QoS in Next-Gen Networks: Investigating Resource Management in Network Slicing and Co-Existence with Wi-Fi

Thesis

Submitted in partial fulfillment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY

by

Saibharath S
ID No. 2017PHXF0100H

Under the Supervision of

Dr. Sudeepta Mishra
and
Under the Co-Supervision of

Dr. Chittaranjan Hota

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
2023

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

Certificate

This is to certify that the thesis entitled, "**QoS in Next-Gen Networks: Investigating Resource Management in Network Slicing and Co-Existence with Wi-Fi**" and submitted by **Saibharath S** ID No. **2017PHXF0100H** for the award of Ph.D. degree of the institute embodies the original work done by him under our supervision.

Smishra

Supervisor
Dr. Sudeepta Mishra
Asst. Professor,
Dept. of CSE,
Indian Institute of Technology Ropar.
Date: 18-10-2023

Co-Supervisor
Dr. Chittaranjan Hota
Senior Professor,
Dept. of CS&IS,
BITS-Pilani, Hyderabad Campus.
Date: 18-10-2023

Declaration of Authorship

I, **Saibharath S**, declare that this Thesis titled, ‘QoS in Next-Gen Networks: Investigating Resource Management in Network Slicing and Co-Existence with Wi-Fi’ and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: S. Saibharath

Date: 18-10-2023

Acknowledgements

I express my deep gratitude and heartfelt thankfulness to my supervisors, Dr. Sudeepta Mishra and Dr. Chittaranjan Hota. Their valuable advice, direction, and continuous encouragement acted as an endless source of motivation throughout the entire course of my thesis work.

I am grateful to my DAC members, Prof. G.Geethakumari and Dr. Paresh Saxena, for their insightful observations and recommendations to significantly enhance this research.

I am thankful to Prof. N L Bhanumurthy, Head of the Department of Computer Science and Information Systems, for all the essential support during my thesis work.

I am thankful to Prof. Tathagata Ray, Departmental Research Committee Convenor, for his encouragement and support.

I express my gratitude to Prof. G. Sundar (Director, BITS Pilani, Hyderabad Campus). I would also like to thank Prof. Venkata Vamsi Krishna Venuganti (Associate Dean, Academic-Graduate Studies and Research Division) and his team's steady support during my Ph.D. work.

I thank my colleagues Gourish Goudar and Anirudh Kasturi for their backing and motivation. I express my sincere gratitude to my parents, brother, in-laws, wife, and kid for their enduring support.

Abstract

Next-generation networks such as 5G and beyond offer faster connection speeds and lower latency, enabling more advanced applications, enhanced reliability, and more efficient spectrum usage. However, several important sub-problems need to be addressed to fully realize their potential. The primary focus of this thesis is on network slicing, which provides customized Quality of Service (QoS) requirements to different use cases. Resource allocation strategies and slicing processes are critical to adapting to varying levels of QoS requirements. In this regard, we propose a solution that employs multiple attribute decision-making with analytical hierarchy processing to maximize stakeholder objectives such as operating efficiency, network performance, and timeliness in QoS-based resource allocation [1]. Additionally, we utilize enhanced Dinic algorithms to compute the maximum possible flows in the network.

Mobile network operators must support varied use cases such as mission-critical traffic, low latency, and ultra-reliable augmented reality. Hence, traffic classification at the Base Station (BS), Radio Access Networks (RAN) partitioning, and application-aware routing are required to meet QoS and Service Level Agreements (SLAs). We study joint QoS and energy savings-based resource allocation strategies [2] and scheduling in network slicing. Prioritizing traffic requests are achieved through standard ML regressors such as gradient boost and random forest. We also investigate QoS-based task offloading at edge servers in network slicing [3]. Another important aspect of 5G is deploying multiple small cells in the micro infrastructure to densify the network. To achieve this, we propose a Swap-based Load Balancing (SLB) with the biasing method that minimizes the load imbalance between access points and maximizes the signal strength of the connected devices. Our results show that the SLB with biasing method reduces the load imbalance by a factor of 22.24% compared to other state-of-the-art algorithms, improving both load imbalance in access points and signal quality among users [4]. Next we examine application-aware routing, which involves identifying, measuring, monitoring, and mapping the application traffic QoS requirements to a specific data path in an SLA class. These routines are applied as add-ons to existing standard routing algorithms, and we study their benefits in terms of bandwidth, latency, and jitter.

Lastly, fair co-existence with Wi-Fi would be necessary for 5G to expand the network spectrum capacity in the unlicensed spectrum. In this regard, we investigate the effects of Wi-Fi selfish users on the cellular network and propose counteraction strategies and network configurations to attain fair co-existence. We deep dive into these side effects in the Duty cycle and Listen-Before-Talk Medium Access Control (MAC) based approaches. Overall, our work deep-dives into QoS aspects in network slicing and resource management to achieve stakeholder objectives. We discuss our proposed solution and its implication in QoS-driven resource allocation, load balancing at the base station, task offloading at edge servers, and co-existence with Wi-Fi.

Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations

QoS	**Q**uality **o**f **S**ervice
SLA	**S**ervice **L**evel **A**greement
4G	**F**ourth **G**eneration of cellular technology
5G	**F**ifth **G**eneration of cellular technology
Wi-**F**i	**W**ireless **F**idelity
NS	**N**etwork **S**licing
CSMA	**C**arrier **S**ense **M**ultiple **A**ccess
RTS	**R**equest **T**o **S**end
CTS	**C**lear **T**o **S**end
MIMO	**M**ultiple-**I**nput- **M**ultiple-**O**utput
mmWave	**m**illi**m**eter**W**ave
SDN	**S**oftware-**D**efined **N**etworking
NFV	**N**etwork **F**unctions **V**irtualization
LTE	**L**ong-**T**erm **E**volution
RAT	**R**adio **A**ccess **T**echnology
SLB	**S**wap **B**ased **L**oad **B**alancing
LBT	**L**isten-**B**efore-**T**alk
DC	**D**uty-**C**ycling
CC	**C**ognitive **C**ycles
BS	**B**ase **S**tation
CR	**C**ognitive **R**adio
ML	**M**achine **L**earning
S1AP	**S1** **A**pplication **P**rotocol
EE	**E**nergy **E**fficiency
VB	**V**irtual **B**ackbone

Chapter 1

Introduction

The next generation of wireless communication technologies is a transformative advancement in the world of mobile communication, promising faster data speeds, lower latency, and higher connectivity than ever before. As more and more devices become connected to the internet and consume large amounts of data, the importance of ensuring Quality of Service (QoS) becomes increasingly critical [5].

In this chapter, we begin by studying the QoS specifications and the history of QoS from the initial releases. Then, the chapter delves into the need for QoS investigation in 5G, the fifth-generation technology standard for cellular networks.

We then investigate the two prominent use cases for 5G technologies: network slicing and 5G-Wi-Fi co-existence in the unlicensed spectrum. Network slicing involves creating multiple logical and virtualized networks over a common multi-domain infrastructure based on custom QoS requirements. The chapter discusses important concepts such as network slicing, resource allocation, RAN partitioning, and scheduling to meet custom QoS requirements. In 5G-Wi-Fi co-existence, both technologies must coexist fairly in the unlicensed spectrum to avoid interference and ensure optimal performance for both.

Next, this chapter explores the role of QoS while load balancing in the 5G micro infrastructure. Due to the concentration of mobile devices around specific Access Points (APs), these APs can become hotspots, leading to intermittent connectivity issues. This chapter highlights the need for proper load balancing and device association to avoid these issues and ensure a seamless experience. Additionally, investigating QoS attributes, such as signal strength during the association process, is necessary to support optimal performance.

Then, the chapter explores the implications of application-aware routing to support QoS and Service Level Agreements (SLAs). This involves routing traffic based on the specific requirements of each application or user to ensure the desired level of service is provided. The chapter highlights the importance of considering system requirements in application-aware routing to support QoS and SLAs.

Finally, the chapter provides a comprehensive overview of the problem statement addressed in this thesis. We detail our overall contributions and provide a Chapter-wise thesis outline of our work.

1.1 QoS Specifications

The concept of QoS earmarked from the 3GPP Release 1997 (R97) [6] and has evolved in R99 [7], R4, and R5/6 [8]. Through the QoS profile suite, we can delineate the elements of QoS.

> A *Quality of Service (QoS) profile suite* is a predefined collection of performance and service level profiles within a network or system. These profiles specify parameters like bandwidth, latency, and priority levels, tailored to meet specific application or service requirements, ensuring consistent and efficient network resource allocation.

An aggregated base station QoS profile suite comprises attributes such as traffic class, guaranteed bit rate, maximum bit rate, residual bit error ratio, transfer delay, delivery order, source statistics descriptor and signalling overhead, and allocation and retention priority [9].

1.1.1 QoS Support in 4G

The 4G architecture consists of the Core Network (CN), Evolved Universal Terrestrial Radio Access Networks (E-UTRAN), and Mobile Equipment (ME). The concept of QoS in 4G is based on bearer services. Bearer service is a transmission path from the radio interface through the network infrastructure. These are allocated dynamically or through subscription-based mechanisms. The QoS features in 4G [10],[11] are described below:

Access Class Bearing (ACB): This feature helps in terminal class prioritization, especially when the network is overloaded or in emergencies. It constitutes Access Classes (AC) ranging between 0 and 15. AC 0-9 is for all terminals. When the network is massively overloaded, the regular access classes are either barred or set with a particular blocking probability.

Allocation and Retention Policy (ARP): ARP governs the pre-emption capability of traffic flows and pre-emption vulnerability prediction. ARP is a constituent in the subscribed QoS profile for the default bearer.

QoS Class Identifier (QCI): QCI ranging from 1-254 is used to prioritize scheduling and to queue admitted values. These inbuilt fields help us to track the priorities. However, standardization is in progress, and there is a need for a special service layer that can understand different user behaviour like mission-critical users and enterprise business users [12].

1.1.2 QoXphere - QoS Management model

QoXphere, a QoS management model, constitutes various layers:

i) Intrinsic layer - key performance parameters contribute towards Key Performance Indicators (KPIs) are identified to evaluate the Network Performance (NP) for different Classes of Service (CoS).

ii) In the Perceived layer, the Key Quality Indicator (KQI) is identified based on the problem set. The quality perceived by the users is measured through the QoE by comparing the required, offered, and delivered QoS to the customers.

iii) Assessed QoS provides the key risk indicators, user satisfaction level, SLAs, and attrition rate of users through churn probability.

iv) The Business layer concentrates on Key Business Objectives (KBO) such as revenue and margin, average revenue per user, and operational efficiency.

An enhancement to QoXphere [13], a global QoS management model, has been proposed to enhance the next-generation networks. It involves a case study involving Wi-Fi technology using probes in universities, residential areas, and commercial buildings. Basic ML techniques [14] are ideated for removing anomalies through unsupervised learning and inferring relevant KQI for users through inductive supervised learning.

1.1.3 Differentiated Services and QoS Class Identifier

The S1 Application Protocol ($S1AP$) provides signalling [15] between E-UTRAN and Evolved Packed Core (EPC). Attributes like QCI, allocation and retention priority, and the pre-emption vulnerability function are present in s1ap messages. During E-UTRAN Radio Access Bearer (E-RAB) setup or modification, packets include necessary QoS parameters. These packets are sent by the Mobility Management Entity (MME) and used by the Evolved Node B (eNodeB) to assign resources for one or several E-RABs.

The 3rd Generation Partnership Project (3GPP) has prompted cellular networks to enable QoS-based architecture. This work [16] provides mapping recommendations from Quality Class Identifier (QCI) to Differentiated services (DiffServ) and vice versa. However, the approach is partially incompatible due to the following reasons:

- QCI cannot be strictly mapped to Diffserv as one-to-one mapping is lacking. Diffserv doesn't support all the features supported by QCI, and hence a group of QCI is mapped through a poor approximation.
- In the QCI structure, multiple dimensions cross one another. For example, IP Multimedia Subsystem (IMS) signalling (QCI-5) with high priority and low tolerance levels [16], but being Non-Guaranteed Bit Rate (GBR) belongs to the signalling category. Conversation voice (QCI-1) has a lower priority and high tolerance than IMS signalling yet benefits from a GBR, which leads to inconsistency among them.
- QCI represents flows in terms of multi-dimensional needs. The multi-dimensional logic is different between QCI and Diffserv.

Subsequently, for providing end-to-end capabilities, we need unified traffic classes represented through acceptable delay, throughput, jitter, etc.

1.2 Need for QoS Investigation in Next-Gen Technologies

5G has introduced state-of-the-art technology suites and a mindful culmination of prospective techniques and tools from the prior generation of cellular networks. 5G is crucial in automated vehicles and the Internet of Things (IoT).

The researchers have already established that 5G supports minimum viable services such as a) Extreme Mobile BroadBand (xMBB), b) Massive Machine Type Communications (M-MTC), and c) Ultra-Reliable Machine Type Communications (U-MTC) [17]. On the other hand, the authors [18] cite that although power infrastructure and sophisticated technologies were available in the previous generation, the network was purposed sub-optimally, which they attribute to orthogonal business requirements. The quality of Experience (QoE) of end-users often conflicts, leading to reduced user satisfaction and substandard Quality of Business (QoBiZ).

5G adopts various designs and solutions from physical to application layers to accomplish these service requirements and manage these challenges. Some prominent technologies in the physical layer in 5G are Massive Multiple-Input Multiple-Output (MIMO), mmWave MIMO, and Non-Orthogonal Multiple Access (NOMA). Similarly, the transport and core networks need to be programmable, intelligent, powered by Artificial Intelligence and Machine Learning (ML) techniques, provide virtualization, and enable cloud technology. The Software Defined Networking (SDN) principles in transport and application layers, mmWave backhaul with multiband boosters, and mmWave long-haul could play a crucial role.

The 5G network consists of three central units [19]: a) RAN is built through millimetre-wave and massive MIMO to increase bandwidth and reduce latency, b) Core network relies on software solutions, virtualization, and cloud adoption, c) SDN-enabled transport network for better traffic engineering, cloud integration, network intelligence, and monitoring. Backhaul transportation could be either wired or through wireless channels by employing mmWave MIMO technologies.

The authors [20] consider the 5G mobile platform to be a conglomeration of various wireless Radio Access Technologies (RATs), encompassing multiple radio access and wireless technologies. Hence, the authors regard that the design solutions for interoperability and seamless Quality of Service (QoS) experience for end-users while handling the explosive growth of mobile traffic volumes need to be examined. Similarly, the 5G core and transport network has widespread softwarization and seeks end-to-end path optimization that targets the application's QoS requirements and Service Level Agreements (SLAs).

1.2.1 Importance of QoS-based Scheduling in 5G

There are multi-fold reasons why QoS-based scheduling in 5G [21],[22] should be investigated further,

- We know that 5G provides ubiquitous connectivity, ultra-low latency, better reliability, and high data rates. Scheduling becomes paramount to afford such high QoS guarantees and satisfy the SLAs.
- 5G builds a platform for different wireless technologies to collaborate. This environment deals with a collection of multiple RATs with diverse implementations. Hence, network packet scheduling needs to consider the architecture variability across technologies.
- Scheduling would play a crucial role in dealing with Ultra-Reliable Low Latency Communications (URLLC) and carrier aggregation in automated vehicles and the Internet of Things.
- The 5G platform supports xMBB, M-MTC, and U-MTC communications. It also provides mission-critical services such as push to talk feature in the military.
- 5G services are orchestrated through different deployment models like Network slicing, Cloud Radio access networks, etc.,

To realize seamless communication and QoS guarantees, radio access technologies, resource allocation, and network packet scheduling would have to perform a significant part and hence, needs to be analyzed.

1.3 Network Slicing

Full-fledged digital transformation radically changes how services are delivered in the Fifth Generation of cellular networks. The potential of cloud computing, software-defined networking, and network function virtualization with existing network infrastructure is powering and merchandising 5G toward advanced and unexplored products and services for commercial establishments. For achieving these use cases in a scalable, elastic, and cost-efficient manner, Network Slicing (NS) [23],[24] is increasingly becoming popular.

On top of the common physical network infrastructure, multiple virtual network links are created, and these are assigned to slices to cater to the SLA and QoS requirements of the end-users. A Network Slice (NS) is an independent, end-to-end logical network. It could comprise the user device, last-mile connectivity, core network, and transport layers. NS is a formation of automated virtual logical networks over physical substrate networks to provide customers with specific services.

> *Network slicing* is a technique in modern networking, particularly in 5G and beyond, where a single physical network is divided into multiple virtual networks, or ”slices.” Each slice is isolated and customized to meet specific performance, security, and resource requirements, catering to diverse services and applications on the same infrastructure [25].

1.3.1 Driver for Network Slicing

The variety of use cases serviced by a network expanded with 4G has even more inflated in 5G [26]. The use case could be mobile broadband, fixed wireless, or machine-type communications. These are the drivers for NS to support customized offerings with flexibility, reduced risk, and operating cost efficiency. Diverse QoS can be supported by each offering in isolation and helps the network operators to monetize it appropriately.

NS offers multi-dimensional capabilities: i) Network as a Service (NaaS) is a cloud computing service model that provides users access to a virtualized network infrastructure. As an end-user customer, NS should enable NaaS to meet SLAs. ii) As a system user, the user strives for better resource management, which NS realizes through traffic steering and resource partitioning. iii) As a business-management stakeholder, NS via automation and isolation should render operating efficiency and reduced Time To Market (TTM) [27].

1.3.2 Network Slicing Architecture

The architecture for commercial setup shown in Fig. 1.1 is organized through the four layers. a) Shared Infrastructure Layer - The lowest layer which provides the network hardware infrastructure. b) Slices: For instance, presence of Core and RAN logical networks, c) Management and Orchestration Layer - provides slice provisioning, analytics, and 3 M’s (Manage, Monitor, and Metrics Evaluation), and d) Top layer exposes NaaS, customized offerings, and services to the consumer [28]. Here, the control plane describes how the packets must be handled in the data plane. To run virtual networks on top of the shared infrastructure, we have an extra optional layer to create slices and operate as a transparent proxy controller between the physical switches and logical controllers. The different layers should work end-to-end and have inherent challenges. Let’s review the types of slices and challenges in the sub-sections below.

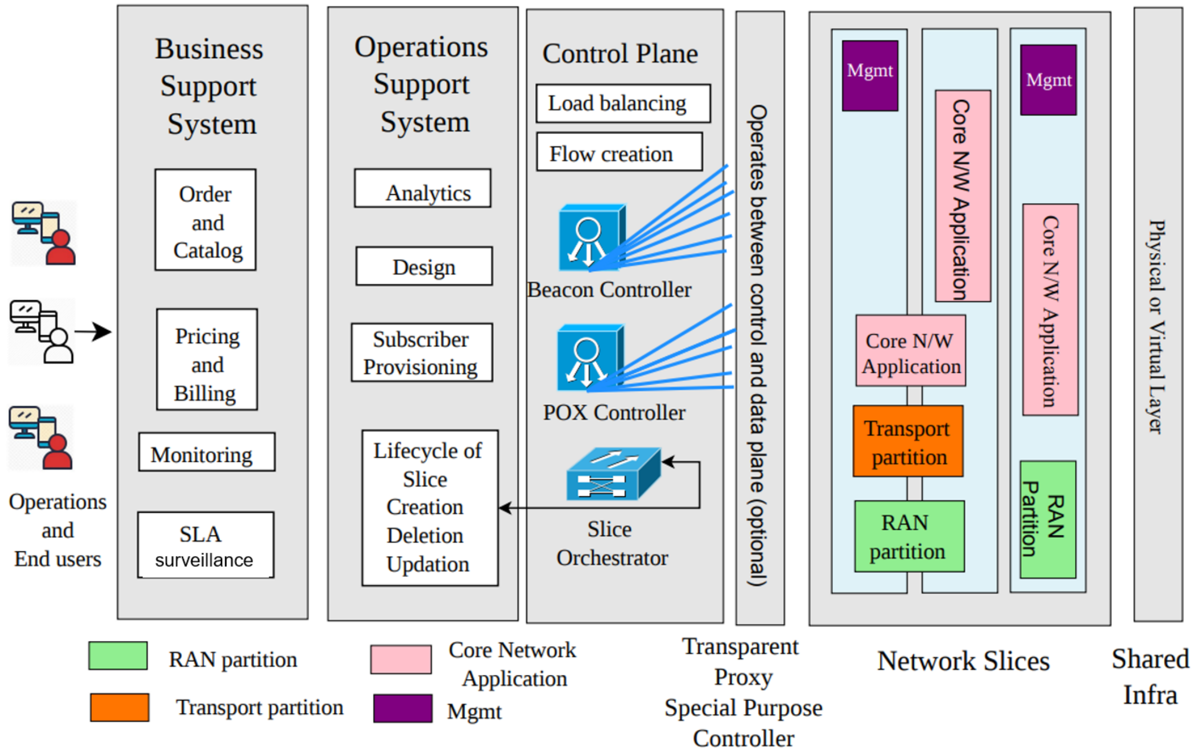


FIGURE 1.1: Network Slicing Architecture [2]

1.3.3 Radio Access Network and Core Network slices

RAN allocates the radio resources in the network. Radio resources are limited. Hence, we need a slicing policy that helps share, reserve, and isolate the resources between slices. RAN's primary functions include resource management, observability, and interconnecting with the core network slices. RAN has to provide slice-aware allocation [29] while connecting to core network nodes. It should also support the concept of slice identifiers in 4G and 5G. RAN slicing builds on top of dynamic radio resource partitioning and works in conjunction with existing QoS principles. The idea is to start today and evolve tomorrow from existing infrastructure, such as 4G, to 5G.

The core network's slices are flexible and powered by Network Functions Virtualization (NFV) and SDN technologies. It can provide both isolation and share network functions across slices. 5GC extends the capability of current EPC in 4G [30], and it unlearns its shortcomings, making it a scalable, flexible, and powerful framework to control slices.

1.3.4 Empowering Network Slicing with Automation

With billions of connected devices with varied network requirements, manual maintenance and upgrade of NS policies can introduce bugs. Automation can mitigate these risks, which is highly recommended for building and maintaining large-scale systems [31]. The upgrades can be smooth, and slices can be delivered on-demand with agility through the orchestration and automation in 5G. It can be built through model-driven orchestration using templates. Automation can help in the authorization of customers to slices and assurance.

1.3.5 Network Slicing Example

NS brings in new business opportunities [32]. Let's consider a residential area with net-enabled smart homes. The residents will be using mobile broadband connectivity through their smartphones. Each residence would possess fixed wireless access, which the individuals would share. Entertainment systems like TeleVision (TV) and gaming require high-speed fixed-wired broadband. Home security systems and detectors need low latency and highly reliable machine-type communications. Slice(s) can handle the above different network services for a residential area or town. The network slice can prioritize critical services and high-priority traffic flows during peak load and congestion.

Another example is a slice that can be designed for a specific commercial service, such as extreme mobile broadband service. It can also be drafted for a scenario composed of co-existing multiple diverse network services. For example, a smart, well-connected metro-operated bus would have i) commuters, who could use fixed wireless access for their mobile-internet usage, ii) embedded devices like Closed Circuit Televisions (CCTVs), for monitoring the surrounding environment, and these require medium bandwidth for real-time video streaming capabilities, iii) the automated transit teller machines, which notify passengers in the current and next bus stops, require the bus stops to be Global Positioning System (GPS) enabled and support low latency transactions for accuracy, and iv) the communication between remote control centre, automated vehicle driving system or manual bus operator, and alarms for emergency purposes are mission-critical in nature. NS puts forward a structure, operation, and revenue model in the above business case studies for enterprises.

The notable challenges [33] in Network slicing are as follows.
a) Formal assurance in the allocation of slices.

b) Consistently and concurrently meeting SLAs in NS.

c) Seamless automation and end-to-end management of slices.

d) Isolation, observability, and integration of RAN, Network, and Transport slices.

1.4 Co-existence of Wi-Fi and 5G in the unlicensed spectrum

In the licensed spectrum [34], many small cells close to each other are deployed in dense networks. These cells are activated based on the high traffic demands in a specific period. However, deploying such a large number of small cells increases the total cost of operation.

5G can operate in the unlicensed spectrum to expand its network capacity. The 5G New Radio-Unlicensed (NR-U) could advance the private networks and alleviate spectrum constraints to deliver better performance. This unlicensed spectrum provides more uplink and downlink allocation, bandwidth, and frequency bands in 5G to satisfy its ultra-dense and scalability requirements.

5G NR-U can provide both license-assisted and standalone use of unlicensed spectrum, sometimes referred to as Anchored and Standalone NR-U, respectively.

These features have unlocked new opportunities for the industrial IoT and this greenfield spectrum provide flexible ways to apply in indoor and outdoor environments. Furthermore, critical use cases like Time Sensitive Networking and industrial IoT can use synchronized sharing, multi transmission-reception points (TRP) with coordinated multipoint communications (CoMP) deployed with NR-U in controlled settings.

NR-U with synchronized sharing can reduce latency and improve fairness to all access technologies within the same spectrum. The anchored and standalone NR-Us will help MNO to deliver better performance [35].

Wi-Fi is a prominent wireless technology operating in these unlicensed bands. The provisioning of unlicensed spectrum at 5GHz can be utilized by both cellular and Wi-Fi users, leading to better coverage and more frequency bands for operators. Wi-Fi Cellular Co-existence is another promising deployment scenario. However, it has its own challenges.

Due to the proximity of cellular and Wi-Fi spectrum channels, utilization of both Wi-Fi and NR can cause interference during operation. The co-existence of 5G New Radio (NR) and Wi-Fi devices in these bands lack the means of communication for negotiation and coordination among them.

Listen-Before-Talk (LBT) and Duty-Cycling (DC) are two standard Medium Access Control (MAC) mechanisms that are applied to enable co-existence. Due to a lack of coordination, network utilization could become unfair with the existence of selfish devices. The selfish users could maximize their throughput and affect other Wi-Fi and NR users consecutively.

1.5 Need for Load Balancing at the base station in 5G micro infrastructure

5G uses a millimetre wave (mmWave) spectrum to satisfy the ultra-high bandwidth demands in urban areas. However, mmWave suffers from a low range. So, the operators tend to densify their networks with small cells to provide ubiquitous connectivity and reliable coverage. Some of these small cells, such as micro or picocells, could handle the bulk of wireless network traffic while other cells remain idle. As time progresses, more mobile devices throng towards the overloaded micro or pico cells, thus creating hotspots. It leads to experiencing intermittent, unstable connectivity and high packet jitter in these small cells. We could re-associate some wireless devices connected from overloaded to the reachable underutilized cells in the vicinity to overcome this effect. Despite being obvious to suggest, load balancing involves several challenges when, where, and how to perform re-association because choosing the wrong movement might hurt network performance. In this work, we aim to reduce load imbalance through traffic distribution and potentially enhance the micro and picocell's performance and client experience.

The load on the micro or picocell quantifies its usage in a 5G network. It is defined through several metrics in the literature. The primitive one is the number of devices connected to the cell. By default, devices connect to the AP, which offers the best Received Signal Strength Indicator (RSSI). An alternative to signal strength is channel utilization, which indicates the residual bandwidth. The other choice is throughput measurement between the 5G cells and the devices described by the number of packets transmitted in a timeslot.

Reactive Load Balancing (LB) methodologies are preferred as they minimize the re-association among small cells. Broadly, LB is classified into the below approaches,

- Admission Control: Devices on arrival are admitted based on the utilization and remaining capacity.
- Association Management: LB decisions are primarily controlled through the 5G APs and higher management planes.
- Transmission Range control: The cell's transmission ranges are usually artificially reconfigured, causing a weakened signal beyond the selective serving distance [36]. Unintentionally, it could block a device if it doesn't find another cell and end up as an orphan.
- Association Control: A device-centric approach where the mobile terminal collects and chooses from macro, micro, or picocells based on the metrics.

1.6 Application-Aware Routing

There is a need to holistically examine and monitor the QoS performance of the network and its devices through different layers to meet stakeholder objectives. The application-aware routing is a method to administer the network from an application point of view. Here, the prime focus is applying QoS constraints and the maximization of relevant utility functions in SLAs.

Application-aware routing [37] tracks network and path characteristics along the data plane and utilizes the gathered information to compute the traffic's optimal data paths.

The characteristics comprise QoS parameters such as latency, bandwidth, packet loss, jitter, and link load. We have several benefits in applying application-aware routing to the network ecosystem [38] [39], such as:

- Apart from the standard route prefix and link-state information applied in conventional traffic, the network traffic path should support the various levels of latency, bandwidth, and other QoS parameters described in an application SLA.
- Dynamic load balancing based on the monitored load of links leads to reduced network costs.
- Application-aware routing could increase the performance of the application without upgrading hardware and software components.

1.7 Problem Description

With high-end requirements and growing traffic consumption from mobile devices, we have identified the following problem statements, which are paramount for providing a seamless experience, optimal network utilization, and QoS for customers.

- 5G would conglomerate multiple tiers and RATs [20]. There is a need for traffic classification algorithms [12] and identifying the right set of attributes for segregating the packets in 5G for improved differentiated services and to meet the QoS requirements.
- It is essential to achieve network requirements in terms of QoS and SLA agreements and as well be energy-efficient. There is a necessity to investigate the joint objective of Energy savings and QoS in the communication networks [20]. Scheduling and resource allocation of traffic for tailored offerings in network slices must be investigated across RAN, transport, and core network layers. The significance has increased with the need to support differentiated services such as mission-critical use cases in 5G.
- 5G micro infrastructure comprising micro and picocells would be pivotal in densifying the network to provide ample coverage. However, a disproportional association of mobile devices with these small cells would cause hotspots and load imbalance. A few micro or picocells suffer from network congestion in such a network. While many others are underutilized, experience lower throughput, and operate below the potential network capacity. To mitigate this drawback, a concrete load balancing policy eliminating hotspots and network congestion and offering better signal strength for mobiles would be essential.
- There is a need for a robust methodology to facilitate the tracking of QoS performance and meet multi-objective QoS metrics. Hence, the application-aware routing should be investigated, where meeting SLA boundaries needs to be monitored. Similarly, task offloading to Multi-access Edge Computing servers in RAN slices needs to be studied under the purview of QoS.
- 5G operates in the unlicensed spectrum to expand its network capacity. When co-existing with a Wi-Fi network, a fair co-existence is essential. However, the presence of selfish nodes could impact the network. It is crucial to investigate standard MAC-based mechanisms such as Listen-Before-Talk and Duty-Cycling on the effect of these selfish users on fair co-existence and QoS of the mobile devices across technologies. The study needs to be

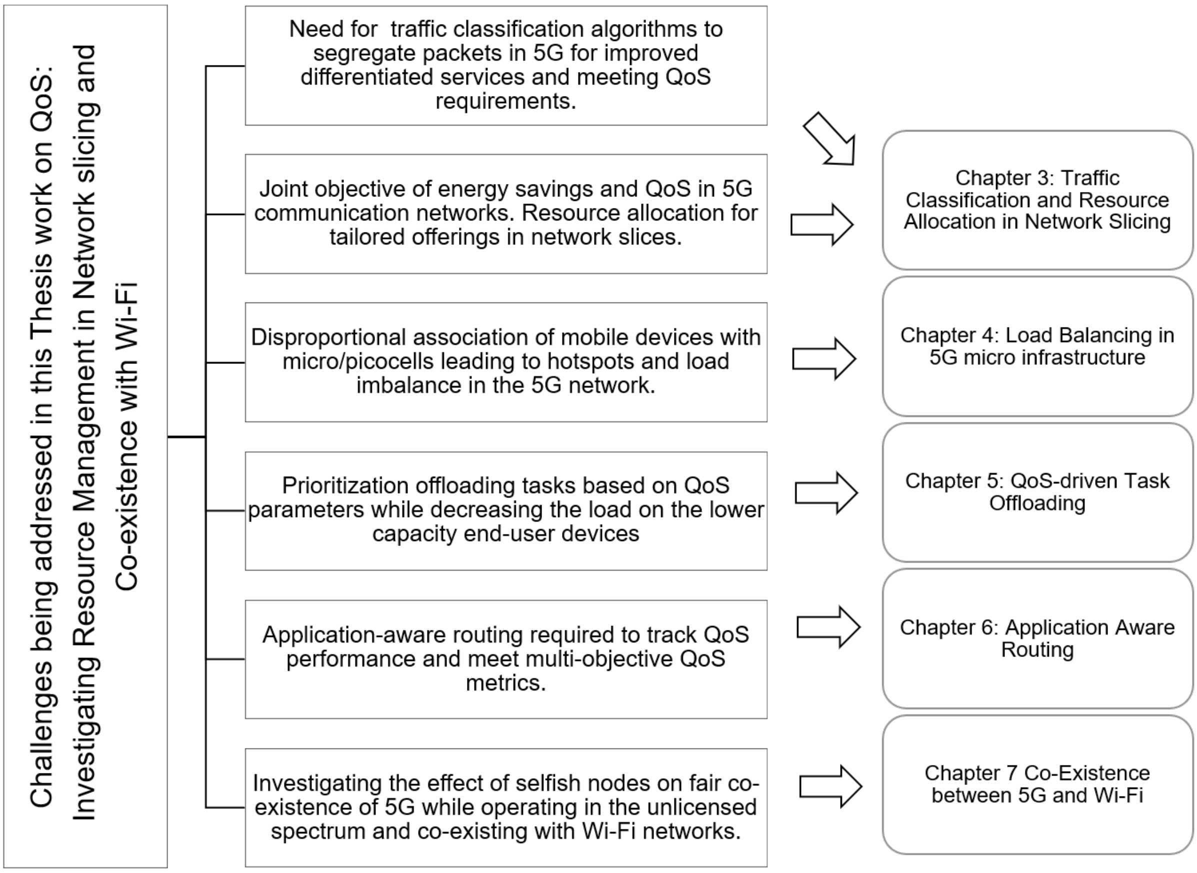

Challenges being addressed in this Thesis work on QoS: Investigating Resource Management in Network slicing and Co-existence with Wi-Fi
Need for traffic classification algorithms to segregate packets in 5G for improved differentiated services and meeting QoS requirements.
Joint objective of energy savings and QoS in 5G communication networks. Resource allocation for tailored offerings in network slices.
Disproportional association of mobile devices with micro/picocells leading to hotspots and load imbalance in the 5G network.
Prioritization offloading tasks based on QoS parameters while decreasing the load on the lower capacity end-user devices
Application-aware routing required to track QoS performance and meet multi-objective QoS metrics.
Investigating the effect of selfish nodes on fair co-existence of 5G while operating in the unlicensed spectrum and co-existing with Wi-Fi networks.
Chapter 3: Traffic Classification and Resource Allocation in Network Slicing
Chapter 4: Load Balancing in 5G micro infrastructure
Chapter 5: QoS-driven Task Offloading
Chapter 6: Application Aware Routing
Chapter 7 Co-Existence between 5G and Wi-Fi

FIGURE 1.2: Challenges and problems being investigated in this work

detailed to cover different network configurations. Possible counteraction mechanisms need to be explored to avert impact on the users.

The challenges being investigated and the corresponding chapters are summarized in Fig. 1.2.

1.8 Contributions

A flow chart to visually describe the technical work and proposed techniques to address the problem are summarized in Fig. 1.3.

Our principal contributions to this research work are:

- For resource allocation of network elements to the slice, we proposed an approach that is influenced by Multiple Attribute Decision Making (MADM), Analytical Hierarchy Processing (AHP) for slice assignment and enhanced Dinic's Maximum Flow Method to find maximum possible virtual paths for allocations [1].

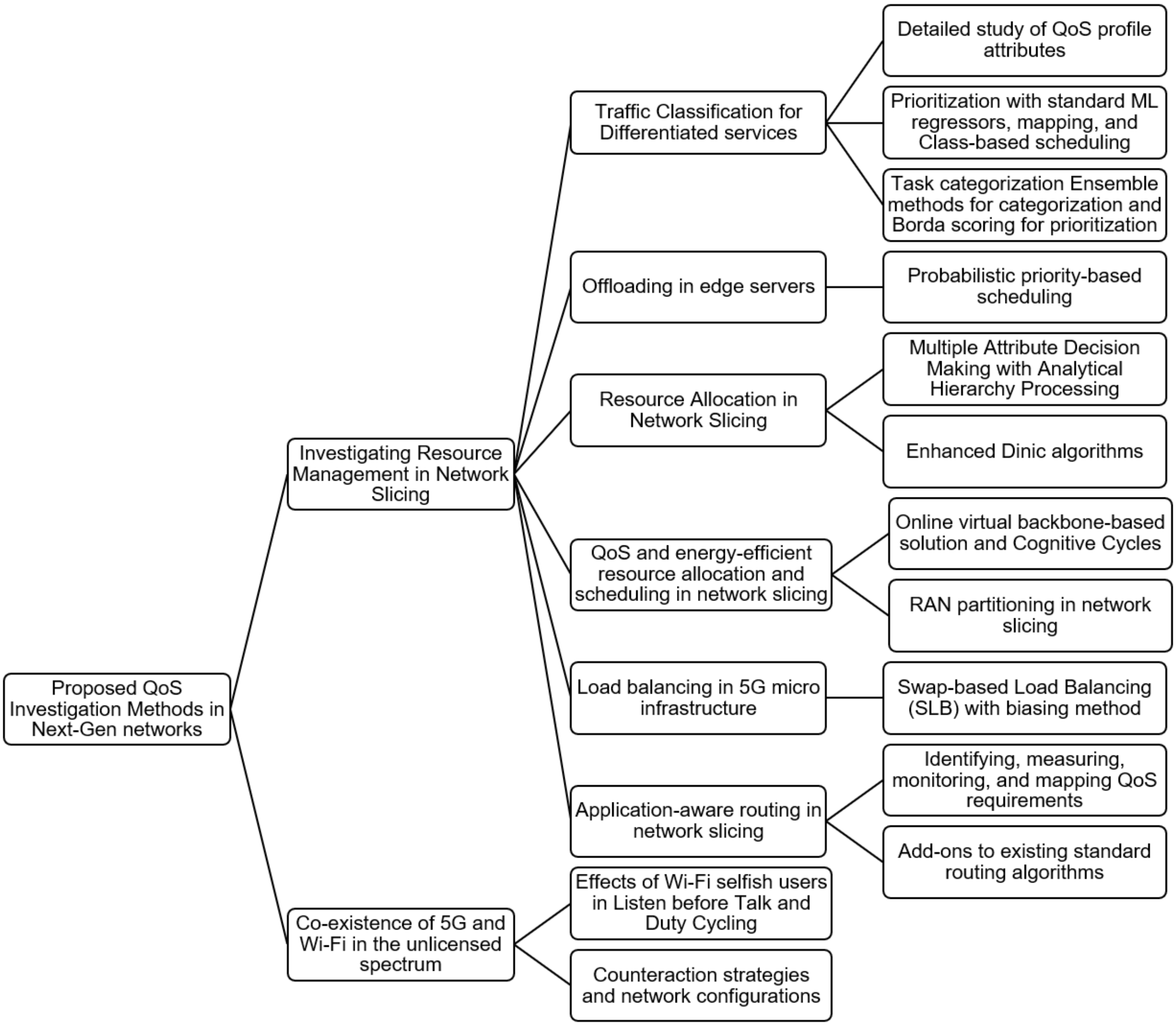

Proposed QoS Investigation Methods in Next-Gen networks
Investigating Resource Management in Network Slicing
Co-existence of 5G and Wi-Fi in the unlicensed spectrum
Traffic Classification for Differentiated services
Offloading in edge servers
Resource Allocation in Network Slicing
QoS and energy-efficient resource allocation and scheduling in network slicing
Load balancing in 5G micro infrastructure
Application-aware routing in network slicing
Effects of Wi-Fi selfish users in Listen before Talk and Duty Cycling
Counteraction strategies and network configurations
Detailed study of QoS profile attributes
Prioritization with standard ML regressors, mapping, and Class-based scheduling
Task categorization Ensemble methods for categorization and Borda scoring for prioritization
Probabilistic priority-based scheduling
Multiple Attribute Decision Making with Analytical Hierarchy Processing
Enhanced Dinic algorithms
Online virtual backbone-based solution and Cognitive Cycles
RAN partitioning in network slicing
Swap-based Load Balancing (SLB) with biasing method
Identifying, measuring, monitoring, and mapping QoS requirements
Add-ons to existing standard routing algorithms

FIGURE 1.3: Proposed QoS Investigation - Methods and Techniques

- A wholesome study of viable QoS attributes for traffic classification and priority class derivation is exemplified. As far as we can say, no other existing work has covered the QoS aspects at this depth and breadth. We illustrate an applied class-based probabilistic priority scheduling through traffic classification results from the wide-ranging QoS attributes [2].

- A customized collective application of the Virtual Backbone (VB) for route path creation and Cognitive cycles (CC) for re-configuration to bring in greater energy efficiency in slices [2].

- To improve load balancing at access points and signal strength issues in micro infrastructure, we propose an extreme Swap-based Load Balancing (SLB) algorithm between APs, which minimizes the load imbalance at cell edges. SLB with biasing delivers both lesser load imbalance in APs and signal quality amongst users [4].

- Task offloading results in the remote execution of tasks, thereby reducing the load on the lower-capacity end-user devices and mobile instruments. We propose an ensemble method for QoS-based task assignment to edge servers in network slices in an SDN setup. An enhanced weighted Borda scoring is presented to categorize the task into its priority class. We present a probabilistic, priority-driven Kafka-topic consumer which schedules the offloaded tasks in the edge containers [3].

- We detail the methodology for tracking, measuring, mapping, and monitoring QoS metrics to achieve application-aware routing.

- Fair co-existence of Wi-Fi and 5G is necessary. We study the side effects of selfish users through channel sensing and acquisition time under different network configurations and medium access mechanisms such as duty cycle and Listen-Before-Talk. We analyze the impact of QoS through metrics such as throughput due to selfishness. We explore counteraction mechanisms in the co-existence setup to overcome selfishness, which was earlier adopted in the Wi-Fi-only network. Lastly, we recommend network configurations and counteraction mechanisms that promote co-existence and shield legitimate users.

1.9 Thesis Outline

The chapter mapping and the QoS attributes being investigated and optimized are represented in Fig. 1.4.

Chapter 2 discusses a literature survey of QoS-based resource allocation and scheduling. We go through the related works on network slicing and how it tries to accomplish tailored offerings for the customers and meet their SLAs. We discuss the existing load-balancing methodologies that devices associate with access points. We also discuss the existing literature on QoS-centric task offloading and application-aware routing methodologies. We iterate the study in another deployment scenario of a 5G-Wi-Fi coexisting network on fair co-existence.

Chapter 3 focuses on traffic categorization and resource allocation in network slicing.

Firstly, the core parameters of Quality of Experience (QoE) to end-user systems, Network Performance, and Operating efficiency are carefully investigated while placing network virtual functions and determining the nodes, links, and resources for assignment across RAN, transport, and core networks.

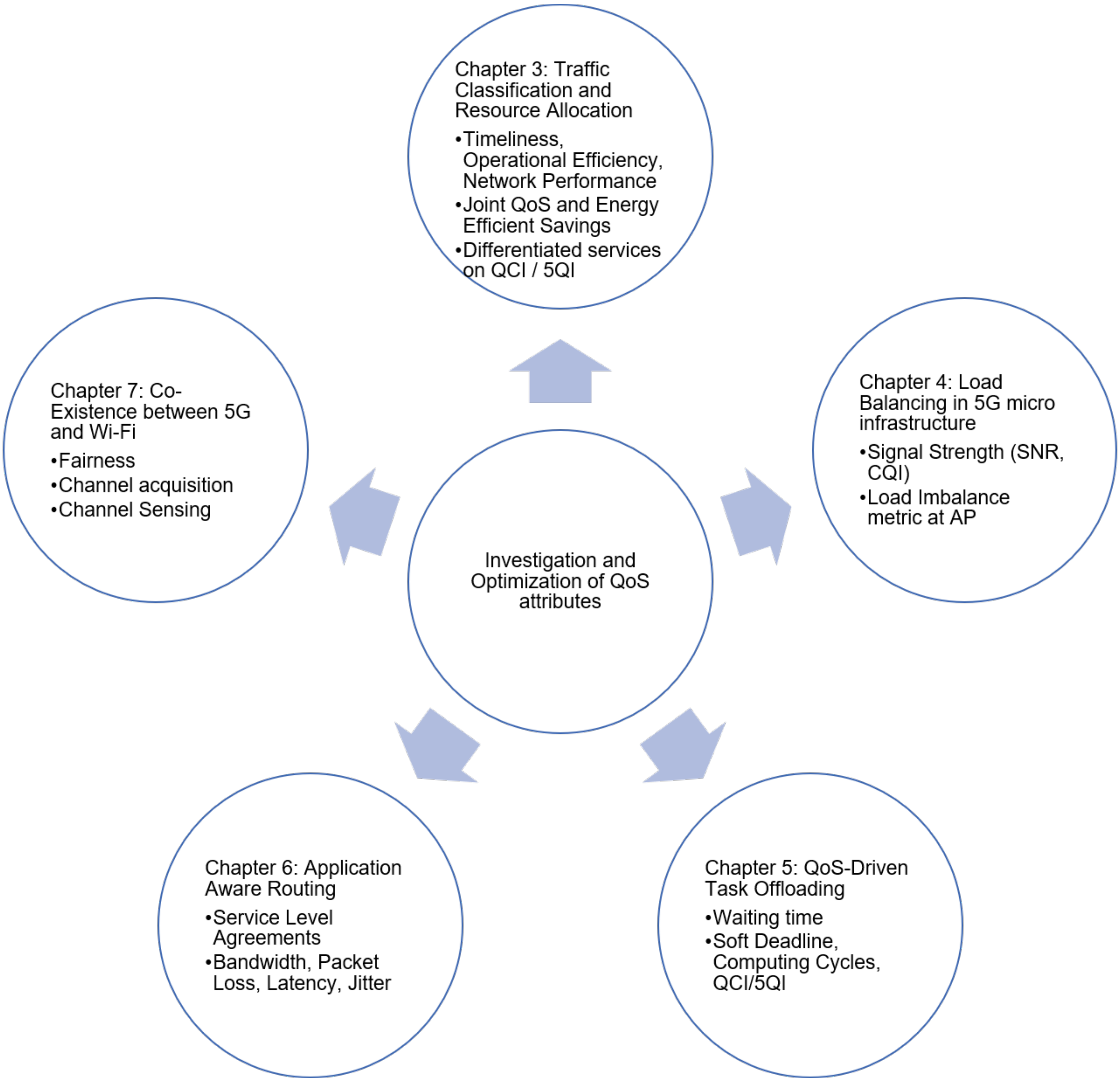

Chapter 3: Traffic Classification and Resource Allocation
•Timeliness, Operational Efficiency, Network Performance
•Joint QoS and Energy Efficient Savings
•Differentiated services on QCI / 5QI
Chapter 7: Co-Existence between 5G and Wi-Fi
•Fairness
•Channel acquisition
•Channel Sensing
Investigation and Optimization of QoS attributes
Chapter 4: Load Balancing in 5G micro infrastructure
•Signal Strength (SNR, CQI)
•Load Imbalance metric at AP
Chapter 6: Application Aware Routing
•Service Level Agreements
•Bandwidth, Packet Loss, Latency, Jitter
Chapter 5: QoS-Driven Task Offloading
•Waiting time
•Soft Deadline, Computing Cycles, QCI/5QI

Figure 1.4: Chapter-wise Investigation and Optimization of QoS attributes

Secondly, it details QoS attributes from S1AP and Internet Protocol (IP). Virtual Backbone with Cognitive Cycles (CC) based approaches are proposed for route allocation in network slices targeting joint QoS and energy efficiency. Standard ML regression algorithms determine the priority used for class-based priority scheduling of packets at RAN.

Chapter 4 presents our work on swap-based load balancing. Here, we first detail the standard one-way traffic distribution, which is based on the signal strength of devices and the load of the access point through the biasing factor. Then, we formalize our proposed two-way extreme load balancing, which targets minimizing the load imbalance factor and enhancing the Channel Quality Index (CQI) and Signal-to-Noise Ratio (SNR) metrics of users. We validate our work through a dataset from an Irish mobile operator and discuss the performance gain over other candidate approaches.

