo Tornatore, Lin Bai, and Yuefeng Ji. Dynamic 5g ran slice adjustment and migration based on traffic prediction in wdm metro-aggregation networks. *Journal of Optical Communications and Networking*, 12(12):403–413, 2020.

[7] Chair of research group minimally invasive interdisciplinary therapeutical interventions, 2024. `https://web.med.tum.de/en/miti/home/` [Accessed: (01/03/2024)].

[8] Amina Hentati, Amin Ebrahimzadeh, Roch H. Glitho, Fatna Belqasmi, and Rabeb Mizouni. Remote robotic surgery: Joint placement and scheduling of vnf-fgs. In *2022 18th International Conference on Network and Service Management (CNSM)*, pages 205–211, 2022.

[9] Alcardo Alex Barakabitze, Arslan Ahmad, Rashid Mijumbi, and Andrew Hines. 5g network slicing using sdn and nfv: A survey of taxonomy, architectures and future challenges. *Computer Networks*, 167:106984, 2020.

[10] Moustafa M. Nasralla, Sohaib Bin Altaf Khattak, Ikram Ur Rehman, and Muddesar Iqbal. Exploring the role of 6g technology in enhancing quality of experience for m-health multimedia applications: A comprehensive survey. *Sensors*, 23(13), 2023.

[11] Ali Behravan, Vijaya Yajnanarayana, Musa Furkan Keskin, Hui Chen, Deep Shrestha, Traian E. Abrudan, Tommy Svensson, Kim Schindhelm, Andreas Wolfgang, Simon Lindberg, and Henk Wymeersch. Positioning and sensing in 6g: Gaps, challenges, and opportunities. *IEEE Vehicular Technology Magazine*, 18(1):40–48, 2023.

[12] Ahmed M. Alwakeel, Abdulrahman K. Alnaim, and Eduardo B. Fernandez. Toward a reference architecture for nfv. In *2019 2nd International Conference on Computer Applications Information Security (ICCAIS)*, pages 1–6, 2019.

[13] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, 18(1):236–262, 2016.

[14] Cisco. Cisco annual internet report (2018–2023) white paper, 2020. `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html` [Accessed: (01/03/2024)].

[15] Ericsson. Ericsson mobility report, 2023. `https://www.ericsson.com/4ae12c/assets/local/reports-papers/mobility-report/documents/2023/ericsson-mobility-report-november-2023.pdf` [Accessed: (01/03/2024)].

[16] Sahrish Khan Tayyaba and Munam Ali Shah. Resource allocation in sdn based 5g cellular networks. *Peer-to-Peer Networking and Applications*, 12(2):514–538, 2019.

[17] Shimaa A. Abdel Hakeem, Hanan H. Hussein, and HyungWon Kim. Vision and research directions of 6g technologies and applications. *Journal of King Saud University - Computer and Information Sciences*, 34(6, Part A):2419–2442, 2022.

[18] Anna Wernhart, Susanne Gahbauer, and Daniela Haluza. ehealth and telemedicine: Practices and beliefs among healthcare professionals and medical students at a medical university. *PLOS ONE*, 14:e0213067, 2 2019.

[19] Abid Haleem, Mohd Javaid, Ravi Pratap Singh, and Rajiv Suman. Telemedicine for healthcare: Capabilities, features, barriers, and applications. *Sensors International*, 2:100117, 2021.

[20] Angelos I. Stoumpos, Fotis Kitsios, and Michael A. Talias. Digital transformation in healthcare: Technology acceptance and its applications. *International Journal of Environmental Research and Public Health*, 20(4):3407, 2023.

[21] Oliver Holland, Eckehard Steinbach, R. Venkatesha Prasad, Qian Liu, Zaher Dawy, Adnan Aijaz, Nikolaos Pappas, Kishor Chandra, Vijay S. Rao, Sharief Oteafy, Mohamad Eid, Mark Luden, Amit Bhardwaj, Xun Liu, Joachim Sachs, and José Araújo. The ieee 1918.1 "tactile internet" standards working group and its standards. *Proceedings of the IEEE*, 107(2):256–279, 2019.

[22] Nattakorn Promwongsa, Amin Ebrahimzadeh, Diala Naboulsi, Somayeh Kianpisheh, Fatna Belqasmi, Roch Glitho, Noel Crespi, and Omar Alfandi. A comprehensive survey of the tactile internet: State-of-the-art and research directions. *IEEE Communications Surveys Tutorials*, 23(1):472–523, 2021.

[23] Gerhard P. Fettweis. The tactile internet: Applications and challenges. *IEEE Vehicular Technology Magazine*, 9(1):64–70, 2014.

[24] Martin Maier, Mahfuzulhoq Chowdhury, Bhaskar Prasad Rimal, and Dung Pham Van. The tactile internet: vision, recent progress, and open challenges. *IEEE Communications Magazine*, 54(5):138–145, 2016.