Chapter 5 studies task offloading to Multi-access Edge Computing servers in RAN slices. We present an ensemble categorization and probabilistic prioritized task execution at edge servers. Mininet, Flowvisor, and SDN controllers such as Beacon and POX constitute the SDN setup. The offloaded tasks are categorized and placed at relevant Kafka topics and executed through docker containers.

Chapter 6 examines application-aware routing, which is enabled by measuring, mapping, and allocating paths while meeting SLA boundaries. Firstly, we estimate the key QoS parameters such as latency, packet loss, and jitter of the data path, and we also compute the notional value of the above metrics. The second step is to map each data route against the SLA class definition of users. Finally, we suggest a heuristic QoS KPI-driven path forwarding scheme through SDN. We exemplify the approach through the QoS framework constituting SDN controllers to direct the forwarding plane, special-purpose network slice controller and management unit, and virtualized portable NFV modules to monitor metrics and suggest path allocation.

Chapter 7 discusses Listen Before Talk and Duty Cycle-based medium-access-control approaches for Wi-Fi and 5G Co-existence, respectively. First, we brief about the potential of 5G transmission co-existing with that of Wi-Fi in the unlicensed spectrum. We also study and quantify the effects of selfish users on the QoS of other legitimate users, preferred counteraction methodologies, and network deployment configuration.

Chapter 8 summarizes the work, iterates the specific contributions, and presents future work.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we first thoroughly review resource allocation in NS and the use of cognitive cycles in resource management. We also study energy-efficient resource management of QoS and EE to achieve optimal resource utilization, power savings, and customer satisfaction. We also dive into traffic classification and identify the gaps in the existing research during resource allocation and management stages. We then study related work in load balancing of mobile devices across access points and the impact on their QoS. The existing works on task offloading and application-aware routing are discussed. Finally, we deep-dive into the co-existence of the cellular network with Wi-Fi.

2.2 Resource Allocation in Network Slicing

The NS problem is examined as a min-cost feasible slice embedding [40] problem. The slice may consist of several Virtual Network Functions (VNF) like user and control planes, Base Band Unit, and edge caching. The NS problem is formulated through the Virtual Network Embedding (VNE) Problem, which minimizes the resource utilization cost. The basic NS problem, much similar to VNE, is viewed as the optimal placement of VNFs at resource nodes and link capacity reservations for their interconnections with the extra link and node capacity constraints.

VNE can be viewed as the allocation of virtual resources, and it can be divided into sub-problems: VNoM (Virtual Node Mapping) and VLiM (Virtual Link Mapping) where these virtual nodes and links are mapped to the underneath physical network. VNE problem deals with how these virtual resources can be realized on the substrate resources [41].

VNE problem is best described through three parameters: static vs. dynamic, centralized vs. distributed, and concise vs. redundant. The VNE problem description is a combination of options from the above classes. In a static, centralized, and concise environment, the system doesn't have to consider backup nodes or edges as part of its solution. It is computed in a centralized way in an offline mode.

The VNE problem can be reduced to a multi-way separator problem, which is NP-hard. When given a virtual node mapping, allocating multiple virtual links optimally to physical links is NP-Hard as it reduces to an unsplittable flow problem. Hence, heuristic and meta-heuristic solutions can be devised. As an extension to VNE, the NS problem is NP-hard [40], and heuristic approaches are suggested to achieve near-optimal solutions.

In the NS problem, a business infrastructure service request may arrive dynamically. The online mode of path computation for VNE is studied in [42]. This work investigates persistent requests with uniform demand. It proposes a deterministic competitive algorithm for requests involving routing and processing, where the accrued benefit to the network operators is maximized. The major contributions are to introduce a new service model of parking a request in standby mode, and it provides a worse-case lower bound in servicing such requests.

Telecom networks use and analyze many QoS parameters to consider network performance [43]. Caching at cell edges with limited capacity constraints is studied under the umbrella of vertical RAN Slicing [44]. The limited storage caching is treated as a bi-convex problem, and it studies the slice coordination issues. The problem considers cache storage and backhaul capacity in its solution model, and it proposes centralized and distributed cost-minimization algorithms. These existing works solely focus on the benefit function, which defines the total cost-benefit to the network operator under a segment of capacity and node processing-related constraints.

Network slicing provides a dedicated virtual network over the common physical infrastructure. Traffic modelling is scheming into different categories by using the physical parameters of measured traffic. Source-Traffic modelling using the Poisson process coupled with the Markov model is discussed in [45]. So far, there is a limited research effort in traffic aggregation, and there is a need to handle aggregation for each slice independently.

The basic NS problem is constructed as assigning optimal virtual links over the underlying infrastructure is reduced to a multi-commodity flow optimization problem. This work defines the basic NS problem as a VNE problem with location requirements. However, a study on the NS problem that considers different types of slices or the QoS management aspects [40] is not presented to the best of our knowledge.

Authors formulate NS as a bi-convex optimization problem [44], and they examine RAN vertical slicing and coordination issues. Minimization of the overall cost incurred is considered, and a heuristic-based solution for near-global convergence is devised. However, the work mainly focusses on the use of cache slicing, and non-trivial aspects such as RAN, core, and transport network node assignments are overlooked.

In our work, we deep-dive into resource allocation strategies in Network slicing. First, we propose a resource allocation algorithm in slicing through enhanced Dinic maximum flow, multiple attribute decision-making, and analytical hierarchical processing. We study and maximize stakeholder objectives such as operational efficiency, timeliness, and network performance in Chapter 3.

2.2.1 Energy Efficient Resource Management

Energy awareness is important for any application and network deployment. An initial work, a basic energy-aware wireless scheduling system, is proposed in [46], which formulates the average energy per packet using an M/M/1 model.

Conventional EE approaches [20] could fold into these categories: a) Energy harvesting techniques extract energy from the surrounding environment like solar, wind, and mechanical vibrations. b) Dynamic Power Savings defines the ideal way to save power is to switch off the transceivers whenever there is no need to transmit or receive, and c) Relay and Cooperative (RC) Transmissions deploy multiple relay nodes. Hence, there are multiple channels for communication between source and destination [47]. Finding an optimal relay placement strategy within the 5G RAN systems that can achieve radio interference, spectral efficiency, and EE management goals remains an unsolved research area.

The list of challenges identified are i) When designing Dynamic Power Savings (DPS) for heavily loaded 5G RAN systems and maintaining a wide range of QoS requirements remains an area for further exploration. ii) Retaining high QoS performance in RC communications while preserving better throughput and reduced latency is an open area in RC communications.

Hybrid approaches that address EE are:

i) Joint Cell Association and Power Control (JCAPC) - A combination of cell association can be fused with prioritized power control schemes depending on the desired objective. Diverse transmit powers of BSs can lead to uneven distribution of load. Channel access-aware user association scheme can boost spectral efficiency in downlink transmissions and load balancing among BSs [48].

ii) Bio-Inspired Resource Interference Resource Management (RIRM) Based Techniques - The behaviour of biological organisms are modelled in radio interfaces to construct an algorithm that maximizes energy-aware throughput as networks profit. Profitability is characterized through attainable bio-behaviors with preferences in allocation with the objective of energy and spectrum efficiency of the entire RAN subsystem [49].

iii) Integrated Spectrum and Energy Harvesting Techniques - Nodes are equipped with Energy Harvesting Cognitive Radio (EHCR) modules. It provides the devices and networks with the ability to harvest energy and sense spectrum simultaneously. EHCR can convert ambient energy into electricity at the same time, probe and use primary channels [50].

Cluster-based protocols and virtual backbone tree-based routing methods are applied in Wireless Sensor Networks (WSNs). These are energy-efficient routing algorithms deployed to improve the lifetime of WSNs. The routed messages to the target node are routed via backbone tree nodes. The tree nodes are selected based on the fitness factor of the nodes. The fitness factor is composed of the energy of the node, distance with the upstream parent node, and angle of its communication. Our approach derives inspiration from the fitness factor and tree nodes from the WSN backbone routing for energy efficiency [51].

2.2.2 QoS & Energy Efficient Resource Allocation

Maximizing the energy efficiency while guaranteeing the QoS requirements between the user and RAN are studied in [52]. It applies a weighted Tchebycheff method for converting Multi-Objective Optimization (MOOP) into Single Objective Optimization (SOOP). SOOP constitutes several quasiconvex fractional functions (QFFs), and the proposed iterative algorithm aims to minimize the maximum of QFFs. EE for every user is maximized, however, w.r.t QoS, this work is based on bandwidth allocation only.

The EARTH energy model is applied for the virtual base stations [53]. When there are no incoming messages in the queue, the node goes to sleeping mode. The average power consumption is studied as a function of active processing time and idle sleeping time. In our work, we apply the EARTH model. Our work also uses the Dynamic Power Savings technique. However, only the provisioned nodes remain active as decided by the scheduling algorithm. Scheduling rules allocate routing paths as per QoS requirements and meet the demand of current traffic with a buffer to accommodate a potential sparse increase in the load. The paths and nodes are dynamically activated by CCs, which carry out performance monitoring. CC plays a major role in shuffling and changes to the network assignments. It can assess overall load, QoS and proactively satisfy the demands of EE, QoS, resource utilization, and operational efficiency.

2.2.3 QoS attributes in scheduling

Many existing QoS-based scheduling research works have focussed on a single objective, such as bandwidth or latency. We have put related works and the parameters the authors are improving in Table 2.1. For example, authors [54] focus on bearer services for guaranteed and non-guaranteed bit rates. Similarly, a priority based Scheduling to provide Differentiated QoS is proposed based on the delay factor [55]. A scheduling framework is discussed to guarantee a packet delay below the QCI's PDB [56].

While many others consider QoS-based scheduling as multi-objective optimization. However, these works have not identified the relevant attributes extensively. For example, the proposed multi-traffic scheduling only aims at minimizing delay, packet loss, and improving data rate [59].

For NS, based on QoS for static and dynamic allocation are studied under throughput and delay [40]. For real-time connections, the authors guarantee data rate with waiting delay violation probability [60]. In 5G and beyond, it would be a collective platform of wireless technologies. Hence, there is a need for a comprehensive study and analysis of QoS attributes for scheduling.

Table 2.1: Existing works in QoS and Energy Efficiency

Related work	**QoS**	**Energy-Efficiency**	**Other Objectives**
Guanding Yu et al. [52]	Investigates bandwidth allocation only	Energy efficiency is maximized through Tchebycheff method	✗
K. Suganthi et al. [57]	✗	Randomized virtual backbone tree for the reduction in rapid energy depletion	Theoretical derivation on probabilistic bounds for connected nodes to the backbone are computed.
Siddique et al. [48]	✗	Cell association fused with prioritized power control schemes	Achieves load balancing and spectral efficiency
Olwal et al. [49]	Only throughput metric considered	Maximizes energy-aware throughput as networks profit	Spectral efficiency achieved through bio-inspired algorithms
Liu et al. [50]	✗	Use of Energy harvesting Cognitive Radio methods	Spectral efficiency, to convert ambient energy into electricity
[58],[55],[56]	Singular objective such as GBR / non-GBR [58], and delay [55][56] are studied	✗	Scheduling framework proposed
H. Wu et al. [59]	Multiple QoS attributes such as delay, packet loss, and improving data rate studied	✗	Downlink traffic studied through genetic algorithm

2.2.4 Cognitive Cycles in resource management

Despite the evolution of the cellular network, a lack of intelligence and autonomous capabilities persist as a stumbling block to deploying, supporting, and scaling next-gen apps. Cognitive Radio (CR) explores the use of underutilized licensed channels by secondary users. CC enables a node to learn, gain knowledge from prior experience, and act to adapt to the dynamic network conditions [61]. Q-Learning [62] based CC model helps the BS to expand or shrink its coverage. Thereby, traffic offloading decisions are empowered through CC.

> *Cognitive cycles (CC)* are a set of cascading recurring patterns. Each CC senses the current situation and interprets it about ongoing goals. Then, it selects an internal or external action in response.

2.2.5 IP and Cellular Traffic Classification

Traditional techniques to perform Traffic Classification are Payload inspection, statistical, and behavioural methods. Currently, ML is gaining traction in this field [63]. Network traffic classification can provide troubleshooting capabilities, build security policies, and manage QoS to guarantee overall user satisfaction. Authors have used multiple ML algorithms not limited to Bayesian-based classifiers, neural networks, decision trees, and clustering-based techniques like expectation-maximization and K-means based approaches. In our work, the features are identified and selected in a well-detailed and fine-grained manner. We also focus on a unified approach for prioritization in the end-to-end QoS management between cellular networks and the Internet.

Traffic classification based on priority queue is discussed in the Data Traffic Model for 5G Slicing. It aggregates M2M data through the Packet Data Convergence Protocol (PDCP). Based on QCI values, it aggregates data at the Radio Network in the PDCP layer. Extended labelled mobile network data constituting three levels of traffic identification is discussed in [64]. It uses flow statistic levels to classify the packets, and the accuracy of such methods is below par. Hence, to segregate the users and their network packets, traffic classification algorithms need to be studied for 5G [12].

2.2.6 RAN partitioning and isolation

Radio resource partitioning is a systematic process that happens every allocation window T. Let $m_{i,j,k} \in M$ is binary, indicates if Physical Resource Block j (PRB_j) is allocated to k^{th} slice. Here, $i \in \{1, 2, ..., T\}$ in the time domain and $j \in \{1, 2, ...F\}$ in the frequency domain.

The overall resource partitioning problem is about maximizing $\sum_{i=1}^{T} \sum_{j=1}^{F} \sum_{k=1}^{K} m_{i,j,k} \mid m_{i,j,k} \in \{0, 1\}$, where slice k could request a set of contiguous locations, virtual Resource Blocks (vRBs), or granular virtual transport block size.

Partitioning of RAN can be achieved by arranging RF carriers into time and frequency resource grids. The physical resource splitting technique divides the frequency band into different subcarriers. The isolation at the logical level delimits the capacity of logical components referred to as PRBs. Differentiated services, prioritization, and Sub Carrier Spacing (SCS) can be accommodated in PRBs.

Isolation of slices is achieved in infrastructure and management levels. In the former, RAN partitioning, splitting transport domains through Multiprotocol Label Switching (MPLS) or Virtual Local Area Network (VLAN) tagging, and Virtual Network Functions(VNFs) in the data centre across different or shared compute nodes provide end-to-end isolation. Virtual Machines

(VM) or Docker containers across availability zones provide hard isolation and reliability in the data centre. Similarly, Flex ethernet with Time-Sensitive Networking and MPLS-Transport Engineering in the network layer provides a balanced approach between hard and soft isolation in the transport domain.

In the latter, the management and orchestration are through multitenancy. Each tenant should have an exclusive administration of end-to-end slices through network domain controllers. The provider manager administers the shared part, where tenant managers operate functions running inside a specific NS. Currently, Virtualized Infrastructure Manager (VIM) and NFV orchestrator by NFV Management and Orchestration (MANO) enable multitenancy.

The work highlights [65] the need for slices to operate independently and isolate performance, dependability, and security between them. For example, congestion in one NS instance shouldn't affect the KPIs of other slices. Similarly, faults originating in an NS instance don't impact another.

The authors [66] highlight the importance of resource partitioning and apply a heuristic two-dimensional knapsack optimization solution for maximizing the unallocated sequential resource blocks and admission of slices. The RAN and CN domains provide Network Slice Subnet Instances (NSSI), combined through backhaul links to form an NS instance. Here, domain controllers such as SDN controllers or Multi-Protocol Label Switching (MPLS) management carry out slice subnet management for E2E operation.

Various radio resource management schemes under spectral and EE, minimal interference, and hybrid models have been discussed [20]. However, QoS challenges like explosive growth in traffic volumes need to be addressed holistically. Some constraining gaps in QoS are as follows:

i) 5G RAN systems would conglomerate multiple tiers and RATs [20]. 5G system needs optimally designed resource allocation, scheduling, and load balancing. The design should consider the distribution of UEs, network slicing, and traffic patterns. Also, with explosive growth, it is essential to segregate the packets in 5G through traffic classification algorithms and QoS attributes to prioritize channel allocation and packet scheduling.

ii) MC user needs higher availability and resilience. In [22], the authors have focused on the end-to-end reliability of mission-critical traffic. It studies the co-existence of mission-critical and best-effort traffic. The work admits that there is a considerable design consideration in supporting 5G mission-critical traffic.

The thesis delves into QoS profile attributes from S1AP and IP protocol in Chapter 3. We apply standard traffic regression algorithms to compute the priority of s1ap requests while scheduling. Virtual backbone and cognitive cycles techniques are used to create a joint QoS and energy-efficient resource allocation and scheduling in network slicing. RAN partitioning and isolation are studied extensively.

2.3 Load Balancing and the Impact on QoS of mobile devices

The presence of network hotspots can cause unstable connectivity and jitter issues to the mobile devices connected to the overloaded AP. The problem of load balancing is extensively studied in the literature. Dynamic LB is considered a stochastic optimal control problem and indicates Least Relative Load Routing (LRLR) as asymptotically optimal for a homogenous load [67]. Farzi et al. [68] study a zone-based load balancing for HetNets comprising macro and small cells. The authors propose the transfer of devices from overloaded to underloaded cells through a Cournot game, where the optimal load distribution of each cell is the Nash Equilibrium Solution (NES).

Jie Cui et al. [69] aim to select the best response time for devices considering the overall usage threshold in data and control plane during load balancing. Sahoo et al. [70] propose load balancing for Multi-Controller SDN in IoT through the multi-criteria decision-making method Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) considering the overall latency, bandwidth, and processing load.

Bejerano [71] study max-min fair bandwidth allocation and ensuring fairness for the device users based on load balancing and association control. The devices don't greedily associate with AP based on their best RSSI. It proposes an approximation-based algorithm for greedy users and weighted demand users.

Attiah [72] presents a reinforcement learning framework for optimizing neighbour cell relational parameters to balance the traffic between different cells better. However, this research doesn't consider the user mobility and scale of the LTE network.

The authors [73] explore load balancing between commercial eNodeB and Wi-Fi AP, and parameters such as spectral efficiency, SINR, available capacity, etc., are discussed. Gen Liang et al. [74] identify access selection and traffic distribution as indispensable for the overall system performance improvement. They strive to minimize transmission delays and distribute the traffic flow of users to achieve an optimal transmission rate in Heterogeneous Wireless Networks (HWNs).

Garcia [75] categorizes APs into fair, gull (overloaded), and willing (underloaded). It utilizes the channel utilization method and the Available Admission Capacity (AAC) metric. AP in gull condition sends SOS to other $n-1$ APs. Willing APs respond with a new AAC, and the gull AP selects the optimal AP for transferring the device. However, for n APs, it suffers a high message overhead of $O(n^2)$ for each migration. Also, when altering the transmission range, unpredictable station transfers may lead to orphaned devices.

Kawada et al. [76], propose a trigger-based dynamic load balancing technique. AP selection algorithm minimizes the highest AP usage by re-distributing the load from the highest bottleneck AP. However, load balancing all the overloaded APs may not be possible in fluctuating traffic.

Yun [77] proposes a cell breathing technique with a centralized agent that evaluates the average load. The overloaded APs contract their perceived transmission range, and the underloaded APs enhance their beacon strength. The average load of a global network will often mislead LB decisions. Ye et al. [78] apply a biased load as a function of the signal-to-noise plus interference ratio on the underloaded APs to promote fair resource utilization.

Asakura et al.[79] examine dynamic traffic distribution between the APs, where the traffic flow consists of an active and backup flow. A VPN server keeps track of all the APs and switches the path between primary and secondary sub-flows based on congestion and RSSI. In [80], SDN controllers have a global vision of the network, and they implement a round-robin load balancing strategy among the access nodes.

5G micro infrastructure constitutes several small cells deployed in ultra-dense areas. So, load balancing at access points is vital for better utilization and avoiding hotspots. There is a need for an in-depth study of QoS attributes such as the Signal-to-Interference-plus-Noise Ratio (SINR) and CQI of mobile devices. There is a need for tuning the standard load balancing algorithms and performing optimization to minimize load imbalance that would promote better traffic distribution and analysis without affecting the QoS of devices are paramount.

In this thesis, we propose and investigate swap-based load balancing with the biasing technique in Chapter 4, which will converge load imbalance at the access point towards zero and ensure good signal strength of the connected mobile devices during traffic distribution.

2.4 Task Offloading techniques

LTE Spectrum is approaching Shannon's capacity, and the cellular traffic from indoor locations with negligible mobility is rising rapidly. Better traffic planning and utilization can be achieved when we consider the nature of mobility patterns. Mostly, the current offloading techniques are based on localized BS control, and it leads to heavy control signalling. Decoupling data and control plane while offloading can lead to effective throughput usage.

Cartmell et al. [81], have come up with a Converged Gateway (CGW) component that enables the Selective Internet Protocol Traffic Offload (SIPTO) function. This work applies policy-based SIPTO, which relieves the core network from the additional load, and this component intermediates directly between eNodeB and the application server. Once the packet reaches Local SIPTO, through CGW, it performs packet inspection and identifies flows for offloading. Then, the packet is translated via the Network Address Translation unit and forwarded to the application server on the public internet. Through this methodology, we are diverting the flows from entering the core networks. Here, the SIPTO is refraining from decoupled control and data streams. Additionally, inherent dependence on localized control leads to high signalling overhead and sub-optimal offloading.

Elgendi et al. [82], propose a 3-tier offloading model for managing and operating dense networks. The authors contribute to User Rate-Perceived (URP) algorithm, Femotcell IP Access (FIPA), and Selective Local Controller Traffic Offload (SLCTO) techniques. URP monitors the user traffic flows and decides whether to offload or maintain the existing connection. FIPA introduces additional femtocells to the existing network infrastructure, and the traffic flows are migrated to these cells. The 3-tier offloading model constitutes physical, control, and management layers. The physical layer consists of network infrastructure, and the control layer controls the physical layer's equipment through its Local Controller (LC) APIs. This model leads to a single point of failure in the global controller of the management layer in network monitoring, network topology maintenance, and placement of local controllers while managing the lifecycle of network functions.

Park et al. [83], have proposed SDN-based traffic offloading, where the controller can regularly monitor the packet drops in the links. The idea is to automatically re-route such traffic flows on the pretext of loss detection, which can occur due to congestion.

The controller monitors every node and port through *ofp_queue_stats* messages from the OpenFlow protocol. On finding a packet drop, the controller can utilize the re-routing module to compute the alternate paths. Results show that this work would reduce the average loss rate in the system. The authors also identify the need for thresholds for packet losses, which triggers the re-routing procedure. Otherwise, the ping pong effect can be observed, which could degrade the network performance. This procedure solely identifies the packet drop and subsequently triggers offloading. This work doesn't consider the vital parameters of QoS.

Alameddine et al. [84], study the dynamic task offloading and scheduling under three subproblems, which are i) task offloading, ii) application-aware resource allocation, and iii) task scheduling. Application server instances are hosted in the edge servers. The tasks are offloaded to the nearby edge server, which the application instance processes. The task scheduling and offloading in this work are centered on the IoT for ultra-low latency and formulated as a mixed-integer problem. The work utilizes logic-based blender decomposition.

Olfa Chabbouh [85] presents a joint service offloading and scheduling strategy in Heterogeneous Cloud Radio Access Networks (HCRANs). The scheduling scheme aims to minimize the task execution time. Here, an edge cloud is added to the remote radio head near the mobile end-user. When the computational load shoots up in the resource-depleted mobile device, the tasks are offloaded for remote execution.

Authors [86] have studied traffic offloading under different streams. Offloading in Mobile Cloud Computing (MCC) has been researched based on communication and computation costs. There is a necessity to scrutinize offloading through a concrete set of QoS Parameters.

We investigate task offloading based on QoS attributes such as QCI, Allocation and retention priority, soft deadlines, and computation cycles in Chapter 5. We propose an ensemble categorizer through Borda scoring and methodologies such as single and multiple attribute categorizers. We also reduce the waiting time for certain types of tasks through probabilistic priority-based scheduling.

2.5 Application Aware routing through SDN, NFV, and NS methodologies

We attempt to summarize the study on QoS metrics and frameworks through NS, SDN, and NFV in the upcoming sub-sections. Next, we focus on the existing literature and its effect on quality assurance standards and mechanisms for data transmission. We then tabulate the different literature comparing their proposed QoS frameworks and the focus areas. Finally, we attempt to identify the existing gaps for application-aware routing methodologies to deliver better-customized NS settings.

2.5.1 QoS in Software Defined Networking

A design to enable service differentiation and efficient network resource use through SDN controllers is studied in [87]. The authors have designed resource monitoring, route calculation, call admission control, and resource reservation modules through OpenvSwitch and POX controllers. However, this solution doesn't guarantee end-to-end delay and lacks performance analysis and admission control mechanisms. Dutra et al. [88] propose ensuring QoS guarantees without over-committing the network resources to users who request high bandwidth and low jitter. The simulations employ Open vSwitches (OVSs) in the network, where the algorithm strives to minimize the active OVS. The authors [89] put forward a QoS provisioning architecture where the user queries through Resource Reservation Protocol (RSVP). The SDN framework enables the QoS settings through admin and RSVP providers. The result shows the average total data transfer times for larger files performed better with QoS setting against a Non-QoS setting.

2.5.2 QoS through Network Function Virtualization (NFV)

The NFV functions operating on virtual machines facilitate Service Function Chaining (SFC). However, processing and queuing delays may differ with virtual CPU, virtual memory, and the overall traffic load. The packet delays could be irregular; a prediction method based on random forest regression is proposed [90]. A QoS-assured SFC is presented here to reduce the latency and bandwidth consumption. The authors implement routing modules in the OpenStack cloud operating system, which exchanges the results with the OpenContrail to improve the overall QoS by decreasing the delay time for medical application data [91]. Wide area management system estimates, gathers and investigates data in power systems through SDN and NFV. The work [92] is implemented in Mininet setup and Ryu controllers to realize the quality of service (QoS) requirements like reducing round-trip latency and packet loss and exploiting the network resources optimally. The increase in traffic volume and service demand in telecom operators are addressed through NFV functions deployed as virtual machines in cloud environments. The authors studied VNF placement through the Mixed Integer Programming (MIP) model and solved it using the Gurobi solver for a topology of 28 nodes and 41 links. However, this model takes a significant time to converge and can be challenging to apply for real-time evaluations [93].

2.5.3 Application aware routing

Bagaa et al. [94] have shared three solutions for multipath forwarding. In this setup, an orchestrator communicates the resource demands among the access points and switches. The SDN controller executes the proposed algorithms to optimize resource allocation.

The first solution is Full Paths Re-computation (FPR), which applies a linear integer programming technique that leverages branch and bound methods. However, it is not time efficient as it consumes exponential time. The second solution, Heuristic Paths Re-computation (HPR), explores Dijkstra's shortest path algorithm, and the output comprises activated switches and assigned paths. The above solutions reduce operational expenditure but ignore several pivotal QoS KPIs. Further, the work doesn't describe the algorithm design for multiple controller planes and distributed coordination of the computed paths. The third solution, Partial Paths Computation (PPR) is a greedy algorithmic design where the shortest routing paths are computed and allocated for the new requests while ignoring the previous allocation, which would become suboptimal with time.

Diffserv defines a scalable mechanism for classifying the network traffic and providing QoS for IP networks. It classifies through QoS tolerance limits comprising attributes such as packet loss, delay, and jitter. For each service class, like signalling, streaming, and real-time - low latency data, the tolerance limits are baselined [95].

Network Situation Aware Framework (NSAF) [96] applies a genetic algorithm for examining every application's QoS requirements, observes the current status, identifies the violation, and finds the suitable path for fulfilling the specifications. This framework lies between the control and application layers and acts as an intermediator between them. It manages the QoS requirement of the application and controls the SDN controller.

NSAF handles 12 different application types described by DiffServ classes. It employs the digest algorithm and makes use of a T-score based on average, a standard deviation of QoS attributes.

A latency-aware optimization model (*directMIN*) for provisioning 5G slices in cloud networks is studied [97]. This work proposes *networkAware* mode to balance the trade-off between resource/energy consumption and realized latency.

A slice-tailored resource allocation, scheduling, and selection of Backhaul (BH) link maximize the total BH throughput. This work complements adaptive routing and small cell-related operations focusing on throughput-oriented slices [32].

The focus of this work is limited to analyzing primitive metrics such as bandwidth allocation and overall time to bring up switches for a slice [98]. The existing literature has not studied the QoS holistically to cater to tailored offerings for network setup in unison with SDN, NFV, and slicing to the best of our knowledge. In Chapter 6, we study QoS through holistic metrics collection, monitoring, and analysis through application-aware QoS routing in SDN and NFV for customized requirements in a 5G network slicing setup, which would be vital to meet the SLA requirements.

2.6 Fair Co-existence and QoS in 5G Wi-Fi Co-existence

Wi-Fi 6/6E and Private 5G are complementary solutions as both support dense IoT environments, new applications, and use cases targeting high throughput, low latency, and high capacity. Private 5G networks support wide-range indoor and outdoor deployments. It enables high-capacity throughput and coverage in both dedicated and shared spectrum. Particularly, Licensed-Assisted Access leverages the 5 GHz unlicensed band in combination with the licensed spectrum to deliver a performance gain for mobile devices. In contrast, Wi-Fi 6/6E targets short-range indoor deployment, upholding similar throughputs.

Wi-Fi 6/6E and Private 5G together can grow its prospects across multiple initiatives. For example, in education, students can learn better with immersive learning (virtual reality) serviced by Wi-Fi. A cellular base station can connect buildings and track transportation across the entire campus. The Wi-Fi 5G co-existence is exciting and of particular interest to the research community due to fair proportional user sharing.

2.6.1 Challenges in 5G Wi-Fi Co-existence

The frequency bands of 5G and Wi-Fi are in close proximity. The unchaining of frequency bands leads to interference, and these technologies lack a means of coordination among them. The prominent mechanisms that facilitate fair co-existence between Wi-Fi and cellular are Medium Access Control (MAC) based, which are a) Listen-Before-Talk (LBT) and b) Duty-Cycling (DC).

LBT is a technique where radio transmitters first sense if the channel is busy. The device can start transmitting the data if it finds the channel idle. This methodology applies two algorithms, namely Carrier Sensing and Energy detection, to determine whether the radio channel is free. If it detects the channel is occupied, the user waits a random amount of time before re-applying the technique.

The LBT technique is combined with the complementary Distribution co-ordination Function (DCF). Here, when the device wants to transmit, it waits for a small amount of time equal to DIFS (Distributed Inter-Frame Space) if it finds the channel idle. Then, the device again senses the medium. The device sends a Request to Send (RTS) signal if the medium is still free. Once the user receives the Clear to Send (CTS) message from the BS, the user starts the data transmission. In case of a failed transmission, the user enters a randomized truncated exponential backoff period.

In a binary exponential backoff algorithm, after c collisions, each retransmission is delayed by a random number of slots between zero and $2^{log_2(CW_{min})+c}$. After a successful transmission, the backoff counter is reset to zero. The truncated variant implies that the exponentiation stops after a specific backoff counter value increases. CW_{min} and CW_{max} are minimum and maximum contention windows of the truncated Carrier Sense Multiple Access (CSMA) / Collision Avoidance (CA) exponential backoff. A busy channel does not cause a backoff, but only causes the backoff counter (b) to freeze.

$$b = rand(1, \min\{2^{log_2(CW_{min})+c}, CW_{max}\}) \tag{2.1}$$

LBT squanders critical time resources over the air to decide the user contention when implemented inaccurately. The devices require transmitters to sense over time to acquire the channel. It could lead to battery drain in mobile devices.

Duty-Cycling (DC) allows cellular systems to transmit only in a portion (α) of fair time in a period T. The rest of the time $(1 - \alpha)$ would be utilized by Wi-Fi devices for communication. Here, cellular devices do not sense the carrier before transmission. Instead, it achieves data transmission in a deterministic manner controlled by a centralized access mechanism in the cellular BS.

Wherein the Wi-Fi users follow carrier sensing and DCF mechanisms. Hence, these users transmit when the carrier is free and follow exponential backoff if the channel is occupied. Additionally, Wi-Fi devices are unaware of the presence of a cellular system.

The main concern of duty cycling is that, since cellular devices do not perform carrier sensing before transmission, ongoing Wi-Fi transmissions at the end of the current period $((1 - \alpha)T)$ spill into the cellular portion of the next period (αT). It leads to interference of Wi-Fi packets with cellular transmissions at the start of the next period. This interference occurs due to no coordination among Wi-Fi and cellular systems.

2.6.2 Inter Technology Communication

LAA-as-a-Service (LAAS) is applied in the network slicing framework to improve RAN performance across slicing [99]. This work studies dynamic radio topologies through nomadic movable access nodes with LAAS to complement the ultra-dense networks for various key performance indicators (KPI) based on the deployment requirements.

Here, the licensed channel remains the primary carrier, and the unlicensed band can provide the best-effort service to boost availability, reliability, and performance. While License Assisted Access (LAA) is actuated through LBT, challenges such as delay in LBT in heterogeneous environments and hidden terminal problem between Wi-Fi and cellular technology has to be studied.

The system parameters such as initial backoff duration, transmission opportunity (TXOP) of LAA, packet length, and transmission rate of Wi-Fi are tuned for maximizing the total network sum rate of LAA-Wi-Fi co-existence. This study also compares the various scenarios wherein the network performance of LAA-Wi-Fi. and LAA-Wi-Fi considers duty-cycle-based transmissions for the cross-technology co-habitants.

In the physical layer, this work [100] studies the co-existence of Wi-Fi and cellular networks under the same frequency and time domain. The power domain is used to accommodate the heterogeneous neighbour.

Inherently, the LTE scheduler blacklists the usage of specific resource blocks in each time slot. Wherein the Wi-Fi interleaves payload-generating bits in the proper position with low-power constellation points. The power of unaffected and affected subcarriers differs by 13 dB.

The authors in [34] study MAC-based duty cycling in the LTE-U and Wi-Fi co-existence. This work analyzes the performance of LTE-U interference and Wi-Fi performance in terms of throughput and fairness. In duty-cycling, Wi-Fi employs DCF CSMA/CA with exponential back-off. When the LTE-U transmission with Interference to Noise Ratio (INR) is higher than -62 dbm or a neighbour Wi-Fi station transmission with INR greater than -82 dbm, it will cause the interfered Wi-Fi station to freeze its back-off timer. The authors refer to weak interference as the LTE-U interference with INR less than -62 dbm and strong interference with INR greater than -62 dbm. Firstly, this work demonstrates that this co-existence using simple air time-sharing is generally unfair to Wi-Fi during weak interference. Secondly, it shows that co-existence can achieve fair sharing under strong interference for some (T, α). Thirdly, this work illustrates that fairness degrades linearly when Wi-Fi payload length or LTE-U to Wi-Fi collision probability increases. The authors recommend a version of the LBT mechanism for LTE-U networks to overcome the observed unfairness.

The authors suggest that LBT has the potential to serve as a unified global solution framework. It identifies frame-based and load-based equipment as add-ons that can be deployed along with LBT for low interference and higher efficiency.

Frame-based LBT (FLBT), similar to Carrier Sense Adaptive Transmission (CSAT), decomposes air time channels into continuous frames with a fixed duration. FLBT divides each frame into an idle (10-100 μs) and channel occupancy period (1-10 ms). The LTE must remain muted in this idle period, and after it completes, it invokes clear channel assessment. When the channel is idle, LTE can transmit.

Load-based LBT doesn't follow a fixed frame structure. If the channel is idle during channel sensing, it enters an extended clear channel assessment of 34 μs instead of transmitting immediately. If the channel is still idle, it tries to acquire the channel. This mechanism avoids one node monopolizing the channel usage.

The authors [101] focus on real-time deployment aspects of coexisting networks. It studies fundamental parameters such as latency of Wi-Fi connection affected due to co-existence. Similarly, it analyzes the energy-sensing threshold of Wi-Fi that affects the latency and throughput of its devices. The work outlines the diversified concerns in the co-existence of cellular and IEEE 802.11 technologies in the unlicensed bands in coordination, fairness conditions, transmission techniques, regulatory requirements, deployment scenarios, and standardization efforts [102].

This work [103] examines wireless interference identification techniques in co-existence management amongst heterogeneous technologies. The methods discussed are limited to hardware or the physical layer.

2.6.3 Selfishness in CSMA/CA

In this work, we discuss the effect of selfish nodes on the LAA-Wi-Fi co-existence. It is a crucial problem to address since we discuss the co-existence, and that to be fair.

In conventional CSMA/CA Networks, Mario et al. [104] study the greedy behaviour of nodes, where users are static and mutually reachable to avoid hidden terminal problems. Esmalifalak et al. [105] assume apriori knowledge on the number of devices in the network to gauge the optimal throughput of selfish nodes. Thereby, this work calculates the transmission probabilities to optimize network traffic. In reality, these assumptions are unviable. Here, the selfish nodes aim to increase their common transmission resource by decreasing the backoff window size value.

Vaidya [106] proposes a methodology where the receiver-assigned backoff value is validated by the sender to identify the misbehaviour of the former node. However, these require changes to the current protocol.

The network performance impact is studied for IEEE 802.11 when the selfish nodes are programmed not to obey exponential backoff [107]. Not only malicious, but selfish users can increase its throughput, significantly impacting well-behaved user operations. The well-behaved user and misbehaving nodes are modelled to show how the access probability of selfish nodes increases against regular users. Mainly, denial of service for legitimate nodes due to selfish users is presented. Additionally, the work discusses a Nash Equilibrium while obtaining equitable throughput among selfish users when multiple selfish users are present in the Wi-Fi network.

The authors [108] show when a selfish user exhibits perpetual fixed window backoff patterns, these selfish users affect the channel acquisition of legitimate users. However, if an intermittent fixed window backoff pattern or perpetual reduced exponential backoff is applied, the misbehaving node converges with the legitimate node on acquisition and throughput.

CRISP [109] and Bayesian estimation [110] are counteraction strategies applied when selfish users are present in the Wi-Fi-only network. Here, legitimate Wi-Fi users also use a greedy approach to obtain equal bandwidth as selfish Wi-Fi users. However, by applying these methods in a co-existing network, all Wi-Fi users may be greedy and cellular users would be legitimate. Hence, these strategies don't work in a co-existing network.

We perform an in-depth investigation of the detrimental effects of selfish Wi-Fi users on the co-existing network in Chapter 7. For both duty-cycling and listen-before-talk, we study the counteraction action strategies for various network configurations.

2.7 Chapter summary

This chapter undertook a review of resource allocation strategies in NS and the significance of QoS-based scheduling and energy efficiency. We also examined the QoS attributes in S1AP and IP protocols and application-aware routing procedures. We studied the importance of SINR, CQI, QoS of devices, and load balancing in the 5G micro infrastructure. We deep-dived into co-existence challenges for 5G use cases in the unlicensed spectrum.

Chapter 3

Traffic Classification and Resource Allocation in Network Slicing

3.1 Introduction

Network slicing is a key enabler for 5G for supporting custom requirements. The objective of the chapter is two-fold. Firstly, the resource allocation in network slicing needs to factor in operational efficiency, network performance, and timeliness KPIs. We implement QoS-based resource allocation in network slicing, combining MADM with AHP to maximize stakeholder objectives. The maximum possible flows are computed through Enhanced Dinic algorithms.

Secondly, we deep dive and consolidate a list of QoS parameters across the Internet and Radio Access Technologies (RATs) such as 4G and 5G. We apply standard ML regressors to compute the priority of the network packets and compare their results. Subsequently, we propose a probabilistic class-based scheduling and theoretical results of this algorithm is detailed for M/D/1 and M/D/c models. For resource allocation, we propose a novel Virtual Backbone and Cognitive cycle based solution. The fitness function comprises a list of parameters around energy savings, latency, bandwidth, and cost to the operator, and is studied as a multi-objective optimization problem.

The system model and problem formulation are discussed in sections 3.2 and 3.3, respectively. The proposed algorithm for managing stakeholder objectives and its results are examined in section 3.4. The detailed study of QoS parameters in S1AP and IP protocols and the use of these attributes in traffic classification are studied in section 3.5. Class based scheduling is discussed in section 3.6. Virtual backbone and Cognitive cycle-based solutions for joint QoS and energy savings are discussed in section 3.7. The simulation setup for Network slicing is presented in section 3.8.

The results and analysis for QoS-based resource allocation through analytical hierarchical processing and MADM are studied in 3.9. The results of traffic classification are studied in 3.10. The results of virtual back and cognitive cycle-based solutions for joining QoS and energy savings are presented in 3.11.

In this chapter, we have addressed two distinct problems. The first one involves studying resource allocation to meet stakeholder objectives related to timeliness, operating efficiency, and network performance. The second problem, still within the domain of resource allocation, explores the joint study of QoS attributes for achieving differentiated services and energy savings. Due to the distinct nature of these problems, we have employed different approaches here.

3.2 Network Slicing System Model

The system is modelled with the following groups of parameters:

Fixed parameters: The physical network is represented as $G = \{N, E, \zeta, \omega\}$ where N and E represent Nodes and Edges in the network. ζ_n and $\zeta_{n,m}$ indicate the capacity of node n and link (n, m), where capacity refers to the throughput supported by the access point and connection link, respectively. Once a fraction of capacity is allocated, the allocatable capacity reduces. Let the residual capacities of nodes and links be represented as Υ_n and $\Upsilon_{n,m}$. $\omega_{u,v}$ stands for the weight between two dissimilar nodes u and v. $\omega_{_{U,AI}}$ would be first-hop weight between the end-user and air interface.

The weight assigned to the air interface (w_{ai}) and edges (w_{e_i}), $i \in \{T, CN\}$ directly corresponds to the operational cost in air interface selection, transport cost, and core network routing. The overall operational cost of a network slice is defined as the sum of the products of the fractional allocation percentages of the air interface, edges, application server, and their respective weights (costs). Here, we refer to dissimilar nodes as the nodes within the network that may possess distinct characteristics, configurations, or functionalities. More specifically, these differences can encompass variations in capacity and functional aspects, including elements like application servers, air interfaces, and other node-related disparities.

Input variables: Inputs are the slice(s) to be allocated. A slice is denoted through S. NS are set of slices (logical networks) over a physical substrate,

$$S_1 \cup S_2 \cup S_3 \cup S_4.. \cup S_s$$

A slice request consists of demand $\hat{d}$ (in terms of bandwidth) and resources needed (e.g., an application or network functions hosted on the core network). The slice request would contain a detailed specification and SLA covering parameters like reliability, security, peak bandwidth, average bandwidth, and acceptable latency.

The Slice request can be expressed as a tuple $(\hat{d}, R, SLA)$, where:

- $\hat{d}$ represents the bandwidth demand.
- R is a set that includes resources like air interfaces (AI), core network nodes (CN), and application servers (SV).
- SLA encompasses service level agreement parameters such as Latency (L_{req}) of a request, operational cost (OC), and reliability parameters such as availability (Av).

Control parameters: The selection of nodes and edges along the path are the tuning knobs. Control parameters encompass choices related to selecting or adjusting elements such as the air interface, nodes, edges along the path, and application servers. These choices directly impact the achieved bandwidth, latency, and operational cost of the network slice.

In a network, a path P refers to the route or sequence of network elements and connections taken by data or traffic as it travels from a source to a destination. This path includes various components and stages, which may include:

- Source: The starting point of the data transmission, often a device or User Terminal (UT) that initiates the communication.
- Air Interfaces (AI): These are wireless communication interfaces or access points that facilitate the wireless transmission of data. Air interfaces are commonly used in wireless networks such as Wi-Fi, cellular networks, and satellite communications.

- Transport Network: This part of the path involves the intermediate network infrastructure that transports data between different locations. It may include switches, routers, and optical fiber links, depending on the network technology. Nodes are represented by $\{T_1, ..T_t\}$. Edges between air interface and first node in the transport network is denoted by $E_{_{AI,T_1}}$.

- Core Network Nodes: These are key nodes or network devices that play a central role in routing and forwarding data within the network. Core network nodes are responsible for efficient data transport across the network, and nodes are represented by $\{CN_1, ..CN_{cn}\}$. Edges between egress node from transport and ingress node in core network is denoted by $E_{_{T_t,CN_1}}$.

- Application Server (SV): An application server is a dedicated server or software component that hosts and provides various network functions or applications. It may serve as the endpoint for specific services or applications in the network. Here network functions can be hosted such as baseband processing, data processing, or other specialized functions.

The path in a network represents the complete journey of data or traffic as it traverses through these components, starting from the source, passing through air interfaces, transport and core network nodes, and eventually reaching the application server where it may interact with network functions or applications. This path is crucial for understanding how data flows within a network and for optimizing network performance, reliability, and latency based on the specific requirements of the applications and services being used.

Output variables: Outputs are the virtual paths of an individual slice represented by a function ψ, whose arguments would be a tuple of Path P and Slice S. ψ returns a boolean value. In this output setup, a virtual path P on physical network G can be allocated to only one slice. However, the proposed algorithm can be extended for nested virtual path allocation.

$$\psi(P_i, S_j) = \begin{cases} True, & \text{if } P_i \,\epsilon\, slice\, S_j\, allocation \\ False, & \text{otherwise} \end{cases} \tag{3.1}$$

Let ϑ be $|P| \times |AI|$ matrix containing the allocated percentage of air interfaces AI in the path P. Similarly ℓ and η are 2D matrices $|P| \times |E|$, and $|P| \times |N|$ respectively, indicating the allocated portion of edges E and reserved capacity of Nodes N to the path P. The values in the above matrices would be between 0 and 1, indicating the fraction allocation of the paths.

Cost of slice S from an edge E with a demand $\hat{d}$ (percentage of usage of total capacity) is $\hat{d}_s w_{_E}$. The cost of air interface network selection AI is $\hat{d}_s w_{_{AI}}$.

3.3 Problem Formulation

Providing end-to-end services through NS is a combined optimization problem between different network components. Below, the optimization constitutes operational efficiency (the cost to the network operator), Network Performance, and customer QoS satisfaction (Bandwidth, Latency).

The cost of a Slice S_i comprises of three components:
i) Cost of Network Selection:

$$C_{NS} = \textstyle\sum_{p=1}^{P} \psi(p, S_i) \cdot \sum_{ai=1}^{AI} [\vartheta_{p,ai} \cdot w_{ai}]$$

ii) Cost of Transport and Core Network routing:

$$C_{T,CN} = \textstyle\sum_{p=1}^{P} \psi(p, S_i) \cdot \sum_{e_t=1}^{T+CN} [\ell_{p,e_t} \cdot w_{e_t}]$$

iii) Cost of the application server C_{SV}.

The Operational Cost (OC) of all input slices (NS) to the network operator and the constraints can be formulated as:

$$OC = \sum_{s=1}^{NS} \sum_{p=1}^{P} \psi(p, s) \cdot \left[\sum_{a=1}^{AI} \vartheta_{p,a} w_a + \sum_{e_t=1}^{T+CN} \ell_{p,e_t} w_{e_t} \right] +_{C_{_A}} \tag{3.2}$$

$$\forall ai, \sum_{p=1}^{P} \vartheta_{p,ai} \leq 1, \forall S_i, \sum_{p=1}^{P} \psi(p, S_i) \cdot \sum_{ai=1}^{AI} [\vartheta_{p,ai} \times \zeta_{ai}] = \hat{d}_{S_i} \tag{3.3}$$

$$\forall e, \sum_{p=1}^{P} \ell_{p,e} \leq 1, \forall S_i, \sum_{p=1}^{P} \psi(p, S_i) \cdot \sum_{e_t=1}^{T,CN} [\ell_{p,e_t} \times \zeta_{e_t}] = \hat{d}_{S_i} \tag{3.4}$$

$$\forall n, \sum_{p=1}^{P} \eta_{p,n} \leq 1 \tag{3.5}$$

The equations (3.3), (3.4), and (3.5) bounds maximum allocation along each edge and node. (3.5) is applicable for the nodes in transport and core network part of slice allocation. (3.3) and (3.4) provide constraints around bandwidth allocated to each slice.

Bandwidth aside, QoS related parameters such as request latency and path length has to be minimized, and reliability has to be maximized. To improve the operational efficiency, the cost to the operator needs to be maximized as shown in (3.2). For a given request, the end-to-end latency would span from RAN, Transport (T), Core Network (CN), and application processing time.

$$L_{req} = 2 \cdot \left[L_{ai} + \sum_{e_t=1}^{T,CN} L_{e_t} \right] + L_{SV} \tag{3.6}$$

$$L_{ai} = \frac{Length(Pkt)}{\vartheta_{ut,ai} \cdot \zeta_{ai}} + T_{pr_{ai}} + \frac{(\frac{T_{pr_{ai}}\lambda}{m})^{\sqrt{2(m+1)}-1}}{m - \lambda T_{pr_{ai}}} \cdot \frac{Cv_a^2 + Cv_p^2}{2} \tag{3.7}$$

Latency on a given $e(u,v)$ in (3.7) involves the transmission time of the source node, propagation delay over the link, queuing delay, and processing time on the target node. Propagation delay involves the time taken to transfer the packet size $Length(Pkt)$ over an allocated edge. In the given model, $M/M/m$ queuing model is considered, where m is the number of parallel processing units, $T_{pr_{ai}}$ is the processing time of a unit, λ is the rate of arrival, and Cv_a and Cv_p indicates the coefficient of variation of service time and average inter-arrival time. The latency of a request is denoted by L_{req}. The total latency would be round trip time comprising nodes in the traversed path of RAN, Transport, and backhaul links as shown in (3.6) and (3.7). The overall path length (PL) of a request in path p_i is

$$PL = 2 + \sum_{e_t=1}^{T,CN} \ell_{p_i,e_t} \tag{3.8}$$

The availability of a path would depend on all the nodes serially between the air interface, transport, core, and application server. $Av = \rho_{AI} \cdot \rho_T \cdot \rho_{CN} \cdot \rho_{SV}$.

From above, multiple attributes are being optimized, where the path should be allocated such that QoS, QoE, and QoBiz parameters are optimized.

The operating cost is a financial metric representing the expenses incurred by the network operator in maintaining and operating the network infrastructure, including factors like power consumption, maintenance, management, and other associated costs.

These are all components that contribute to the overall operating cost, and when formulated into an equation, the result would be expressed in a specific currency unit (e.g., dollars) as it represents the financial impact on the network operator's resources.

Eqns. (3.3)-(3.4) represents the bandwidth allocation in megabits per second (Mbps) or gigabits per second (Gbps) for a given slice based on the demand.

In Eqn (3.5), the allocated fractional bandwidth from a specific link should be less than or equal to its residual bandwidth, which is again represented in Mbps or Gbps. The unit of overall latency in Eqn (3.6) is represented in milliseconds (ms).

Table 3.1: QoS-Based Analytic Hierarchy Process

Timeliness	***Network Performance***	***Operational Efficiency***
Latency	Availability	Cost
Path Length	Throughput	Benefit

3.4 Proposed QoS-Driven Resource Allocation Algorithm for Tailored Offerings for end-to-end Network Slicing

A hybrid MADM and Analytical Hierarchy Process (AHP) with levels of QoS parameters are proposed to address this problem. It is tabulated in Table 3.1.

We propose an online algorithm which processes the slice allocation request. The algorithm operates in real-time, allocating network slices as incoming requests arrive based on available bandwidth. In summary, the term ”online” indicates that the algorithm operates continuously, responding to network slice requests in real-time as they arrive rather than processing them in a batch or offline mode.

Algorithm 1: ENHANCED_DINIC_PATH_FINDER(UT, SV)

```
$Paths = \phi$, $total = 0$
while (BFS(UT,SV)) do
    $p_t \leftarrow \phi$ ;
    $flow \leftarrow$ SendFlow($UT$,$\infty$,$SV$,$p_t$);
    if !$p_t$.isEmpty() then
        $Paths$.add(pair($flow$,$p_t$));
        $total$ += $flow$;
    else
        return $Paths$ ;
```

Algorithm 2: BFS(s, t)

```
$\forall$ n, $level[i] = -1$; $level[s] = 0$ ; $Q$.push($s$) ;
while !Q.isempty() do
    $u \leftarrow Q$.pop() ;
    for each $e : adj[u]$ do
        if $level[e.v] < 0$ && $\Upsilon_e < \zeta_e$ then
            level[$e.v$] = level[$e.u$] + 1 ;
            $Q$.push($e.v$) ;
return level[t] < 0 ? false : true ;
```

The proposed approach begins by invoking Algorithm 1, called "ENHANCED_DINIC_PATH_FINDER," to find the maximum possible paths from a user terminal to an application server. The User Terminal set (UT) represents a collection of user terminals in the network, which acts as the source of slice requests, denoted as $UT = \{u_1, u_2, u_3, ..., u_n\}$. The Application Server set (SV) represents application servers that serve as the destination of slice requests, defined as $SV = \{sv_1, sv_2, sv_3, ..., sv_n\}$. These are the input parameters of this algorithm.

In essence, this algorithm 1 invokes a path-finding component within a network flow optimization process. It uses the Breadth First Search (BFS) algorithm to explore paths between user terminals and application servers, sending flow along these paths and keeping track of the discovered paths and their associated flow values in the *Paths* set. The algorithm terminates when no more paths can be found and returns the set of discovered paths along with the total flow value.

The pseudocode for Algorithm 1 can be elaborated as follows:

- We initialize two variables: *Paths* as an empty set and *total* as 0.
- It enters a while loop that continues until the Breadth-First Search (BFS) algorithm between the user terminals (UT) and application servers (SV) returns true, indicating that there are still paths to explore.
- Inside the loop, we create an empty path variable p_t. Then, we invoke the SendFlow function with parameters (UT, ∞, SV, p_t).
- The output i.e., the path p_t variable is checked if it is not empty. If p_t is not empty, a valid flow path has been found. We add a pair consisting of the flow value ($flow$) and the path (p_t) to the set of *Paths*. We update the *total* flow value by adding the newly found $flow$.
- If the path p_t is empty, algorithm returns the set of *Paths*. This implies that no further paths can be found, and the algorithm terminates.

Algorithm 2 employs BFS to determine flows between source and destination and to form a level graph. Each node is assigned a level, representing its shortest path length from the source. The level graph then triggers Algorithm 3, known as SEARCH_PATH, to locate multiple flows in the graph. This process continues until a blocking flow is reached. The algorithm traverses the graph, considering the residual capacities of nodes and edges. SEARCH_PATH recursively

Algorithm 3: SEARCH_PATH(u, $flow$, t, p_t)

```
if u==t then return $flow$ ;
$edgeList \leftarrow adj$[u] ;
while !$edgeList.isempty()$ do
    $e \leftarrow edgeList$.get($R$.nextInt($edgeList$.size()) ;
    if $level[e.v] < 0$ && $\Upsilon_e < \zeta_e$ then
        $currFlow$ = min($flow$,$\Upsilon_e - \zeta_e$);
        $tempFlow$ = SEARCH_PATH($e.V$,$currFlow$,t,$p_t$) ;
        if $tempFlow < 0$ && !$p_t.isempty()$ then
            $\Upsilon_e$ += $tempFlow$ ;
            adj[$v$][$e.rev$] -= $tempFlow$ ;
            $p_t$.add($e$) ;
            return $tempFlow$ ;
        else
            $p_t \leftarrow \phi$ ;
```

identifies paths, which are collected in the *Paths* set. These paths are finite alternatives and are evaluated based on various decision-making attributes.

The pseudocode of the SEARCH_PATH algorithm, which finds a path in a graph from node u to node t while respecting certain constraints, is explained below:

- If node u is target t, we return the flow.
- We get the edges connected to u in *edgeList*.
- While *edgeList* is not empty, we randomly select an edge e from *edgeList*. We check if $e.v$ is unvisited ($level[e.v] < 0$) and if capacity constraints are met.
- If constraints are met, we calculate *currFlow* as the minimum of requested bandwidth and available capacity. Recursively, we then call SEARCH_PATH function.
- If a valid path is found, we update capacities and add e to p_t. We return the flow.

This algorithm efficiently explores paths from u to t, adjusting flows and respecting constraints to find a suitable path or return false if none exists.

The assessment results are stored in matrix A, where A_{ij} represents the value of alternative i against attribute j in Algorithm 4. To make comparisons across different attributes and ensure consistency, matrix A is normalized using the Enhanced Max-Min method (EMM), where 0 represents the worst rank, and 1 represents the best rank. Before evaluation, weights are

Algorithm 4: ALLOC_SLICE($S.Req$)

```
$Paths \leftarrow PATH\_FINDER(S.Req.src, S.Req.res)$ ;
$A \leftarrow populateAttri\_AlternatesM(Paths, Attr)$ ;
$\hat{A} \leftarrow normalize\_EMM(A)$ ;
$allocated \leftarrow 0$ ;
while $allocated < \hat{d}_S$ do
    $Wgt_A, Wgt_H \leftarrow AssignWgtEntropy(H, Attr)$ ;
    $WH \leftarrow eval\_Attr\_Within\_Hierarchy(\hat{A}, Wgt_A, H)$ ;
    $\hat{WH} \leftarrow normalize\_EMM(WH)$ ;
    $AH \leftarrow eval\_Across\_Hierarchy(\hat{WH}, Wgt_H, H)$ ;
    $newflow \leftarrow 0$, $BP \leftarrow AH.nextBestPath()$ ;
    $pathflow \leftarrow BP.left$, $P \leftarrow BP.right$ ;
    if $pathflow > \hat{d}_S$ then
        if $allocated > 0$ then
            $newflow \leftarrow (\hat{d}_S - allocated)$; ;
        else
            $newflow \leftarrow \hat{d}_S.half()$ ;
    else
        if $pathflow \geq (\hat{d}_S - allocated)$ then
            $newflow \leftarrow (\hat{d}_S - allocated)$ ;
        else
            $newflow \leftarrow pathflow$ ;
$allocatedFlow + = newflow$ ;
$\vartheta(P, p_{AI}) \leftarrow pathflow/\zeta_{AI}$ ;
$\Upsilon_{AI} \leftarrow \Upsilon_{AI} - pathflow$ ;
$\psi(P, S) \leftarrow 1$ ;
for $e = p_{AI}.next()$ to $S.Req.res$ do
    $\ell(P, e) \leftarrow pathflow/\zeta_e$ ;
    $\eta(P, e.u) \leftarrow pathflow/\zeta_{e.u}$ ;
    $\Upsilon_e \leftarrow \Upsilon_e - pathflow$ ;
    $\Upsilon_{e.u} \leftarrow \Upsilon_{e.u} - pathflow$ ;
```

computed using the entropy function. Within each group, a Simple Additive Weighing (SAW) method is used to compute C_{SAW} for each path against each attribute. At the hierarchy level, these values are aggregated. The attributes within each hierarchy are detailed in Table 3.1.

The table represents the proposed hierarchy for evaluating Quality of Service (QoS) in network performance. This hierarchy helps in a comprehensive QoS assessment. It has three main categories:

- Timeliness (Latency, Path length): Focuses on network responsiveness, measuring attributes like latency (delay) and path length (distance).

- Network Performance (Throughput, Availability): Evaluates overall network performance, considering throughput (data processing capacity) and availability (uptime).

- Operational Efficiency (Cost, Benefit): Assesses network efficiency and cost-effectiveness by examining costs and overall benefits.

The finest path with the highest coefficient value across the hierarchy is selected. This process continues until the slice request's demand is met. $\psi(P, S)$ is set for the allocated paths, and the residual capacities of nodes (Υ_N) along the path are reduced. Matrices $\vartheta(p, ai)$, $\ell(p, e)$, and $\eta(p, n)$ capture the allocated percentages, while $\psi(P, S)$ represents the final outcome of the slice allocation problem.

On slice request arrival, we invoke SEARCH_PATH, which computes the maximum possible paths between the requested source and destination. The time complexity is $O(EV^2)$. All the paths are evaluated against each criterion C, and the decision matrix A is attained, which takes $O(PC)$ time.

$$Wgt_A = 1 - \frac{1}{lnP} \cdot \sum_{p=1}^{P} [A_{pj} ln(A_{pj})] \tag{3.9}$$

$$WH_{H_i,p} = \sum_{j=1}^{|H_{Attr}|} Wgt_{A,j} \hat{A}_{pj} \tag{3.10}$$

$$C_{SAW} = \sum_{j=1}^{|H|} Wgt_{H,j} \hat{WH}_{pj} \tag{3.11}$$

$$EMM = \begin{cases} 1 - \dfrac{|A_{pj} - max(A_{pj})|}{max(A_{pj}) - min(A_{pj})}, & \text{Upward attributes} \\ 1 - \dfrac{|A_{pj} - min(A_{pj})|}{max(A_{pj}) - min(A_{pj}))}, & \text{Downward attributes} \end{cases} \tag{3.12}$$

$$Av \geq 1 - (1 - (\rho_{AI} \cdot \rho_T \cdot \rho_{CN} \cdot \rho_{SV}))^2 \tag{3.13}$$

The entropy function to calculate the weight of attributes (3.9) takes just O(P) time. The decision matrix is normalized through the EMM method denoted in (3.12).

In the proposed algorithm, the Enhanced Max-Min method (EMM) is used to normalize the values of decision-making attributes to a common scale, where 0 represents the worst, and one represents the best. This normalization process ensures that attributes with different measurement units or scales can be effectively compared and combined in the decision-making process. The weights assigned through the entropy function and the Simple Additive Weighing

(SAW) method help prioritize the attributes and alternatives based on their importance and performance.

When the identified paths identified by Algorithm 1 are fewer, the relativity in normalization could be skewed. Hence, EMM is tweaked during such runs, where min is set to a minimum acceptable value. The well-known compensatory algorithm, Simple Additive weighing, is applied within attributes in the hierarchy and across the hierarchy as shown in (3.10) and (3.11). It consumes O(C) and O(H) time, respectively. Our algorithm provides at least one-to-one redundancy of paths during average traffic. The availability is shown in (3.13).

Algorithm 1 details Breadth-First Search (BFS) Layering: The Enhanced Dinic's algorithm starts by constructing a level graph using BFS. This step has a time complexity of $O(V + E)$, where V is the number of vertices and E is the number of edges in the network. Algorithm 2 controls the Blocking Flow Phase. The layered graph finds augmenting paths in each blocking flow phase. Each BFS takes $O(V + E)$ time. However, each augmenting path is found in O(V) time since each vertex can be visited at most once, and each edge can be examined at most twice (once for forward and once for backward edges). Algorithm 3 defines the number of Blocking Flow Phases: In the worst case, the number of blocking flow phases can be $O(V)$, leading to a worst-case complexity of $O(EV^2)$. Overall, the Enhanced Dinic algorithm's time complexity is typically $O(EV^2)$, which can be significantly improved for networks with certain characteristics. In Algorithm 4, each path P is evaluated against each criterion C, and the decision matrix A is attained, which takes $O(PC)$. The entropy function to calculate the weight of attributes takes just $O(P)$ time.

3.5 QoS attributes and Traffic Classification

Differentiated services play a critical role in ensuring Quality of Service (QoS) in cellular networks. The QCI (QoS Class Identifier) values, ranging from 1 to 254, are defined within the S1AP protocol. The setting of QCI values lacks centralized control and validation, making it a non-standard parameter. The existing literature does not provide a detailed and comprehensive study of QoS attributes.

As 5G would conglomerate multiple tiers and RATs [20]. There is a need for traffic classification algorithms [12] and identifying the right set of attributes for segregating the packets in 5G for improved differentiated services and to meet the QoS requirements.

Our approach for traffic classification is as follows:

a) Data Collection:

The S1 Application Protocol (S1AP) facilitates signalling between E-UTRAN and the Evolved Packet Core (EPC). Within S1AP messages, attributes such as QCI, allocation retention priority, and pre-emption vulnerability function are encompassed. These attributes are particularly relevant during the E-Ultran Radio Access Bearer (E-RAB) establishment or modification, where packets carry essential Quality of Service (QoS) parameters. These packets, dispatched by the Mobility Management Entity (MME), are utilized by the eNodeB to allocate resources for one or multiple E-RABs.

Initially, data was sourced from various origins in the form of packets. Subsequently, these packets underwent analysis using Wireshark. Applying Wireshark's robust expressions, filters, and column selectors, the pertinent fields were extracted and exported in CSV format. Specifically, the analysis encompassed PCAP files related to VOIP calls, captures for traffic analysis, initial Context Setup, iPhones employing VoLTE on their respective networks, E-UTRAN Radio Access Bearer (E-RAB) Management procedures, UE Capability Information, Tracking Area Update requests, E-RAB Modify Requests, Ciphered messages, and Activate Default EPS Bearer Context requests. The evaluation focused on selecting sources, particularly prioritizing materials like the Initial Context Setup Request, Attach Accept, and Activate Default EPS Bearer Context Request, all of which are extensively documented [111].

b) Feature Extraction and Cleaning:

Attributes considered for learning are tabulated. Table 3.2 lists the parameters along with its measurement units and PRotoCol (PRC). Most QoS attributes are present in the S1 Application Protocol, and the rest are from IP Differentiated Services. Categorical values such as QCI and DSCP are elaborated into more fields due to their inherent importance. For instance, from QCI - the parameters are Resource Type ($GBR/Non-GBR$) and Service Types. The standard QCI characteristics are broken down as per [112].

Recently, while establishing the formal specification for QoS in 5G, QCI is extended as 5QI (5G QoS Identifier). 5QI is a pointer to a set of QoS characteristics such as priority level, packet delay or packet error rate [113],

The difference between QCI and 5QI is conceptually exactly the same. Hence, in the thesis, we have referred to the QCI metric as it is more prevalent in real-world packets, as 5QI is just

Table 3.2: Attributes used in Traffic Classification

Parameters	**Measurement units**	**PRC**
QCI	[0.1-10]	s1ap
Packet Length	Length in Bytes	na
Info	Description	na
uE aggregate MBR UL	Bits per sec	s1ap
uE aggregate MBR DL	Bits per sec	s1ap
ARP Priority	High(1)-Low(15)	s1ap
Pre Emption Capability	0,1	s1ap
Pre Emption Vulnerability	0,1	s1ap
QoS Delay Class	1-4	s1ap
Reliability Class	1-4	s1ap
Peak throughput	Octets per sec	s1ap
Precedence Class	0-7	s1ap
Traffic class	1-4	s1ap
Delivery Order	Delivery Order no	s1ap
Erroneous SDU delivery	Whether Discarded	s1ap
Maximum SDU	Size in Octets	s1ap
Maximum Bit Rate for Uplink	kbps	s1ap
Maximum Bit Rate for Downlink	kbps	s1ap
Residual Bit Error Rate	Fractional Value	s1ap
SDU Error Ratio	Fractional Value	s1ap
Transfer Delay	ms	s1ap
Traffic Handling Priority	Relative Importance	s1ap
GBR for uplink	kbps	s1ap
GBR for downlink	kbps	s1ap
ECN	Whether Enabled	ip
Src Statistic Descriptor	0,1	s1ap
Maximum Bit rate Downlink (Extended)	Mbps	s1ap
Guaranteed Bit Rate Downlink (Extended)	Mbps	s1ap
Maximum Bit Rate Uplink (Extended)	Mbps	s1ap
Guaranteed Bit Rate for Uplink (Extended)	Mbps	s1ap
Radio Priority	Value	s1ap
Packet Flow Identifier	1-4	s1ap
APN Aggregate Maximum Bit Rate Downlink	kbps	s1ap
APN Aggrgate Maximum Bit Rate Uplink	kbps	s1ap
Total APN Aggrgate Maximum Bit Rate Downlink Extended	Mbps	s1ap
Total APN Aggrgate Maximum Bit Rate Uplink Extended	Mbps	s1ap
Service Type	Description	s1ap
DSCP	Default,CS1 to CS7	ip
Resource Type	GBR/Non-GBR	Der
Is Mission Critical	Boolean	Der
Is Low Latency	Boolean	Der
Forwarding type based on DSCP	AF,DF,EF	Der

formalized. The only difference is that 5QI applies to a flow carried at some point in a bearer, while QCI applies to a bearer within which certain types of flows are expected.

The service type being descriptive is replaced by a set of boolean attributes. New attributes to indicate if these packets are mission-critical and belong to the low latency category are framed. The DSCP field is categorized into assured, expedited, and default forwarding. The output attribute would be a priority that can hold continuous fractional values between [0.1,10], with 0.1 being the highest priority and 10 being the lowest.

c) Algorithm Selection and Model Construction:

The traffic analysis is performed through supervised techniques like Support Vector Machine, Random Forest, and Gradient Boosting. These are compared to select the best regressor.

3.6 Priority Class Based Packet Scheduling

We observed in the earlier section the traffic classification module determines the priority of the packets. It is important to treat priority levels of [0-2) like emergency since it consists of mission-critical users, along with delay-sensitive and IP Multimedia Subsystem (IMS) packets. Packets with priority levels 8+ can be scheduled only when the network is available without contention. These packets form a significant population. We model priority-based scheduling as a class-based formulation. Priority levels [0-2] are treated as Class A. Classes C [5-8) & B [2-5), which encompass GBR and Non-GBR flows, constitute a division into B1, B2, C1, and C2 sub-classes as depicted in Figure 3.1. Packets can be ordered as a function of priority arrival time denoted through Ordering Function $Ord(Packets)$.

The classes can be properly defined as follows:

- Class A (Priority Levels 0-2): This class represents the highest priority packets, including mission-critical users, delay-sensitive packets, and IMS packets. These packets are treated as emergency traffic.
- Class B (Priority Levels 2-5): Class B encompasses both GBR (Guaranteed Bit Rate). It is further divided into two sub-classes, B1 and B2. Here, B1 represents conversational voice and relative gaming flows, where low packet loss and latency are needed. B2 consists of other guaranteed bit flows like buffered streaming.

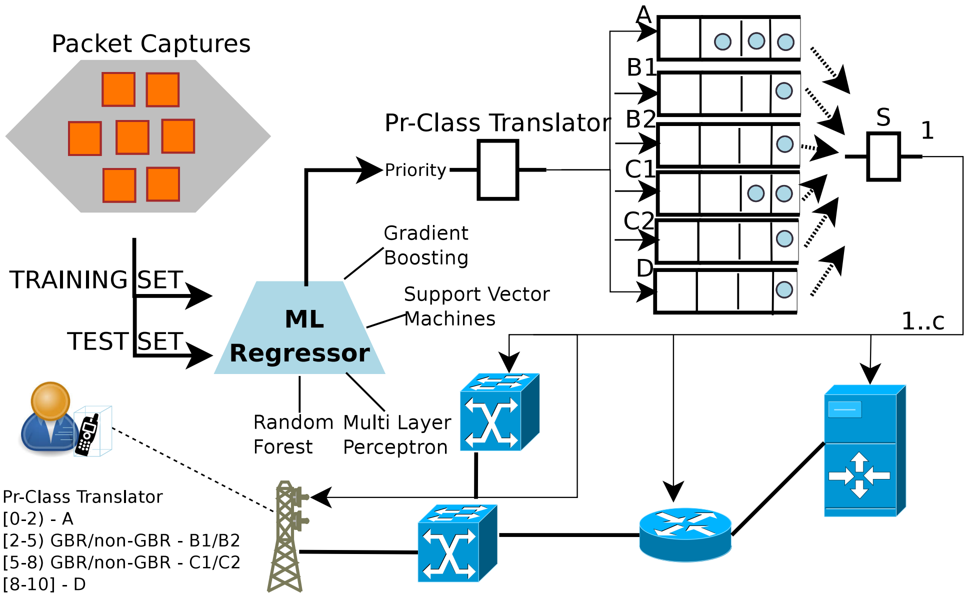


FIGURE 3.1: Packet Based Priority Scheduling

- Class C (Priority Levels 5-8): This class includes Non-GBR flows and is divided into two sub-classes, C1 and C2. C1 represents Non-GBR conversation voice and interactive gaming. C2 consists of Non-GBR TCP-based applications.

- Class D (Priority Levels 8+): This class represents the lowest priority packets. Packets with priority levels 8+ can be scheduled only when the network is available without contention.

The priority-based scheduling is modelled using these classes, with different levels of priority assigned to each class.

Approach 1

Here, the approach for scheduling the packets is deterministic. We have six sub-divisions. We refer to sub-divisions or classes as Units of Work (UoW) to be serviced. In a scenario where every sub-division has a packet to be scheduled:

–*At t_0, class A is slotted.* (Round 1)

–*At t_1, class A is slotted again.* (Round 2)

–*At t_2, class B GBR is slotted.*

–*At t_3, class A is slotted.* (Round 3)

–*At t_4, class B GBR is slotted.*

–*At t_5, class B non-GBR is slotted.*

–*At t_6, class A is slotted.* (Round 4) ...

–*At t_9, class C GBR is slotted.*

–*At* t_{10}*, class A is slotted.* (Round 5) ...

–*At* t_{13}*, class C GBR is slotted.*

–*At* t_{14}*, class C non-GBR is slotted.*

–*At* t_{15}*, class A is slotted.* (Round 6) ...

–*At* t_{20}*, class D is slotted.*

In total, it consists of six rounds, and it gets repeated after six rounds. On round i, it covers i UoWs in i timeslots in the order of priority of UoW. Class A is of the highest priority, and Class D is of the lowest one. Class A is serviced in every round, whereas class D is allotted only in the sixth round. If a particular subclass doesn't have traffic to be serviced, timeslots are not unutilized but rather identified in advance through an indicator. Through this indicator, schedulers denote if the UoW needs to be serviced in the round. Hence, every timeslot is effectively utilized as long as traffic is in the switch.

Approach 2

This approach is based on randomized probability. Much like the deterministic approach, we have the same concept of classes and UoW. All six sub-class have been assigned a probability: $p_1 > p_2 > p_3 > p_4 > p_5 > p_6$, such that the sum of probabilities =1. A sub-class is chosen based on the outcome of the random variable X. W.r.t every subclass, one can apply geometric distribution following with success as p_i and failure as $(1 - p_i)$. As approach 1, if the sub-class j is selected, and if it doesn't have packets to be serviced, the subsequent next highest priority sub-class holding the packets is serviced. W.k.t in a typical system, the arrival rate of higher priority packets is much lower than the default ones. Hence, the exponential distribution is considered for the packet arrivals whose priority range lies in [0-10].

Packets were scheduled based on priority dealt through precedence classes. Considering the packets are found in all queues, class j queue would be scheduled for $2\sum_i^n (j \geq i : 1 : 0)$ times in every $n(n+1)$ timeslots. The probability of the data traffic to be picked up for processing and transmission is:

For each class $j, p_j = \dfrac{2(n-j+1)}{n(n+1)}$. Assuming the traffic follows exponential distribution and the model is M/M/1, the probability distribution function of the amount of traffic in each class is:

$$\kappa(i) = \begin{cases} 1 - e^{-2x}, & \text{if class } i == 1 \\ e^{-2(i-1)x} - e^{-2ix}, & \text{if class } i \geq 2 \end{cases} \tag{3.14}$$

The overall utilization from each class is given by, $\rho(i) = \dfrac{\kappa(i)}{2\mu p_i}$.

$$\rho(i) = \begin{cases} \dfrac{n(n+1)(1-e^{-2x})}{4\mu(n-i+1)}, & \text{class } i == 1 \\ \dfrac{n(n+1)(e^{-2(i-1)x}-e^{-2ix})}{4\mu(n-i+1)}, & \text{class } i \geq 2 \end{cases} \tag{3.15}$$

The average time spent for a packet in the queue q belonging to the class i, $W_q(C_i) = \dfrac{\kappa(i)}{\mu p_i(2\mu p_i - \kappa(i))}$. Similarly, the expected number of messages in the queue q of class i is, $L_q = \dfrac{\kappa^2(i)(n-i+1)}{\mu^2 n^2(n+1)^2 - \mu n(n+1)\kappa(i)}$

For M/M/c queuing model, the overall utilization from each class is given by , $\rho(i) = \dfrac{\kappa(i)}{2\mu c p_i}$. From the same consumption queue, the scheduler dispatches processing for c servers. The expected number of messages in the queue q of class i is, $L_q = \dfrac{\varrho_0(\kappa(i))^{c+1}}{c!\,(2\mu\rho(i))^{c-1}(2\mu\rho(i)-\kappa(i))^2}$, where ϱ_0 denotes the probability that there are 0 packets in the system, $\varrho_0 = \dfrac{1}{\sum\limits_{m=0}^{c-1} \dfrac{(c\rho)^m}{m!} + \dfrac{(c\rho)^c}{c!(1-\rho)}}$

3.7 Virtual Backbone formation

At first, the backbone is initialized with zero nodes. The setup is considered to be a bipartite graph with the user terminals and points of interest on the left. The network in the middle, as shown in Figure 3.2, needs to provide coverage connecting the resources. The middle layer is composed of backbone nodes in radio access, transport, and core slices. When the request allocation is fulfilled, Backbone nodes are elected based on their fitness function (fit) and proximity (pr) to the target node t. Fitness is a function of multiple objectives which are varied through the path traversal.

Thresholds (Th) are established for both fitness and proximity functions, above which the nodes are privileged to be selected as the next hop tree node. The computed fitness and proximity values would be needed in the next-hop selection.

Next Hop Selection Function: While dealing with next-hop and when we find the presence of multiple nodes in the transmission range whose valuation is higher than both thresholds Th_{pr} and Th_{fit}, we do the following:

— The Tree node with the highest value for the proximity function pr is selected as the next hop.

— When there are no tree nodes, the node with the highest fitness value fit is elected as a Tree node and chosen as the next hop.

When a slice allocation request is received, paths are established from source to target nodes, and the path allocation is invoked until the requested bandwidth is allocated. The algorithm selects the next-hop node based on proximity and fitness functions and establishes that communication happens entirely via tree nodes. The virtual backbone ensures nodes that are not part of the backbone can operate passively. Backbone should select the best out of the immediate possible alternatives during the search. Energy is conserved for the nodes that aren't part of the backbone. This solution can be classified as a typical dynamic power-saving technique with QoS guarantees. When there are node outages or as part of the global CCs re-structuring, the virtual backbones would be altered.

To prevent the overuse of tree nodes, causing congestion, the residual capacities are updated once the tree nodes are allocated for a slice request. Subsequently, the revised residual capacity is used for fitness function computation. Hence, this would lead to lower fitness and not be used for servicing subsequent slice allocation requests.

As indicated in Fig. 3.2, the backbone tree nodes span across RAN, Transport, and Core Network sub-slices. It connects sources to resources, thereby forming an end-to-end NS.

Through the cognitive cycles, the network is observed and assessed. It identifies the path populated through the virtual backbone and can be optimised through re-structuring. Here, we apply the Dijkstra algorithm during cognitive cycles on the existing virtual backbone. Recombination involves selecting paths and using Dijkstra's algorithm to find the shortest path for each request. The cognitive cycle mimics natural evolution by iteratively selecting paths and nodes from the population generated in the initialization phase.

In summary, Cognitive cycles (CC) enable a node to learn, gain knowledge from prior experience, and act to adapt to the dynamic network conditions [61]. CC are a set of cascading recurring patterns. Each CC senses the current situation and interprets it about ongoing goals. Then, it selects an internal or external action in response.

Cognitive cycles can be used to monitor performance and gain insights on the network. In this work, we have used cognitive cycles to reconfigure the network path allocation to yield better

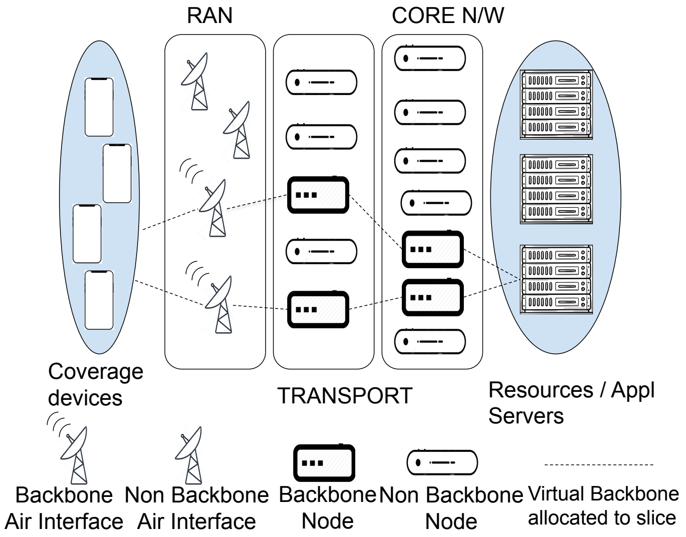


FIGURE 3.2: Virtual Backbone in Network Slicing

performance periodically. We use the dynamic power savings technique, with inspiration from EARTH model. The idea is that only the backbone nodes are meant to be always active. Other nodes can be in sleep mode, which is not used for routing. This helps in reducing the power consumption of non-backbone nodes.

3.8 Simulation Setup - Mininet, Flowvisor, and Controllers for Network Slicing

The system setup consists of:

a) Traffic analyzer: The captured and filtered traffic are fed into the traffic analyzer component. Here, as we discussed, the ML techniques are applied to derive the different QoE classes of users. Tools like Jupyter Notebook and python API are used.

b) Channel Assignment and Priority Scheduling Engine: The network is segmented as per slice requests. The proposed packet scheduling algorithm governs the priority routing, and channel assignment (virtual backbone creation and assignment) provides the routing infrastructure setup.

c) CC Analyzer: CCs orient themselves and observe the network slice performance. The analysis is performed based on QoS metrics, which decide the logical clustering of network nodes and the re-orientation of network slices.

The same network slicing setup is used across the work. It comprises Mininet, POX and Beacon controllers, and Flowvisor. Let us discuss a few details on these entities.

We have emulated the network of hosts, links, and switches through Mininet. This tool provides rapid prototyping for Software Defined Networks (SDN) and acts as the data plane in the slice. The bash process emulates the hosts running on the network namespace, and it is composed of a private network interface. Switches are software-based OpenvSwitch or OpenFlow reference switches. Links are virtual ethernet pairs which connect the Mininet switches and hosts. Notably, Mininet-Wi-Fi extends virtualized access points to this ecosystem.

The control plane consists of POX and Beacon controllers. These controllers are implemented in Python and Java, respectively. These controllers adapt OpenFlow devices into a hub, switch, load balancer, and firewall devices, and yield faster deployment and prototyping.

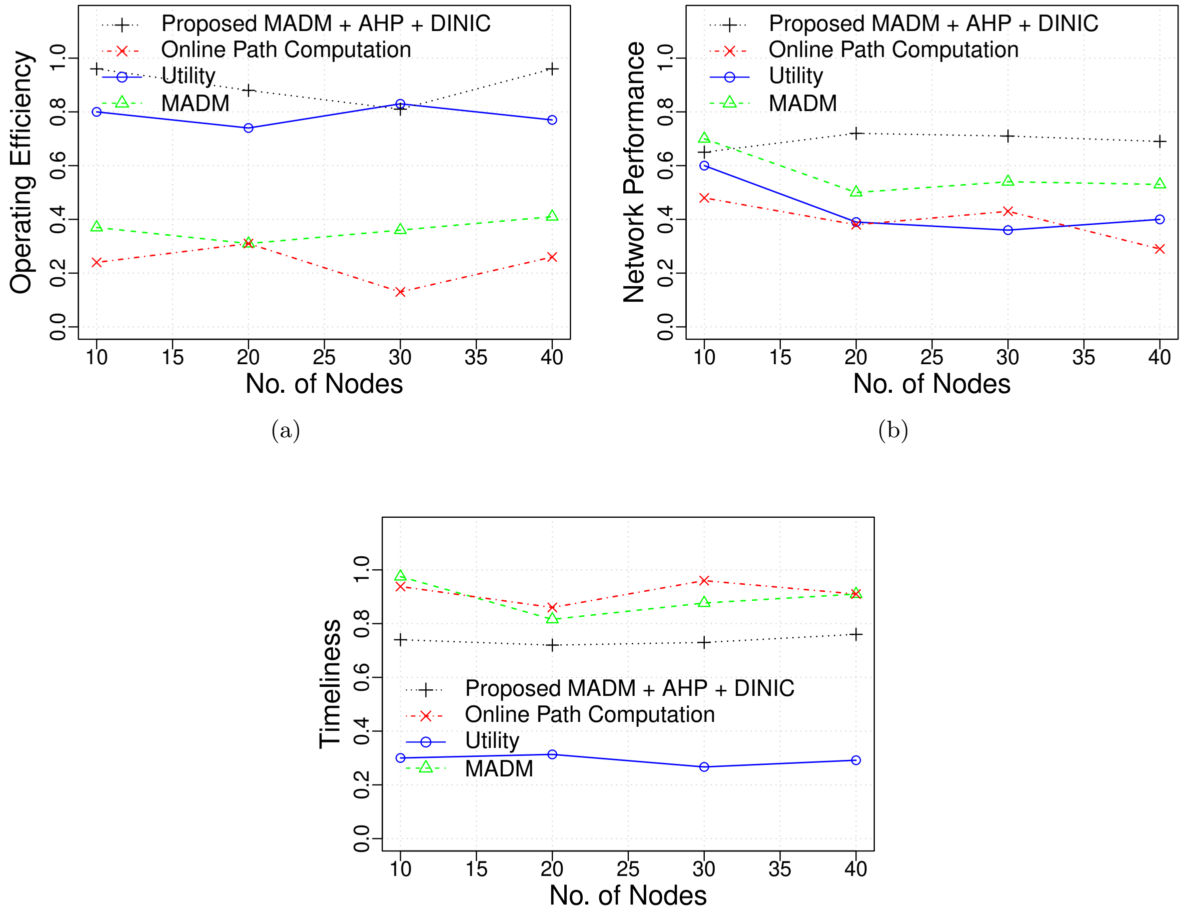
Flowvisor, a special-purpose controller, creates slices of virtual network resources, and it delegates control of each network slice to the configured controller. These slices constitute a combination of layer 1 - switch ports, layer 2 - source and destination Ethernet address, layer 3 - source and destination IP address and layer 4 - TCP or UDP port. The Flowvisor enforces isolation between the configured slices.

Virtual backbone creation and re-orientation are simulated through Flowvisor and the control plane. RAN partitioning is achieved by allocating a set of subcarriers (in the frequency domain) for the slice request in the allotted time domain.

3.9 Simulation Results and Analysis: QoS-Driven AHP-based Resource Allocation

The results of QoS-Driven and MADM-AHP-based resource allocation for stakeholder objective is discussed in this section. A linear topology is implemented in Mininet, wherein each switch is connected to a single host and interconnected in a straight line. The network configuration is established using commands like $"sudo\,mn\,--topo\,linear,4\,--link\,tc,bw=x,delay=y\,ms"$, to generate the Mininet network. This configuration allows for specifying desired bandwidth, delay, and packet loss parameters for individual links.

The network's performance is assessed using tools such as IPERF, which measures the available bandwidth, and by conducting controlled ping floods to calculate latency and assess packet loss.



(a)

(b)

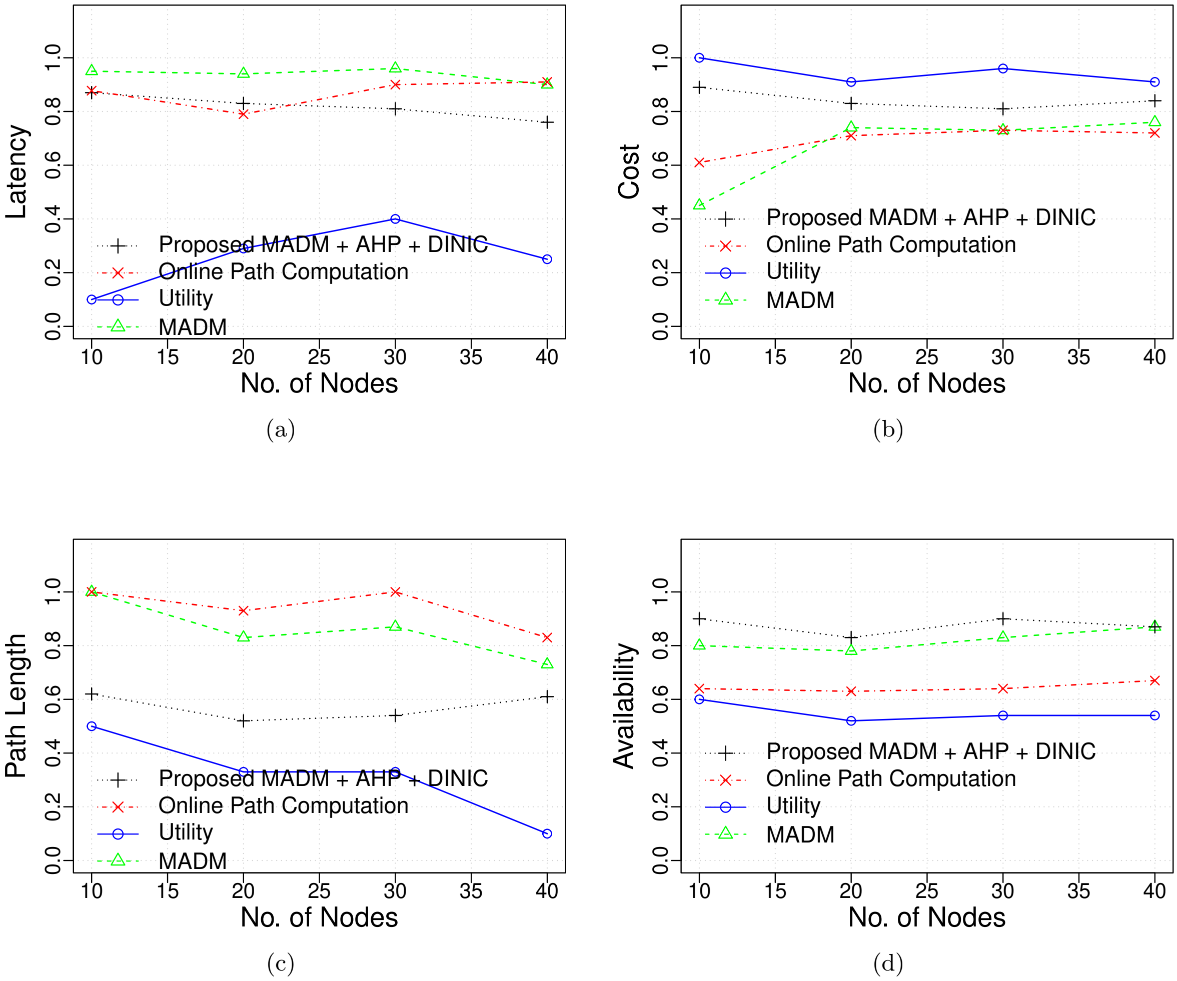
(c)

FIGURE 3.3: Satisfaction Levels of Network Slice Allocation

The number of hops determines the path length, indicating the count of intermediate devices data traverses between the source and destination.

Eq 3.10 computes the entropy function of a path across a hierarchy. Hierarchy consists of attributes $j = 1$ to $|H_{Attr}|$. For e.g., for hierarchy - Timelines, j = {latency, path length}. For a given path p on a hierarchy H_i, the entropy function is denoted by $WH_{H_i,p}$. Here, hierarchy is denoted by H={Timeliness, Operating Efficiency, Network performance}. The individual attributes encompassed in them are represented in Table 3.1, and their notation and equations are represented in Sections 3.3 and 3.4.

The satisfaction level is the normalized value represented between 0 to 1 that the path can achieve on a given hierarchy. Hence, it is used as a metric in the comparison. This normalization is achieved through the well-described Enhanced Max min method in Eq. 3.13. Satisfaction is an absolute measure between 0 and 1, which $WH_{H_i,p}$ can reach for any hierarchy H_i and path p.



(a)

(b)

(c)

(d)

FIGURE 3.4: Satisfactory levels and performance of against QoS parameters

a) Our proposed algorithm is compared against the utility theory (cost function), pure MADM approach, and online path selection algorithms [42]. In Fig. 3.3, the x-axis represents the number of nodes in the graph, and the y-axis measures the satisfaction level (SL). For every configuration, 4 simulation trials with independent topology and weights are run, and we compute the mean results against each parameter. SL lies between 0 and 1 computed as per the equations (3.9)-(3.13) which follows relative normalization. In Fig. 3.3, we compared these algorithms using Operational Efficiency, Timeliness, and Network Performance.

We can infer that the satisfaction index of the proposed algorithm strikes a balance and optimizes for different stakeholders of NS. Utility Theory focuses on Operational Efficiency leading to poor timeliness and network performance. Online Path Selection in [42] has mixed results across these hierarchies.

In this context, we've normalized Operating Efficiency and Network Performance on a scale from 0 to 1. When we juxtapose our proposed approach with the MADM method, it becomes evident that our approach outperformed the latter. This was manifested in the form of improved Operating Efficiency and Network Performance achieved by our approach in Fig. 3.3. Even with timeliness, and the proposed approach is closely behind pure MADM.

b) The proposed approach and well-known algorithms are compared against individual criteria (Latency, Cost, Path Length, Availability) in Fig. 3.4.

Cost is the financial investment required to create, maintain, and operate network paths and slices. This includes capital expenditures (CAPEX) for initial infrastructure setup and operating expenses (OPEX) for ongoing management. Cost considerations include equipment, energy consumption, maintenance, and leasing network resources. Computation in a Realistic Network: To compute the cost of a network slice, you would need to consider factors like hardware and software expenses, energy consumption, and maintenance costs. These can be quantified based on the pricing of network components, power consumption rates, and expected maintenance frequency.

Network performance refers to the ability of a network slice to meet certain quality of service (QoS) and performance requirements. This includes bandwidth, latency, packet loss, jitter, and throughput metrics. Different applications and services require varying levels of network performance to function optimally. In this research work, for the Analytic Hierarchical structure, Throughput and Availability are considered to be constituents of Network Performance. We have carved out another domain, Timeliness to capture Latency, Path length, etc.. Computation in a Realistic Network: Network performance metrics can be measured through various tools and techniques. Bandwidth can be measured using tools like iPerf. Latency can be measured using ping or specialized latency measurement tools.

Availability refers to the ability of a network slice to remain operational and accessible to users. High availability ensures that services provided by the slice are accessible with minimal downtime. Availability is often measured as a percentage of time the service is up and running. Computation in a Realistic Network: Availability can be calculated by monitoring the slice's uptime and downtime over a specific period. The formula for availability is:

$$Availability(\%) = (Uptime/(Uptime + Downtime)) * 100 \tag{3.16}$$

Uptime and downtime can be measured using monitoring systems that track the state of the network slice and detect outages.

In our study, we define availability through the probability of failure in links along the path of the slice. The proposed algorithm defines a primary and secondary (next best) path to the slice. This strengthens the fault tolerance in the slice. Theoretical availability is defined as (1 − Probability of failure along both the allocated paths of the slice). In a realistic network slicing scenario, these factors are interrelated. For example, higher network performance might come at a higher cost due to the need for more advanced hardware or increased resource allocation. Similarly, ensuring high availability might require redundant infrastructure, leading to increased costs. Balancing these factors is crucial to designing effective network slices that meet the needs of various applications and services.

$\hat{A}$ values are referred to here to evaluate the compensatory algorithms using EMM. Online Path Selection Algorithm [42] outperforms all the algorithms in terms of path length. In terms of cost, the proposed algorithm is comparable with utility theory. Pure MADM fares well with most QoS parameters except the overall cost incurred.

3.10 Results and Analysis - Traffic Classification

The data orientation of the captured packets represents an aggregated function of Resource Type and Allocation Retention Priority in Tables 3.3 and 3.4. The regression function of predicting the priority computed against a set of 5, 10, 15, and 20 QoS parameters. The number of QoS parameters is denoted on the x-axis. The experiment is performed on the s1ap dataset [111] Wireshark and pcap captures. The train-test split evaluation is through the Scikit-learn library by configuring the random seed value. Two different sets of runs are assessed:

i) The training set would comprise 40% of the dataset, while the rest represents the testing set.

ii) Training dataset with 70% of total captures and test set with the remaining 30%.

We use the default settings of each of the ML regression algorithms. Emerged outputs of the algorithms are compared against standard metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). MAE takes the average of this error from every sample in a dataset. RMSE is the square root on the average of the square of the difference between the

Table 3.3: Insights on Data distribution - Mean values on group function of Resource Type.

Attributes	***GBR***	***Non GBR***	***Attributes***	***GBR***	***Non GBR***
QCI	30.94	16.82	Priority	2.32	5.73
ARP Priority	1.714	6.84	MBR DL (Mbps)	128	96
Pre-Emption capability	0.22	0.46	Packet Length	268.8	404.3

Table 3.4: Insights on Data distribution - Mean values on group function of ARP Priority.