[25] Kwang Soon Kim, Dong Ku Kim, Chan-Byoung Chae, Sunghyun Choi, Young-Chai Ko, Jonghyun Kim, Yeon-Geun Lim, Minho Yang, Sundo Kim, Byungju Lim, Kwanghoon Lee, and Kyung Lin Ryu. Ultrareliable and low-latency communication techniques for tactile internet services. *Proceedings of the IEEE*, 107(2):376–393, 2019.

[26] Anutusha Dogra, Rakesh Kumar Jha, and Shubha Jain. A survey on beyond 5g network with the advent of 6g: Architecture and emerging technologies. *IEEE Access*, 9:67512–67547, 2021.

[27] Davit Harutyunyan, Nashid Shahriar, Raouf Boutaba, and Roberto Riggio. Latency-aware service function chain placement in 5g mobile networks. In *2019 IEEE Conference on Network Softwarization (NetSoft)*, pages 133–141, 2019.

[28] Vincenzo Eramo, Mostafa Ammar, and Francesco Giacinto Lavacca. Migration energy aware reconfigurations of virtual network function instances in nfv architectures. *IEEE Access*, 5:4927–4938, 2017.

[29] Yicen Liu, Yu Lu, Xi Li, Zhigang Yao, and Donghao Zhao. On dynamic service function chain reconfiguration in iot networks. *IEEE Internet of Things Journal*, 7(11):10969–10984, 2020.

[30] Somayeh Kianpisheh and Roch H. Glitho. Joint admission control and resource allocation with parallel vnf processing for time-constrained chains of virtual network functions. *IEEE Access*, 9:162553–162571, 2021.

[31] Keigo Akahoshi, Fujun He, and E. Oki. Service deployment model with virtual network function resizing. *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2021.

[32] Milad Ghaznavi, Aimal Khan, Nashid Shahriar, Khalid Alsubhi, Reaz Ahmed, and Raouf Boutaba. Elastic virtual network function placement. In *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pages 255–260, 2015.

[33] Khalid Ali and Manar Jammal. Proactive vnf scaling and placement in 5g o-ran using ml. *IEEE Transactions on Network and Service Management*, 21(1):174–186, 2024.

[34] Amir Varasteh, B. Madiwalar, Amaury Van Bemten, W. Kellerer, and Carmen Mas-Machuca. Holu: Power-aware and delay-constrained vnf placement and chaining. *IEEE Transactions on Network and Service Management*, 18:1524–1539, 2021.

[35] Vincenzo Eramo and Tiziana Catena. Application of an innovative convolutional/lstm neural network for computing resource allocation in nfv network architectures. *IEEE Transactions on Network and Service Management*, 19(3):2929–2943, 2022.

[36] Jia Chen, Xin Cheng, Jing Chen, and Hongke Zhang. A lightweight sfc embedding framework in sdn/nfv-enabled wireless network based on reinforcement learning. *IEEE Systems Journal*, 16(3):3817–3828, 2022.

[37] Narges Gholipoor, Hamid Saeedi, Nader Mokari, and Eduard A. Jorswieck. E2e qos guarantee for the tactile internet via joint nfv and radio resource allocation. *IEEE Transactions on Network and Service Management*, 17(3):1788–1804, 2020.

[38] Eugina Jordan. Open ran functional splits, explained, 2021. `https://www.5gtechnologyworld.com/open-ran-functional-splits-explained/` [Accessed: (01/03/2024)].

[39] Meng Wang, B. Cheng, Shangguang Wang, and Junliang Chen. Availability- and traffic-aware placement of parallelized sfc in data center networks. *IEEE Transactions on Network and Service Management*, 18:182–194, 2021.

[40] Riccardo Guerzoni, Zoran Despotovic, Riccardo Trivisonno, and Ishan Vaishnavi. Modeling reliability requirements in coordinated node and link mapping. In *2014 IEEE 33rd International Symposium on Reliable Distributed Systems*, pages 321–330, 2014.

[41] E. F. Moore. The shortest path through a maze. In *Proceedings of an International Symposium on the Theory of Switching, Part II*, pages 285–292, Cambridge, MA, 1959. Harvard University Press.

[42] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction To Algorithms*. MIT Press, 2001. Greedy Algorithms.

[43] Gurobi optimizer. `https://www.gurobi.com/solutions/gurobi-optimizer/` [Accessed: (01/03/2024)].

[44] Nattakorn Promwongsa, Amin Ebrahimzadeh, Roch H. Glitho, and Noel Crespi. Joint vnf placement and scheduling for latency-sensitive services. *IEEE Transactions on Network Science and Engineering*, 9(4):2432–2449, 2022.