Attributes	***1***	***2***	***9***	***15***
QCI	4.3	2	9.0	9.0
Priority	1.7	2.8	9.0	9.0
Length	367.8	312.0	290.8	512.0
Pre-Emption capability	0.2	0	1	0

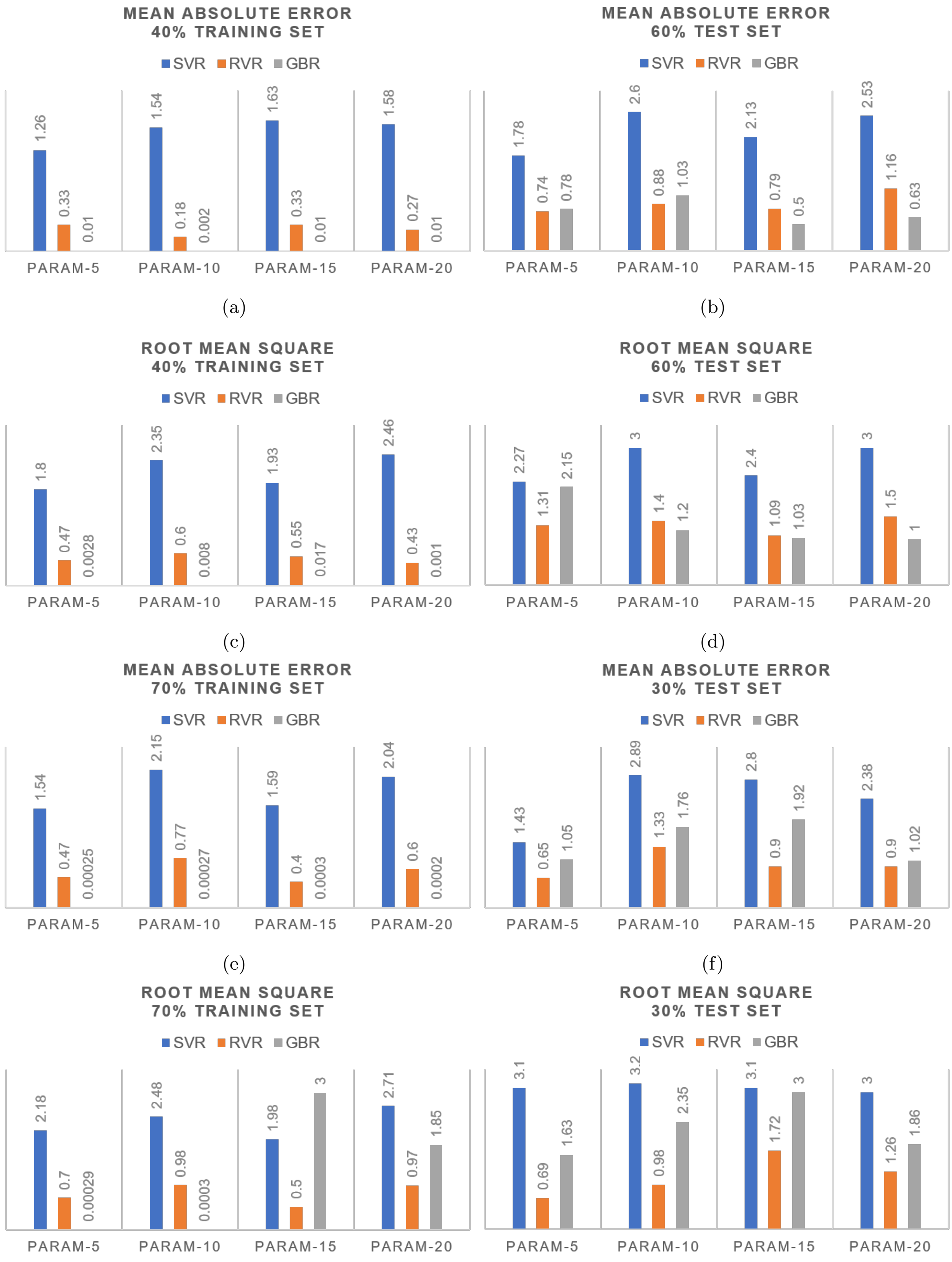
original and predicted values of the data. It indicates the spread of the error. Fig. 3.5 represents how each regressor fares against the other.

Support Vector Regressor (SVR) produces acceptable error rates, however, it doesn't perform relatively with the others. Random Forest Regressor (RVR) and Gradient Boosting Regressor (GBR) provide comparable results with negligible error rates. We would recommend RVR and GBR for further traffic classification. Against the five-parameter QoS run, prominent attributes like QCI, DSCP, Is Mission Critical, Resource Type and Allocation Retention Parameter were considered from the tabulated QoS parameters. The runs with 5 and 10 parameters provided satisfactory outcomes, and it can be perceived higher parameters lead to overfitting from Fig 3.5. When repeated with different hyperparameters like depth and a number of estimators, we could notice the QCI feature, being a prime constituent attribute, plays a key role in traffic prioritization.

3.11 Results and Analysis - Virtual backbone

In this section, we discuss the results of the virtual backbone based approach for joint QoS and energy savings.

The simulation settings are captured in Table 3.5. The field setting constitutes different combinations of Nodes [500,1000,2000], and Slice Requests [25,50,100] indicated by FS_1 - FS_5 in Fig. 3.6. It captures the mean and confidence intervals of no. of nodes consumed for intermediate



(a) (b)

(c) (d)

(e) (f)

(g) (h)

FIGURE 3.5: Error in predicting priority through Regression-based ML algorithms. Mean Absolute Error for 40% training and 60% test are shown in (a) and (b), respectively. Root mean square error for 40% training and 60% test plotted in (c) and (d) respectively. Mean Absolute Error for 70% training and 30% test are shown in (e) and (f), respectively. Root mean square error for 70% training and 30% test plotted in (g) and (h) respectively.

routing. 98% confidence intervals are computed across 1000 Monte Carlo runs in each field setting. In every run, we invoke the Erdos Renyi graph generator to fix the positions of nodes within the testbed. Among these runs, the slice resources are located near the centre of the field setting. In this section, we compare the following algorithms:

i) Proposed Online Virtual Backbone solution, ii) Routing Information Protocol based on minimum hops (RIP-Min Hops), iii) Online Path computation and Function Placement which relies on Dijkstra shortest path algorithm (PC-FP), and iv) Proposed Cognitive Cycles Based Approach (CC Cycles).

The time complexity of constructing a virtual backbone for a slice request is $O(V + E)$, and space complexity is $O(V)$ since an extra visited array of size V is required. The search strategy is similar to the Depth-first search. For the candidate algorithms, RIP - Min Hops and Dijkstra shortest path, the time complexity is $O(V^2)$. When the input graph is represented using an adjacency list, the time complexity is $O(E \ log\, V)$ with the help of a binary heap. The reformation of the virtual backbone using cognitive cycles inherently uses Dijkstra's algorithm for the broken path. The time complexity is $O(E' \ log\, V')$, where $E' \subset E$ and $V' \subset V$.

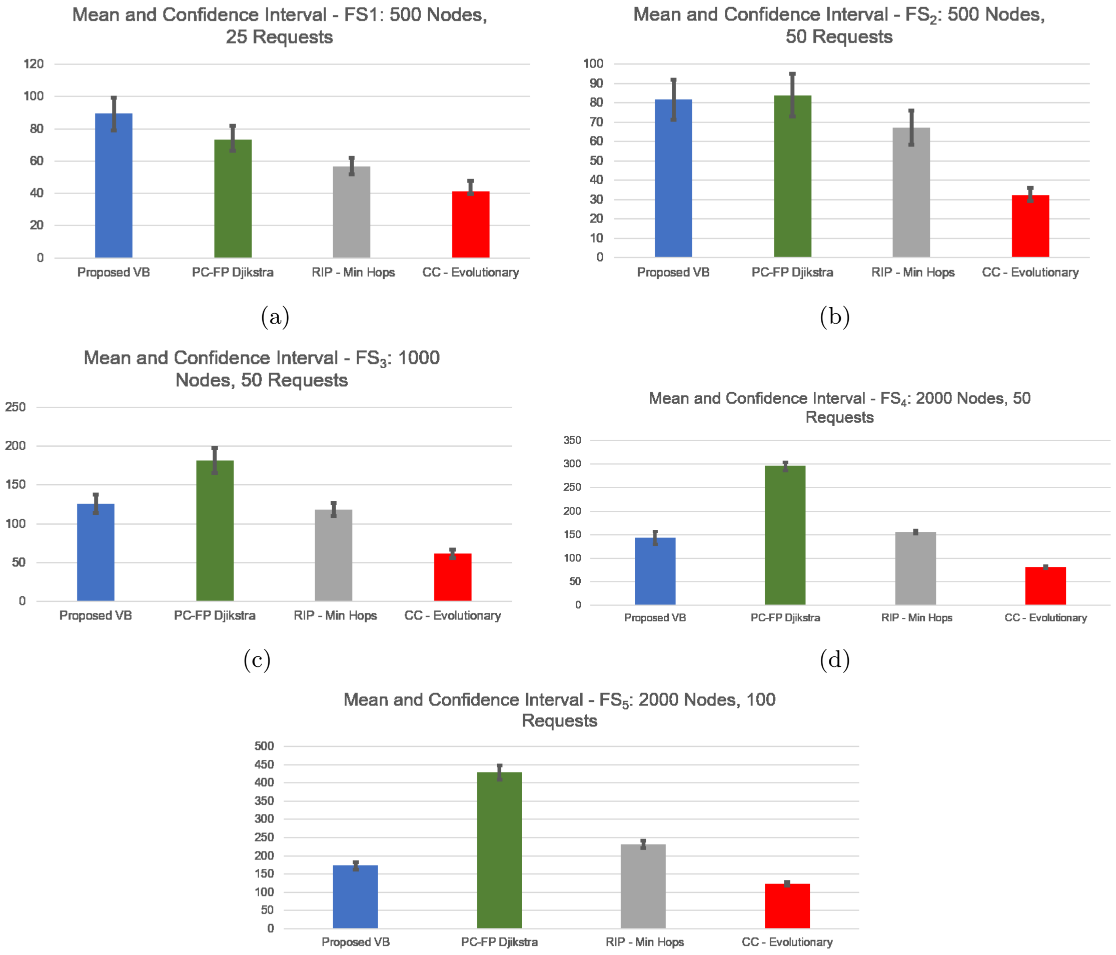
Every slice in the experiment is assigned a total of 20MHz bandwidth. For example, Wi-Fi uses Orthogonal Frequency Division Multiplexing (OFDM). Here, 64 OFDM subcarriers span over 20MHz of bandwidth. 11 subcarriers are used as a guard band between two adjacent channels, 4 are pilot subcarriers, and the centre subcarrier is inactive. Effectively, 48 subcarriers are applied for slice transmission. Since the Sub Carrier Spacing amounts to 312.5kHz, 15.323 MHz bandwidth is effectively achieved when all data sub-carriers are allotted. Using 64-QAM with one spacing stream, MCS Index 6, 800ns guard interval, and 3/4 coding rate, the data rate is $\approx 58.5 Mb/s$. When n slices (RAN partitions) share the sub-carriers, it reduces to $\approx {}^{58.5}/_{n}\, Mb/s$.

From Fig. 3.6, we can observe the proposed virtual backbone solution consumes no. of routing nodes in the same range as compared to RIP-Min Hops and PC-FP. It should also be noted the proposed virtual backbone solution is online-based, and it has a view only on its next-hop selection, whereas the other two algorithms have a global view of the field setting.

VB solution provides a leaner confidence interval indicating that the number of nodes employed is at close quarters across the independent runs in every field configuration. In terms of no. of nodes utilized, one can also observe RIP-Min hops faring better than PC-FP.

Table 3.5: Simulation Settings - Virtual Backbone

Parameters	**Value**
Nodes	[500, 1000, 2000]
Slice Requests	[25, 50, 100]
Simulation Runs per each setting	1000 Monte Carlo runs
Location of Slice resources	Center of testbed
TestField area	10000 sq. feet area
Transmission Range (TR) of AP	1000 sq. feet area
Location of user or slice requests	Erdos Renyi Graph Generator in TR for the APs in the testbed
Erdos Renyi Probability to access nodes within TR	1
Fitness function threshold (Th_{fit})	top 60% percentile
Fitness function scalar co-efficients	20% capacity, 20% distance, 20% energy, 20% delay, 20% cost
Proximity fn threshold (Th_{pr})	top 60% percentile
Node selection in a slice	randomly selected connected graph consisting of nodes with an allocated capacity. Originates from source to resources
fit, pr co-efficients	0.6 and 0.4, summative additive weighing applied


Mean and Confidence Interval - FS1: 500 Nodes, 25 Requests

(a)

Mean and Confidence Interval - FS$_2$: 500 Nodes, 50 Requests

(b)

Mean and Confidence Interval - FS$_3$: 1000 Nodes, 50 Requests

(c)

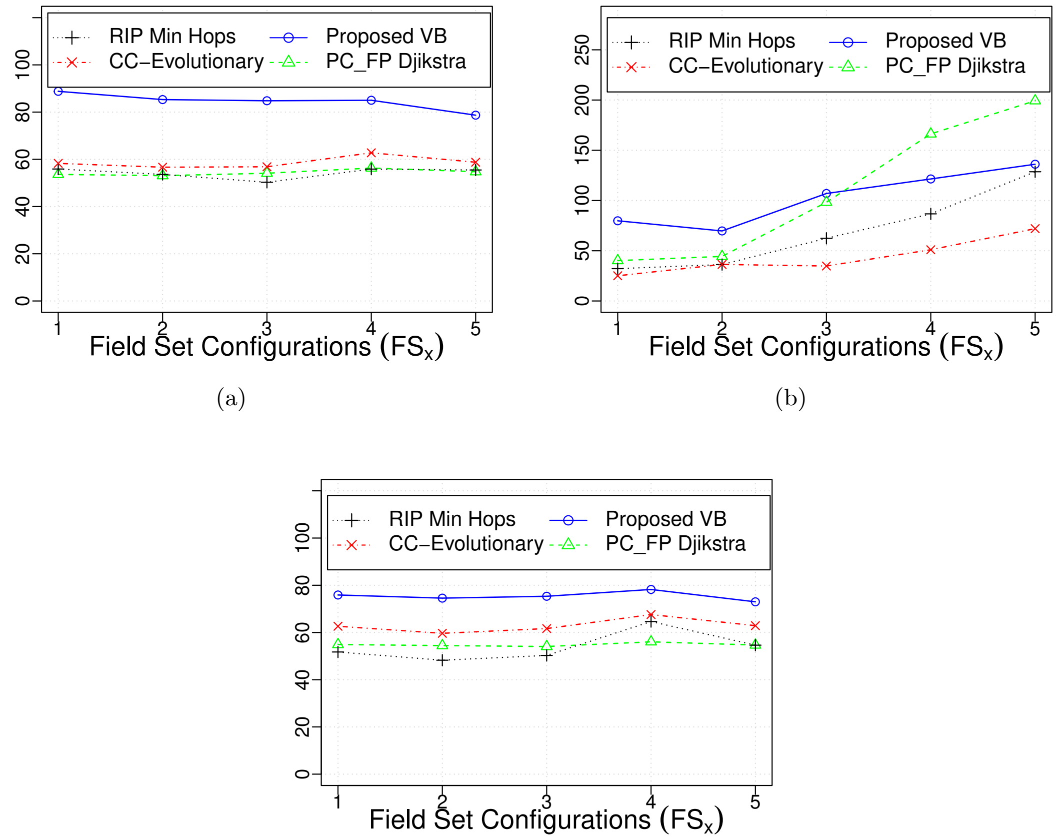
Mean and Confidence Interval - FS$_4$: 2000 Nodes, 50 Requests

(d)

Mean and Confidence Interval - FS$_5$: 2000 Nodes, 100 Requests

(e)

Figure 3.6: Node Utilization against different configuration (Mean and 98% Confidence Interval)



(a)

(b)

(c)

FIGURE 3.7: (a) Percentage of Routing nodes Greater than Threshold (b) No. of Routing Nodes whose energy is greater than the threshold (c) Percentage of the Energy level of routing nodes

The Cognitive cycle-based approach optimizes on top of the virtual backbone and attains the least number of nodes for routing in Fig. 3.6. This evolutionary algorithm uses the proposed VB and random paths during its initialization phase. In the recombination phase, a randomized variant of PC-FP is employed, which helps it to produce far better results in terms of node consumption.

Thresholds are set for both fitness and proximity functions. In Fig. 3.7a), when we take a closer look at the percentage of routing nodes with Open vSwitch, their fitness and proximities are greater than the threshold ($fit, pr > Th_{fit}, Th_{pr}$), the proposed virtual backbone solution employs the highest. All the other approaches are comparable to each other, as shown in Figure 3.7a. Out of the total nodes, which are part of the final solution, VB has the highest percentage of nodes greater than proximity and fitness thresholds. In this regard, it fares better than all the other algorithms.

Fig. 3.7b) depicts the number of routing nodes that exceed the threshold metrics. One could notice PC-FP has the highest numbers among other solutions. It is obvious that this algorithm has the highest node consumption (it can be seen in Fig. 3.6, the mean number of nodes consumed is far greater than others). Due to this ripple effect, PC-FP has the highest value as compared to other candidate algorithms. Even here, we could observe the proposed VB solution producing better results.

Assuming the nodes are assigned random energy levels between 0 and 100 during the initialization phase of each field setting, the percentage of static energy levels of elected routing nodes are shown in Fig. 3.7c). Based on the application use case, the energy levels can also be an attribute of the fitness function. Again here, the virtual backbone yields better results.

3.12 Conclusion

The first objective of the chapter was around the study of end-to-end resource allocation in network slicing for tailored offerings. Our employed algorithm tries to bring in fairness and efficiency. This approach covers breadth and depth in online virtual paths discovery, evaluation, ranking, and assignments for slice allocation. The proposed approach took inspiration from the Dinic algorithm and MADM for addressing the formulated multi-objective constraint optimization problem. This approach uses simple and low-complexity techniques to allocate the virtual paths for slices. Detailed Simulation results prove that the proposed algorithm performs well in fulfilling stakeholder goals when compared with other candidate algorithms. We published this work [1] in the 91st IEEE Vehicular Technology Conference, VTC Spring 2020.

The next goal of the chapter was to jointly investigate QoS and energy savings. QoS is ensured in a two-fold manner. Firstly, traffic classification predicts packet priority. A comprehensive fine-grained list of QoS attributes is identified from S1AP and DSCP protocols. ML algorithms such as Random Forest, Gradient Boosting, SVM, and MLP are applied using these attributes. Random Forest and Gradient Boosting record lesser mean absolute percentage error of 0.79% and 1.22%, offering higher accuracy than other compared approaches. Following this, Class-based priority scheduling routes packets based on the predicted priority. Secondly, QoS metrics such as bandwidth, latency, jitter, and path length are an integral part of the fitness function. The fitness function plays a significant role in node selection during resource allocation in NS.

EE is achieved by reducing the number of nodes employed for routing. Virtual backbone and CC-based approaches are proposed to bring in energy savings in slices. Our experimental results show the basic VB utilizes 17.8% and the Cognitive cycle-based VB tree uses 8.4% nodes, respectively, which are lesser than other standard approaches. We published this work [2] in the Computer Communications, Elsevier journal, 2023.

Chapter 4

Swap-based Load Balancing in Radio Access Networks

4.1 Introduction

5G leverages millimeter waves (mmWave) to meet the high bandwidth demands in densely populated urban areas. As mmWave has a low range, 5G operators densify their networks with small cells to provide seamless connectivity and reliable coverage. In a real network setup, some microcells could handle most of the traffic while others remain idle. It causes overloaded cells to experience intermittent, unstable connectivity and high packet jitter.

In this chapter, we examine the load imbalance at the base station and the signal strength of the connected devices. The terms base station, access points, and cells are used interchangeably.

Here, we refer to traffic flow as the channel connection between the mobile device and the serving AP, which requires a certain bandwidth. Thus, the load of a traffic flow on a given small cell from a device would be the number of radio blocks allocated to service, i.e., the total data rate required to service the bandwidth requirements based on the Modulation Coding Scheme (MCS) between the device and the AP. The load on the micro or picocell is the sum-rate function of all its connections. The control plane drives the proposed reactive algorithm, which can be classified under association management.

In this work, we intend to ensure balanced loads among APs with heterogeneous access points. The proposed algorithm would be suitable for both homogenous APs like micro cell only deployments and heterogeneous networks (HetNets) like macro-small cell or micro-pico combinations applicable to 5G deployments.

We use parameters such as threshold, load per unit capacity, and load imbalance to classify APs, which are described in the next section. Furthermore, we aim for better signal quality for traffic flows that belong to the data plane. The control plane tracks the status of the APs in its coverage area.

Chapter 4, which discusses swap-based load balancing in radio access networks, can be related to the resource allocation aspect discussed in Chapter 3 on network slicing in the following ways:

a) Optimizing Resource Utilization: In Chapter 4, load balancing techniques are explored to distribute traffic and resources more efficiently. This relates to the resource allocation in Chapter 3 because effective load balancing helps in optimizing the utilization of network resources allocated to different slices.

b) Enhancing Network Performance: The load balancing techniques detailed in Chapter 4 aim to minimize load imbalance and improve metrics such as the Channel Quality Index (CQI) and Signal-to-Noise Ratio (SNR). These improvements directly impact network performance, which is a key consideration in resource allocation, as discussed in Chapter 3.

c) QoS and QoE Considerations: Chapter 3 emphasizes Quality of Service (QoS) attributes, while Chapter 4's load balancing techniques can influence QoS by ensuring a more balanced distribution of traffic and resources, ultimately affecting the end-user Quality of Experience (QoE) discussed in Chapter 3.

In summary, Chapter 4's discussion on load balancing complements the resource allocation and network performance optimization goals outlined in Chapter 3, creating a cohesive approach to managing network resources and ensuring a better user experience.

We establish the system model, variables, and constraints in Section 4.2. Then, we analyze one-way load balancing and propose a swap-based load balancing in Sections 4.3 and 4.5. Finally, section 4.6 compares the results of the discussed approaches.

4.2 Load Balancing System Model

The mathematical formulation of load distribution in radio access devices is a triple (F, P, R). Here F is a finite set of traffic flow of user devices, P is a finite set of macro or small cells and $(R(f) \subset P : f \in F)$ is a set of reachable APs. The bandwidth requirement is a vector $(B(f) : f \in F)$, where $(\forall f,\ B(f) > 0)$. The Signal-to-Interference-plus-Noise Ratio between the user connection f and the corresponding cell p is given by $SINR_{f,p}$. Let $\lambda(f, p)$ be the total data rate load, i.e., the number of resource blocks consumed for serving $B(f)$ by AP p based on Channel Quality Index (CQI), coding, and modulation rate between the device and AP.

The neighbourhood vector $R(f)$ denotes the cells that are reachable to the traffic flow f, with equitable RSSI so that it can be associated with $R(f)$. An *association* already exists $(\forall f \in F, R(f) \neq \emptyset)$ for admitted traffic flows in their respective neighbourhoods. The associated access point of a traffic flow is indicated by $A(f) \subset P : f \in F,\ A(F) \in R(f)$. A microcell can serve around 200 users, and a picocell can serve 32 to 64 users approximately [114]. The capacity of a micro cell (p_1) or picocell (p_2) is denoted by ζ_p, where $\zeta_{p_1} > \zeta_{p_2}$. The *total load* at an access point $p \in P$ is given by $q(p) = \sum \lambda(f, p)$, where $A(f) = p$. The Network Slice (NS) controller or control plane computes the load per unit capacity in its coverage area.

Load per unit capacity(ϕ): The total sum of loads $(\lambda(f, p))$ of all the traffic flows associated with the access points divided by the sum of the capacities (ζ_p) of all the access points in a given coverage area (C). It tracks the utilization factor of the coverage.

$$\phi_C = \frac{\sum_p q(p)}{\sum_p \zeta_p} : p \in P, q(p) = \sum \lambda(f, p) \text{ where } A(f) = p \tag{4.1}$$

The **threshold** or balanced state of an AP $(p \in P)$ in a given coverage area is the product of load per unit capacity (ϕ) and capacity of the access point $(\zeta_p : p \in P)$.

Load Imbalance of an AP(δ): The difference of the total load of an AP $(p : p \in P)$ and its computed threshold in the given coverage area (C).

$$\delta_C(p) = |q(p) - \phi_C \cdot \zeta_p| \tag{4.2}$$

NS controller in its coverage area (C) determines the state ($S(p) : p \in P$) of the APs as overloaded, fair, and underloaded,

$$S_C(p) = \begin{cases} \mathit{Overloaded}, & \text{if } (\phi_C \cdot \zeta_p + \epsilon) \ < \ q(p) \\ \mathit{Fair}, & \text{if } (\phi_C \cdot \zeta_p - \epsilon) \leq q(p) \leq (\phi_C \cdot \zeta_p + \epsilon) \\ \mathit{Underloaded}, & \text{if } q(p) \ < \ (\phi_C \cdot \zeta_p - \ \epsilon) \end{cases} \tag{4.3}$$

In 5G networks, micro and pico cells would co-exist; we strive to sustain the overall load proportional to the capacity of such cells. Meanwhile, we target to reduce imbalance among homogenous micro or pico cells only deployments. Here, ϵ is the margin above and below the threshold.

Total Load Imbalance(ψ): The absolute sum of the load imbalance of every access point ($\forall p \in P$) participating in the given coverage area (C).

$$\psi_C = \sum_p \mid \delta_C(p) \mid : p \in P \tag{4.4}$$

The **migration** (M) of a traffic flow (f) at a specific instant (t) from one access point AP (u) to another AP (v) is represented as $M(f,t) = (u,v)$. At time t, the traffic flow f is associated with AP u, and at time $(t+1)$, it is associated with AP v. In this context, u and v represent different access points within a network, and f is the traffic flow being relocated from u to v. Hence, at instant t, $A(f) = u$, and at $(t+1)$, $A(f) = v$.

$L_{f,p}$ quantifies the delay or latency experienced by the device associated with the access point (AP) p when handling a traffic flow (f) over the channel.

To begin with, we formulate the LB problem,

$$\forall p \ \mathit{minimize} \ \delta_C(p) \ : \ p \in P$$

$$\mathit{by} \, M(f,t) = (u,v) : f \in F, v \in R(f)$$

$$\mathit{minimize} \ L_{f,p}, \ \mathit{maximize} \ SINR_{f,p}$$

$$0 \leq t \leq T, u \, \text{and} \, v \in P, \text{and} \, u \neq v$$

The derived entity ψ_C is *minimized.* Further, we reduce latency $L_{f,p}$ and improve the signal strength $SINR_{f,p}$ of the traffic flow (f) while selecting the underloaded macro or small cell (p).

The constraints primarily revolve around ensuring that devices can establish a connection with the migrated access point, as indicated by the condition $v \in R(f)$. Additionally, the latency ($L_{f,p}$) and SINR ($SINR_{f,p}$) constraints are indirectly accounted for by introducing a biasing factor into the optimization process. This biasing factor helps strike a balance between these constraints and other objectives.

4.3 One-Way Traffic Distribution

One-way load balancing mechanism allows only unidirectional transfer of traffic flows from overloaded to reachable underloaded cell. The control plane would pick a traffic flow ($f \in F, \text{at } t+1 \Rightarrow A(f) = u, S_C(u)\, is\, Overloaded : u \in P$) randomly from overloaded cell and offload the traffic flow ($f \in F, \text{at } (t+1) \Rightarrow A(f) = v$) to the chosen underloaded cell ($v \in R(f), S_C(v)\, is\, Underloaded : v \in P$). The control plane filters the underloaded cells with available admissible capacity ($\alpha(k) \geq \lambda(f,k)$). $\lambda(f,k)$ denotes the total data rate or bandwidth consumption on cell k to support user experienced data rate $B(f)$ of the traffic flow f.

Algorithm 5: 1-way load balancing

```
function loadBalance $(f, u, A, q, \lambda, t, C, P)$
$n \leftarrow \{R(f) - u\}$;
while $(k : (n : S_c(n) == Underloaded))$ do
    $\alpha(k) = \phi_C \cdot \zeta_p - q(p)$;
    $Bias(k) = (1 + {}^{\alpha(k)}/\max_k \alpha(k)(1 - {}^{L_{f,p}}/\max_k L_{f,p}) \times \beta)$;
$h \leftarrow \arg\max_k SINR_{f,k} \times Bias(k)$
  such that $\alpha(k) \geq \lambda(f,k)\,\&\, S_c(k) == Underloaded$;
if $h \neq \emptyset$ then
    Migrate $M(f,t) = (u,h)$;
    Update $q(h), A(f), q(u)$;
else
    LB is not possible. Try $SLB$
```

We define ($\alpha(p) : p \in P, S_C(p) == Underloaded$) as the difference between the threshold and the current load of the underloaded access point in a given coverage area.

$$\alpha(p) = \phi_C \cdot \zeta_p - q(p) : p \in P \tag{4.5}$$

In Algorithm 5, among the reachable underloaded APs with available admissible capacity, instead of selecting the AP with the highest SINR to the potential underloaded AP, we elect the AP with the highest absolute product of $SINR_{f,p}$ and biased factor based on available admissible capacity and latency. The biasing factor is a normalized function, $(1 + {}^{\alpha(k)}/\max_k \alpha(k)(1 - {}^{L_{f,p}}/\max_k L_{f,p}) \times \beta)$, to provide a chance for associating flow to heavily underloaded APs and lower round-trip response time. $0 < \beta < 1$, ensures the association doesn't overtly incline towards heavily underloaded with low signal quality and doesn't connect with AP, enduring excessive latency. At an instant t, the coverage area reaches a steady state when the 1-way algorithm can't initiate any more migrations, and it is proved through Lemma 4.4 and Theorem 4.5. The average number of processed overloaded APs (p) through Algorithm 5 is $O({}^{|P|}/2 - \epsilon)$. This complexity is based on the real-world random selection of overloaded access points using a threshold that roughly bisects the total access point count. On average, the algorithm addresses roughly half of the overloaded access points, considering the ϵ factor, which represents a margin above and below the threshold.

Lemma 4.1. *In 1-way LB, $\nexists p \in P$ where before load balancing, $S_C(p) ==$ Underloaded and after $LB, S_C(p) ==$ Overloaded.*

Proof. In Algorithm 5, $(p : S_C(p)\, is\, Underloaded)$ cannot be chosen to receive f unless $\alpha(p) \geq \lambda(f,p)$, where w.k.t $\alpha(p) = \zeta_p \phi_C - q(p)$. Further, after LB, $q(p) = q(p) + \lambda(f,p)$ is updated. Thus, $q(p) \leq \zeta_p \phi_C$ and by *eq.*(4.3) proves post LB, $S_C(p)! = Overloaded.$ □

Theorem 4.2. *1-way LB reaches a steady state after $\forall p \in P : S_C(p) ==$ Overloaded are processed by control plane CP for a given time t in a coverage area C.*

Proof. To verify, we need to show $\nexists M(f,t) : f \in F$ after $\forall p \in P$, $S_C(p)\, is\, Overloaded$ are processed by Algorithm 5. Let's prove it by contradiction. Pre & post LB $\forall f \in F : A(f) = u$, and after *steady state* $\exists f : A(f) = (u)\, \&\, S_C(u)\, is\, Overloaded$, and $v \in P : S_C(v)\, is\, Underloaded$ and $M(f,t) = (u,v)$ is possible.

During 1-way LB, the Algorithm 5 has been processed for $\forall f \in F, A(f) = p \in P$, where $q(p) \geq \zeta_p \phi_C$. Either f should have got transferred or not transferred during LB.

Consider f is transferred, then during LB, it is migrated to $x : S_C(x)\, is\, Underloaded \rightarrow A(f) = x$, which contradicts the initial assumption that $A(f) = u$ post-LB. Now, consider f is not transferred during $LB \rightarrow \nexists v : v \in R(f)$ and v is underloaded with $\alpha(v) \geq \lambda(f,p)$. Clearly, v doesn't have admissible capacity ($\alpha(v)$) to accept f, and from Lemma 4.4, v in $S_C(v)\, is\, Underloaded$ before LB, can't become overloaded post LB. Hence, $\forall p$ where $S_C(p)\, is\, Underloaded$ can't accept anymore f from the overloaded APs. This disproves the initial statement $M(f,t) = (u,v)$ exists. At $t+1$ in C, when the field setting of F, P, $\forall f \in F$, $R(f), \lambda(f,p)$, and $A(f)$ changes, it would necessitate further migrations.

□

4.4 Swap-based Load Balancing

Migrating f is not possible through 1-way LB, when $\forall p \in P : S_C(p)\, is\, Underloaded$ and $\forall f, \alpha(p) < \lambda(f,p)$. The Exchange or Swap Load Balancing (SLB) is executed after 1-way LB, explores the possibility of migrating $M(f,t) = (u,v)$, by including a reverse transfer of reachable traffic flows K such that $K \subset F : A(K) = v$, $S_C(v)\, is\, Underloaded$, $u \in R(K)$ from underloaded to overloaded AP (u) denoted by $M(K,t) = (v,u)$ in Algorithm 6.

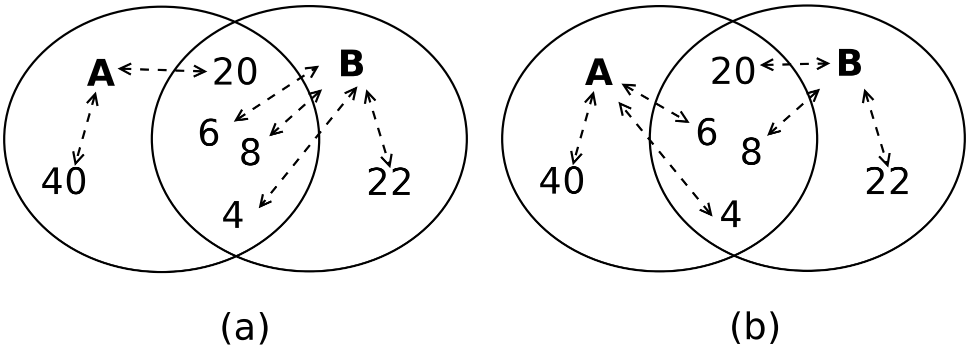


Figure 4.1: Swap Based Load Balancing

Like 1-way LB, SLB ensures no underloaded APs become overloaded. In Fig. 4.1a, we present an example with two homogenous picocells, A and B, with capacity x, and let us assume the MCI index is equivalent. Loads on picocells, $q(A) = 60$ constitutes $\{f_{40}, f_{20}\}$, and $q(B) = 40$ comprises $\{f_{22}, f_8, f_6, f_4\}$. Hence, $\phi(C) = {}^{100}\!/\!_{x+x}, \delta_C(A) = 10$, and $\delta_C(B) = |-10| = 10$. Then, $\alpha(B) = 10$ and $\psi_C = 20$. Since $(\alpha(B) < f_{40}\, \&\alpha(B) <\; f_{20})$, reducing ψ_C through 1-way LB is not feasible.

However, through SLB, we can transfer $M(f_{20},t) = (A,B)$ and transfer back $M(\{f_6, f_4\},t) = (B,A)$ represented in Fig. 4.1b. Thus, $(q(A) = q(B) = 50) \rightarrow \psi_C = 0$.

Though it finds an AP for migration, it leads to twice the handoff cost, and CP should study the cost before executing SLB. The message overhead demands APs to periodically update CP

accounting $O(|P|)$. Through SLB, no underloaded APs become overloaded, which is validated through Theorem 4.6.

Theorem 4.3. *In SLB, $\nexists p \in P$ where before $SLB, S_C(p) == Underloaded$ & after $SLB, S_C(p) == Overloaded$.*

Proof. Swapping is invoked post 1-way LB. Through Lemma 4.4, the above statement is proved for 1-way LB. In SLB, at a minimum, for a flow f from $p : S_C(p)\, is\, Underloaded$, we return atleast $\tau(f,p)$ from underloaded nodes.

$$\tau(f,p) = q(p) + \lambda(f,p) - \zeta_p\phi_C \tag{4.6}$$

Hence, the current load of p at instant $t+1$ post 2-way migrations would be,

$$q_{t+1}(p) \leq q_t(p) + \lambda(f,p) - \tau(f,p) \tag{4.7}$$

Substituting $\tau(f,p)$ from eq. 3,9,

$$q_{t+1}(p) \leq q_t(p) + \lambda(f,p) - q(p) - \lambda(f,p) + \zeta_p\phi_C \rightarrow q_{t+1}(p) \leq \zeta_p\phi_C \tag{4.8}$$

Thus, $q(p) \leq \zeta_p\phi_C$ & by *eq.*(3) proves post LB, $S_C(p)! = Overloaded$. □

Algorithm 6: Swap based LB

```
function SLB $(f, u, A, q, \lambda, t, C, P)$
$\rho(f,u) = \zeta_u\phi_C - q(u) + \lambda(f,u)$ $n \leftarrow \{R(f) - u\}$;
while $(p : (n : S_c(n) == Underloaded))$ do
    $\tau(f,p) = \alpha(p) - \lambda(f,p)$;
    if $\tau(f,p) > 0 \,\&\, \tau(f,p) < (\zeta_v - q(v))$ then
        Compute $K$ through $0 - 1$ $Knapsack\ DP$ such that
          $\tau(f,p) \leq \mu(f,v) \leq \rho(f,u) \leq \lambda(f,u)$ where $\mu(f,p) = \sum_k \lambda(k,p) : A(k) = p$
$l \leftarrow \arg\min_p \mu(f,p)$;
if $l \neq \emptyset \,\&\, \mu(l,f) < \lambda(f,u)$ then
    Migrate $M(f,t) = (u,l)$;
    Migrate $M(K,t) = (l,u)$;
    Update $q(l), A(f), q(u), A(K)$;
else
    LB is not possible through $SLB$
```

By accepting $f : A(f) = u$ from $S_C(u) = Overloaded$, if $\lambda(f,v) - \alpha(v) > 0$, we denote the exceeded threshold offset by $\tau(f,v)$, where $\tau(f,v) = \lambda(f,v) - \alpha(v) : S_C(v)\, is\, Underloaded.$ At a minimum, $\sum_k \lambda(k,v) : k \in K \geq \tau(f,v) : A(k) = v$ is migrated back through $M(K,t) = (v,u)$.

Table 4.1: Traffic Flow selection

Flow	Weight	1	2	3	4	5	6	7	8	9	10
f_8	8	0	0	0	0	0	0	0	8	8	8
$f_4^{\checkmark}$	4	0	0	0	4	4	4	4	8	8	8
$f_6^{\checkmark}$	6	0	0	4	4	4	6	6	8	8	10

In strict case, if $q(u) - \lambda(f,u) \leq \zeta_u \phi_C$, the maximum load which can be return transferred would be $(\zeta_u \phi_C - q(u) + \lambda(f,u))$ denoted by $\rho(f,u)$. A loose bound would be returning a load lesser than the accepted flow ($\sum_k \lambda(k,u) : k \in K < \lambda(f,u)$). A stricter constraint minimizes ϕ_C than the latter.

Hence, for v to accept the traffic flow f, the total load of traffic flows K to be transferred back from v lies in,

$$\tau(f,v) \; \leq \; \sum_k \lambda(k,u) : k \in K \; \leq \rho(f,u,v) \leq \lambda(f,u) \tag{4.9}$$

In Algorithm 6, CP selects underloaded AP (l) with the least return transfer to minimize the offloading cost.

$$l \leftarrow \arg\min_p \sum_k^K \lambda(k,p) : A(k) = p \,\&\, S_c(p) == \mathit{Underloaded} \tag{4.10}$$

Computing K through Dynamic Programming (DP) is detailed in Algorithm 7.

4.4.1 Dynamic Programming Solution

The selection of traffic flows to be migrated in the reverse transfer can be reduced to the *0-1 Knapsack problem* where the value and weight of the picked item (traffic flow) are equivalents. Hence, the set K is computed through the *Dynamic Programming* technique. The DP matrix for Fig. 4.2a is given below in Table 4.1.

Here, in this example, f_4 and f_6 are chosen through the DP matrix. DP solution is explained in Algorithm 7.

We've defined ϵ to represent the margin above and below the load threshold. Access points with loads falling within this lower and upper bound range are considered to be load balanced. It's worth noting that there are no established citations in the literature that explicitly define the ϵ value. The choice of ϵ in load balancing algorithms can vary based on specific network or system requirements, objectives, and characteristics.

Algorithm 7: DP Matrix and exchange flows

```
fn DP $(\tau(f,v), l \leftarrow \lambda(f,v), p : (p : S_c(p) == Underloaded)\ o : (o : S_c(o) == Overloaded))$
$D[\,][\,] \leftarrow \phi$, $link[\,] \leftarrow \phi$, $i \leftarrow 0$;
for $f : f \in F\&(A(f) = p\&f \in R(o))$ do
    $i+ = 1$, $link[i] = f$ ;
    for $(j = 0;\ j \leq l; j+ = 1)$ do
        if $l \leq j$ then
            $D[i][j] = max(D[i-1][j-l] + l, D[i-1][j])$
        else
            $D[i][j] = D[i-1][j-1]$
$tl \leftarrow \arg\min_{q=\tau(f,v)}^{l}(D[i][q] \geq \tau(f,v) : D[i][q])$;
if $\exists tl\,\&\,D[i][tl] \geq \tau(f,v)$ then
    $w \leftarrow D[i][tl]$
    $K \leftarrow \phi$
    for $k = i; k > 0\,\&w > 0; k--$ do
        if $D[k][w]! = D[k-1][w]$ then
            $K \leftarrow K \cup f_{k-1}$
            $w \leftarrow w - \lambda(k-1, v)$
    return $K$;
```

Factors like network topology, workload distribution, and primary performance goals (such as minimizing response time and maximizing throughput) should be considered. In general, an acceptable ϵ value might range from 5-7%. However, when facing more stringent requirements, there's a tradeoff to consider between performance, resource utilization, and system demands. Therefore, observing the network environment is essential for determining the acceptable range.

The time complexity of Algorithm 5 can be summarized as follows. The main loop iterates through overloaded access points and has a time complexity of $O(|P|)$, where $|P|$ is the total number of access points. Arithmetic calculations, comparisons, and SINR computations within the loop are typically $O(1)$ constant time operations. In Algorithm 6, Computing $\rho(f,u)$ and initializing variables takes constant time, $O(1)$. The main loop iterates through underloaded access points and has a time complexity of $O(|P|)$, where $|P|$ is the total number of access points. The calculation of $\tau(f,p)$ for each underloaded access point is a constant time operation $O(1)$ within the loop. The time complexity of solving the 0-1 knapsack problem using dynamic programming in Algorithm 7 is $O(nW)$, where n is the number of traffic flows, and W is the maximum load capacity of the knapsack. This time complexity arises from filling a 2D table of size $(n+1) \times (W+1)$.

4.5 Results and Analysis

The simulation is carried out in the Mininet 2.3.0 setup [115]. Network slices are built through Flowvisor. The nodes and switches from the data plane are connected to the created slices. The slices are supervised by Beacon or POX controllers. In Flowvisor, the dataset contributed by the Irish mobile operator [116] - 4G dataset and [117] - 5G dataset is evaluated in Java language.

An agent program resides in every AP. This agent measures the details concerned to APs and devices associated with it. The metrics captured are the number of devices connected, their geo-location information, throughput consumption, current RSSI, and CQI with the connected AP. It sends the maximum serviceable capacity, geo-position of AP. The agent periodically reports the above metrics to the NS controller like FlowVisor or control plane. The load balancing algorithm operating in Flowvisor (Java) or Beacon Controller (Java) computes and makes the load balancing decisions. The controllers send the devices to be migrated to the agent program of monitored APs in their coverage area. The agent program performs the handover.

The sum of the uplink and downlink bandwidth impact of devices on the serving cell is considered as the load on the cell. When the device migrates, based on CQI, modulation, and coding rate, the channel bandwidth consumption on the cell would vary for the same usable network throughput.

In 5G, one New Radio Resource Block (RB) contains 12 sub-carriers in a frequency domain. The New Radio works with 100 MHz channel bandwidth for lower bands $< 6GHz$ and 400 MHz channel bandwidth for higher bands in mmWave ranges.

Like the frequency domain parameter $\triangle f$, 5G NR has a parameter for the time domain. For instance, if $= 0$, $\triangle f = 15$ KHz: One resource block is 180KHz (15 x 12 sub-carriers) in the frequency domain and 1ms in the time domain. Similarly, If $= 1$, $\triangle f = 30$ KHz: One resource block is 360KHz (30 x 12 sub-carriers) in the frequency domain and 0.5ms in the time domain.

Mapping the Channel Bandwidth (in MHz) to a number of Resource Blocks is shown in Table 4.2. MCS defines bits transmitted per resource block, as shown in Table 4.3.

The attributes in the dataset are:

- Timestamp of the sample

- Longitude and Latitude

Table 4.2: Mapping Channel Bandwidth and Resource Blocks

μ	1	2	3
$\triangle f = 2\mu \times 15KHz$	30KHz	60KHz	120KHz
Min RB	24	24	24
Max RB	275	275	275
Min Channel Bandwidth(MHz)	8.64	17.28	34.56
Max Channel Bandwidth(MHz)	99	198	396

Table 4.3: Modulation Coding Scheme

CQI	Modulation	Code Rate	Bits per RB
6	16 QAM	0.6016	2.4064
7	64 QAM	0.4551	2.7306
8	64 QAM	0.5537	3.3222
9	64 QAM	0.6504	3.9022

- Cellular Operator Name
- Serving Cell for Mobile Device
- Network Mode
- Downlink Bit Rate (Rate measured at the device) (kbps)
- Uplink Bit Rate (Rate measured at the device) (kbps)
- Ping Statistics (average, minimum, maximum, standard deviation and loss)
- Signal strength (signal quality) is measured across all resource elements, including interference from all sources (dB).
- SNR: value for signal-to-noise ratio (dB).
- RSSI represents a received power, including a serving cell and interference and noise from other sources.
- RSRQ Represents a ratio between RSRP and Received Signal Strength Indicator (RSSI).
- CQI: value for CQI of a mobile device. It indicates the data rate that could be transmitted over a channel as the function of SINR and UE's receiver characteristics.
- RSRP represents an average power over cell-specific reference symbols. Used for measuring cell signal strength/coverage (dBm).

Table 4.4: Simulation Settings - Load Balancing

Parameter	***Value/Description***
Number of Traces	135
Average Trace Duration	15 minutes per trace
β value	0.3
Serving Cell Type	Homogeneous serving cells with heterogeneous loads
Modulation Types	QAM (16, 64, and 256)
Handline missing metrics	Simple Regressor library in Java
Mobility Patterns	Cumulative patterns for bus, car, static, pedestrian

The tabular view of the simulation parameters is present in Table 4.4.

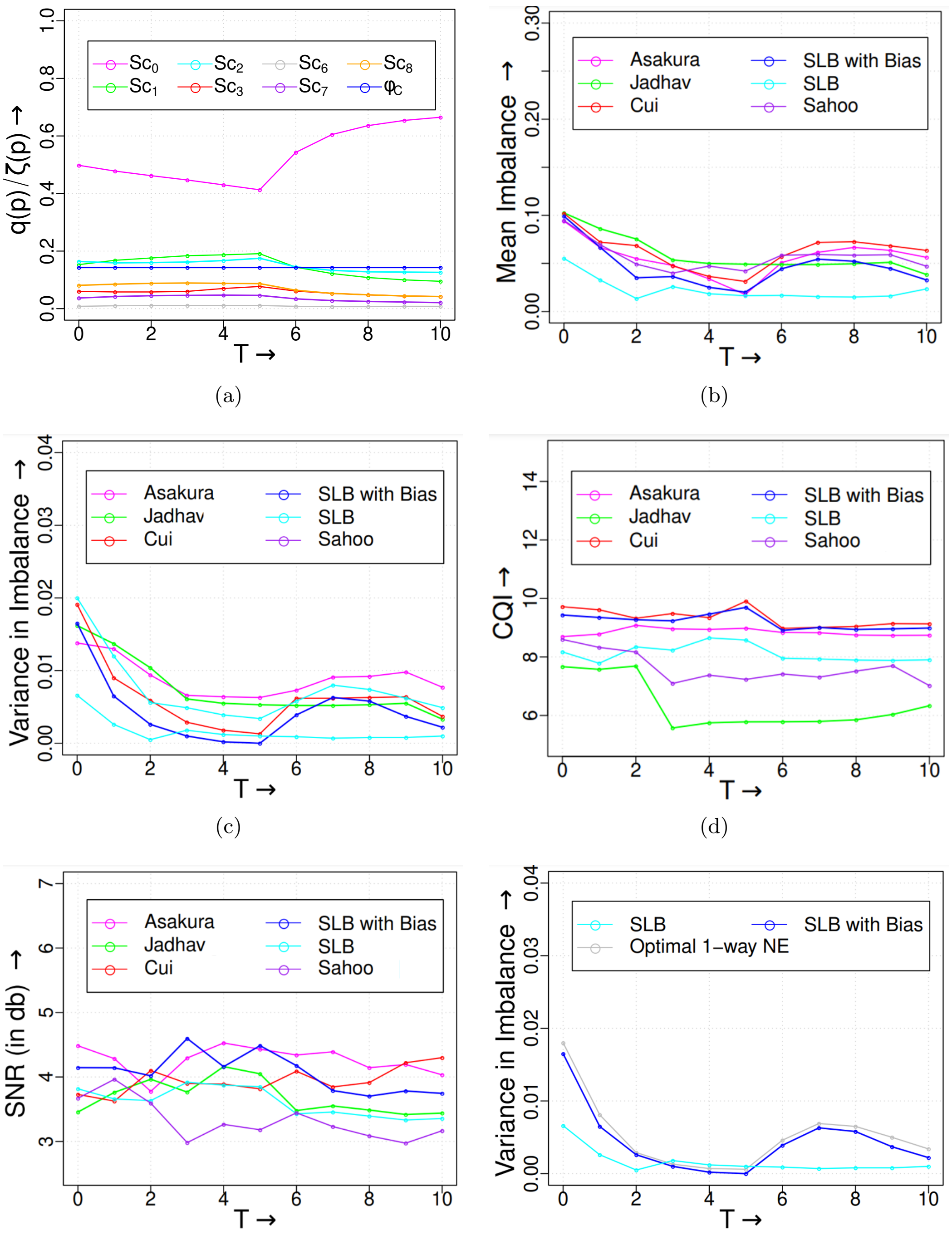
The dataset consists of 135 traces, with an average of 15 minutes per trace, and β is initialized to 0.3. The x-axis marks the timeslots, where every time slot remarks the 15-minute trace. The proposed approach is applicable to heterogeneous serving cells and loads. For evaluation, homogenous serving cells with heterogeneous loads are considered with capacities ζ_{sc} and $\forall f, f \in F : R(f) := P$ as per the dataset. Hence, the threshold would be the same for each of the serving cells. We applied Quadrature Amplitude Modulation (QAM) - (16, 64, and 256 QAM) and Channel Quality Index (CQI) in the dataset for determining the load at destination AP. Since the channel and context metrics of devices to unconnected APs are not present in the dataset, these are populated through the Simple Regressor library in Java.

In our simulation, we observe the arrival of real-time traffic from various devices. Initially, in Fig. 4.2a, we can see that a significant portion of the traffic load is handled by Sc_0 without the application of load balancing. However, when load balancing techniques are introduced, we can clearly see the impact in Fig. 4.2b and 4.2c, where both the mean and variance of the load imbalance factor are significantly reduced.

Fig. 4.2 evaluates the cumulative (bus, car, static, and pedestrian) mobility patterns. Fig. 4.2a plot the $q(p)/\zeta(p)$ under no traffic distribution. The serving cells that lie above and below the threshold are overloaded and underloaded, respectively. We could observe that the resource utilization is not uniform, and hence, there is a need for traffic re-distribution in this dataset.

The proposed swap-based algorithm with and without biasing is compared against the candidate algorithms like Asakura [79], Jadhav [80], Farzi [68], Cui [69], and Sahoo [70].

While not cellular algorithms, the works of Cui et al. [69] and Sahoo et al. [70] indirectly relate to our research. Cui's focus on best response time for devices has a direct correlation to SINR



(a) (b)

(c) (d)

(e) (f)

FIGURE 4.2: Evaluation on Cumulative Mobile Pattern

and CQI, which are key attributes in our biasing-based algorithm. Similarly, Sahoo et al. applied TOPSIS techniques for multi-attribute optimization, aligning with our QoS-centric approach.

For more direct comparisons, we evaluated our approach against specific cellular algorithms. Garcia et al.'s work [75] presents a cellular algorithm for load balancing in access points, specifically focusing on optimal 1-way Nash Equilibrium. In contrast, our approach compared against Jadhav's round-robin load balancing with random strategies, which incurs less overhead.

Furthermore, we benchmarked our approach against Asakura et al. [79], a prominent primary load balancing cellular algorithm featuring backup flows for each primary flow. Additionally, we assessed Kawada et al.'s algorithm [76], which rebalances only the most overloaded access point, as mentioned in the literature. Our proposed approach, SLB, demonstrated superior performance in comparison.

In this study, we conducted thorough comparisons with the aforementioned state-of-the-art algorithms, as illustrated in Fig 4.2. The selection of these algorithms was based on their relevance to our research objectives and problem domain.

Fig. 4.2b and 4.2c compare the mean imbalance and variance. SLB without biasing delivers the least imbalance among all other contestants. However, SLB without biasing suffer from poor SNR and CQI values, as shown in Fig. 4.2d and 4.2e. Cell association through signal strength and latency [79],[69] provides a better SNR and CQI but a higher load imbalance since it is swayed just by signal quality. Our multi-dimensional approach, SLB with biasing, offers comparable SNR and CQI for the migrated devices compared to [79] and [69]. It also achieves a lesser imbalance among the serving cells than candidate algorithms. Thus, SLB with biasing offers a good trade-off between load imbalance and signal quality and exhibits better results in each of the parameters shown in Fig. 4.2b to Fig. 4.2e.

To outline the overall effectiveness, we first measure the load imbalance reduction percentage of SLB with biasing against other candidates in every time slot. Then, we determine the minimum percent ratio in which SLB outperforms state-of-the-art approaches in every round. Finally, we compute the moving average of minimum percentage gain across the ten timeslots to state the overall improvement. Overall, SLB with biasing reduces the imbalance by a factor of 7.14% compared to the optimal uni-transfer algorithm. Against other state-of-the-art algorithms, it outperforms them by a factor of 22.24%.

4.6 Chapter Summary

The chapter on swap-based load balancing presented a comprehensive approach to managing network traffic distribution and optimizing performance. The key points underscore the significance of the research work.

- One-way and Two-way Traffic Distribution: The chapter begins by detailing the standard one-way traffic distribution method, which considers factors like signal strength and access point load. This approach provides a foundation for understanding traditional load balancing methods.
- Proposed Two-way Swap Load Balancing: The chapter introduces a novel approach, the two-way extreme load balancing technique, designed to minimize load imbalances and improve critical metrics such as Channel Quality Index (CQI) and Signal-to-Noise Ratio (SNR) for users. This innovation sets the stage for improved network performance.
- Validation with Real Data and Comparison with Other Algorithms: The work is validated using a dataset from an Irish mobile operator, adding a practical dimension to the research. Real-world data evaluation strengthens the credibility of the proposed load balancing techniques. The research rigorously compares the proposed swap-based algorithm, with and without biasing, against other candidate algorithms proposed by various researchers. The results highlight the effectiveness of the approach, especially when biasing is applied.
- 0-1 Knapsack Algorithm and Biasing Techniques: The implementation of swap-based load balancing employs the 0-1 Knapsack algorithm to determine which devices should be exchanged. The chapter explores biasing techniques based on signal strength and associated access points. These techniques are vital in controlling how load balancing is executed, leading to better load distribution and performance optimization.
- Tighter and Looser Variants: The tighter variant maintains equilibrium with minimal exchange, while the looser variant allows for more dynamic load balancing, adapting to the network's needs.
- Control-Plane-Driven Methodology: The chapter discusses a control-plane-driven load balancing methodology applicable to Software-Defined Networking (SDN)-enabled architectures in WLANs, Radio communications, and SD-WAN deployments. This adds relevance to modern network infrastructure.
- Signal Quality Improvement and Reducing Load Imbalance: Swap-based load balancing decreases the imbalance among access points (APs) and enhances signal quality for connected devices. These outcomes directly impact user experience and network efficiency.

The chapter emphasizes that SLB without biasing achieves the least load imbalance among all other contestants. This underscores the practical significance and superiority of the proposed load balancing technique.

The chapter provides a detailed exploration of swap-based load balancing, supported by empirical data, algorithmic insights, and a focus on network performance improvement. It contributes to the field by offering innovative solutions to address load balancing challenges in modern network environments. We published this work [4] in IEEE Wireless Communications Letters Journal in November 2021.

Chapter 5

QoS Driven Task Offloading in RAN Slicing

5.1 Introduction

The storage and computation power of mobile devices are not able to scale up to run resources-hungry applications. A possible alternative could be to offload the requests to remote resource-rich VM instances, possibly in the cloud or container-based technology at the edge. It could involve offloading cost, network delay cost and scheduling delay. Hence, the system design should consider when to offload and when not to in the paradigm of Multi-access Edge Computing (MEC).

Requests can be offloaded to an edge server only when they can cater to the specific application processing or Virtual Network Function (VNF). Such offloaded requests to the edge server, need to be scheduled at the application servers or VNF and the resources need to be subsequently provisioned. Hence, there is a need for joint task offloading with resource allocation in the edge servers.

As we know, when the applications in the mobile devices become resource-intensive, an option is to offload total or part of the application workflow to a resource-rich cloud infrastructure. The pitfall would be a long latency associated with it. A cloudlet solution was proposed to bring down the latency. The cloudlets can act as a valid solution, particularly for small or medium resources. However, due to its inherent architecture and placement, it would be difficult to satisfy QoS in difficult circumstances [85].

The improvised version of cloudlet constitutes next-gen technology: Mobile Edge Cloud Computing, where cloud computing technology is at the edge of the Radio Access Networks. This version is an extension of the Cloud Radio Access Network (C-RAN). C-RAN is a cloud computing-based, centralized, clean and collaborative radio access network. C-RAN comprises of three units: i) Remote Radio Heads (RRH), ii) Base Band Unit (BBU) pool and iii) High-bandwidth, high speed, low latency fiber transport or fronthaul link. The fronthaul link connects RRH to the BBU cloud pool. In traditional architecture, radio and baseband functions are located in physical BSs. Whereas, in C-RAN, the functions are virtual and deployed as cloud services.

In summary, the chapter discusses requests for task offloading from mobile devices, the tasks themselves, and the computational resources available at the edge, particularly within the context of Mobile Edge Computing (MEC) and Network Slicing (NS).

- Requests: These are requests related to task offloading in the context of mobile edge computing and network slicing. Specifically, they refer to requests to offload tasks from devices to edge servers for processing. These requests are made to improve the performance and resource utilization of resource-constrained devices.
- Tasks: The tasks referred to are computing tasks or workloads generated by mobile devices that can be offloaded to edge servers. These tasks are typically application processing or Virtual Network Functions (VNFs) that require computational resources for execution.
- Resources: The resources in question primarily pertain to the edge servers located within the Radio Access Network (RAN). These edge servers are equipped with computational capacity and are responsible for handling the offloaded tasks efficiently. The architecture also involves other network components like Remote Radio Heads (RRH), Base Band Unit (BBU) pools, and high-speed fiber transport links.

In the context of a Mobile Edge Computing (MEC) system with edge servers installed at Radio Access Network (RAN) slices, the problem statement is to optimize the task offloading process from mobile devices to these edge servers based on Quality of Service (QoS) attributes. Specifically, this chapter aims to categorize incoming tasks into different priority levels and efficiently process them at edge servers. The task placement needs to consider attributes such as QoS Class Identifier (QCI), Allocation and Retention Priority (ARP), Access Class Bearing (ACB), Differentiated Services Code Point (DSCP), soft deadlines, and computing cycles.

In this chapter,

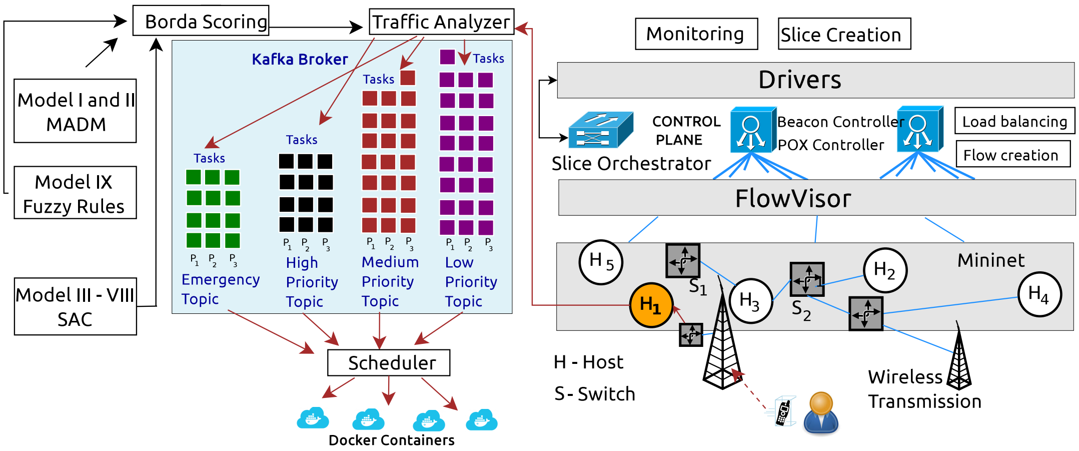


FIGURE 5.1: Network Slice setup for Task Offloading

- We present a novel task categorizer for offloading that considers the QoS attributes. We compare the proposed ensemble method with other multiple and single attribute categorization approaches.

- We present Kafka topic based priority queuing and resource provisioning of offloaded tasks to the edge servers hosted in container-based technology.

In this work, we realise slices through flowvisor and controllers from SDN technologies depicted in Fig. 5.1. The network slices are created and configured through flowvisor.

5.2 Task Offloading System Model

Edge servers are installed at RAN slice. Tasks from mobile devices traverse through the radio device to the co-located edge host. The traffic analyzer inspects the task and routes it to the appropriate Kafka topic. Tasks are executed as containers.

5.2.1 Computation Task Model

We formulate the offloading scheme for mobile User Equipments (UEs) as a triple (U, T, S). Here U is a finite set of UEs, T is a finite set of tasks. $t_{i,u} \in T$ is a task belongs to user $u \in U$ and is identified by i. Tasks are executed by the processing units $s \in S$. In this work, we consider discrete tasks modelled by a set of attributes,

$$t_{i,u} = \{Q_i, \zeta_i, \alpha_i, \hat{d}_i, \theta_i, \mu_i\} \tag{5.1}$$

Q_i represents QCI ranging from 1-254 are used for prioritization in scheduling and queuing of admitted values.

ζ_i denotes the Allocation and Retention Priority (ARP) which governs through the ARP priority (1-15). It is a constituent in subscribed QoS profile for the default bearer.

α_i depicts the Access Class Bearing (ACB) feature used for terminal class prioritization. When the network is massively overloaded, the regular access classes are either barred or set a particular blocking probability.

$\hat{d}_i$ denotes the Differentiated Services Code Point (DSCP) attribute of the task $t_{i,u}$ belonging to user u.

θ_i, μ_i represent the soft deadline (latency requirements) and computing cycles required for processing by $p \in P$.

In this model, we inspect based on parameters such as QCI, ARP, and ACB. QCI fills in for other parameters, such as Guaranteed Bit Rate (GBR) and Maximum Bit Rate (MBR), and these are not autonomously examined. In this chapter, we follow the notation t_i and $t_{i,u}$ interchangeably.

5.2.2 QoS Task Scheduling Model

When task offloading requests arrive, the task should be characterized by attributes in equation 5.1. We construct a priority scheduling model based on QoS parameters as follows,

- In this work, we structure task processing through Kafka topics $\tau = \{\tau_E, \tau_H, \tau_M, \tau_L\}$ which stands for Emergency, High, Medium, and Low priority topics, respectively. This can be generalized or extended to $\tau = \{\tau_1, \tau_2, \tau_3, ..\tau_p\}$ with τ_1 as the higher priority and τ_p being the lower priority.
- The number of pending tasks to be executed in each topic k is given by σ_k.

  $$\sigma_k = \text{Latest pushed offset} - \text{Current offset read}$$

  The offset is a simple integer number that is used by Kafka to maintain the current position of a consumer.

- ϵ_i is the Earliest Finish Time (EF) of task t_i based on the current load $\{\sigma_E, \sigma_H, \sigma_M, \sigma_L\}$ in the Kafka topics. It is computed based on: a) the unconsumed number of tasks in each topic, b) the topic in which the task t_i is placed, and c) the frequency with which messages are consumed from the topics. Further, ϵ_i is defined later in the discussion through the equation 6.6. The actual completion time would also depend on the future arrivals of higher priority tasks that are going to be scheduled before this particular task.

The Earliest Finish Time of a task is the expected completion time of that task, taking into account the current set of tasks that have not yet been processed, the system's scheduling strategy, and the time required for task execution. EF represents the point in time at which the task is anticipated to finish. Importantly, the calculation of EF focuses solely on the existing tasks that have not been completed and does not consider potential tasks with higher priority that might arise in the future. This approach ensures that the EF is based on the current workload, without being influenced by potential future tasks that could alter the task's completion time.

5.3 Solution Framework

In this section, we discuss the proposed QoS driven prioritized offloading mechanism. The offloading algorithm consists of three parts: (a) Offloaded Task placement (b) Task Categorizer (c) QoS driven Prioritized Task scheduling.

5.3.1 Offloaded Task Placement

In task classification, the incoming task $t_{i,u} = \{Q_i, \zeta_i, \alpha_i, \hat{d}_i, \theta_i, \mu_i\}$ is categorized to place in one of the topics of $\tau = \{\tau_E, \tau_H, \tau_M, \tau_L\}$.

Algorithm 8 examines the incoming offloaded tasks T and rejects the task whose deadline $(\theta_i < c : t_i \in T)$ have already expired where c denotes the current time. Then, the task classifier analyzes and computes the priority of the task t_i, and it returns the priority of traffic class $p \in \{E, H, M, L\}$. If the task is classified as emergency, it is instantly placed in the emergency topic τ_E. For other traffic classes p, if the earliest finish time is less than the deadline $\hat{d}_i$, the tasks are placed in the topic τ_p where $p \in \{H, M, L\}$,. The earliest finish time for task $t_{i,u}$ is denoted by $F_{\varkappa(p,t_i)}$ where $p \in \{H, M, L\}$.

Algorithm 8: Offloaded Task Placement in Kafka Topic

```
function taskPlacement $(T, \tau)$
while $(t_{i,u} \in T\, from\, u \in U)$ do
    if $\theta_i < c$ then
        reject $t_{i,u}$;
    $p \rightarrow task\_categorizer(t_{i,u}, \tau)$
    if $p \in E$ then
        place $t_i$ in $\tau_E$
    $\epsilon_i \rightarrow F_{\varkappa(p,t_i)}$
    if $\theta_i < \epsilon_i$ then
        reject $t_{i,u}$;
    else
        place $t_{i,u}$ in $\tau_p : p \in \{H, M, L\}$
```

5.3.2 Task Categorizer

The proposed traffic categorizer is an ensemble method achieved over two stages. First, we apply mathematic models which evaluate the QoS attributes of each task and choose the appropriate priority. Second, we integrate it and implement a weighted voting scheme for deducing the traffic priority.

Models

Model I and II are based on the multiple attributes decision making (MADM) [118]. The attributes are $J = \{Q, \zeta, \alpha, \hat{d}, \theta, \mu\}$ which defines the tasks. Let a_{ij} represent the mapped value ξ_j for task t_i with respect to j^{th} attribute.

$$a_{ij} = \xi_j(t_i) \tag{5.2}$$

$\hat{a}_{ij}$ is the normalized matrix using the Max-Min function,

$$\hat{a}_{ij} = \frac{a_{ij} - min_i(a_{ij})}{max_i(a_{ij}) - min_i(a_{ij})}$$

for upward attributes. Max and Min functions are reversed for downward attributes. Model I and II use Simple Additive Weighting (SAW) and Multiplicative Additive Weighting (MEW) represented in equation 5.3 and 5.4 respectively.

$$R_{SAW} = \sum_{j=1}^{J} w_j \hat{a_{ij}} \tag{5.3}$$

$$R_{MEW} = \sum_{j=1}^{J} \hat{a}_{ij}^{w_j} \tag{5.4}$$

TABLE 5.1: Standard QoS Class Identifier (QCI) values and priority level

QCI Value	***Priority Level***	***Services***
1	2	GBR Conversational voice
2	4	GBR Conversational video
3	3	GBR Relative Gaming
4	5	GBR Non-Conversational video
65	0.7	Mission critical user plane Push to talk
66	0.7	Non Mission critical user Push to talk
5	1	IMS Signalling
6	6	Non-GBR Video (Buffered Streaming)
7	7	Non-GBR Voice, Interactive Gaming
8	8	Non-GBR TCP based apps
9	9	Default
69	0.5	Mission Critical Delay Sensitive signalling
70	5.5	Mission Critical Data

Every attribute is provided with homogeneous weights. The range of R_{SAW} co-efficient values are divided into four equal parts and are classified into {E,H,M,L} respectively The top portion of $R_{SAW} \rightarrow E$ and lowest $R_{SAW} \rightarrow L$.

The mapping function ξ_j is prescribed for the transformation of attribute j based on its relative priority. For example, every standard QCI (Q) values can be coded into a relative priority where 0.5 is the highest priority and 9 implying the lowest priority level. Based on the priority level range, the mapping function classifies it into $\{E, H, M, L\}$ or $\{1, 2, 3, 4\}$ values, and are stored in decision matrix a_{ij}. In Table 5.1, we tabulate the standard QCI values in LTE to its relative priority level [112].

Similarly, ACB (α) has values AC12 for emergency, AC14 for security service, AC10 for consumer emergency and AC0-8 are for regular class. The mapping function (ξ_j) is applied for the actual values for attributes j of the task t_i to classify into one of $\{E, H, M, L\}$.

Models *III-VIII* are designed as Single Attribute Categorization Problem (SACP) for $\forall j \in J$ referred in equation 5.1. Models *III-VIII* are for $\{Q, \zeta, \alpha, \hat{d}, \theta, \mu\}$ based on the single attribute only. Here again, we apply mapping function ($\xi_j : j \in J$) for relative ranking based on the levels of ξ_j we classify into one of $\{E, H, M, L\}$.

For example, in model IV, QCI(Q_i) of task t_i with priority (0-2) are classified as high H, (3-5) as medium M , and (6-9) as low L. The mission critical traffic and signalling category of (Q_i) are treated as emergency E.

Model(IX) utilizes fuzzy rule sets. A fuzzy set is a membership function assigning each object to the priority class. We develop a module of *fuzzy rule base* which is a collection of IF - THEN rules. For example, in access class bearing attribute, **if** $\alpha_i \geq 10$, **then** p ← E.

Voting through Borda Scoring

The models are the voters and the candidates are the topics to choose from. In this work, the collected ballots from each of the models classify the task $t_i \in T$ into suitable topics with a ranking order. For example, for task $t_{i,u}$, M_I can provide the ranking as H→E→M→L, indicating it prefers to place the task $t_{i,u}$ in topic $\tau_H > \tau_E > \tau_M > \tau_L$. In this approach, we apply *Borda Score*(BS). Borda score of τ_H for task t_i against models (M_I-M_{IX}) can be applied as follows:

When the number of topics ranging from τ_1 to τ_p,

$$BS(\tau_k) = \begin{cases} (p-1) \times \#\{l|l\, ranks\, \tau_k\, first\} \\ +(p-2) \times \#\{l|l\, ranks\, \tau_k\, second\} \\ +.. \\ +1 \times \#\{l|l\, ranks\, \tau_k\, second\, to\, last\} \\ +0 \times \#\{l|l\, ranks\, \tau_k\, last\} \end{cases} \tag{5.5}$$

In the current implementation, we use the following topics $\{\tau_E, \tau_H, \tau_M, \tau_L\}$. Also, the Borda Scoring is enhanced by providing weights (wt) to the models. The enhanced Borda Score of τ_H is,

$$BS(\tau_H) = \begin{cases} 3 \times \#\{\sum_r^{R_1} wt_r|r\, ranks\, \tau_H\, first\} \\ +2 \times \#\{\sum_r^{R_2} wt_r|r\, ranks\, \tau_H\, second\} \\ +1 \times \#\{\sum_r^{R_3} wt_r|r\, ranks\, \tau_H\, second\, to\, last\} \\ +0 \times \#\{\sum_r^{R_4} wt_r|r\, ranks\, \tau_H\, last\} \end{cases} \tag{5.6}$$

where $|R_1| + |R_2| + |R_3| + |R_4|$ = number of models. R_1 and R_2 are set of models which rank τ_H first and second respectively. wt_r indicates the weight given for the model for voting. Hence, we sum the weights of the models which provides the same rank and multiply with the points that are given to each model in reverse proportion to their ranking. Similarly the above formula can be applied for computing BS(τ_E), BS(τ_M) and BS(τ_L) for evaluating task t_i. The topic with the highest *Borda Score* is selected for placing the task t_i represented by equation 5.7.

$$\tau_k = \arg\max_p BS(\tau_p) \Leftrightarrow BS(\tau_k) = \max_{p \in \{\tau\}} BS(\tau_p) \tag{5.7}$$

The time complexity of Borda scoring is $O(m * n)$ for calculating the score of a single task, where n is the number of topics to choose from, and m is the number of models. The topic selection for a task is performed using a max function in $O(n)$. If scores are calculated for all tasks ($|T|$), the overall time complexity becomes $O(m * n * |T| + n * |T|) \rightarrow O(m * n * |T|)$. Each model involved in ensemble categorization has different time complexities. Single attribute categorizers, like Access Class Bearing, categorize tasks in $O(1)$ time. Others, like Quality Class Identifier (QCI), Allocation and Retention Priority (ARP), and Differentiated Services Code Point (DSCP), map tasks into ranked topics in $O(1)$ time. Multiple attribute decision-making models like Simple Additive Weighting (SAW) depend on multiple attributes ($|J|$), leading to a complexity of $O(|J|)$. When ensemble categorization has single and multiple attribute decision categorizers, the overall time complexity becomes $O(m * n * |J| * |T|)$.

Borda scoring can be used in ensemble methods, although it is less common than other aggregation techniques like majority voting or averaging. Here are some key reasons why Borda scoring should be considered in ensemble methods:

- Preference Aggregation: Borda scoring is primarily used for preference aggregation when there is a need to rank or prioritize multiple options or choices. In certain ensemble scenarios, such as recommender systems, where the goal is to rank items based on user preferences, Borda scoring can be a suitable choice.

- Ranking-Based Ensembles: If individual models in an ensemble provide ranked predictions or preferences, Borda scoring can be employed to aggregate these rankings effectively. For example, in a collaborative filtering recommendation system, where each model ranks items for a user, Borda scoring can help determine the final ranked list of recommendations.

- Combining Ranking Models: Ensemble methods often involve combining multiple models with different strengths and weaknesses. If some of these models are designed to produce rankings or preferences, Borda scoring can be used to leverage their outputs alongside other models that produce class labels or continuous predictions.

- Customized Aggregation: Borda scoring allows for customized weighting of individual models based on their ranks. This can be valuable when certain models in the ensemble are known to perform better or have more credibility in specific situations.

- Diverse Ensemble Members: In cases where ensemble members provide diverse rankings or preferences, Borda scoring can help capture and combine these diverse viewpoints to make a final decision.

In many classification and regression tasks, simpler aggregation methods like majority voting or averaging are more commonly used because they directly address the objectives of predicting class labels or continuous values. However, when the objective is to rank or prioritize options, Borda scoring can be a relevant choice within an ensemble framework.

5.3.3 Proposed Prioritized Scheduling

Here the offloaded tasks present in the τ_E are consumed immediately. If there are no emergency tasks, the scheduler assigns probability of ρ_H, ρ_M, ρ_L for each of the topic where $\rho_H > \rho_M > \rho_L$ where $\rho_H + \rho_M + \rho_L$=1. As mentioned earlier, σ_H,σ_M,σ_L captures the number of unread tasks lying in the topics where σ_H+σ_M+σ_L would be the total number of unexecuted offloaded tasks. The probability of processing in the next time slot from topic H is

$$Pr(H) = \frac{\rho_H \sigma_H}{\rho_H \sigma_H + \rho_M \sigma_M + \rho_L \sigma_L} \tag{5.8}$$

Similarly, the above probabilities are applied for τ_M and τ_L respectively. The task when dispatched by scheduler processor instantiates a docker container.

The Earliest Finish time ($EF(t_i)$) for task t_i categorized into τ_H would be as follows. Let l be the number of scheduling operations such that $\rho_H * l = \sigma_H + 1$,

$$EF(t_i) = \sum_{v}^{\sigma_E} \mu_v + \sum_{x}^{\sigma_H} \mu_x + \sum_{y}^{l*\rho_M} \mu_y + \sum_{z}^{l*\rho_L} \mu_z \tag{5.9}$$

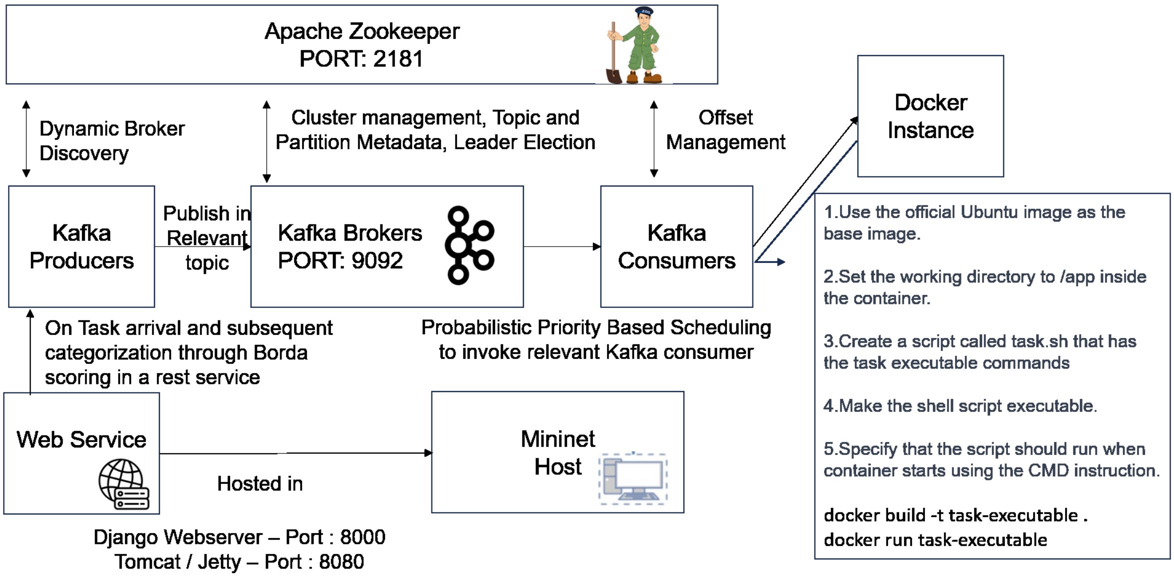
where $u \in \tau_E$, $x \in \tau_H$, $y \in \tau_H$, $z \in \tau_L$, $\sigma_H >> \sigma_M >> \sigma_L$.

To summarize, there are number of offloaded tasks to be executed, which need task prioritization and scheduling at edge servers. Following Quality of Service (QoS) considerations are considered:

Q_i (QCI): Prioritizes tasks using values from 1-254.

ζ_i (ARP): When network resources are scarce or in high demand, ARP is used to determine which services or tasks should receive preferential treatment in terms of resource allocation.

α_i (ACB): Prioritizes tasks based on terminal class, crucial during network congestion.


Apache Zookeeper
PORT: 2181
Dynamic Broker Discovery
Cluster management, Topic and Partition Metadata, Leader Election
Offset Management
Docker Instance
Kafka Producers
Publish in Relevant topic
Kafka Brokers PORT: 9092
Kafka Consumers
On Task arrival and subsequent categorization through Borda scoring in a rest service
Probabilistic Priority Based Scheduling to invoke relevant Kafka consumer
Web Service
Hosted in
Mininet Host
Django Webserver – Port : 8000
Tomcat / Jetty – Port : 8080

1.Use the official Ubuntu image as the base image.

2.Set the working directory to /app inside the container.

3.Create a script called task.sh that has the task executable commands

4.Make the shell script executable.

5.Specify that the script should run when container starts using the CMD instruction.

```
docker build -t task-executable .
docker run task-executable
```

FIGURE 5.2: Mininet Host integration with the Docker and Kafka messaging system

$\hat{d}_i$ (DSCP): Identifies tasks based on specific attributes.

θ_i, μ_i (Soft Deadline and Computing Cycles): Ensure tasks meet latency requirements and have adequate processing resources.

These attributes optimize task management and resource allocation, enhancing system performance.

When offloading tasks with different parameters, we have discussed techniques for categorizing and accuracy of such methods. Our focus during scheduling optimization has been metrics like minimizing waiting times and reducing queue lengths for different priority classes.

5.4 Simulation and Results

The network slicing setup is already well-explained in Chapters 3 and 4 simulation section.

The offloaded tasks are redirected from the end-user devices to the Mininet hosts, which are functioning in the slice. The slices are administered through the respective controllers. The Mininet host operates the web server, which receives the request for task execution and, based on the precedence factor of the request, places it in the appropriate Kafka topic. It is eventually picked up by one of the multiple processing units, which are instantiated as application server instances using the Docker ecosystem. The topics are realised through the Kafka messaging system.

Table 5.2: Accuracy of Task Categorization

Methodologies	***Emergency***	***High***	***Medium***	***Low***
ACB SAC	76.73%	NA	11.79%	91.22%
Proposed Ensemble method	**100%**	**100%**	**64.18%**	**99.37%**
Deadline (EDF) SAC	NA	16.05%	39.35%	75.8%
DSCP SAC	NA	63.45%	64.32%	100%
MADM Model I	**71.17%**	**68.72%**	**88.28%**	**58.18%**
MADM Model II	**100%**	**79.65%**	**100%**	**60.9%**
QCI SAC	30.95%	41.71%	65.59%	75%

Docker uses OS-level virtualization to create containers. Containers are isolated processing units with their software, libraries and configuration files. Container instances are brought up with required dependencies to process the request. Requests for task execution are read from the Kafka topics.

The traffic categorization through ensemble method is compared against the results of multiple attribute decision making, and Single Attribute Categorization (SAC) methods for different priority task classes. The accuracy of the categorizers are tabulated in Table 5.2. The generated test data consists of 10000 tasks of the four priority classes $\{E, M, H, L\}$.

We have depicted in Figure 5.2, how the Mininet host is integrated with Docker and Kafka messaging systems.

The web service is hosted on port 8080 using the Tomcat or Jetty HTTP web server with the Java programming language. We've also experimented with the Django web server, which operates with Python and uses port 8000 to handle requests. When a task arrives at these REST services, we trigger ensemble categorization and Borda scoring. Based on the Borda scoring result, the Kafka producer publishes the task to the relevant topic.

In the background, Apache ZooKeeper is running on port 2181 to assist the Kafka producer in dynamic broker discovery. The Kafka broker operates on port number 9092 and relies on ZooKeeper for cluster management, managing topic and partition metadata, and leader election. A relevant Kafka consumer retrieves tasks from different topics (Emergency, High, Medium, Low) based on the probabilistic priority scheduling outcome. The Kafka consumer also uses ZooKeeper for offset management.

To execute tasks, we create a Docker instance using a Docker run file. This run file uses the official Ubuntu image as the base image. Inside the container, we set up the working application.

An executable script for the task is created and marked as executable. We specify that the task executable should be executed as part of the CMD command, which is used to start the script when the Docker instance starts.

To create and store the Docker image, we use the *docker build* command, ship to *docker hub* and then run the image using the ”docker run” command.

For this experiment, the emergency class consists of emergency & security services of ACB, mission-critical sensitive signalling & data of QCI. High priority tasks include GBR conversation voice, IMS signalling, and mission-critical user (push to talk) services of QCI, Expediated Forwarding from DSCP, and top 10% of earliest deadline tasks. Medium priority tasks include GBR services except for the voice, assured forwarding, and 10% - 50% percentile of the deadline of tasks received. Low priority tasks are non-GBR and default services, default forwarding, and tasks with the bottom 50% percentile deadline. The SAC methods exhibit low accuracy. The ensemble method shows an accuracy of 97.695% across the priority classes.

On average, the proposed methodology has greater accuracy compared to both MADM Model I and II by 26% and 12.5%, respectively. For the experiment, during the enhanced Borda score voting, the weights for the models I - IX in the ensemble method are (1,1,2,0,1,1,2,0,2).

Some of the key observations are listed below:

- The Proposed Ensemble Method achieves the highest accuracy across all priority levels, with 100% accuracy for Emergency and High tasks.
- DSCP SAC also performs well with high accuracy for High, Medium, and Low priority tasks.
- MADM Model II has perfect accuracy for Emergency and Medium tasks but lower accuracy for High and Low tasks. ACB SAC performs well for Emergency and Low tasks but does not provide accurate information for High priority tasks.
- Deadline (EDF) SAC has lower accuracy compared to other methods, especially for Medium and Low tasks. QCI SAC has relatively lower accuracy across all priority levels compared to some other methods.

The choice of methodology depends on the specific requirements and priorities of your task categorization system.

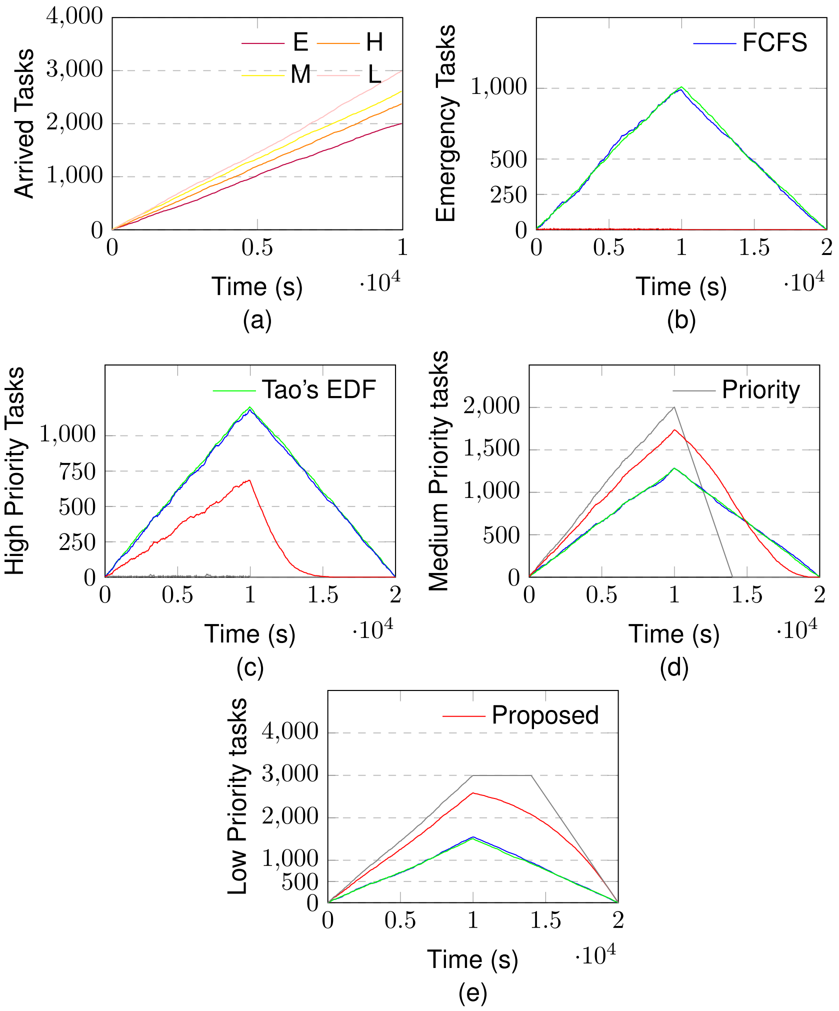


Figure 5.3: Unprocessed tasks for experimental settings - *I*

The Proposed Ensemble Method appears to be the most accurate overall, but the trade-offs between accuracy and other factors like computational complexity should also be considered when selecting a methodology for your application. The computational complexity of the proposed ensemble categorization is well explained in the previous section.

In Fig. 5.3, we plot the performance of the proposed scheduler and compare it against Tao's EDF, standard priority queuing, and FCFS-based approaches. The simulation settings of Fig. 5.3 are captured in Table 5.3. This experiment *I* constitutes the first of the two experimental settings outlined. Tao's EDF is a deadline-based scheduling approach that prioritizes tasks based on their deadlines, ensuring that tasks with imminent deadlines are executed first. Standard priority queuing assigns tasks priorities and executes higher-priority tasks before lower-priority ones. FCFS-based approaches execute tasks in the order they arrive, without considering priorities or

Table 5.3: Experimental Settings - I for Fig 5.3 - Proposed Priority Based Scheduling

Property	***Value***
Number of Tasks	10000
Execution time of a task	2 seconds
Emergency Tasks	20.14%
High Priority Tasks	23.58%
Medium Priority Tasks	26.24%
Low Priority Tasks	30.03%
Task arrival distribution	Poisson
Poisson Mean and Seed parameters	25 and 1000%

deadlines.

We process 10K tasks where tasks arrive in batches. The task population comprises 20% emergency, 23.5% high, 26.25% medium, and 30% low priority tasks. In Fig. 5.3, we plot the number of unprocessed tasks for each priority class against time in seconds. Each task takes an average of 2 seconds to get executed. The Poisson Distribution is used task arrival. Mean is set to value 25, which represents the average number of events in the distribution, and seed, which is an optional parameter, is set to 1000, used to initialize a random number generator for generating random values following the Poisson Distribution with the specified mean for the experiment in Fig. 5.3a. In the simulation, the Docker task execution time of 2 seconds was chosen. This simplifies the experiment, allowing for a fair comparison of scheduling methods (e.g., FCFS, Random, Round Robin, Tao's EDF, standard priority) against the proposed approach without the complexity of varying execution times. While the specific value was selected for practicality, it aligns with typical task durations in the simulation context.

In the context of scheduling algorithms, particularly in First-Come-First-Serve (FCFS) and Earliest Deadline First (EDF) strategies, there is a notable issue with emergency and high-priority tasks experiencing prolonged waiting times. This means that these critical tasks, which are often time-sensitive or of utmost importance, tend to linger in the queue for a significant duration before being processed.

In our proposed approach, we have strived to address this concern effectively. When we compared the number of pending emergency and high-priority tasks using our methodology against the conventional standard priority queuing method, we found that our approach maintains a comparable number of such tasks in the queue. This means that we don't compromise on the prompt processing of these high-priority tasks.

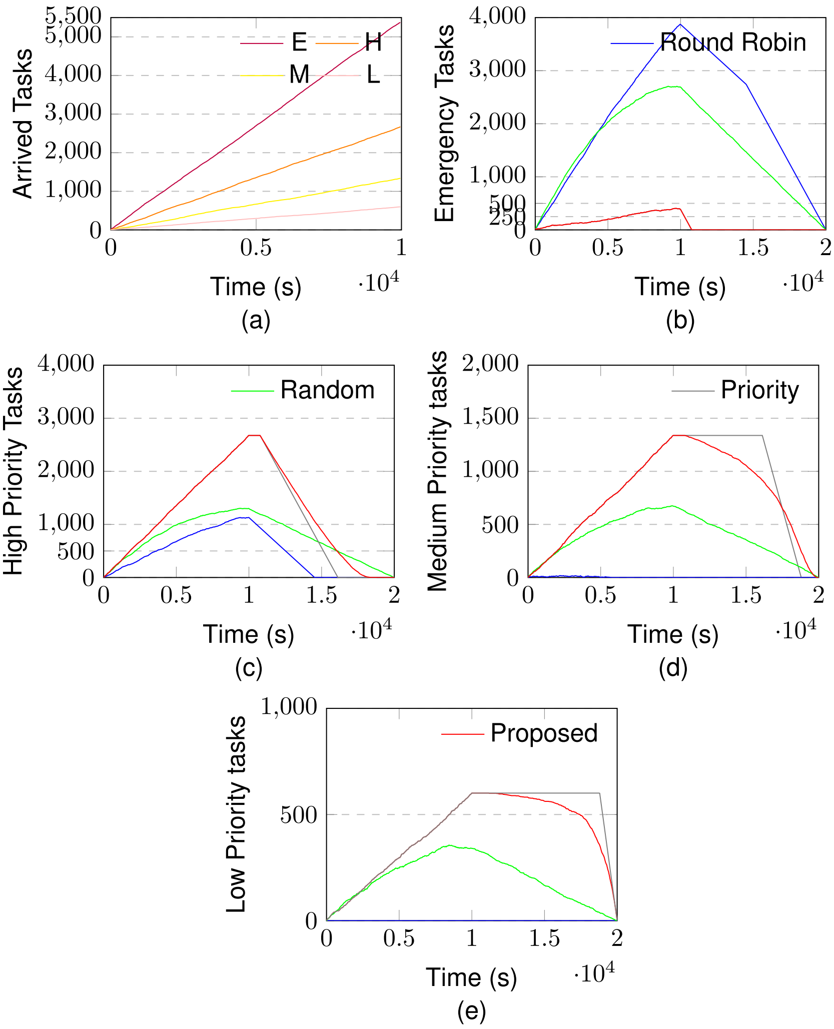


FIGURE 5.4: Unprocessed tasks for experimental settings - *II* (Logarithmic Distribution)

However, our approach brings a significant advantage over standard priority queuing when it comes to medium and low-priority tasks. These tasks often form long queues in traditional priority queuing systems, leading to delays and inefficiencies. In our method, we observed a remarkable reduction in queue buildup for medium and low-priority tasks. Specifically, within the observed time interval from the start of the process to 1.2×10^4 seconds later, we noticed a reduction in queue length by 9% for medium-priority tasks and 5% for low-priority tasks.

This means that our proposed methodology not only ensures that high-priority and emergency tasks are handled promptly, as in standard priority queuing, but it also optimizes the system's overall efficiency by significantly reducing queue congestion for less critical tasks. This outcome makes our approach a promising solution for systems where a balance between prioritizing critical tasks and maintaining overall performance is essential.

Table 5.4: Experimental Settings - *II* for Fig 5.4 - Proposed Priority Based Scheduling

Property	***Value***
Number of Tasks	10000
Execution time of a task	2 seconds
Emergency Tasks	53.33%
High Priority Tasks	26.67%
Medium Priority Tasks	13.33%
Low Priority Tasks	6.66%
Task arrival distribution	Poisson
Poisson Mean and Seed parameters	25 and 1000%

In Experiment II, we designed the task composition according to a logarithmic distribution with an emergency-to-high-to-medium-to-low ratio of 8:4:2:1, as visualized in Fig. 5.4a. The simulation settings of Fig. 5.4 are tabulated in Table 5.4. In this experiment, additionally, we have also compared with Round Robin and Random scheduling algorithms. Round Robin allocates CPU time to tasks in a circular order from the pre-defined set of topics. Random Scheduling, selects tasks for execution in a completely random manner, without any specific order or priority, making it unpredictable. In this experiment, we can observe that both Round Robin and random scheduling encounter a notable problem with queuing a large number of emergency and high-priority tasks.

In Fig. 5.4b, both the Priority and Proposed approaches immediately execute emergency tasks without delay. While handling high-priority tasks, the Proposed approach, while slightly trailing, exhibits comparable performance when compared to priority-based scheduling.

In Fig. 5.4c, during the timeframe spanning from 1×10^4 to 1.6×10^4 seconds, we observe the trend where medium-priority tasks effectively circumvent lengthy queueing. This, in turn, mitigates the risk of medium-priority tasks facing resource starvation, especially for those tasks that arrived earlier in the queue.

Similarly, in the interval stretching from 1×10^4 to 1.8×10^4 seconds in Fig. 5.4d, we note a similar trend where low-priority tasks, which entered the system earlier during the simulation run, receive substantially earlier execution within the Proposed approach in contrast to the priority-based scheduling method.

5.5 Chapter Summary

In this chapter, we presented QoS-driven task offloading through ensemble categorization and probabilistic prioritized task scheduling at edge servers. We apply enhanced weighted Borda scoring for topic selection. We simulate the Network slicing setup through Mininet, Flowvisor, POX and Beacon controllers. The offloaded tasks to edge servers are categorized and placed in Kafka topics, and later processed through docker containers. We published this work [3] in the 18th Annual Consumer Communications Networking Conference (CCNC), 2021.

Chapter 6

Application-aware QoS-Based Routing for 5G Network Slicing

6.1 Introduction

We know that software-defined networking has spurred the paradigm of programmable network structures. SDN has three layers: control, data, and infrastructure planes. SDN has predominantly centered around the functioning and performance aspects of the controller and data plane co-ordination. [119].

The application-aware routing is a method to administer the network from an application point of view. Here, the prime focus is applying QoS constraints and the maximization of relevant utility functions in SLAs.

5G mobile platform comprises of multiple radio access and wireless technologies [49]. Similarly, 5G core and transport network has widespread softwarization and seeks end-to-end path optimization that targets the application's QoS requirements and SLA. In the previous generations of cellular networks, Key Performance Indicators are over-provisioned to meet the application demands. However, to implement network slicing and enable multi-tenancy, 5G would need more robust and reliable co-ordination and harmonization between RAT, transport, and core networks, which are connecting the end-users and application servers.

In this chapter, we apply application-aware routing principles in the state-of-the-art QoS framework, by measuring QoS metrics, mapping, and allocating paths while meeting SLA boundaries. Firstly, we estimate the key QoS parameters such as latency, packet loss, and jitter of the data path, and we also compute the notional value of the above metrics. The second step is to map each data route against the SLA class definition of users. Finally, we discuss QoS Key Performance Indicator (KPI) driven routing scheme through the standard algorithm in SDN.

This chapter's content is organized as follows: Section 6.2 outlines the system model. Section 6.3 discusses the proposed mechanism using application-aware routing principles and the overall workflow. The simulation environment and the QoS framework are described in Section 6.4. Section 6.5 evaluates the results against other candidate solutions. In the last section, we conclude by summarizing our work and defining the future directions.

6.2 Application-aware routing system model and problem statement

Let $G = \{V, E, A, U, S\}$, where V denotes the set of base station, routers and switches in the Transport and Core Network. E represents the links from the wireline communication between these routers, Layer 2, and Layer 3 switches. Let S denote the application server in the core network. Let A be the set of antenna in a base station ($b \in B$) on massive Multiple Input Multiple Output (MIMO) in 5G New Radio communication. The users associated with the BS are denoted by u. Each user can have different client requests, indicated by vector $R(u)$.

The uplink path P_{u_1} of first hop wireless channel and wireline transport can be described as:

$$P_{u_1} = u \cdot \{a_1, a_2, ..a_m\} \cdot b \cdot v_1 \cdot e_1 ... \cdot v_n \cdot e_n \cdot s_1$$

The application server ($s_1 \in S$) receives the data x (vector) through the transport and core (V, E) from the base station b. The message $\bar{x}$ is transmitted through a set of $m \times 1$ antennas $\{a_1, a_2, ..a_m\}$ to user u.

Similarly, the uplink path P_{u_2} for a mobile wireless backhaul would be,

$$P_{u_2} = u \cdot \{a_1 .. a_{m_{b_1}}\} \cdot b_1 .... \{a_1 .. a_{m_{b_k}}\} \cdot b_n \cdot v_1 \cdot e_1 ... \cdot v_n \cdot e_n \cdot s_1$$

where m_{b_k} is the number of antennas in base station b_k. In a mobile wireless backhaul, $k > 0$ and $n >= 0$. In a pure mobile wireless backhaul with an edge server s, $n = 0$. Here, (.) operator indicates the flow of data from one entity to other.

The communication channel between the user and the transmitter exhibits small and large scale fading, both modeled using the Rayleigh fading model. In this context, the received signal y (a vector) at the first-hop wireless base station [120] can be described using Equation 6.1.

In this equation:

- ρ_u represents the normalized scalar downlink transmit power of the user.
- n denotes the additive white Gaussian noise vector.
- The transmitted message is represented by the signal vector x.
- G is the downlink composite channel matrix.
- The set of users is numbered from 1 to $|U|$.

To summarize, Equation 6.1 captures the received signal y at the first-hop base station, taking into account user transmit power, noise, transmitted message, and the characteristics of the channel, which includes both small and large-scale fading based on the Rayleigh fading model.

$$y = \sqrt{\rho_u} G \bar{x} + n \tag{6.1}$$

To meet the users' QoS requirements, let us define the Service Level Agreements (SLA). An SLA class i consists of attributes β_i ,μ_i , Φ_i , and τ_i, where each of the metric has a minimum or maximum acceptable criterion. The desired value for these metrics are described through min and max criteria.

$$SLA_i = \{\beta_i, \mu_i, \Phi_i, \tau_i\}$$

$$\forall j \in SLA_i, \exists (j_{min}, j_{max})$$

β is the requested user bandwidth measured in Mbps. $\tau(P_u)$ is the round-trip latency in path P_u measured in milli seconds, which is defined as the sum of transmission time ($T_{t,u}$) at the antenna/router, propagation time ($T_{p,u}$) over the wired or wireless link, and the processing time (T_{pr}) at the server along the routing path.

$$\tau(P_u) = 2 * \left\{ \sum_{p}^{P_u} (T_{t,u} + T_{p,u}) + T_{pr} \right\} \tag{6.2}$$

Here, Φ is the Signal-to-Interference-plus-Noise Ratio (SINR) defined for the MIMO system for user j. The total SINR is the effective product of all the links traversed in the path.

$$\Phi(u_j, A_j) = \frac{\rho_u ||g_j||^2}{\sum_{i=1, i\neq j}^{K} E\left\{ \left| \frac{g_j^H}{||g_j||} g_i \right|^2 \right\} + E\left\{ \left| \frac{g_j^H}{||g_j||} n \right|^2 \right\}} \tag{6.3}$$

For wired backhaul, the noise and interference is negligible.

Here, μ is the measured periodic jitter. Say, the mean response time is $E(\tau_p)$, and r_k is the Round Trip Time (RTT) along path p during round k. Then, μ can defined as,

$$\mu(p) = \sqrt{\frac{1}{K}((E_\tau - r_1)^2 + (E_\tau - r_2)^2 + .. + (E_\tau - r_K)^2)} \tag{6.4}$$

6.3 Proposed Heuristic Application-aware Routing Methodology

6.3.1 Measurement of QoS Metrics

In the data path, the controller sends the beacon messages periodically over an interval. The one-way latency, Round Trip Time (RTT), and packet loss in the data path are measured. A route would consist of two components: a) the first-hop wireless channel between the end-user device and eNodeB and b) the wireline communication from the access point to the application server through the transport and core network.

6.3.2 Aggregation and Mapping

The average loss, latency, and jitter are computed through a sliding window of measured packet loss and latency. The application-aware routing uses the set of the latest polls in a sliding window to determine the SLA classification of the data path. Based on the measurement and calculation of path loss and latency, along with the bandwidth details, each path may satisfy one or more user-configured SLA classes. The next phase maps an application's traffic to the data plane that renders the desired performance. The data plane should meet the required constraints as per the SLA description in terms of bandwidth, latency, and jitter.

Algorithm 9: Candidate Path Identification

```
function pathIdentification($P, ptrack, n, e_{x,n}, S, SLA_i$)
v[n] = true
if $P.size() < |C|$ then
    return
if $n == S$ then
    if $evalMinQoS('full', e, n, ptrack, SLA_i)$ then
        ptrack.add(n)
        P.put($GUID$,ptrack)
else
    if $evalMinQoS('partial', e, n, ptrack, SLA_i)$ then
        while $m : adj[n]$ do
            if $!v[m]$ then
                ptrack.add(n)
                pathIdentification($P, ptrack, m, e_{n,m}, S$)
return P;
```

Algorithm 10: QoS Evaluation

```
function evalMinQoS($type, e, n, path, SLA_i$)
if $e.isAirInteface()$ then
    Compute $\Phi_e$
    if $\Phi_e < \Phi_{min,SLA_i}$ then return false
Compute $\beta_e$ and $\tau_{path}$
if $\beta_e < \beta_{min,SLA_i} || \tau_{path} < \tau_{min,SLA_i}$ then
    return false
if $type.isFull()$ then
    Compute $\mu$
    if $\mu_{path} < \mu_{min,SLA_i}$ then return false
return true
```

6.3.3 Path identification

Let P be a hash map of paths $< path\_id, nodes >$ selected for evaluation through Algorithm 9. Let $N \leftarrow \{V, A, B\}$ comprise of vertices (routers and switches), and base station antennas. The controller initiates Depth First Search (DFS) in the directed paths from the source. In the path identification phase, we intend to find n possible paths that meet the minimum SLA specifications of the class. Along the route, the path identifier and the visited nodes are stored in the data structure. DFS attributes to the time complexity of $O(V+E)$, and can be applied to find the route in the presence of failures.

Once we traverse a node, we check if the parameters have exceeded the upper bound of the SLA class. For eg., whether the latency has crossed the maximum latency range, if so, rejects the path through Algorithm 10. When the traversal leads to the server node, the path is identified. The paths are assessed against the QoS parameters. When it satisfies the minimum acceptable range in terms of measured KPI and inherent theoretical KPI of the SLA class, the path is selected for further evaluation against the remaining $n-1$ identified connections. The nodes in the selected path to the server are marked as visited to avoid cycles.

6.3.4 Path evaluation

The selected paths are evaluated against the SLA_i class to which it is mapped. Among the paths, the path with the highest rank is elected as shown in Algorithm 11. The rank is calculated based on the Borda scoring of the path among each attribute in the SLA_i class.

Each path from the candidate set (P) is evaluated for each attribute j in the SLA_i class. Every path is ranked in the natural order against each QoS metric. Path identification and evaluation are re-triggered when there are soft network failures or new nodes are deployed. Similarly, re-computation is triggered during periodic monitoring of the QoS metrics phase when the elected paths fall below the minimum SLA values. The Borda scoring can be further extended by providing weights against the natural order rank.

The Borda score of an evaluated path is:

$$BS(\rho_{p_1}) = \begin{cases} \ (|P|-1) \times \#\{j|j\, ranks\, \rho_{p_1}\, first\} \\ +(|P|-2) \times \#\{j|j\, ranks\, \rho_{p_1}\, second\} \\ +... \\ +1 \times \#\{j|j\, ranks\, \rho_{p_1}\, second\, to\, last\} \\ +0 \times \#\{j|j\, ranks\, \rho_{p_1}\, last\} \end{cases} \tag{6.5}$$

The Borda scoring can be further extended by providing weights against the natural order rank.

Algorithm 11: Path Evaluation

```
$\forall p \in P$, Compute $\beta_p, \mu_p, \Phi_p, \tau_p$
$\forall p \in P$, Compute $BS(p)$
$bp \leftarrow \arg\max_p BS(p)$
return $bp$
```

Table 6.1: Network Slices - Core Networks settings

Slices	***TL Bandwidth***	***TL Latency (per hop)***	***TL Packet Loss (per hop)***
Slice 1	500MBps	0.1ms	0.01
Slice 2	500MBps	0.5ms	0.01
Slice 3	500MBps	0.5ms	0.03
Slice 4	500MBps	0.1ms	0.1
Slice 5	1GBps	0.1ms	0.01%
Slice 6	1GBps	0.1ms	0.1%
Slice 7	1GBps	0.5ms	0.1%
Slice 8	250MBps	0.1ms	0.01%

6.4 QoS Framework and Simulation Settings

In this simulation setup, we use the same network slicing setup discussed in the work with Kafka, Docker, and Java Web services. Firstly, we generate the random graph topologies and QoS configurations of edges through the Erdos-Renyi MATLAB module. These topologies and their configurations (in terms of bandwidth, packet loss, and latency) are realized through Mininet network.

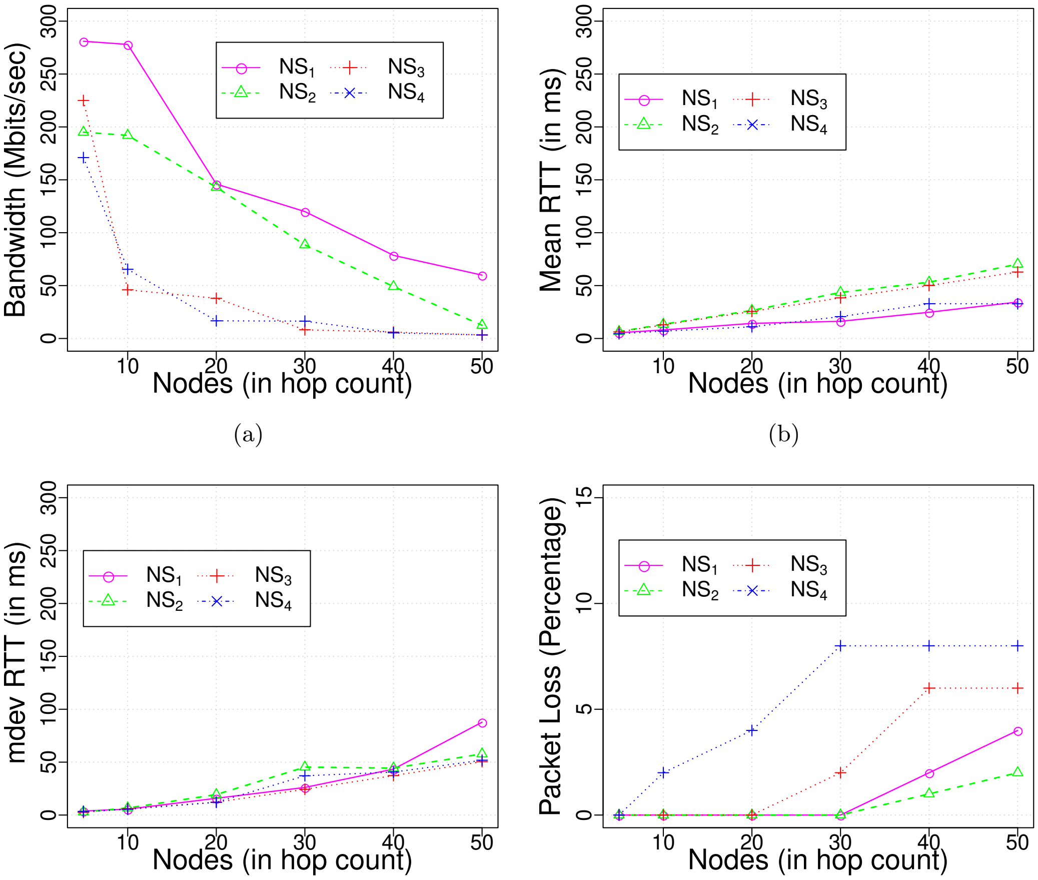
We apply the application-aware routing procedures in the well-established architecture of NS (through FlowVisor), SDN (POX controllers), and NFV modules deployed in Docker instances. Each SDN controller, such as POX, has slice-aware QoS modules, which invoke the virtual network functions in Docker for optimal path allocation.

6.5 Results and Analysis

We have come up with a few samples for NS settings in Tables 6.1 and 6.2, configured through bandwidth, latency, and packet loss at each hop of Transport Network and RAN, respectively. We state some real-world metrics set for various services during the analysis.

Table 6.2: Network Slices - Radio Access Networks settings

Slices	***RAN Bandwidth***	***RAN Latency***	***RAN Packet Loss***
Slice 1	300MBps	0.5ms	0.05%
Slice 2	250MBps	0.5ms	0.05%
Slice 3	250MBps	0.5ms	0.05%
Slice 4	250MBps	0.5ms	0.2%
Slice 5	1GBps	0.5ms	0.05%
Slice 6	1GBps	0.5ms	0.2%
Slice 7	500MBps	1ms	0.2%
Slice 8	250MBps	0.5ms	0.01%



(a)

(b)

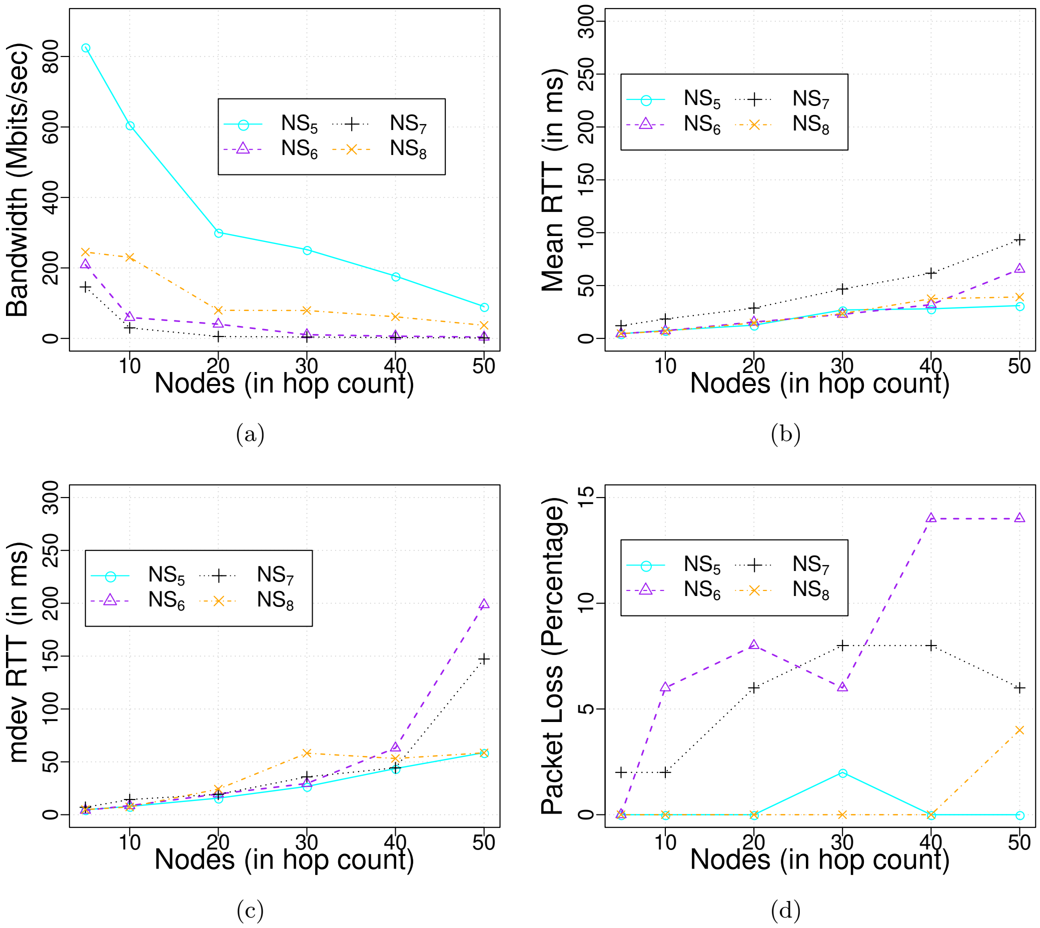
(c)

(d)

Figure 6.1: Evaluation of Network Slices 1- 4 for actual QoS performance

Having more slices with higher end configuration, can indeed saturate the network capacity, depending on the available hardware resources and the specific network topology and traffic patterns. To extend the overall eco-system to support more slices, one can consider following:

- Hardware Resources: Ensuring that our physical hardware (e.g., CPU, RAM, and network



(a) (b)

(c) (d)

FIGURE 6.2: Evaluation of Network Slices 5 - 8 for actual QoS performance

adapters) can handle the increased load. Upgrading or using more powerful hardware can help accommodate a larger number of slices, hosts and links.

- Network Topology Optimization: Carefully designing our network topology to reduce unnecessary link and host saturation. Efficiently structuring our network can help mitigate capacity issues.

- Traffic Management: Implement traffic shaping and Quality of Service (QoS) policies to prioritize traffic and prevent congestion. This can help ensure that critical traffic gets the necessary resources.

In Mininet specifically, for the simulation, one can consider the following approaches:

- Parallelism: Distributing the simulation across multiple Mininet instances or using distributed simulation frameworks if our simulation workload is extremely large.

- Optimized Mininet Settings: Tweaking Mininet parameters, such as the CPU scheduling policy and resource allocation, such as CPU and Memory allocation, link capacity, queue size, host - link properties, to optimize performance for our specific use case.
- Scaling Down: If scaling up hardware resources is not feasible, we can consider downsizing our simulation, reducing the number of hosts and links, or simulating smaller subsets of our network.
- Profiling and Optimization: Profiling our Mininet setup to identify bottlenecks and areas for optimization. Tools like top, htop, and iperf can help in diagnosing performance issues.
- Simulation Timeframe: Adjusting the timeframe of our simulation to avoid overloading the network during the entire simulation period.

By applying these strategies, one can extend Mininet-based network simulation to support a larger number of slices, hosts, and links while maintaining network performance and avoiding saturation issues.

6.5.1 QoS Performance of Network Slices

In Fig. 6.1 and 6.2, we evaluate network slices on their performance. We sequentially plot the perceived bandwidth and mean RTT in Fig. 6.1a and 6.1b. $MDEV$ RTT and the overall Packet Loss (PL) is outlined in Fig. 6.1c and Fig. 6.1d. The x-axis indicates the number of hops between the *src* and *dest*. In the y-axis, we measure the bandwidth in MBits/sec, latency in ms, and packet loss in percentages. Although Slice 2 and 3 have near-identical configurations, we observe bandwidth of Slice 2 is higher than Slice 3, $\beta(NS_2) >> \beta(NS_3)$. It is owing to higher per-hop packet loss probability, $PL(NS_2) \approx 3PL(NS_3)$.

Similarly, the mean RTT doubles between NS_1 and NS_2 due to an increase in the propagation delay from 0.1ms to 0.5ms. Assuming round trip, the total latency comprises nodes in the traversed path of RAN (R), Transport (T), and backhaul links (CN - Core Network) as shown in the (6.6) and (6.7).

$$L_{req} = 2 \cdot \left[L_R + \sum_{e_t=1}^{T,CN} L_{e_t} \right] + L_{SV} \tag{6.6}$$

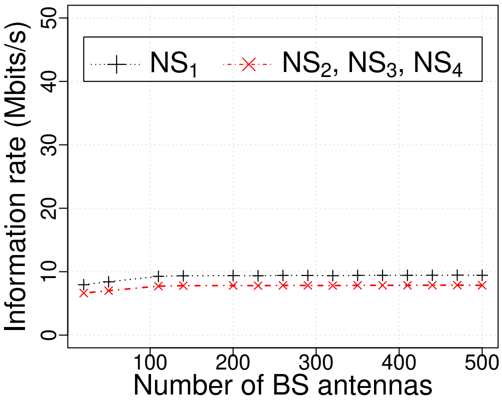


FIGURE 6.3: Information Rate

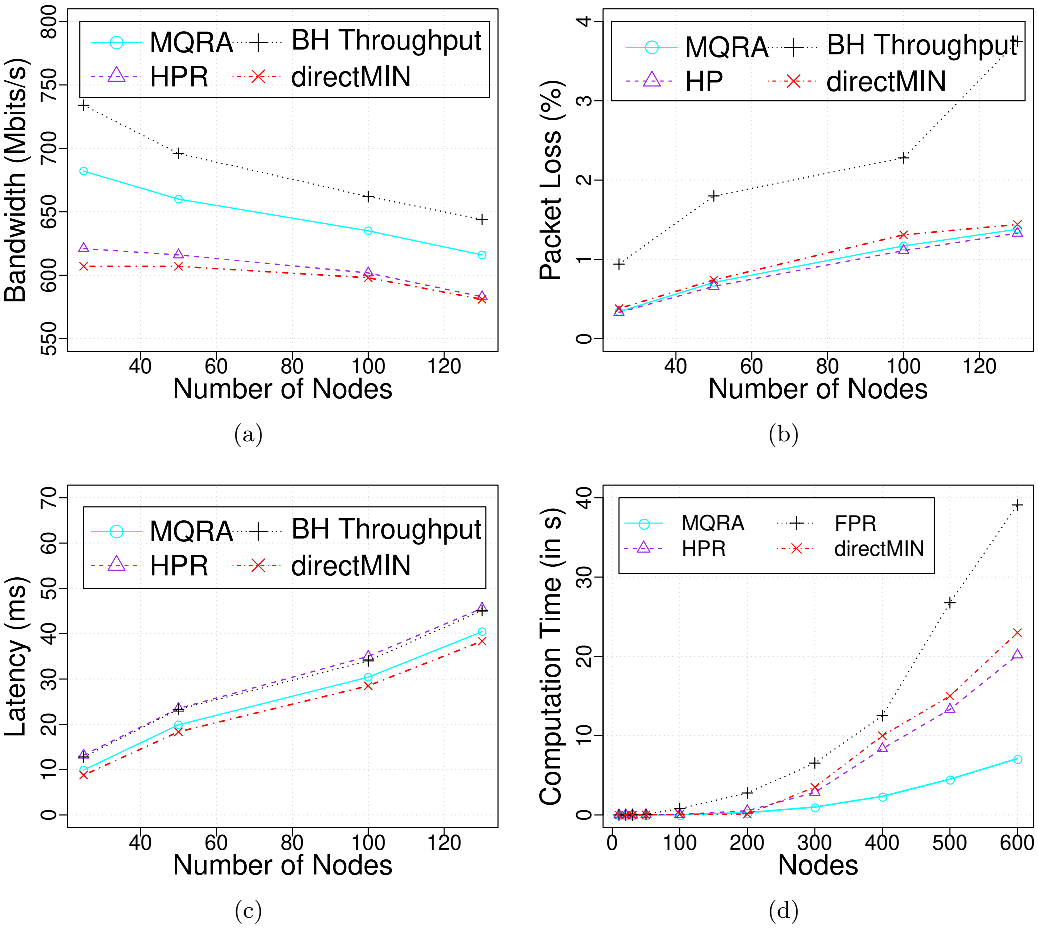
$$L_R = \frac{L(Pkt)}{\vartheta_{ut,R} \cdot \zeta_R} + T_{pr_R} + \frac{(\frac{T_{pr_R}\lambda}{m})^{\sqrt{2(m+1)}-1}}{m - \lambda T_{pr_R}} \cdot \frac{Cv_a^2 + Cv_p^2}{2} \tag{6.7}$$

Latency on a given RAN link between user terminal (ut) to Radio (R) involves the transmission time of the source node, propagation delay over the link, queuing delay, and processing time on the target node. Propagation delay consists of the time taken for transferring the packet size $L(Pkt)$ over an allocated edge. When $M/M/m$ queuing model is considered, where m is the number of parallel processing units, T_{pr_R} is the processing time of a unit, λ is the rate of arrival, and Cv_a and Cv_p indicate the coefficient of variation of service time and average inter-arrival time. Here, in this experiment, though the propagation delay is increased five times since the delay is minimal (0.5ms), the perceived latency (τ) drops only by half due to the effect of other components mentioned in Equations (6.6) and (6.7).

According to surveys, the network should keep packet loss of Voice over Internet Protocol (VoIP) traffic below 1%. For video, between 0.05% and 5% is preferred. When nodes between the source and destination are less than 20, these slice configurations fit in-meeting VoIP requirements. NS_1, NS_2, NS_5, and NS_8 present less packet loss, and these slices can service GBR VOIP when the number of hops increases. The overall *mdev* is around 100 milliseconds in the presence of many nodes.

For low latency reliable communications, a combination of a higher packet loss rate like NS_4 and a greater latency value like NS_3 would not be appropriate and may lead to higher round trip time and jitter. Instead, a slice configuration with both lesser propagation delay like NS_3 and negligible packet loss like NS_4 would be preferred.

NS_1, NS_5, and NS_8 would be best suited as they offer the least latency and packet loss compared to other candidates. The total cost of ownership (TCO) of NS5 is significantly higher than



(a)

(b)

(c)

(d)

FIGURE 6.4: Comparison of path selection algorithms

NS_8. Due to cost considerations around high bandwidth links, for a network setup that focuses only on ultra-reliable low latency communication, a slice like NS_8 would be ideal among the candidate configurations.

The user's information rate for the Massive MIMO system is shown in Fig. 6.3. The x-axis and y-axis represent the number of antennas, and the information rate is calculated in MBits per second. The information rate is computed through Maximal Ratio Combiner (MRC). The total rate experienced is plotted considering the random placement of users in the coverage area, the intensity of the signal, path loss, and increased SINR with more antennas leveraging spatial multiplexity.

6.5.2 Application-Aware Path Selection Algorithms

FPR [94] consumes exponential time, wherein other candidates discussed in this work exhibit a heuristic solution. The HPR [94] explores Dijkstra shortest-path based approach but considers only bandwidth constraints. HPR takes less time than (*directMIN*) [97]. The proposed heuristic application-aware routing shows significant reduction than HPR due to optimization based on QoS metrics during the path selection phase. When the latency or bandwidth is not met during the path identification, further traversal among the route is avoided.

SLA_{exp} is defined for the attributes j with (j_{min}, j_{max}) as follows. β_{exp} as $(100, 500)Mbits/s$, packet loss in $[0.01\%, 0.09\%]$ per hop and τ in [0.1,0.5] milliseconds per link. All the candidate algorithms HPR, directMIN, BH Throughput [32], and studied application-aware routing approach obey the minimum and maximum attribute constraints for the SLA class.

In Fig. 6.4a, 6.4b, and 6.4c, we plot the bandwidth, packet loss, and latency attributes in the y-axis. Though BH Throughput [13] exhibits high bandwidth, it suffers from high packet loss and latency. The directMIN displays minimal latency across the candidates. However, it demonstrates less bandwidth. The latency and packet losses are measured through standard ping flood, changing the size of bytes in an ICMP packet header and sending it over an extended ping. We used the IPerf tool for bandwidth measurement on IP networks. The usage of application-aware approach provides an equitable opportunity for the performance of the path in each attribute. Although it does not give the best performance in bandwidth and latency metrics, the performance of the selected routes is comparable to the single objective optimized approaches. These metrics are proven through repeated 1000 Monte Carlo runs.

The computational time of the application-aware path selection algorithms is discussed in Fig. 6.4d. The proposed heuristic application-aware routing shows a significant reduction than HPR and *directMIN* due to optimization based on QoS metrics during the path selection phase. When latency or bandwidth are not met during path identification, further traversal among the route is avoided. The x-axis represents the number of nodes in the route, and the y-axis represents the computational time. The path traversal from one node to another is randomly distributed between 100 microseconds to 1 milliseconds. For the proposed approach, the number of selected QoS paths for evaluation is capped at ten distinct routes.

The absolute values of bandwidth, packet loss, and latency are normalized on a scale of 0 to 1. We apply an enhanced max-min approach to derive the scale from the upward attribute - Bandwidth, and downward features - Latency and Packet loss: We calculated the difference between the best value of a metric against each algorithm's performance on the metric. We then computed the root mean square deviation against the top values of each metric. The discussed approach exhibits the least mean square deviation of 29.4%.

6.6 Chapter Summary

In this chapter, we accomplish application-aware routing through NS, SDN, and NFV. QoS parameters such as bandwidth, path loss, and latency constraints are assessed. The work discusses how our proposed heuristic application-aware routing methodology can be applied within the well-established QoS architecture through NFV, SDN, and slicing modules applied in 5G. We outlay the performance of network slices on various QoS metrics. We have focused on structuring the performance around the SLAs meant for the slices and how the application-aware approach guarantees the SLAs.

Chapter 7

Co-existence of Wi-Fi and Cellular Networks

7.1 Introduction

5G operates in the unlicensed spectrum to increase capacity. Wi-Fi is another prominent technology in these frequency bands. The proximity can cause interference. These cross-technology lack centralized control, negotiation and coordination between them.

The goal of co-existence is to ensure balance and equitable sharing of resources and communication channels among cross-technology devices in an indoor environment. The performance of co-existence can be measured by the following parameters: a) Channel acquisition rate of Wi-Fi vs. cellular devices. b) Throughput realized by these devices.

At the core of them are Listen-Before-Talk (LBT) medium access mechanism and Duty Cycling (DC) based approaches to drive the co-existence between cellular and other Radio Access Technologies (RATs). LBT operates through two related functions: Carrier Sense can recognize and distinguish the Wi-Fi packet headers. Energy Detection would perform the backoff data transmission based on the energy threshold.

The other standard coexistence MAC approach DC splits the shared radio channel through air time sharing between the Wi-Fi and LTE-U subsystems. In the DC-based approach, cellular devices can transmit signals only in a pre-determined duty cycle when one cellular unlicensed access point and one Wi-Fi base station co-exist. In contrast, Wi-Fi has to contend with

Distributed Coordinated Function (DCF). Wi-Fi users and stations don't have prior knowledge of the DC period or DC transmission operated by Cellular technology.

The cellular devices achieve the duty cycle in a deterministic manner. It has a centralized access control mechanism. The access time and OFDM subcarriers of cellular frames are pre-determined in eNodeB. MAC schedulers can consider radio measurement based on deterministic access time and sub-carriers to plan for necessary QoS. However, since Wi-Fi MAC follows DCF, the access is random and distributed. While improving the efficiency of spectrum usage, the users would experience interference due to heterogeneous RAT systems. The interference mitigation among dense areas would be vital for decoding the signal [121].

Now, network adapters are becoming exceedingly programmable. Selfish users will aim to increase their share of data transmissions.

In this chapter, we study the effect of the selfish behaviour of the nodes in cross-technology communications. There are several methods the selfish user can exploit to trigger an unfairness in the network. The first way is to vary the CCA threshold such that it can disable its carrier sensing. It enables the selfish user to gain more transmission opportunities. Though the channel might be busy, this selfish user starts transmitting immediately, leading to interference among other users. The second approach is misbehaviour caused by selfish users by declining to forward the network packets in route discovery and maintenance processes. The third method is setting the backoff window smaller. In CSMA/CA, the node first listens if the channel is idle for more than Distributed Inter Frame Space (DIFS) timeslot, then it sends the Request to Send signal. In case of a failed transmission, the user enters a randomized truncated exponential backoff period. [122].

The CSAT algorithm, through exponential backoff, relies on a stochastic delay of packet transmissions during collisions [123]. The nodes operate by these rules to maintain a fair co-existence. The selfish user can exploit this backoff window mechanism to attain better transmission opportunities. This could be dangerous in cross-technology deployments. With the absence of centralized control, the node could obtain a larger share of the available bandwidth at the expense of the others. In this chapter, we consider selfish nodes that don't appreciate the regulated exponential backoff. The adverse impact of QoS on other selfish users, the underlying network, and regular users are well studied for LBT and Duty-Cycling mechanisms in this chapter. Moreover, we characterize the backoff mechanisms of many selfish nodes and study their effect on the network. We depict the presence of Wi-Fi selfish user in Cellular Wi-Fi coexistence in Fig. 7.1. To the

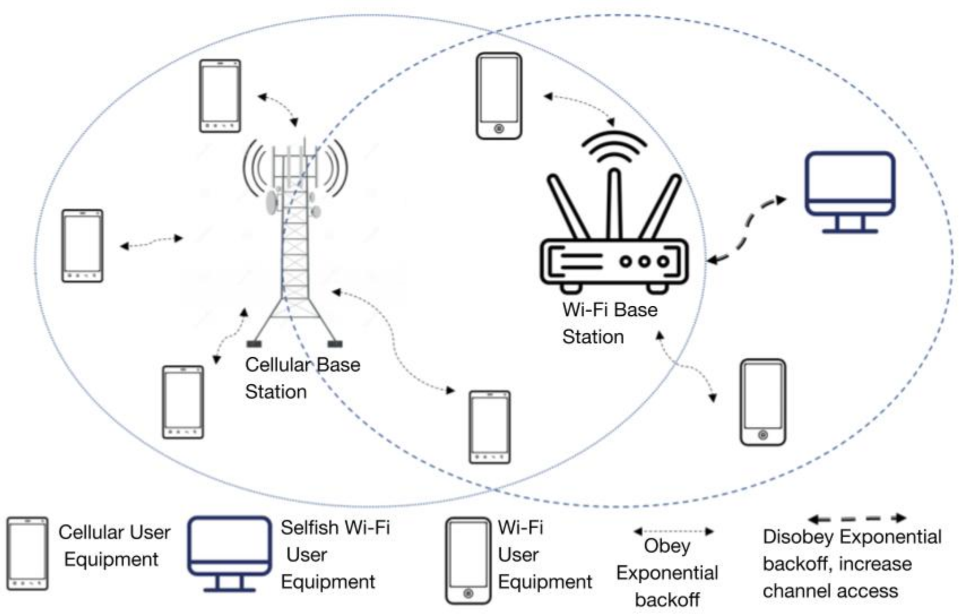


Figure 7.1: Presence of Selfish user in Cellular Wi-Fi Co-existence

best of our knowledge, this is the first discussion on the rational cheating of nodes in the purview of cross-technology between Wi-Fi and 5G.

The rest of the chapter is organised as follows. The system model is described in Section 7.2. Section 7.3 studies the case of backoff mechanisms, nodes, and their influence on the LBT-based co-existence. Section 7.4 discusses the Duty-Cycling based co-existence. Section 7.5 discusses the simulation setup of cross-technology communications. Section 7.6 investigates the impact of selfish users on LBT-based co-co-existing networks and other regular users within the network. Section 7.7 analyses the effects of different degrees of selfishness and backoff patterns in LBT. Section 7.8 discusses the results and analysis of Duty-cycle configuration The counteractions of selfishness are briefed in Section 7.9.

7.2 Listen-Before-Talk based Co-existence System Model

We begin system modelling by considering two sets of wireless nodes, $D = \{d_1, d_2, d_3, ...d_{|D|}\}$ that are the devices serviced by Wi-Fi AP and $L = \{l_1, l_2, l_3, ...l_{|L|}\}$,that are served by the 5G cell. The co-existence among Wi-Fi and 5G cells is studied. Here, we define the selfish as a subset of nodes $C_1 \subset D$ and $C_2 \subset L$. The terms *selfish*, *misbehaving*, and *cheater* are used interchangeably to refer to nodes that disregard the exponential backoff protocol. Conversely, the terms *regular*, *legitimate*, and *honest* are used to describe users who adhere to the exponential backoff protocol.

During generalization, we denote all the users or nodes as N, $N \in \{D, L\}$. The selfish users $C \in \{C_1, C_2\}$ doesn't respect the exponential backoff. This mode of selfishness is most convenient among the cheaters since it doesn't require changes to the protocol's operation.

The static or dynamic backoff is indicated by two dimensional vector B. $B = \{B_1, B_2, B_3, ...B_n\}$, where B_i is the backoff vector for user i, $\forall\, i \in N$. For an user i, the backoff values can be defined as $B_i = \{b_{i,1}, b_{i,2}, ..b_{i,j}, ..b_{i,T}\}$, where j indicates the timeslot during a discrete duration T.

We assume selfish users to be rational, i.e., they want to maximize their benefit by reducing the waiting time to access the channel and increasing the frequency of obtaining the channel access to deliver better throughput.

In particular, the misbehaving nodes want to maximize throughput (r) and minimize the waiting time (τ) for channel acquisition. The strategy of each selfish node (i) would be not to follow the exponential backoff values and alter the backoff values $(b_{i,j})$ such that its expected payoffs (utilities) are maximized.

For the LBT, Request-To-Send (RTS) and Clear-To-Send (CTS) mechanisms is used.

T_{start} be the minimum wait duration normalized to the system slot time before the user can start a transmission.

$$T_{start} = DIFS + T_{RTS} + SIFS + T_{CTS} \tag{7.1}$$

T_{start} is the time involved between transmitting the RTS from the user and obtaining the CTS before data transmission.

$$T_{\hat{b}} = \sum^{\psi} rand(1, \min\{2^{log_2(CW_{min})+\Upsilon}, CW_{max}\}) \tag{7.2}$$

$T_{\hat{b}}$ is the total time spent in the repeated exponential or paused backoffs. A backoff is capped to CW_{max}. The minimum contention window is CW_{min}. The backoff counter increments in each repeated iteration of failed transmission.

When the channel is sensed as busy, the backoff counter is paused. Here Υ is the running backoff counter in eq. 7.2. Once successful transmission occurs, Υ is reset to 0. The backoff window

Table 7.1: Notations - Wi-Fi Co-existence with 5G

Symbol	***Description***
$D = \{d_1, d_2, ...d_{\|D\|}\}$	Set of Wi-Fi nodes
$L = \{l_1, l_2, ...l_{\|L\|}\}$	Set of cellular nodes
B	Two dimensional backoff window vector
N = {D,L}	Total nodes in the co-existing network
$C = \{C_1, C_2\}$	Set of misbehaving nodes
$b_{i,j}$	Backoff window of user i during timeslot j
r_i	Throughput of node i
τ_i	Waiting time of node i

is $b_{i,j} = rand(1, \min\{2^{log_2(CW_{min})+\Upsilon}, CW_{max}\})$. Here ψ is the number of retries to sense the channel before a successful transmission.

$$T_f = \begin{cases} b_{i,j}, \;\; \textit{Channel busy sensed after backoff window} \\ b_{i,j} + T_{RTS} + DIFS, \;\; \textit{Channel occupied amid RTS/CTS} \\ b_{i,j} + T_{RTS} + DIFS + 2 * SIFS + T_{CTS}, \; \textit{Base station} \\ \qquad\qquad \textit{chosen another node for data transmission} \end{cases} \tag{7.3}$$

There are different variants of T_f. Once the backoff window is completed, and immediately channel could be sensed as busy. Or, the medium could be reported as occupied amid RTS/CTS requests. The BS would have to decide on only one of the users to transmit and send CTS signal. In the first case, when the channel is sensed busy, Υ is not incremented. In the rest of the failure cases, Υ is incremented.

7.3 Co-existence Analysis for LBT

7.3.1 Correlation of Backoff Window and Throughput

The channel sensing and acquisition probabilities of a cellular and Wi-Fi node are discussed in this and upcoming sections. The throughput enjoyed by a given node k, which is the average information payload transmitted in a slot time over the average length of time, is computed using Bianchi's model [124] as follows:

$$r_k = \frac{P_k \bar{S}}{P\, t_S + P_{idle} t_{idle} + P_{collision} t_{collision}} \tag{7.4}$$

P_k is the probability that user k successfully acquires the channel in a given time slot, $P_k = \rho(t_k)\,\Pi_{j\neq k}(1-\rho(t_j))$. $\rho(t_k)$ denotes the probability of successfully sensing the channel to be idle for user k at time t. $\bar{S}$ is the average size of the packet and t_S is the time taken to transfer this packet. $\rho(t_k)$ is the channel sensing probability of user k, where $k \in D \cup L$. P_{idle} is the probability of channel being idle - $\Pi_j(1-\rho(t_j))$. t_{idle} is the duration of the idle period in a slot. $P_{collision}$ and $t_{collision}$ is the probability and average time spent in collision. P is the total access probability of all users, $P = \sum_k P_k = 1$.

We consider two separate Markov chains. The first one, with a fixed backoff stage, assumes the misbehaving nodes randomly select their backoff window size between 1 and the fixed value. The sensing probability for cheater i be $\rho(t_c)$. The second chain is for a well-behaved node. The cheater $c \in C$ sensing probability which fixes its backoff values in B_c would be [124],

$$\rho(t_c) = \frac{2}{\mu_{B_c}+1},\ c \in \{C\}$$

μ_{B_c} indicates the average backoff window value of cheater. The throughput of the cheater is,

$$r_c = \frac{\rho(t_c)\alpha_c}{\rho(t_c)\beta_c + \gamma_c}$$

where $\alpha_c = p_{-c}\bar{S}$, $\beta_c = p_{-c}(t_S - t_{idle}) - s_{-c}(t_S - t_c)$, and $\gamma_c = (1-p_{-c}-s_{-c})t_c + s_{-c}t_S + p_{-c}t_{idle}$ with the following substitution [104],

$$p_{-c} = \Pi_{k\neq c}(1-\rho(t_k))$$

$$s_{-c} = \sum_{j\neq c}\rho(t_j)\Pi_{k,d\neq j,d\neq c}(1-\rho(t_k))$$

p_{-c} and s_{-c} ignore about probabilities of transfer from the c^{th} node. p_{-c} is the probability that none of the other nodes transfer during a timeslot. s_{-c} is the sum of the channel acquisition probabilities of each of the other nodes in the network.

It is important to notice here, the selfish user has full control over its backoff window B_i. By varying B_i, node i (cheater) can change its sensing probability $\rho(t_i)$. Let us assume B_i is a continuous variable. We apply first-degree partial differentiation through the first derivation of r_i,

$$\frac{\partial r_i}{\partial B_i} = \frac{\alpha_c \gamma_c}{(\rho(t_c)\beta_c + \gamma_c)^2} \frac{-2}{(\mu_{B_c} + 1)^2}$$

where $t_S \geq t_c$. We conclude that expected received throughput r_i is a strictly decreasing function of B_i (for $\rho(t_j) < 1, j \neq i$). Thus, by unilaterally decreasing its own B_i, a device can increase its throughput. Except, when $\rho(t_k) = 1$ for some device k.

7.3.2 Effect of misbehaviour in co-existing network

We refer to the performance metric, Order Gain, $G(d_1, d_2, t)$ to quantify the gain of backoff misbehavior. It is based on waiting time τ_i, which indicates the average sum of the total number of idle slots from the time node i contends for the channel to acquire it successfully. Once we sum the number of slots spent in failed transmissions T_f of node i and the time spent to acquire the channel T_s, we divide this sum by the number of successful acquisitions to derive τ_i.

Order gain can be defined as,

$$G(d_1, d_2, t) = \log \frac{\varrho(\tau_{d_1} > t)}{\varrho(\tau_{d_2} > t)}$$

$\varrho(\tau_{d_1} > t)$ denotes the probability that the waiting time of d_1 is greater than a given t, showing how often the waiting time of node d_1 is longer than a given value.

The work proves [108] this order gain of a fixed-window backoff misbehaving node over legitimate nodes in a Wi-Fi only network is,

$$G(d_1, d_2, t) = \Theta\left(\frac{t}{ln\, t}\right)$$

$$d_1 \nexists C,\ d_2 \exists C,\ \text{and}\, d_1, d_2 \exists D$$

It reveals that the order gain G of fixed window backoff misbehavior following a uniform integer distribution is an increasing function as $t \rightarrow \infty$ and saturates as the number of users in the network increases. It confirms that this misbehaving node can always get considerable benefits from fixed-window backoff misbehavior.

In the wireless network, let's say $d_x \exists C_1$ is selfish and $\{d_x, d_y\} \exists D$ and $d_y \nexists C_1$. Here d_y is legitimate node. Then for a given t, w.k.t,

$$G(d_x, d_y, t) = \log \frac{\varrho(\tau_{d_x} > t)}{\varrho(\tau_{d_y} > t)} < 1$$

In an LBT-based network, where both Wi-Fi and cellular network follow CSMA/CA with similar CW_{min} and CW_{max}, the following condition holds,

$$G(d_y, l_z, t) = \log \frac{\varrho(\tau_{d_y} > t)}{\varrho(\tau_{l_z} > t)} \approx 1$$

Here, $d_y \exists D$, $l_z \exists L \, \text{and} \, l_z \nexists C_2$. Both d_y and l_z are legitimate nodes lying in the Wi-Fi and cellular network, respectively.

Order gain G follows the transitive additive rule and is proven here [108]. Hence, in a co-existing network where d_x is a selfish Wi-Fi node, and l_z is a cellular non-legitimate node, from the above two equations, we notice:

$$G(d_x, l_z, t) = \log \frac{\varrho(\tau_{d_x} > t)}{\varrho(\tau_{l_z} > t)} < 1$$

We observe here that the presence of a selfish Wi-Fi node in the network increases the waiting time of a legitimate node in the cellular network.

We discuss how equilibrium is achieved among all Wi-Fi users through Cooperation via the Randomized Inclination to Selfish Play (CRISP) strategy which leads the equilibrium to a Sub-game Perfect NE (SPNE) in Section 7.9. It also shows, how this can lead to detrimental effects on the co-existing cellular networks.

7.4 Co-existence Analysis for Duty Cycling

In the duty-cycle mode with time period T, let α be the initial fraction of the time - cellular unlicensed is ON, where the 5G users can communicate. The rest of the duration $(1 - \alpha)T$, Wi-Fi nodes can transmit till the end of the duty-cycle period. Let r_y be the average throughput of node y.

We can assume the cellular reference signal doesn't cause significant interference in Wi-Fi communication during the Cellular Unlicensed OFF period. The Wi-Fi subsystem has no direct

Table 7.2: Simulation Settings for Wi-Fi Co-existence

Parameters	***Values***
Location of Wi-Fi access point	(0,0)
Initialization of backoff window	[1, CW_{min}]
Location of Cellular Base station	(0, rand(0,IBS))
Regular Users MAC	LBT Exponential backoff
CW_{min}	16
Selfish Users MAC	LBT Uniform Integer Distribution backoff [1, CW_{min}]
CW_{max}	1024
Wi-Fi coverage radius (WCR)	30m
Total number of nodes	[4,20]
Minimum Separatable Unit (MSU)	1m
Total number of selfish users	0,1,2,3,4
Duration of simulation	200000 microseconds
Number of Monte Carlo Runs	1000
Position of Wi-Fi users	Straight, Diagonal from BS
DIFS time duration	34 micro seconds
Cellular coverage radius (CCR)	30m
One basic unit of simulation	1 micro second
RTS / CTS time taken	Propagation time between device and base station
Data Slots	1ms,10ms
Inter Base station distance (IBS)	[MSU,rand(min(WCR,CCR))]
SIFS time duration	16 micro seconds
Position of Cellular users	Straight, Diagonal from BS
Distance of Wi-Fi users from BS	rand(MSU,WCR)
Distance of Cellular users from BS	rand(MSU,CCR)

knowledge of unlicensed cellular network presence and simply adheres to the DCF mechanism. Thus, the Wi-Fi node senses the channel during the Cellular unlicensed ON period, detects the channel being busy due to cellular signals, and increments its exponential backoff. During the cellular ON period, Wi-Fi transmission will fail due to substantial cellular signal interference. When cellular signal interference is weak, the Wi-Fi users keep transmitting when possible.

In Duty-cycling, we study the effect of selfish Wi-Fi users on the co-existing network. We do not consider selfish cellular users as the cellular base station centrally manages the channel allocation and can detect the presence of such users who try to gain more throughput by deviating from the agreed protocols.

In Wi-Fi, since the users apply carrier sensing and exponential backoff through CSMA/CA and DCF functions, this method lacks centralized and distributed control over channel allocation.

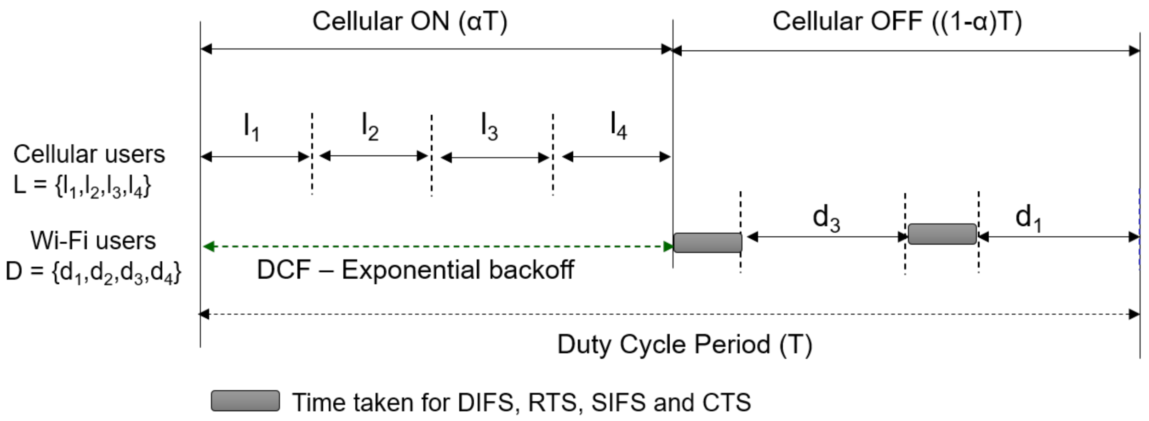


FIGURE 7.2: Duty-Cycling in Co-existence

Hence, we apply selfishness only to Wi-Fi users through a fixed backoff window to maximize the throughput. We discuss different scenarios to study the impact of Wi-Fi selfishness below.

7.4.1 Strict Time Domain Multiplexing (STDM)

In this use case, we consider both Wi-Fi and cellular operators follow a strict Time Domain Multiplexing. Both operators would have agreed mutually on α. When the Cellular - OFF period and duty-cycle period end, the cellular transmission would strictly begin immediately.

The interference (I) occurs when Wi-Fi transmission (IF_2) at the end of the current duty-cycle period overlaps with the start of the cellular portion (IF_1) of the next duty-cycle.

In Theorem 7.1, we prove if the operators mutually agree upon α and the network follows STDM duty-cycling strategy, the impact of Wi-Fi selfish users on co-existence and cellular users is limited to the length of the last Wi-Fi transmission in the Cellular-OFF period.

Theorem 7.1. *When α is agreed and Duty-cycle follows STDM by Wi-Fi and cellular operators, $\exists x,\, x \in D \,\&\, x \in C \rightarrow \forall y \in L$, impact on r_y is limited to the last Wi-Fi transmission in the duty-cycle period.*

Proof. Let the start time of the first duty-cycle period is t_0. The channel acquisition of $\forall y \in L$ lies in $[t_0, t_0 + \alpha T]$ in this cycle. Similarly, for all Wi-Fi devices, the transmission start time of $\forall x \in D$ lies in $[t_0 + \alpha T, t_0 + T]$. Let t_1, where $t_1 = t_0 + T$ is the start of the next duty-cycle period. The Wi-Fi transmission started before $[t_0 + T]$, can end after $[t_0 + T]$. This transmission causes interference between Wi-Fi signal and cellular in $[t_1 + \alpha T]$. Similarly, in $\forall t_i$, at the start of duty-cycle period, the last overlapping Wi-Fi transmission from previous duty-cycle period can impact cellular users.

Provided $IF_2 << \alpha T$, the impact of Wi-Fi selfish users is minimal and strictly only forces interference issues during the last Wi-Fi transmission. Also, the user x, which arbitrarily overlaps with the cellular ON period, could also occur in the absence of selfish nodes. Hence, the impact of Wi-Fi selfish users on cellular transmissions is strictly limited to the last Wi-Fi transmission in the duty-cycle period and would be minimal provided $IF_2 << \alpha T$. □

This theorem and statement hold as long as the Wi-Fi users follow the duty-cycle and obey the start of Wi-Fi transmission only during the $(1 - \alpha T)$ period. If Wi-Fi devices deviate and initiate communication during αT fraction, the medium access mechanism is violated, and cellular users are exposed to Denial of Service attack.

7.4.2 Interference in Duty-Cycling

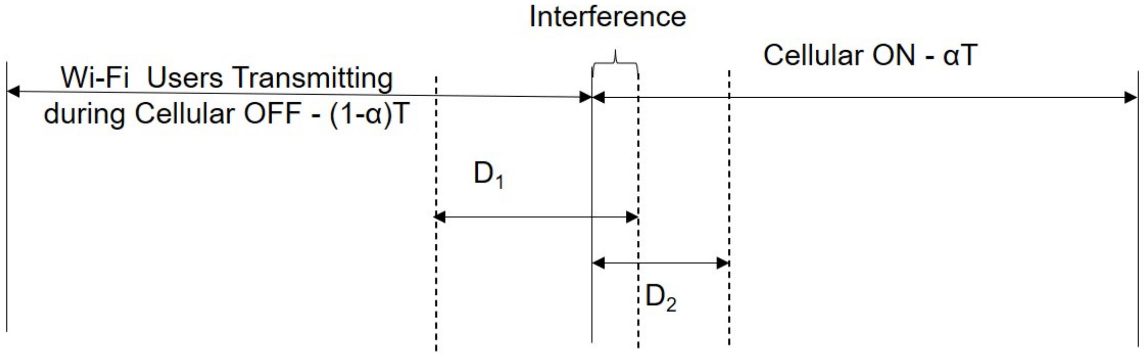


FIGURE 7.3: Interference in STDM-based Duty Cycling

Assuming overall simulation duration is $\digamma$, the expected overlap transmission duration (Given that: $IF_1 << \alpha T$) is,

$$E(I) = \frac{\digamma}{T} \times \frac{IF_1}{2} \tag{7.5}$$

Fig. 7.3 portrays the possibility of interference of Wi-Fi transmissions on cellular signals at the end of the period.

The interference is inversely proportional to duty-cycle period T - more the duty-cycles (T) during simulation duration (F), interference drops. Similarly, the smaller the data transmission duration, the lesser the interference. The total inference in percentages, $I(\%) = \dfrac{E(I) \times 100}{F}$.

7.4.3 Carrier-Sensing Adaptive Transmission (CSAT) with Flexible Duty-Cycling

Here, we study the use case when the cellular operator adjusts its fraction of Cellular being ON in the duty-cycle according to the dynamic Wi-Fi load.

We can classify this methodology into three sub-categories:

a) The overall duty-cycle period is constant. With the increased traffic load in the Wi-Fi, we consider that the cellular network reduces the α fraction linearly.

b) The cellular base station proportionally increases the cellular traffic at the start of the duty-cycle, thereby reducing Wi-Fi transmission comparably. The Duty-cycle period is subsequently increased, and the actual Cellular period is ON is calculated once the cellular signals are on full flow.

c) The overall duty-cycle period is constant. The Cellular ON period is flexible and proportionately decreases with increased Wi-Fi load.

In the first sub-category, when α varies as per Wi-Fi load and the duty-cycle period is constant, the throughput of cellular users are impacted. With a higher likelihood of channel access, selfish users obtain higher channel acquisition in CSMA/CA. Hence, the probability of channel acquisition decreases for legitimate Wi-Fi users. To perform the same m transmissions, the overall time taken for legitimate Wi-Fi users increases, leading to the rise in overall Wi-Fi load or more frequency bands to accommodate selfish nodes.

In the second subcategory of CSAT, the cellular ON period activates after cellular transmissions have reached 100% channel occupancy. And the length of the cellular ON period is not impacted and not altered. Hence, the impact of the Wi-Fi selfish users only exists during the proportional increase of cellular transmissions and decrease of Wi-Fi transmissions. If the proportional strategy is fair, this subcategory achieves an equitable co-existence.

In the third subcategory of CSAT, since the duty-cycle period is fixed, the cellular ON period diminishes. Here, the impact of the Wi-Fi selfish users is again based on the proportional strategy to increase cellular transmissions and decrease Wi-Fi transmissions. If the proportional approach takes time to converge, this use case dilutes to subcategory I, where co-existence is impossible.

If the proportional strategy can converge in negligible time, the third subcategory can achieve an equitable co-existence and is comparable to the second subcategory.

7.5 Simulation

The co-existence setup is built through unlicensed cellular variants in the 5GHz band and Wi-Fi system. MATLAB is used for this realistic discrete-event simulation as the tool also extends support for channel coding schemes. The procedure in our evaluation is pursued as follows:

1. Every LTE-U and Wi-Fi user initializes their request counter to sense the channel or contention window with a random number between [0, CW_{min}]

2. Regular users in both technologies use LBT - CSMA/CA with exponential backoff in LBT. In Duty-cycle, only regular Wi-Fi users apply exponential backoff.

3. When the channel is busy, the backoff counter is incremented by 1. And the new channel sensing time is derived as the $current\, timer$ + $2^{backoff\, counter}$. The contention window is capped to 1024 units.

4. We use continuous event simulation with discrete frames. The total duration of this simulation is 200000 μs, where one basic unit is one μs. DIFS = $34\mu s$, SIFS = $16\mu s$, and the experiment is run with data slots as 1ms and 10ms [21]. Once the user successfully acquires the channel, the user transmits the data in the above slot. The propagation time to send RTS and CTS from the user to the base station or vice versa is computed by dividing distance by the speed of light.

5. Full buffer traffic is considered for the users. The minimum separable unit between the user and the base station is 2m.

We first examine the performance of well-behaved nodes. Here, the location of BS and devices are fixed, and they offset from one another through random grid positioning. We consider the Wi-Fi base station is located at the origin. We perform 1000 Monte Carlo runs. The cellular base station is allocated a position in a Wi-Fi coverage radius with a minimum separable distance in each run.

The experiments are repeated for the total number of nodes in [4,20]. The devices are equally divided between Wi-Fi and cellular base stations. We study the effect of zero, one, and two selfish users in each configuration. When adequate nodes are in the setup, three and four selfish users are considered. This simulation configuration is tabulated in Table 7.2. Throughput is defined as the quantity of data received on a flow divided by the time interval between the initial and latest packet of the flow.

The throughput of the devices is computed as follows. The distance between the base station and the user is used for computing the data rates. Then, we calculate the path loss in dbm for 802.11ax transmission in the 5 GHz band for Wi-Fi. Next, we estimate the SINR and then figure out the data rate by applying the Modulation and Coding Scheme (MCS) for 802.11ax 20 MHz, GI=400ns, and one special stream. Similarly, we compute the path loss of cellular users by combining both Line of Sight (LoS) and non-LoS path loss values as per LoS probability. MCS of a cellular user is determined by the number of bits per PRB per 1 ms for a given SINR.

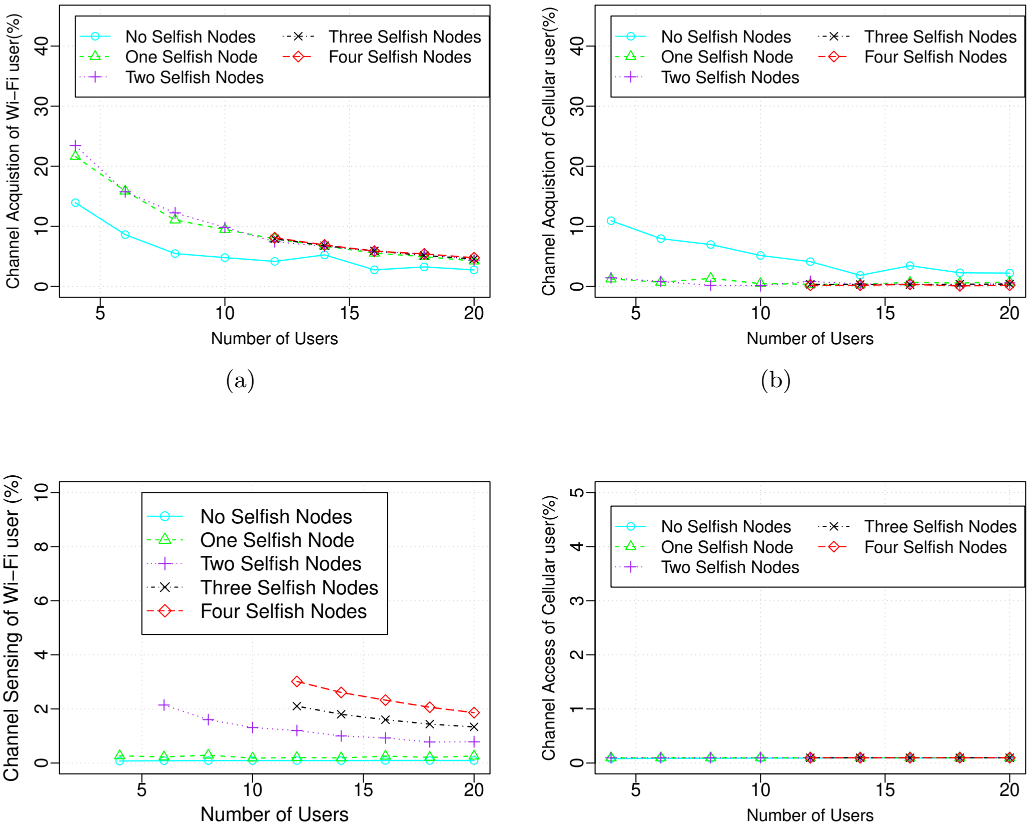
7.6 LBT Results - Effect of selfish users on the co-existing network

In this section, we investigate the effect of selfishness on co-existence. Here the Wi-Fi nodes are programmed to behave selfishly, and we study the impact on the cellular nodes.

7.6.1 Channel acquisition and Channel sensing

In the presence of just legitimate nodes, the Wi-Fi and cellular users obtain near-identical mean channel acquisition and sensing rates. The channel acquisition and sensing rates of legitimate Wi-Fi users are plotted in Fig. 7.5a and 7.5c, and the cellular users in Fig. 7.4b and 7.4d, respectively. We can observe here that the regular users achieve an acquisition ratio of ${}^{Total\,Data\,Occupancy}/_{n}$, where n is the total number of users in the absence of selfish users. The total number of nodes in the setup is denoted on the x-axis.

However, with a selfish user in Wi-Fi, we witness an apparent disparity among the mean acquisition rates of Wi-Fi and cellular users. With an increase in the number of legitimate nodes, although this disparity reduces, we still have a noticeable difference in the channel acquisition rates.



(a)

(b)

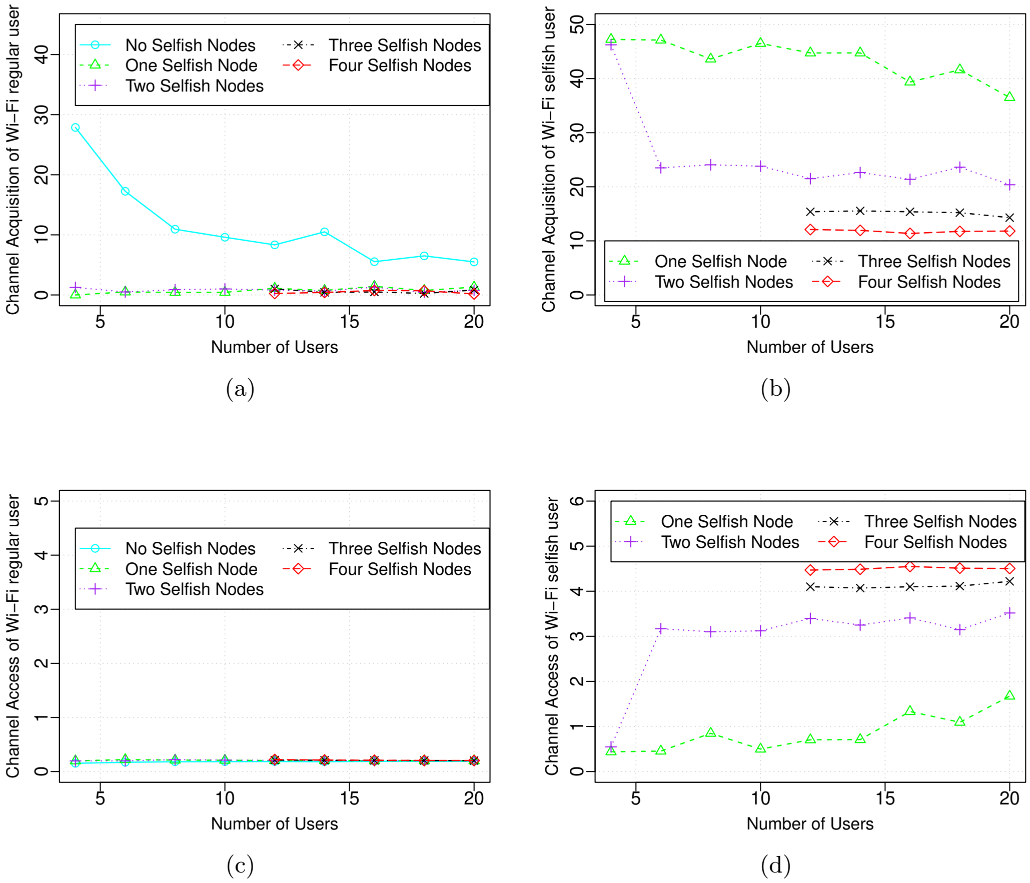
(c)

(d)

Figure 7.4: Effect of Selfish users in Wi-Fi cellular co-existence in LBT

When the number of selfish users increases in the Wi-Fi network, the throughput or acquisition of the co-existing cellular user drops to zero. The plots in Fig. 7.4a and 7.4b demonstrate the side-effect of selfish Wi-Fi users on cellular co-existence through the above observations. Here, on the y-axis, we denote the average channel acquisition rates of Wi-Fi and cellular users, respectively. This channel acquisition for all Wi-Fi users is computed as an average across legitimate and misbehaving nodes.

Fig. 7.4d conveys the average sensing rate of all cellular users. We observe this sensing rate as 0.1 - 0.4% throughout the experiment, and the authors [26] also show this behaviour. The mean channel sensing rates of Wi-Fi users are an increasing function of the number of Wi-Fi selfish users, as observed in Fig. 7.4c. Since the cellular user follows the truncated exponential backoff, the sensing rate is relatively lower than that of Wi-Fi users.



(a)

(b)

(c)

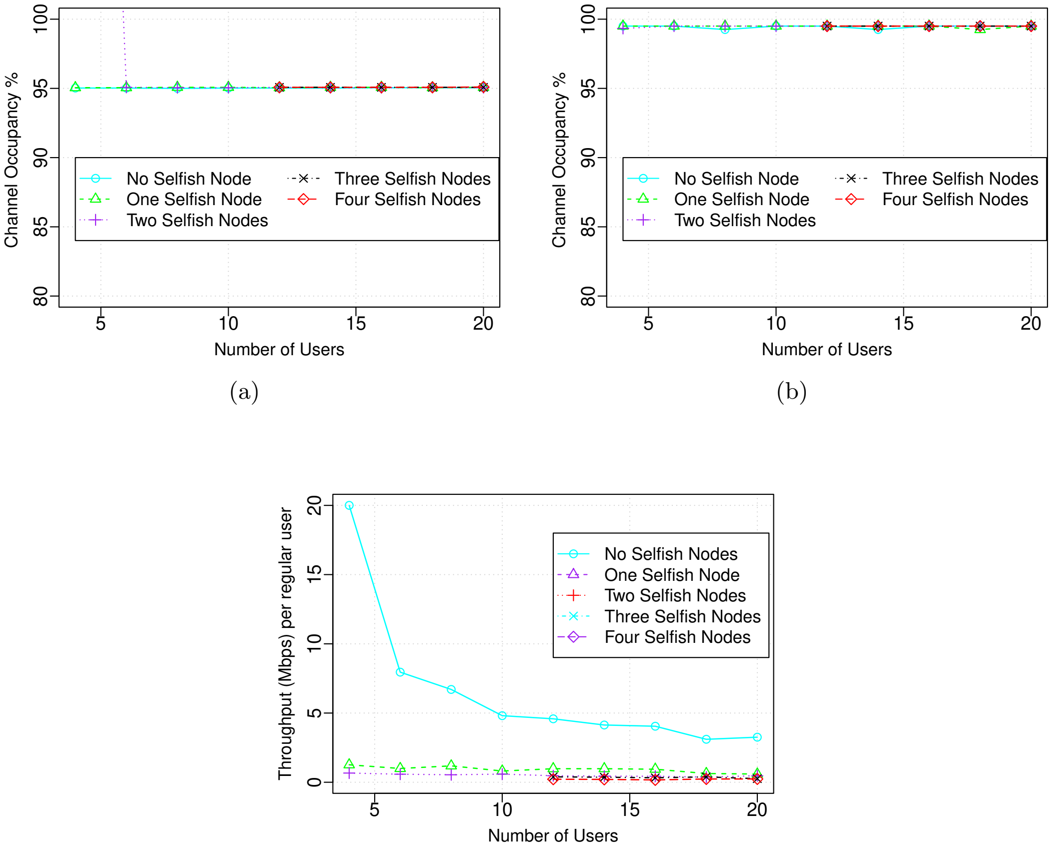
(d)

FIGURE 7.5: Performance of Wi-Fi legitimate and selfish users during co-existence

7.6.2 Impact of selfishness on legitimate Wi-Fi and Cellular nodes

In Fig. 7.5, we study the exploitation of selfish Wi-Fi users and their effect on their counterpart legitimate nodes in the Wi-Fi segment of the co-existing network. In Fig. 7.5b, we observe the exploitation of selfish users, which grabs most of the data transmission opportunities. The maximum channel acquisition rate of a selfish user is realized when there is precisely one selfish user in the system. As the number of selfish users increases in the setup, each selfish user obtains a channel acquisition ratio close to $^{Total\,Data\,Occupancy}/_{n}$, where n is the number of selfish users. In Fig. 7.4 and Fig. 7.5, we consider the data slot to be 1ms.

The channel sensing rate of selfish users is marked in Fig. 7.5d. The sensing probability follows uniform integer distribution with a range $[1, CW_{min}]$. We notice the sensing rate of a selfish user is comparatively less when there are fewer selfish and total users in the system. These users exhibit this behaviour because they transmit the data regularly and do not sense the channel



(a)

(b)

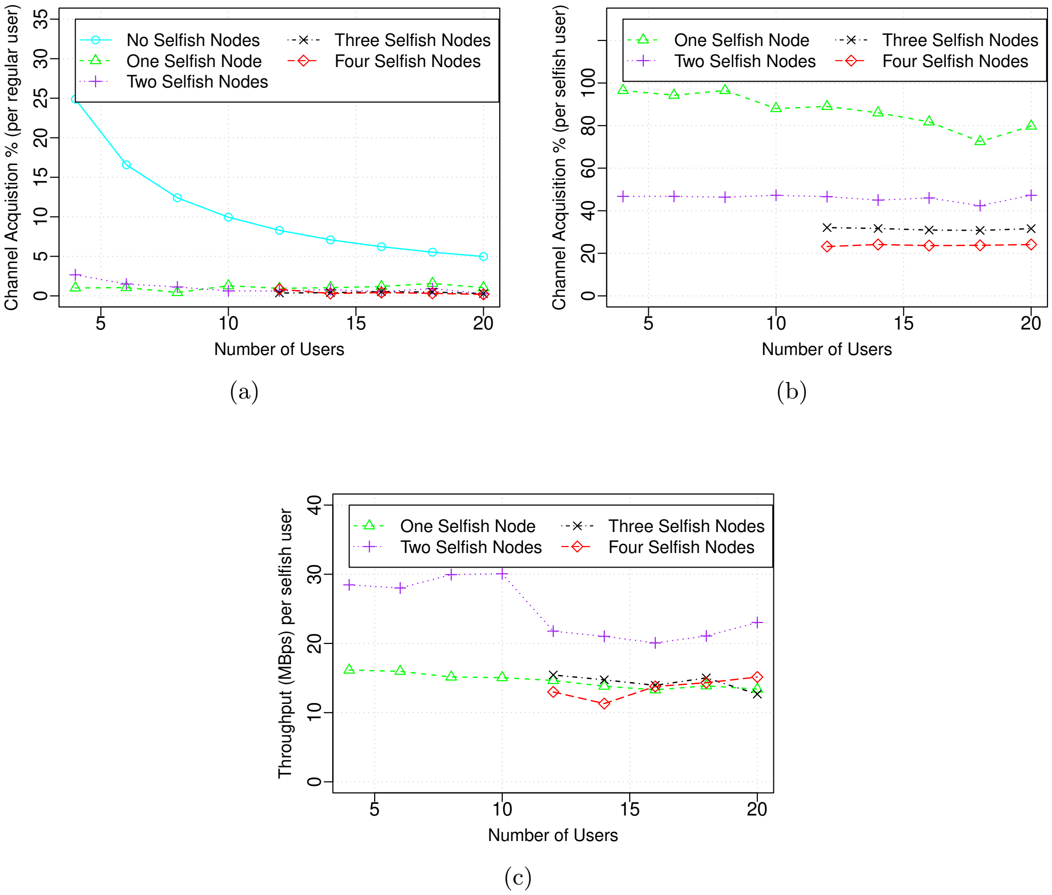
(c)

FIGURE 7.6: Overall Data Transmission Channel Occupancy with Data slots set to (a) 1ms (b) 10ms

during this period. With the increase in the number of selfish and legitimate nodes, we notice that the channel sensing rate of selfish Wi-Fi users converges between 4% to 5% here.

As the number of selfish Wi-Fi users increases, it also reveals a significant channel acquisition dip in regular Wi-Fi users. When the number of total and selfish users increases, the overall acquisition of a regular user tends to zero. In the absence of selfish users, the users have a near-constant sensing rate when the system's total number of users increases. It can be attributed to the exponential backoff mechanism used during channel sensing.

The sensing rate of regular users is always slightly higher when selfishness is present in the system than when it is absent. This observation is due to the deprival of channel occupancy.



(a)

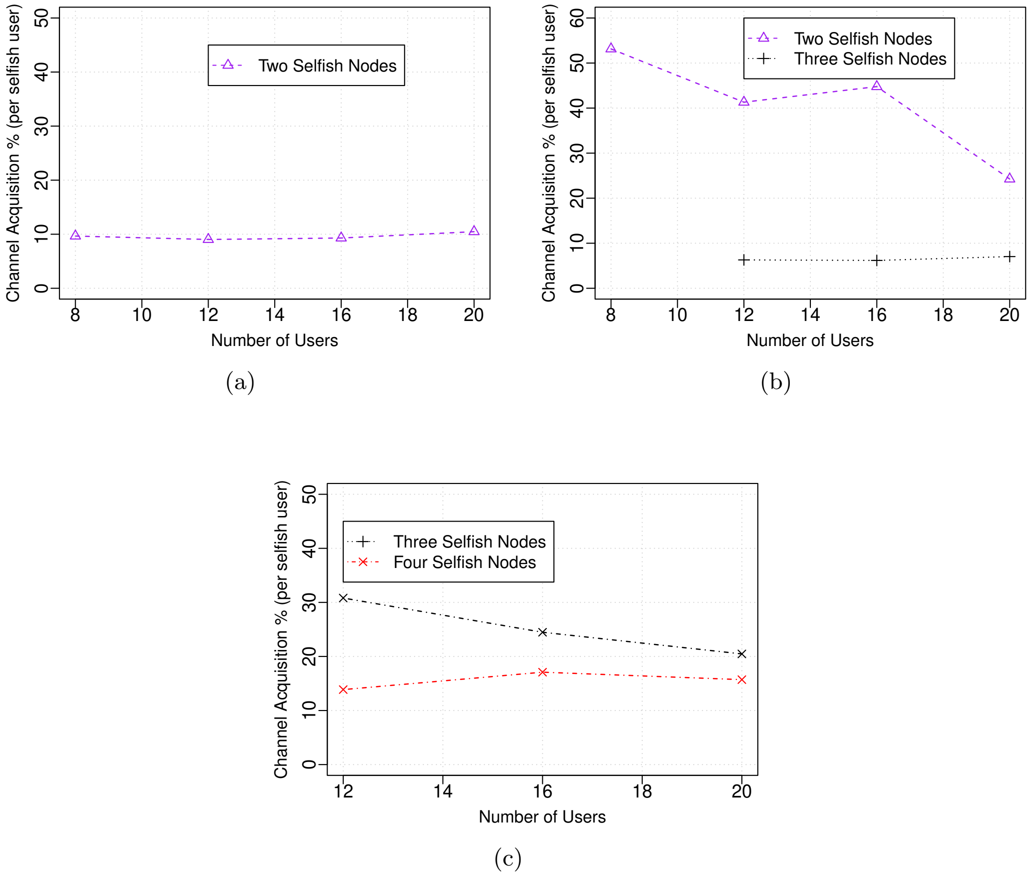
(b)

(c)

Figure 7.7: Performance of regular and selfish user exploitation with Data slots set to 10ms

7.6.3 Data Channel occupancy

Fig. 7.6 depicts the overall data channel occupancy percentages when the data transmission period is 1 ms and 10 ms. The total number of users in the system is denoted on the x-axis. We realize 95% and 99.5% channel utilization for the actual data transmission, respectively. These data slots are configured between 1 ms and 10 ms in the experiments [125]. This configuration range allows optimal utilization as it avoids frequent context switching and does not hold the channel for too long, denying access to other users. The channel acquisition rate of regular and selfish users when the data transmission period is initialized to 10ms is portrayed in Fig. 7.7a and 7.7b. When the number of users is less in the system, the higher data occupancy directly leads to a slight increase in the acquisition rate of these users. However, the rise in acquisition rate becomes negligible when the number of users in the setup increases.



(a)

(b)

(c)

Figure 7.8: Exploitation of Users with different degree of selfishness
a) Channel Sensing - 5% b) Channel Sensing - 7% c) Channel Sensing - 9.5%

7.6.4 Throughput study of legitimate and misbehaving nodes

The mean throughput obtained by legitimate and misbehaving users across both technologies is depicted in Fig. 7.6c and 7.7c.

Initially, these selfish users attain a higher throughput in fewer selfish nodes. However, when there are more selfish users, the throughput of these users rapidly decreases, as shown in Fig. 7.7c. When more selfish users are present, the well-behaved nodes reach near-zero throughputs, the fairness index is zero, and the network may crash. Though equilibrium for selfish nodes can fix the network, the well-behaved user would still suffer. It accomplishes the complete starvation of regular users, as shown in Fig. 7.6c.

7.7 LBT Results - Different Degree of selfishness and backoff patterns

7.7.1 Degree of Selfishness

Fig. 7.8 plots the channel acquisition in percentages for the different degrees of selfishness exhibited. Here, the degrees are differentiated by channel sensing rate. Figures 7.7a, 7.7b, and 7.7c represent channel sensing rates of 5%, 7%, and 9.5%, respectively, during the simulation duration.

We observe from Fig. 7.8a that in the presence of two selfish nodes, a selfish user with a 5% sensing rate can acquire roughly 10% of the channel occupancy. With as many selfish nodes, a selfish user can obtain the data channel for approximately 30 - 50% with a sensing rate of 7%, as shown in Fig. 7.8b. With the increase in the total number of nodes in the system, this channel acquisition drops from 50% to 30%.

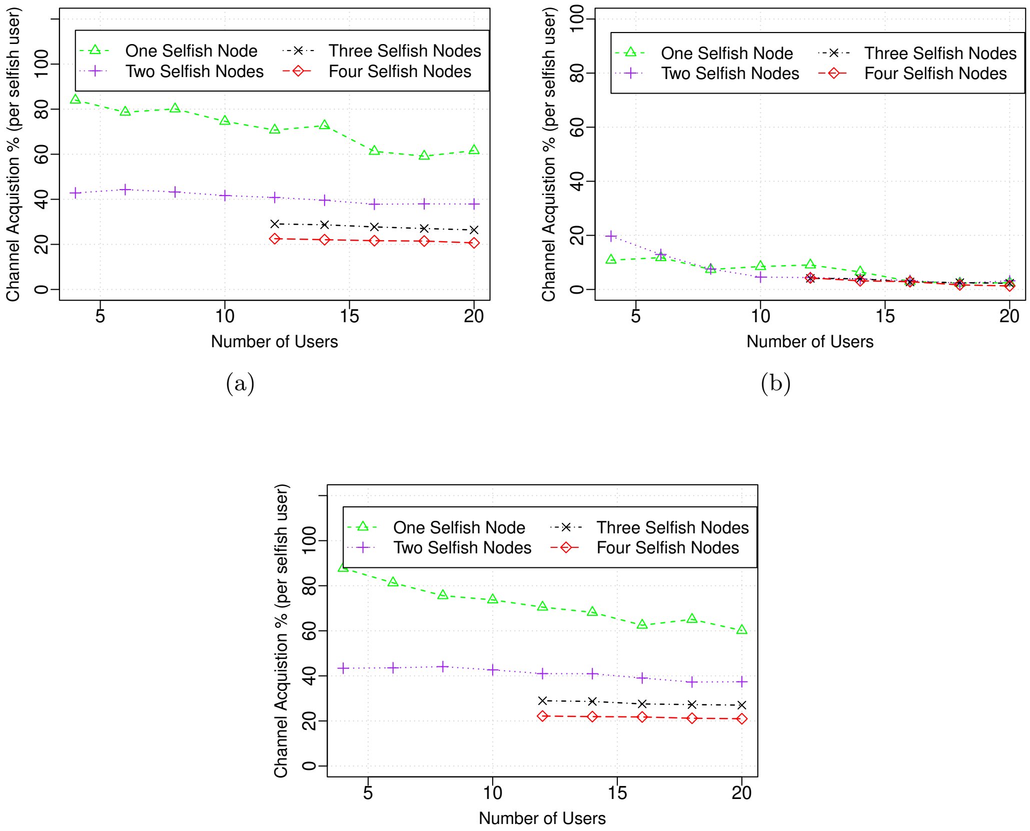
Similarly, with three selfish nodes in the network, a selfish user with a 7% sensing rate acquires only roughly 8% of the data occupancy. But, in a similar setup, when a selfish user exhibits 9.5% carrier sense, it reaches almost 25 - 30% data occupancy when the total number of nodes ranges [4,20].

From the above trends, we can notice how a higher degree of selfishness leads to greater channel exploitation.

We also observe with the increase in the total number of nodes and the number of selfish users, the data occupancy of the selfish user with a specific sensing rate drops linearly from the above figures. To overcome this effect, the selfish user needs to increase its sensing rate to retain the same level of throughput.

7.7.2 Side-Effects of other backoff window patterns

We have applied different backoff patterns for selfish users and observed the behaviour of the nodes in terms of channel sensing and acquisition. We have scrutinized through three distributions: a) Chi-Square Distribution - one-parameter family of curves, which is commonly used in hypothesis testing. b) Normal Distribution - a two-two-parameter family of curves, generally employed as the sum of independent samples from any distribution with finite mean



(a)

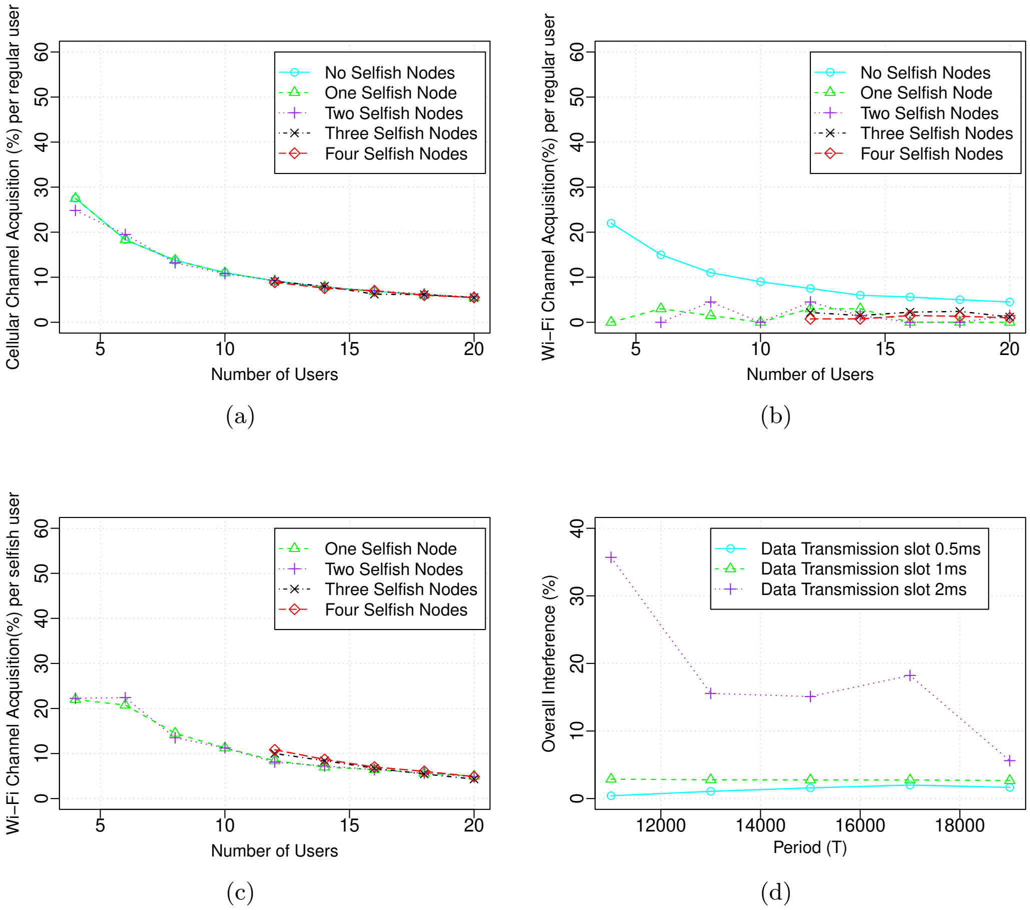
(b)

(c)

FIGURE 7.9: Selfish User Channel Acquisition with Different Backoff distributions a) Chi-Square: $\nu = CW_{min}$ b) Normal: $\mu = CW_{min}, \sigma = \frac{CW_{min}}{2}$ c) Poisson: $\lambda = CW_{MIN}$

and variance, converges to the normal distribution as the sample size reaches infinity. c) Poisson Distribution - a one-parameter family of curves that models the number of times a random event occurs.

In Fig. 7.9, we plot the selfish user channel acquisitions in percentages for the above distributions. We set ν to CW_{min} in Fig. 7.9a, $\mu = CW_{min}$ and $\sigma = CW_{min}/2$ in Fig. 7.9b, and λ to CW_{min} to Fig. 7.9c. We noticed that Chi-Square and Poisson distribution, with its one-parameter setting to a value CW_{min}, can game the setup and extract significant data extraction. The above configurations provide near comparable results as the uniform Integer distribution between [1,CW_{min}] simulated in earlier sub-sections. However, the Normal distribution with mean CW_{min} and standard deviation CW_{min} cannot extract any better than a regular node with this composition. Here, the settings might have to be tweaked by lowering them further.



(a) (b) (c) (d)

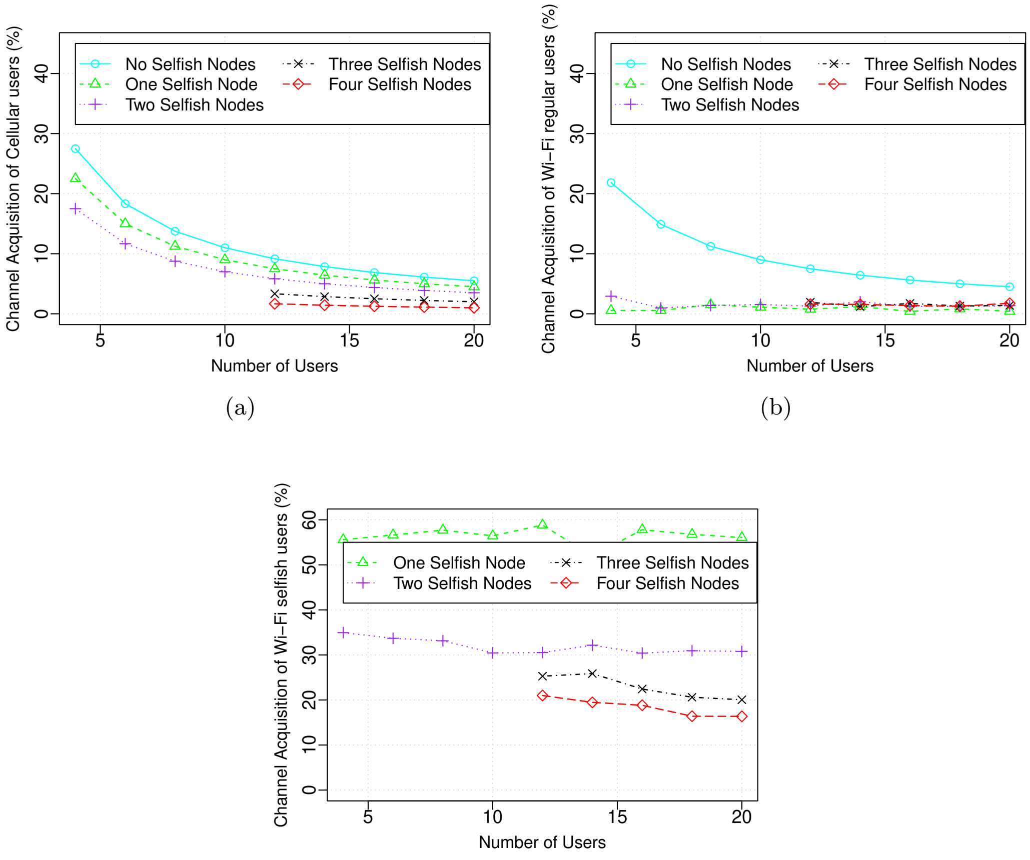
Figure 7.10: Channel Acquisition in STDM-based Duty cycles with $\alpha = 0.5$ and interference

7.8 Duty Cycling Results

7.8.1 Strict Time Domain Multiplexing (STDM) Duty-Cycling

The STDM results are plotted in Fig. 7.10. Here, α is set to 0.5, duty-cycle period T is set to 2×10^4 microseconds, and total duration is 2×10^5 microseconds. We induce misbehaving nodes, which don't obey exponential backoff in Wi-Fi technologies. The channel acquisition of cellular users is presented in Fig. 7.10a. Similarly, regular Wi-Fi and selfish devices are shown in Fig. 7.10b and 7.10c.

We can observe that the channels acquired by legitimate users in cellular technology are proportional to the fraction αT. In Duty-cycle, the average users are shielded by cellular technology. The Wi-Fi selfish ones are unable to penetrate the cellular system. To be precise, the side-effect is limited to the last Wi-Fi transmission in the duty-cycle period, which is quantified in the following sub-section. Here, the cellular base station applies a proportional fair scheduling



(a)

(b)

(c)

FIGURE 7.11: Wi-Fi users Impact on co-existence due to adaptive α as per Wi-Fi load variation

algorithm. Here, it tries to maximize the total throughput of the network during the cellular-ON period while simultaneously allowing all users at least a minimal level of service in accordance with its demanded data rate.

The Wi-Fi users apply Distributed Coordinated Function (DCF), carrier sensing, and exponential backoff. Here, the average Wi-Fi users still suffer in acquiring the channel as the selfish users exploit the network. In the duty-cycle, we could observe that each selfish Wi-Fi user is experiencing a channel acquisition of $^1/_m$ fraction in every duty-cycle period as shown in Fig. 7.10b, m being the number of selfish Wi-Fi users. In Fig. 7.10c, we notice how the regular user's acquisition percentage drops close to zero when the number of selfish nodes increases.

7.8.2 Interference in STDM Duty-Cycling

For the experiments using settings from Table 7.1, the observed average interference is plotted in Fig. 7.10d. The interference mainly occurs at the end of the cellular off period, when Wi-Fi devices transmit. When the cellular-ON period starts, the Wi-Fi device will communicate until its data transmission slots end. Wi-Fi device is unaware of the cellular-ON period, and as per the protocol of Duty-cycle, the cellular device starts transmission. We vary the period T from 11ms to 19ms and plot on the x-axis. The y-axis displays the interference exhibited when the data transmission slots after successfully acquiring Wi-Fi Distributed co-ordinated function is [0.5ms, 1ms, 5ms].

We observe that interference increases with the data transmission duration of Wi-Fi devices. Comparatively, we notice lesser interference when this data transmission duration is set to 0.5ms (lesser value). Similarly, we observe with the increase in the duty-cycle period (T), the interference decreases in Fig. 7.10d. Hence, lower data transmission duration after channel acquisition and a higher duty-cycle period are preferred to achieve minimal interference.

7.8.3 Carrier-Sensing Adaptive Transmission (CSAT) with Flexible Duty-Cycling

In Fig. 7.11, we study the use case when the cellular operator adjusts its part of Cellular being ON in the duty-cycle according to the dynamic Wi-Fi load (the first subcategory). With the increased traffic load in the Wi-Fi, we consider that the cellular network reduces the α fraction linearly. We consider the presence of selfish users in the Wi-Fi network, and Fig. 7.11 plots the effect of cellular users on this co-existence configuration.

We could observe Wi-Fi selfish users having a profound impact on co-existence as they misuse the duty-cycle contract and increase their bandwidth share in the network. As the number of selfish users increases in the network, the α fraction drops to 0.1. This fraction deprives the cellular users of access to the network, leading to a denial of service attack, as shown in Fig. 7.11a. In the above figures, please note that three and four selfish users in the network are considered when at least ten nodes are in the co-existing network.

7.9 Counteraction

There are a few possible selfish behaviours: a) Having smaller CW_{min} and CW_{max}. It results in higher sensing probability. b) Slower increase in CW after the collision. When the range of [1, CW] increases slowly, resulting in better access probability. c) Lesser backoff window values. We discuss three counteraction approaches.

7.9.1 Jamming selfish users

The main idea is to identify the deviating device individually. Here, we assume devices can measure the bandwidth obtained by other users through the broadcast nature of wireless communications. This is possible for legitimate Wi-Fi users, which can deduce the bandwidth acquired by the selfish Wi-Fi user. If a device is measured at a different bandwidth altogether, it deviates.

A simple punishment mode scheme is applied through selective jamming to bring down the bandwidth of the non-cooperative user. For a short period during a selfish player's transmission time, jamming signals are sent by regular users [104]. When a Wi-Fi user is selfish, this punishment has to be applied by another legitimate Wi-Fi user. Across technology, the cellular user can jam a Wi-Fi selfish user only when the legitimate cellular user can comprehend packet headers from the selfish Wi-Fi user and subsequently infer bandwidth details.

We know that the sensing probability $\rho(i)$, for each device i optimizes the overall throughput $R = \{r_1, r_2, ...r_n\}$, where n is the number of users. By choosing system parameter $\rho(i) = \rho_0$, all devices can achieve a unique Nash Equilibrium.

7.9.2 CRISP strategy

Cooperation via Randomized Inclination to Selfish Play (CRISP) [109] is a strategy that applies a limited punishment technique which leads the equilibrium to a Sub-game Perfect NE (SPNE). Regular users could use the deviation detection mechanisms to determine selfish nodes that don't follow CSAT with exponential backoff.

Once detected, other users can enforce a cooperating strategy in the Prisoner's dilemma. It defines punishment and non-punishment modes. Until selfish users are detected, all regular users follow the usual honest approach, i.e., non-punishment (w_h). Once greedy nodes are detected, nodes can apply CRISP strategies through a selfish strategy (w_s) or toggle between w_h and w_s to control selfish users from higher payoff.

Table 7.3: CRISP strategy

State	***Description***	***Strategy***
H	c=0	Follow w_h
S/H	$c > 0$, Toggle mode	$w_h : p_1; w_s : (1 - p_1), p_1 \geq 0$
S/H Phase-up	$c > 0$, c increasing	$w_h : p_2; w_s : (1 - p_2), p_2 \leq p_1$

Let N be the number of devices and c be the number of selfish devices. The successful transmission probability is $P(N, c)$ when there are N users in the network, and c of them are selfish. Let c^* is the threshold where the difference between $P(N, c)$ and $P(N', c + 1)$ is significant for all N, N' and $c \leq c^*$. Although, the legitimate user cannot guess the exact backoff window strategies of other devices. This assumption enables us to state that the legitimate user can infer an increase or presence of selfish players in the network with a certain granularity. When strategy applied in the device i is w_s, it can distinguish the case between $c \leq c^*$ and $c > c^*$.

Similarly, when the strategy applied in the device i is w_h, the user can distinguish between $c = 0$ and $c > 0$. The CRISP strategy is tabulated in Table 7.3. When $c = 0$, all cooperating nodes follow w_h. When $c > 0$ and but remains constant, the regular nodes follow a mixed strategy where w_h is followed with probability $p_1 \geq 0$ and otherwise, w_s.

When $c > 0$ and further increases in the system, the strategy is to increase the likelihood of applying w_s. The probability of using an honest approach is p_2, where $p_2 \leq p_1$.

7.9.3 Counteraction in LBT Technique

When all the players use the CRISP strategies, the equilibrium achieved is fair and Pareto optimal in the LBT technique.

When a Wi-Fi user is selfish, all other legitimate Wi-Fi users can switch to the CRISP strategy. If this selfish user is becoming greedy, all the legitimate users can exhibit a similar approach to achieve Pareto-optimal behaviour.

However, if legitimate cellular users cannot detect this deviation or change in strategy, it would lead to the complete starvation of the cellular network.

In counteraction A, selection jamming is applied to punish selfish users. This method is preferred in a co-existing network as legitimate Wi-Fi can block selfish Wi-Fi users. The co-existence is not at risk even if a legitimate cellular user cannot detect this behaviour. This methodology works as long as the complete Wi-Fi network is not selfish.

In counteraction B, the users switch between punishing and non-punishing modes. These strategies help to achieve a Pareto optimal solution. The selfish users are also encouraged to play CRISP, or they may acquire a lower payoff as other regular users apply CRISP. This methodology is preferred if a legitimate cellular user can detect selfish behaviour by deducing Wi-Fi packet details.

Approach B may be counter-intuitive if a legitimate cellular user cannot detect the Wi-Fi user's misbehaviour. Through CRISP, all legitimate Wi-Fi users would adopt aggressive channel sensing, and cellular users may still follow exponential backoff, leading to starvation. The fixed backoff pattern might be easier to detect and take counteraction against it. So, selfish users need to understand other distributions and use them effectively. By frequently switching backoff patterns, selfish users can deceive counteraction measures, and it will be hard to detect misbehaviour.

7.9.4 Counteraction in Duty-Cycling Technique

Wi-Fi and cellular operators can follow Strict Time Domain Multiplexing (STDM) Duty-Cycling to mitigate selfishness attacks in Duty-cycling. Selfish Wi-Fi users wouldn't be able to intrude into the cellular segment except for the last Wi-Fi transmission in each duty cycle period. This impact is negligible if the duty-cycle period and cellular ON segment are much higher than a specific Wi-Fi transmission. Within the Wi-Fi segment, legitimate Wi-Fi users can follow the CRISP strategy. Here, legitimate Wi-Fi users can apply any deviation mechanism to detect selfish Wi-Fi users not obeying exponential backoff.

When all the players use the CRISP strategies, the equilibrium is fair and Pareto optimal within the Wi-Fi network. In Carrier-Sensing Adaptive Transmission (CSAT), when we set adaptive α, fixed duty-cycle period and is not operator monitored, the above methodology forces $\alpha \rightarrow 0$ and cannot be applied. Hence, CSAT's second and third subcategories with flexible duty-cycling can be preferred. A fair proportional strategy to increase cellular and decrease Wi-Fi transmissions at the start of the cellular ON period can bring the system into equilibrium.

7.10 Chapter Summary

In this chapter, we focus on the fair co-existence of Wi-Fi with 5G cellular users through Listen-Before-Talk or Duty-Cycling approaches to ensure smooth data transmission with relatively less interference. Our work examines the side effects of selfish Wi-Fi users on co-existence and the regular nodes through throughput, channel occupancy, sensing, and acquisition percentages. We also examine different backoff patterns that could be adopted for selfishness and their effect on the network.

Chapter 8

Conclusion

In this research, we analyzed in breadth and depth QoS aspects in the field of Next-Gen networks such as 5G, and we have quantified the results of our work. The conclusion is summarized below:

8.0.1 Classification for Traffic prioritization

A detailed study of QoS attributes around S1AP and IP protocols is identified. The significance of these QoS parameters is described. The work describes using these QoS attributes for traffic classification using machine learning techniques. Data collection involved gathering packets from various sources and inspecting them through Wireshark [111]. The relevant fields are exported to CSV format for feature extraction and cleaning. The QoS attributes are tabulated, and categorical values such as QCI and DSCP are elaborated into more fields. The traffic analysis was performed using supervised techniques like Support Vector Machine, Random Forest, and Gradient Boosting. The results are compared using standard metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The QCI feature is found to be a prime constituent attribute that plays a key role in traffic prioritization. The study suggests that using higher parameters may lead to overfitting. Our study of viable QoS attributes for traffic classification and priority class derivation is comprehensive, and we have demonstrated an applied class-based probabilistic priority scheduling through traffic classification results.

8.0.2 Resource Allocation in Network Slicing

We focused on two important problems in Resource Allocation in Network slicing: Firstly, we analyzed the implementation of QoS-based resource allocation in network slicing for tailored offerings. The proposed approach considers multiple objectives, such as network performance, operating efficiency, and timeliness KPIs. The simulation results show that the proposed algorithm performs well in fulfilling stakeholder's goals and outperforms other candidate algorithms. Secondly, this work focused on the critical challenges of QoS, energy savings, and network slicing that are vital to 5G. We have proposed solutions to these challenges using virtual backbone and cognitive cycle-based approaches for route allocation in network slices. In addition, we have investigated the joint objective of Energy savings and QoS in communication networks, where we achieved EE by reducing the number of nodes employed for routing. Our experimental results show that our proposed solutions are effective and perform better than other standard approaches.

8.0.3 Load balancing in 5G micro infrastructure

5G operators use small cells to densify their networks and ensure seamless connectivity and reliable coverage. However, in a real network setup, some microcells handle most of the traffic while others remain idle, causing overloaded cells to experience intermittent, unstable connectivity and high packet jitter. The one-way load balancing mechanism allows only unidirectional transfer of traffic flows from overloaded to reachable underloaded cells. To address this issue, we proposed an extreme Swap-based Load Balancing (SLB) algorithm between APs, which minimizes the load imbalance at cell edges and improves signal strength issues in the micro infrastructure. Our proposed algorithm improves fairness among APs and signal quality among devices in 5G deployments.

8.0.4 Application-Aware Routing

Implementing application-aware routing in the 5G platform is crucial in meeting stakeholder objectives. The QoS parameters of latency, bandwidth, packet loss, and jitter are holistically examined and monitored in network slicing. The proposed heuristic application-aware routing approach reduced computational time and performed on the above QoS metrics compared to other candidate algorithms.

8.0.5 Cellular Co-existence with Wi-Fi

In the unlicensed spectrum, 5G operators would have interference issues with the Wi-Fi spectrum. The fair co-existence of Wi-Fi and 5G is crucial, and counteraction mechanisms are explored to overcome the impact of selfish nodes on legitimate users. This study analyzed the impact of QoS through metrics such as throughput and channel acquisition due to selfishness under different network configurations in the medium access mechanisms such as duty cycle and Listen-Before-Talk. We recommended network configurations and counteraction mechanisms that promote co-existence and shield legitimate users.

Chapter 9

Summary and Future Work

9.1 Specific Contributions

9.1.1 Swap-based Load Balancing for 5G micro-infrastructure

Our work proposes an extreme Swap-based Load Balancing (SLB) algorithm between APs to minimize the load imbalance at cell edges and improve signal strength issues in the micro infrastructure. Our algorithm aims to ensure fairness among APs with heterogeneous loads and is suitable for both homogeneous and heterogeneous networks applicable to 5G deployments. Our algorithm uses threshold, load per unit capacity, and load imbalance parameters to classify APs. The one-way load balancing mechanism allows only unidirectional transfer of traffic flows from overloaded to reachable underloaded cells. We discuss the rules on swap-based load balancing at overloaded and underloaded cells. A 0-1 Knapsack dynamic programming-based solution is applied to return exchanged load. Parameters such as minimum and maximum loads to be exchanged during SLB are established. To evaluate the effectiveness of our proposed algorithm, we measured the load imbalance reduction percentage of SLB with biasing against other candidates in every time slot. We found that SLB with biasing outperforms state-of-the-art approaches by a factor of 22.24%. Overall, our proposed algorithm reduces the imbalance by a factor of 7.14% compared to the optimal uni-transfer algorithm. Our work provides a unique outlook on traffic distribution through exchanging or re-arranging devices between overloaded and underloaded APs. The proposed algorithm improves fairness among APs and signal quality among devices in 5G deployments.

9.1.2 Traffic Classification and Resource Allocation in Network Slicing

This work makes two specific contributions to resource allocation in network slicing for 5G deployments. Firstly, the study systematically investigates the allocation of network resources for the slices. It considers core parameters of Quality of Experience (QoE) to end-user systems, Network Performance, and Operating efficiency while placing network virtual functions and determining the nodes, links, and resources for assignment to these slices. The proposed approach uses Multi-Attribute Decision Making and Analytical Hierarchical Processing to maximize stakeholder objectives and Enhanced Dinic algorithms to compute the maximum possible flows for network slices. The simulation results show that the proposed algorithm performs well in fulfilling stakeholder's goals and outperforms other candidate algorithms. Secondly, with the objective of joint QoS and energy efficiency, we comprehensively study viable QoS attributes for traffic classification and priority class derivation. The proposed class-based probabilistic priority scheduling algorithm based on ML regression algorithms can be applied to any network. Finally, the proposed virtual backbone and cognitive cycle-based approaches for route allocation in network slices targeting joint QoS and energy efficiency provide an effective solution for energy savings.

9.1.3 Application aware routing

The technical contributions of this work centre on applying application-aware routing principles in the state-of-the-art QoS framework. This involved measuring QoS metrics, mapping, and allocating paths while meeting SLA boundaries. The proposed heuristic application-aware routing approach showed significant reductions in computational time compared to other candidate algorithms, such as HPR and directMIN. The methodology for tracking, measuring, mapping, and monitoring QoS metrics to achieve application-aware routing was detailed. The approach estimated key QoS parameters such as latency, packet loss, and jitter of the data path and computed the notional value of the metrics. It also mapped each data route against the SLA class definition of users and discussed QoS Key Performance Indicator (KPI)-driven routing schemes through the standard algorithm in SDN. Through repeated 1000 Monte Carlo runs, the performance of the selected routes was comparable to single-objective optimized approaches, which was proven by the evaluation of the bandwidth, packet loss, and latency attributes. Overall, implementing application-aware routing would lead to improved network performance and meeting the QoS requirements of the SLA classes.

9.1.4 QoS-driven Task Offloading

The work scrutinizes Multi-access Edge Computing by presenting a QoS-based methodology for offloading requests from mobile devices to resource-rich edge servers. By considering QoS attributes, we ensure that the critical tasks receive the highest priority and are executed promptly. The task categorization is implemented through the ensemble technique and Borda scoring. We also present a Kafka-based queuing system with probabilistic priority-based scheduling that avoids piling tasks in queues while executing these tasks.

9.1.5 Wi-Fi Cellular Co-existence

The study focused on investigating the effect of the selfish behaviour of nodes in cross-technology communications, particularly on the co-existence of Wi-Fi and 5G. The impact of standard MAC-based mechanisms such as Listen-Before-Talk and Duty-Cycling on fair co-existence and QoS of the devices across technologies is analyzed. The study characterizes the backoff mechanisms of many selfish nodes and their effect on the network. It depicts the presence of selfish Wi-Fi users in co-existing networks. To the best of our knowledge, this is the first discussion on the rational cheating of nodes in the purview of cross-technology between Wi-Fi and 5G. Overall, this study provides insights into the impact of selfish users and offers recommendations for a fair co-existence of Wi-Fi and 5G.

9.2 Future Scope of Work

In this section, we detail the next steps or actions that can be taken to build upon and expand the current research findings and contributions. We describe potential opportunities for further exploration and investigation and can help guide researchers toward identifying new research questions and directions.

9.2.1 Swap-Based Load Balancing

The swap-based load balancing technique, especially the two-way extreme load balancing approach, can be highly beneficial for 6G networks and next-generation networks. Here are some reasons why:

- Increased device density: With the advent of 6G and next-generation networks, we can expect to see a massive increase in the number of connected devices. This will put even greater pressure on network resources, making load balancing techniques like swap-based load balancing necessary to ensure that resources are allocated optimally.
- Energy efficiency: Load balancing techniques can also contribute to energy efficiency in 6G and next-generation networks. By balancing the load across network resources, we can reduce the energy consumption of individual resources, which is critical for achieving sustainability goals.

Overall, swap-based load balancing techniques can help ensure the performance, reliability, and efficiency of 6G and next-generation networks, which will play a critical role in supporting the applications and services of the future.

9.2.2 Resource allocation in Network Slicing

Our proposed algorithm addressed a multi-objective constraint optimization problem using a combination of the Dinic algorithm, Multiple Attribute Decision Making, and Analytical Hierarchical Processing. Future work can explore how other optimization techniques, such as game theory or reinforcement learning, can be applied to network slicing problems.

As network traffic and user demands continue to evolve, future work can explore how network slicing algorithms can be made more dynamic to respond to changing network conditions in 6G. This can involve developing techniques for re-juggling resource allocations in stressful traffic demands and dynamic slicing to optimize network performance.

While we have already detailed QoS attributes from S1AP and IP, additional attributes, such as network load, could be considered to enhance QoS.

As with any network solution, it is essential to consider security and privacy implications in network slicing. Future work could explore how to design network slicing to better protect against cyber attacks and safeguard user privacy while still maintaining optimal QoS and energy efficiency.

Combining the problem statements, such as load balancing of devices and resource allocation of network elements, studying them together will add computational complexity. The convergence time will be longer with integer optimization techniques. AI-based optimization techniques can be applied to study the problem together. Also, some system models had to be dealt with independently when exploring a horizontal QoS theme. With researchers zooming into specific research problems with assumptions and constraints, unifying them needs more investigation.

9.2.3 Co-existence between Wi-Fi and 5G

In the current study, the focus was on co-existence between Wi-Fi and 5G. We studied counteraction mechanisms to avoid selfish user attacks and to promote fair co-existence in Listen Before Talk and Duty Cycling approaches. However, in real-world deployments, there may be multiple vendors providing Wi-Fi and 5G solutions. Future work could explore the impact of vendor-specific implementations on coexistence and develop counteraction mechanisms that work across vendors. Similarly, Machine learning algorithms could be used to develop more efficient counteraction mechanisms. For instance, an algorithm could learn from historical data to predict the likelihood of interference and adjust network parameters accordingly.

Game theory can be used in Cellular co-existence with Wi-Fi to analyze the behaviour of selfish users, design strategies to incentive cooperation and achieve a Pareto optimal outcome. Nash Equilibrium and Stackelberg competition are two approaches to modelling competition between selfish users in game theory. We can study the CRISP strategy for counteraction on these standard game theory models.

In Nash Equilibrium, each player chooses their strategy independently, assuming that the other players' strategies are fixed. In this scenario, there may be multiple Nash Equilibria, where no player can unilaterally improve their payoff by changing their strategy. In the current work context, if Wi-Fi users behave selfishly, they may choose their backoff mechanism based on their own payoff. Similarly, cellular operators after detecting deviations in Wi-Fi network, may adapt their parameters to optimize the cellular user's payoff. In this scenario, the Nash Equilibrium is the set of strategies where no player can increase their payoff by unilaterally deviating from their current strategy.

On the other hand, Stackelberg competition models competition between two players where one player (the leader) makes a decision first, and the other player (the follower) observes the leader's decision before making their own decision. In this scenario, the leader's decision can affect the follower's payoff, and the leader chooses their strategy to maximize their own payoff, taking into account the follower's response. In the context of the current work, suppose the selfish Wi-Fi user is the leader, and they choose their backoff parameters to maximize their own payoff. In this scenario, the selfish Wi-Fi user can affect the other Wi-Fi user's payoff and the cellular network. Hence, the cellular operator can detect the deviation and adapt their parameters to tune the channel acquisition of cellular users. Similarly, other Wi-Fi users can choose their backoff parameters to optimize their payoff, taking into account the selfish Wi-Fi user leader.

In future work, we can study CRISP counteraction on both Nash Equilibrium and Stackelberg competition, which can be used to model competition between selfish users in the current work. However, the specific approach depends on the decision-making process and the interdependence of the players' strategies.

Bibliography

[1] Saibharath S, Sudeepta Mishra, and Chittaranjan Hota. “Quality of Service Driven Resource Allocation in Network Slicing”. In: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 2020, pp. 1–5.

[2] Saibharath S, Sudeepta Mishra, and Chittaranjan Hota. “Joint QoS and energy-efficient resource allocation and scheduling in 5G Network Slicing”. In: *Computer Communications* 202 (2023), pp. 110–123. ISSN: 0140-3664.

[3] Saibharath S, Sudeepta Mishra, and Chittaranjan Hota. “QoS Driven Task Offloading and Resource Allocation at Edge Servers in RAN Slicing”. In: *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*. 2021, pp. 1–4.

[4] Saibharath S, Sudeepta Mishra, and Chittaranjan Hota. “Swap-Based Load Balancing for Fairness in Radio Access Networks”. In: *IEEE Wireless Communications Letters* 10.11 (2021), pp. 2412–2416.

[5] Mamta Agiwal, Abhishek Roy, and Navrati Saxena. “Next Generation 5G Wireless Networks: A Comprehensive Survey”. In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 1617–1655.

[6] ETSI TS 101 113 V7.5.0 (2000-07). “Technical Specification, ”Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Service description; Stage 1”. In: *GSM 02.60 version 7.5.0 Release* (1998).

[7] 3G TS 22.105 3.10.0 (2001-10). “Technical Specification. 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service aspects; Services and Service Capabilities”. In: *3G TS 22.105 version 3.10.0 Release* (1999).

[8] 3GPP TS 23.207 V5.8.0 (2003-06). "Technical Specification, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; End-to-end Quality of Service (QoS) concept and architecture". In: *3GPP TS 23.207 version 5.8.0 Release 5* (2003).

[9] A. Kunz et al. "Analysis of the QoS requirements under radio access network planning aspects for GPRS/EDGE and UMTS". In: *2005 International Conference on Wireless Networks, Communications and Mobile Computing*. Vol. 1. 2005, 386–391 vol.1.

[10] 3GPP TS 36.300. "Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2 Release 11". In: *3GPP TS 36.300 version 14.3.0 Release 14* (2017).

[11] 3GPP 23.401. "Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access Release 12". In: *3GPP TS 36.300 version 14.3.0 Release 14* (2015).

[12] *White paper: Quality of Service in 4G/5G networks – Prerequisite for Critical Communications Service*. Tech. rep. Airbus Defence and Space, Airbus, 2017.

[13] Eva Ibarrola et al. "A new global quality of service model: QoXphere". In: *IEEE Communications Magazine* 52.1 (2014), pp. 193–199.

[14] Eva Ibarrola et al. "A Machine Learning Management Model for QoE Enhancement in Next-Generation Wireless Ecosystems". In: *2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*. 2018, pp. 1–8.

[15] 3GPP TS 36.413. "S1 Application Protocol (S1AP)". In: *3GPP TS 36.413 version 12.3.0 Release 12* (2009).

[16] Tim Szigeti Jerome Henry. "Diffserv to QCI Mapping". In: *Network Working Group* (2019).

[17] Yuan Zhang, Ying Wang, and Weidong Zhang. "Energy Efficient Resource Allocation for Heterogeneous Cloud Radio Access Networks With User Cooperation and QoS Guarantees". In: Apr. 2016.

[18] Eva Ibarrola et al. "QOXPHERE: A new QoS framework for future networks". In: *2013 Proceedings of ITU Kaleidoscope: Building Sustainable Communities*. 2013, pp. 1–7.

[19] Akshay Jain, Elena Lopez-Aguilera, and Ilker Demirkol. "Evolutionary 4G/5G Network Architecture Assisted Efficient Handover Signaling". In: *IEEE Access* 7 (2019), pp. 256–283.

[20] Thomas O. Olwal, Karim Djouani, and Anish M. Kurien. "A Survey of Resource Management Toward 5G Radio Access Networks". In: *IEEE Communications Surveys Tutorials* 18.3 (2016), pp. 1656–1686.

[21] Ahlem Saddoud et al. "5G radio resource management approach for multi-traffic IoT communications". In: *Computer Networks* 166 (2020), p. 106936. ISSN: 1389-1286.

[22] Vitaly Petrov et al. "Achieving End-to-End Reliability of Mission-Critical Traffic in Softwarized 5G Networks". In: *IEEE Journal on Selected Areas in Communications* 36.3 (2018), pp. 485–501.

[23] Xenofon Foukas et al. "Network Slicing in 5G: Survey and Challenges". In: *IEEE Communications Magazine* 55.5 (2017), pp. 94–100.

[24] Alexandros Kaloxylos. "A Survey and an Analysis of Network Slicing in 5G Networks". In: *IEEE Communications Standards Magazine* 2.1 (2018), pp. 60–65.

[25] Peter Rost et al. "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks". In: *IEEE Communications Magazine* 55.5 (2017), pp. 72–79.

[26] Bruno Chatras, U Steve Tsang Kwong, and Nicolas Bihannic. "NFV enabling network slicing for 5G". In: *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*. 2017, pp. 219–225.

[27] "Network Slicing for 5G Networks". In: *5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management.* 2018, pp. 327–370.

[28] Taewhan Yoo. "Network slicing architecture for 5G network". In: *2016 International Conference on Information and Communication Technology Convergence (ICTC)*. 2016, pp. 1010–1014.

[29] Malla Reddy Sama et al. "Service-Based Slice Selection Function for 5G". In: *2016 IEEE Global Communications Conference (GLOBECOM)*. 2016, pp. 1–6.

[30] Cesar A. Sierra Franco, Jose Roberto B. de Marca, and Glaucio L. Siqueira. "A Cognitive and Cooperative SON Framework for 5G Mobile Radio Access Networks". In: *2016 IEEE Globecom Workshops (GC Wkshps)*. 2016, pp. 1–6.

[31] Ved P. Kafle et al. "Consideration On Automation of 5G Network Slicing with Machine Learning". In: *2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*. 2018, pp. 1–8.

[32] Emmanouil Pateromichelakis et al. "Slice-Tailored Joint Path Selection Scheduling in mm-Wave Small Cell Dense Networks". In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 2017, pp. 1–6.

[33] Alcardo Alex Barakabitze et al. "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges". In: *Computer Networks* 167 (2020), p. 106984.

[34] Alireza Babaei, Jennifer Andreoli-Fang, and Belal Hamzeh. "Wi-Fi coexistence with duty cycled LTE-U". In: *Wireless Communications and Mobile Computing* 2017 (Jan. 2017).

[35] Gabriel Brown. *White paper: Exploring 5G New Radio: Use Cases, Capabilities Timeline*. Tech. rep. Qualcomm, 2016.

[36] İlhan Demirci and Ömer Korçak. "Cell breathing algorithms for load balancing in Wi-Fi/cellular heterogeneous networks". In: *Computer Networks* 134 (2018), pp. 140–151. ISSN: 1389-1286.

[37] Yoann Desmouceaux et al. "6LB: Scalable and Application-Aware Load Balancing with Segment Routing". In: *IEEE/ACM Transactions on Networking* 26.2 (2018), pp. 819–834.

[38] Li-Chia Cheng, Kuochen Wang, and Yi-Huai Hsu. "Application-aware Routing Scheme for SDN-based cloud datacenters". In: *2015 Seventh International Conference on Ubiquitous and Future Networks*. 2015, pp. 820–825.

[39] *White paper: Radio Aware Routing: Enabling Communications on the Move*. Tech. rep. Cisco Systems, 2022.

[40] Spyridon Vassilaras et al. "The Algorithmic Aspects of Network Slicing". In: *IEEE Communications Magazine* 55.8 (2017), pp. 112–119.

[41] Andreas Fischer et al. "Virtual Network Embedding: A Survey". In: *IEEE Communications Surveys Tutorials* 15.4 (2013), pp. 1888–1906.

[42] Boaz Patt-Shamir Guy Even Moti Medina. "On-Line Path Computation and Function Placement in SDNs". In: *Stabilization, Safety, and Security of Distributed Systems, Springer International Publishing* (2016).

[43] Mohd. Noor ; Seo Young Min ; Lee Young Ki ; Kang Sang Bum ; Choi Sun Woong ; Jang Yeong Min Zaman Chowdhury Mostafa ; Islam. "Characterizing QoS Parameters and Application of Soft-QoS Scheme for 3G Wireless Networks". In: *International Conference on Advanced Communication Technology* (2018).

[44] Phuong Luu Vo et al. "Slicing the Edge: Resource Allocation for RAN Network Slicing". In: *IEEE Wireless Communications Letters* 7.6 (2018), pp. 970–973.

[45] Safdar Nawaz Khan Marwat et al. "Data aggregation of mobile M2M traffic in relay enhanced LTE-A networks". In: *EURASIP Journal on Wireless Communications and Networking* 2016 (Apr. 2016).

[46] C. Schurgers, O. Aberthorne, and M.B. Srivastava. "Modulation scaling for energy aware communication systems". In: *ISLPED'01: Proceedings of the 2001 International Symposium on Low Power Electronics and Design (IEEE Cat. No.01TH8581).* 2001, pp. 96–99.

[47] Masoomeh Torabzadeh. "Green Massive MIMO Scheduling for 5G Traffic with Fairness". In: *Proceedings of the 2017 2nd International Conference on Communication and Information Systems.* New York, NY, USA: Association for Computing Machinery, 2017, 37–42.

[48] Uzma Siddique et al. "Channel-Access-Aware User Association With Interference Coordination in Two-Tier Downlink Cellular Networks". In: *IEEE Transactions on Vehicular Technology* 65.7 (2016), pp. 5579–5594.

[49] Thomas Otieno Olwal et al. "Joint queue-perturbed and weakly coupled power control for wireless backbone networks". In: *International Journal of Applied Mathematics and Computer Science* 22.3 (2012), pp. 749–764.

[50] Guoqing Liu et al. "Interference Alignment for Partially Connected Downlink MIMO Heterogeneous Networks". In: *IEEE Transactions on Communications* 63.2 (2015), pp. 551–564.

[51] Bosheng Zhou, A. Marshall, and Tsung-Han Lee. "An energy-aware virtual backbone tree for wireless sensor networks". In: *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.* Vol. 1. 2005.

[52] Guanding Yu et al. "Multi-Objective Energy-Efficient Resource Allocation for Multi-RAT Heterogeneous Networks". In: *IEEE Journal on Selected Areas in Communications* 33.10 (2015), pp. 2118–2127.

[53] Tao Zhao et al. "Energy-delay tradeoffs of virtual base stations with a computational-resource-aware energy consumption model". In: *2014 IEEE International Conference on Communication Systems*, pp. 26–30.

[54] Weixin Tian Cheng Wang Lili Ma Weihong Fu Qingliang Kong. "A QoS-Aware Scheduling Algorithm Based on Service Type for LTE Downlink". In: 2013.

[55] Joseph S. Gomes et al. "Scheduling Algorithms For Policy Driven QoS Support in HSDPA Networks". In: *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring.* 2007, pp. 799–803.

[56] Pablo Ameigeiras et al. "3GPP QoS-based scheduling framework for LTE". In: *EURASIP Journal on Wireless Communications and Networking* 2016 (Mar. 2016).

[57] "Randomized fault-tolerant virtual backbone tree to improve the lifetime of wireless sensor networks". In: *Computers Electrical Engineering* 48 (2015), pp. 286–297.

[58] Weihong Fu et al. "A QoS-Aware Scheduling Algorithm Based on Service Type for LTE Downlink". In: *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).* Atlantis Press, 2013/03, pp. 2201–2205.

[59] Huayue Wu et al. "QoS-based scheduling algorithm for downlink multi-traffic in ultra high throughput WLAN". In: *2012 IEEE Second International Conference on Consumer Electronics - Berlin (ICCE-Berlin).* 2012, pp. 163–167.

[60] W. Ajib, D. Haccoun, and J.-F. Frigon. "An efficient QoS-based scheduling algorithm for MIMO wireless systems". In: *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.* Vol. 3. 2005, pp. 1579–1583.

[61] Kok-Lim Alvin Yau et al. "Cognition-Inspired 5G Cellular Networks: A Review and the Road Ahead". In: *IEEE Access* 6 (2018), pp. 35072–35090.

[62] Ioan-Sorin Comsa, Antonio De-Domenico, and Dimitri Ktenas. "QoS-Driven Scheduling in 5G Radio Access Networks - A Reinforcement Learning Approach". In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference.* 2017, pp. 1–7.

[63] Fannia Pacheco et al. "Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey". In: *IEEE Communications Surveys Tutorials* 21.2 (2019), pp. 1988–2014.

[64] "Extending labeled mobile network traffic data by three levels traffic identification fusion". In: *Future Generation Computer Systems* 88 (2018), pp. 453–466.

[65] Andres J. Gonzalez et al. "The Isolation Concept in the 5G Network Slicing". In: *2020 European Conference on Networks and Communications (EuCNC)*. 2020, pp. 12–16.

[66] Chia-Yu Chang, Navid Nikaein, and Thrasyvoulos Spyropoulos. "Radio access network resource slicing for flexible service execution". In: *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2018, pp. 668–673.

[67] Murat Alanyali and Bruce Hajek. "On Simple Algorithms for Dynamic Load Balancing". In: *Mathematics of Operations Research*. 1995, pp. 230–238.

[68] Yousefi S. Bagherzadeh J Farzi S. "Zone-based load balancing in two-tier heterogeneous cellular networks: a game theoretic approach". In: *Telecommun Syst 70*. 2019, pp. 105–121.

[69] Jie Cui et al. "A Load-Balancing Mechanism for Distributed SDN Control Plane Using Response Time". In: *IEEE Transactions on Network and Service Management* 15.4 (2018), pp. 1197–1206.

[70] Kshira Sagar Sahoo et al. "ESMLB: Efficient Switch Migration-Based Load Balancing for Multicontroller SDN in IoT". In: *IEEE Internet of Things Journal* 7.7 (2020), pp. 5852–5860.

[71] Yigal Bejerano, Seung-Jae Han, and Li Li. "Fairness and Load Balancing in Wireless LANs Using Association Control". In: *IEEE/ACM Transactions on Networking* 15.3 (2007), pp. 560–573.

[72] Kareem Attiah et al. "Load Balancing in Cellular Networks: A Reinforcement Learning Approach". In: *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 2020, pp. 1–6.

[73] Petros Sioutis George Agapiou Christos Tsirakis Panagiotis Matzoros. "Load balancing in 5G Networks". In: *MATEC Web of Conferences*. Vol. 125. 2017.

[74] Gen Liang and Minyi Chen. "Access Selection Algorithm Based on Traffic Distribution with Delay Optimization in Heterogeneous Wireless Networks". In: *2020 International Wireless Communications and Mobile Computing (IWCMC)*. 2020, pp. 839–844.

[75] Eduard Garcia, Rafael Vidal, and Josep Paradells. "Cooperative load balancing in IEEE 802.11 networks with cell breathing". In: *2008 IEEE Symposium on Computers and Communications*. 2008, pp. 1133–1140.

[76] Masahiro Kawada, Morihiko Tamai, and Keiichi Yasumoto. "A trigger-based dynamic load balancing method for WLANs using virtualized network interfaces". In: *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. 2013, pp. 1091–1096.

[77] Yun Li et al. “A Novel Load Balancing Algorithm in IEEE 802.11 Wireless LANs with Cell Breathing”. In: *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. 2009, pp. 1–4.

[78] Qiaoyang Ye et al. “User Association for Load Balancing in Heterogeneous Cellular Networks”. In: *IEEE Transactions on Wireless Communications* 12.6 (2013), pp. 2706–2716.

[79] Ryuji Asakura et al. “A Traffic Distribution System Among Multiple Terminals Using MPTCP in Multihomed Network Environment”. In: *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 1. 2019, pp. 900–903.

[80] Kiran A Jadhav, Mohammed Moin Mulla, and D. G Narayan. “An Efficient Load Balancing Mechanism in Software Defined Networks”. In: *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*. 2020, pp. 116–122.

[81] John Cartmell, John McNally, and Bartosz Balazinski. “Local selected IP Traffic Offload Reducing traffic congestion within the mobile core network”. In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. 2013, pp. 809–812.

[82] Dharmendra Sharma Ibrahim Elgendi Kumudu S. Munasinghe and Abbas Jamalipour. “Traffic offloading techniques for 5G cellular: a three-tiered SDN architecture”. In: vol. 71. 2016, 583–593.

[83] “Efficient Routing for Traffic Offloading in Software-defined Network”. In: *Procedia Computer Science* 34 (2014), pp. 674–679.

[84] Hyame Assem Alameddine et al. “Dynamic Task Offloading and Scheduling for Low-Latency IoT Services in Multi-Access Edge Computing”. In: *IEEE Journal on Selected Areas in Communications* 37.3 (2019), pp. 668–682.

[85] Ben Rejeb S. Choukair Z Chabbouh O. “A strategy for joint service offloading and scheduling in heterogeneous cloud radio access networks”. In: 2017.

[86] Vincenzo Sciancalepore et al. “Mobile traffic forecasting for maximizing 5G network slicing resource utilization”. In: *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. 2017, pp. 1–9.

[87] Slavica Tomovic, Neeli Prasad, and Igor Radusinovic. “SDN control framework for QoS provisioning”. In: *2014 22nd Telecommunications Forum Telfor (TELFOR)*. 2014, pp. 111–114.

[88] Diego Leonel Cadette Dutra et al. “Ensuring End-to-End QoS Based on Multi-Paths Routing Using SDN Technology”. In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 2017, pp. 1–6.

[89] Alexandre T. Oliveira et al. “SDN-Based Architecture for Providing QoS to High Performance Distributed Applications”. In: *2018 IEEE Symposium on Computers and Communications (ISCC)*. 2018, pp. 00602–00607.

[90] Tsung-Han Lei et al. “Deploying QoS-assured service function chains with stochastic prediction models on VNF latency”. In: *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. 2017, pp. 1–6.

[91] “QoS improvisation of delay sensitive communication using SDN based multipath routing for medical applications”. In: *Future Generation Computer Systems* 93 (2019), pp. 256–265.

[92] Mohammad Rezaee and Mohammad Hossein Yaghmaee Moghaddam. “SDN-Based Quality of Service Networking for Wide Area Measurement System”. In: *IEEE Transactions on Industrial Informatics* 16.5 (2020), pp. 3018–3028.

[93] Dejene Boru Oljira et al. “A model for QoS-aware VNF placement and provisioning”. In: *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. 2017, pp. 1–7.

[94] Miloud Bagaa et al. “On SDN-Driven Network Optimization and QoS Aware Routing Using Multiple Paths”. In: *IEEE Transactions on Wireless Communications* 19.7 (2020), pp. 4700–4714.

[95] J. Babiarz J. Chan and F. Notel Baker. *White paper: Aggregation of Diffserv Service Classes*. Tech. rep. Network Working Group,Internet Engineering Task Force (IETF), 2018.

[96] Joonseok Park, Jeseung Hwang, and Keunhyuk Yeom. “NSAF: An Approach for Ensuring Application-Aware Routing Based on Network QoS of Applications in SDN”. In: *Mobile Information Systems* 2019 (Apr. 2019), pp. 1–16.

[97] “Latency and energy-aware provisioning of network slices in cloud networks”. In: *Computer Communications* 157 (2020), pp. 1–19.

[98] Zhaogang Shu and Tarik Taleb. “A Novel QoS Framework for Network Slicing in 5G and Beyond Networks Based on SDN and NFV”. In: *IEEE Network* 34.3 (2020), pp. 256–263.

[99] Yayu Gao. “LTE-LAA and WiFi in 5G NR Unlicensed: Fairness, Optimization and Win-Win Solution”. In: *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation*. 2019, pp. 1638–1643.

[100] Piotr Gawlowicz et al. “Punched Cards over the Air: Cross-Technology Communication Between LTE-U/LAA and WiFi”. In: *2020 IEEE 21st International Symposium on ”A World of Wireless, Mobile and Multimedia Networks” (WoWMoM)*. 2020, pp. 297–306.

[101] Vanlin Sathya et al. “Wi-Fi/LTE-U Coexistence: Real-Time Issues and Solutions”. In: *IEEE Access* 8 (2020), pp. 9221–9234.

[102] Rony Kumer Saha. “Coexistence of Cellular and IEEE 802.11 Technologies in Unlicensed Spectrum Bands -A Survey”. In: *IEEE Open Journal of the Communications Society* 2 (2021), pp. 1996–2028.

[103] Ayesha Hasan and Bilal Muhammad Khan. “Coexistence Management in Wireless Networks-A Survey”. In: *IEEE Access* 10 (2022), pp. 38600–38624.

[104] Mario Cagalj et al. “On selfish behavior in CSMA/CA networks”. In: vol. 4. Apr. 2005, 2513 –2524 vol. 4. ISBN: 0-7803-8968-9.

[105] Mohammad Esmalifalak et al. “Detecting Stealthy False Data Injection Using Machine Learning in Smart Grid”. In: *IEEE Systems Journal* 11.3 (2017), pp. 1644–1652.

[106] P. Kyasanur and N.H. Vaidya. “Selfish MAC layer misbehavior in wireless networks”. In: *IEEE Transactions on Mobile Computing* 4.5 (2005), pp. 502–516.

[107] Kyung-Joon Park et al. “Malicious or Selfish? Analysis of Carrier Sense Misbehavior in IEEE 802.11 WLAN”. In: *Quality of Service in Heterogeneous Networks*. Ed. by Novella Bartolini et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 351–362.

[108] Zhuo Lu, Wenye Wang, and Cliff Wang. “On Order Gain of Backoff Misbehaving Nodes in CSMA/CA-based Wireless Networks”. In: *2010 Proceedings IEEE INFOCOM*. 2010, pp. 1–9.

[109] Jerzy Konorski. “A Game-Theoretic Study of CSMA/CA Under a Backoff Attack”. In: *IEEE/ACM Transactions on Networking* 14.6 (2006), pp. 1167–1178.

[110] A. Lopez Toledo, T. Vercauteren, and Xiaodong Wang. “Adaptive Optimization of IEEE 802.11 DCF Based on Bayesian Estimation of the Number of Competing Terminals”. In: *IEEE Transactions on Mobile Computing* 5.9 (2006), pp. 1283–1296.

[111] S. Saibharath. *s1ap QoS Dataset.* Tech. rep. Mendeley Data, V2, 2021.

[112] 3GPP$_TS_2$3.203. *Technical Specification: Policy and charging control architecture v16.0.* Tech. rep. 3GPP TS 23.203 v10.2.0., 2010.

[113] 3GPP$_TS_2$3.203 $-$ *h*20. *Technical Specification: Policy and charging control architecture v23.20.* Tech. rep. 3GPP TS 23.203 v23203-h20., 2021.

[114] Xueyuan Wang, Esma Turgut, and M. Cenk Gursoy. "Coverage in Downlink Heterogeneous mmWave Cellular Networks With User-Centric Small Cell Deployment". In: *IEEE Transactions on Vehicular Technology* 68.4 (2019), pp. 3513–3533.

[115] Ramon Fontes et al. "Mininet-WiFi: Emulating Software-Defined Wireless Networks". In: *2nd International Workshop on Management of SDN and NFV Systems, 2015(ManSDN/NFV 2015).* Barcelona, Spain, Nov. 2015.

[116] Darijo Raca et al. "Beyond Throughput: A 4G LTE Dataset with Channel and Context Metrics". In: Association for Computing Machinery, 2018, 460–465. ISBN: 9781450351928.

[117] Darijo et al. Raca. "Beyond throughput: the next Generation a 5G dataset with channel and context metrics". In: Association for Computing Machinery, 2020.

[118] Lusheng Wang and Geng-Sheng G.S. Kuo. "Mathematical Modeling for Network Selection in Heterogeneous Wireless Networks — A Tutorial". In: *IEEE Communications Surveys Tutorials* 15.1 (2013), pp. 271–292.

[119] Manar Jammal et al. "Software defined networking: State of the art and research challenges". In: *Computer Networks* 72 (2014), pp. 74–98. ISSN: 1389-1286.

[120] Tasher Ali Sheikh, Joyatri Bora, and Md Anwar Hussain. "Massive MIMO system lower bound spectral efficiency analysis with precoding and perfect CSI". In: *Digital Communications and Networks* 7.3 (2021), pp. 342–351. ISSN: 2352-8648.

[121] Charitha Madapatha et al. "On Integrated Access and Backhaul Networks: Current Status and Potentials". In: *IEEE Open Journal of the Communications Society* 1 (2020), pp. 1374–1389.

[122] Haiping Li et al. "Study on Throughput of Network with Selfish Nodes Based on IEEE 802.15.4". In: *Advanced Research on Computer Education, Simulation and Modeling.* Ed. by Song Lin and Xiong Huang. 2011.

[123] Anand M. Baswade et al. “LTE-U and Wi-Fi hidden terminal problem: How serious is it for deployment consideration?” In: *2018 10th International Conference on Communication Systems Networks (COMSNETS)*. 2018, pp. 33–40.

[124] G. Bianchi. “Performance analysis of the IEEE 802.11 distributed coordination function”. In: *IEEE Journal on Selected Areas in Communications* 18 (3 2000), pp. 535–547.

[125] Xuyu Wang, Shiwen Mao, and Michelle X. Gong. “A SURVEY OF LTE WI-FI COEXISTENCE IN UNLICENSED BANDS”. In: 20.3 (2017), 17–23.

Appendix A

List of Publications

A.1 Published Journals

- Saibharath S, Sudeepta Mishra, and Chittaranjan Hota, "Swap-Based Load Balancing for Fairness in Radio Access Networks, " in *IEEE Wireless Communications Letters (WCL)*, IEEE, Vol. 10 (11), Nov 2021, pp. 2412-2416. (doi: 10.1109/LWC.2021.3101983)

- Saibharath S, Sudeepta Mishra, and Chittaranjan Hota, "Joint QoS and Energy-Efficient Resource Allocation and Scheduling in 5G Network Slicing, " in *Computer Communications, Elsevier Journal*, Vol. 202, pp. 110-123, 2023. (doi: 10.1016/j.comcom.2023.02.009)

A.2 Published Conferences

- Saibharath S, Sudeepta Mishra, and Chittaranjan Hota, "Quality of Service Driven Resource Allocation in Network Slicing, " in 91st *Vehicular Technology Conference (VTC 2020-Spring)*, May 2020.

- Saibharath S, Sudeepta Mishra, and Chittaranjan Hota, "QoS Driven Task Offloading and Resource Allocation at Edge Servers in RAN Slicing," in 18th *IEEE Consumer Communications and Networking Conference (IEEE CCNC 2021)*, Jan 2021.

- Saibharath S, Sudeepta Mishra, Gazal Arora, Anamika Sharma, and Chittaranjan Hota, "Selfish Users in Listen-Before-Talk Co-existence of Cellular-Wi-Fi Networks: Counteraction Methods," in *International Workshop on Resource Allocation and Cooperation in Wireless*

Networks (RAWNET) workshop held at 21st International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt), August 2023.

A.3 Manuscripts under Review

- Saibharath S, Sudeepta Mishra, Gazal Arora, Anamika Sharma and Chittaranjan Hota, "Mitigating Selfish User Impact on Wi-Fi and Cellular Co-existence through Duty-cycling."
- Saibharath S, Sudeepta Mishra, and Chittaranjan Hota, "Application-aware QoS based routing for 5G Network Slicing."

Appendix B

Biographical Sketch

B.1 Candidate Biography

Saibharath is pursuing his Ph.D. in the Department of Computer Science and Information Systems at Birla Institute of Science and Technology, Pilani, Hyderabad Campus. He completed his Bachelor of Engineering (BE) in Computer Science from Madras Institute of Technology, Anna University, and Master of Engineering (ME) in Computer Science from BITS Pilani, Hyderabad Campus. His main areas of research are distributed systems and computer networks.

B.2 Supervisors Biography

Dr. Sudeepta Mishra is an Assistant Professor in the Department of Computer Science and Engineering at IIT Ropar since Feb 2020. He earned his B.E. degree in Computer Science and Engineering from B.P.U.T. (Biju Pattnaik University of Technology), Rourkela, M.Tech. degree in Computer Science and Engineering from the KIIT (Kalinga Institute of Industrial Technology) University, Bhubaneswar, and a Ph.D. degree in Computer Science and Engineering from the Indian Institute of Technology Madras. Prior to joining IIT Ropar, he worked as an Assistant Professor in the Department of Computer Science and Information Systems at BITS Pilani, Hyderabad campus from May 2017 to Jan 2020. He also worked at TATA Consultancy Services Ltd. from Jun 2008 to Dec 2010 as a Systems Engineer. His research interests include, but not limited to wireless networking such as cellular networks and wireless ad-hoc networks; the Internet of Things.

Dr. Chittaranjan Hota, is a Senior Professor of Computer Science and Information Systems at BITS Pilani, Hyderabad. He completed his Ph.D from BITS, Pilani and has more than thirty years of academic, research and administrative experience in Indian and universities abroad. His Ph.D work was on QoS assurances in IP-Virtual Private Networks, a significant part of which was carried out at University of New South Wales, Sydney, Australia during his visit to Network Research Laboratory at UNSW under Indo-Australian joint research program. Prior to Ph.D, he had earned B.E in Computer Engineering, and M.E in Computer Sc. and Engineering.