Handover Mechanism to Meet Low-Latency Application Requirement in 5G

Undergraduate Work

Yeison Eduardo Caicedo Garzon
Andres Sebastian Garzon Jimenez

Advisor: Msc. Fulvio Yesid Vivas Cantero
Co-Advisor: PhD. Oscar Mauricio Caicedo Rendón

Department of Telematics
Faculty of Electronic and Telecommunications Engineering
Universidad del Cauca
Popayan, Cauca, 2022

Handover Mechanism to Meet Low-Latency Application Requirement in 5G

Yeison Eduardo Caicedo Garzon
Andres Sebastian Garzon Jimenez

Undergraduate work presented to the Faculty of Electronic
and Telecommunications Engineering of the
Universidad del Cauca to obtain the title of:
Electronics and Telecommunications Engineer

Advisor: Msc. Fulvio Yesid Vivas Cantero
Co-Advisor: PhD. Oscar Mauricio Caicedo Rendón

Department of Telematics
Faculty of Electronic and Telecommunications Engineering
Universidad del Cauca
Popayan, Cauca, 2022

Agradecimientos

En primera medida es imperativo agradecer a Dios, a nuestros familiares por su constante apoyo y entrega a lo largo del desarrollo de nuestra vida personal. A nuestros amigos y colegas del programa de Ingeniería Electrónica y Telecomunicaciones por su importante compañía durante la duración de este proyecto de investigación. A nuestro director Msc. Fulvio Yesid Vivas Cantero y a nuestro codirector PhD. Oscar Mauricio Caicedo Rendón, quienes fueron nuestros guías y mentores a lo largo de este proyecto y junto al semillero de investigación COMSOCAUCA nos brindaron de todas las bases investigativas imprescindibles para su culminación. Además mencionar, a todo el equipo del laboratorio Telco 2.0 por proveernos con los recursos hardware necesarios para el desarrollo del proyecto. Finalmente, queremos resaltar lo orgullosos que nos sentimos por haber realizado nuestro primer paso como ingenieros en la ilustre Universidad del Cauca, que nos proveyó de las mejores luces en nuestra formación.

Abstract

One of the main objectives of Fifth Generation (5G) mobile communication networks is to support Low-Latency applications up to 1ms End-to-End (E2E). To enable Low-Latency Communications (LLC), 5G has adopted solutions such as Network Slicing (NS) along with access point densification (also called gNB). NS allows the creation of customized logical networks (slices) in the gNBs according to the Quality of Service (QoS) requirements of one or more applications. On the other hand, gNBs/slices densification increases network coverage and capacity. Although these two solutions allow better resource management, it generates frequent gNB changes (Handover Management - HM) due to the mobility of the user equipment (UE). Consequently, this HM in the 5G network has two difficulties to enable LLC. First is the uncertainty in meeting the QoS requirements of the application in the target gNB, given the unknown availability of resources in the slices. Second, the interruption of up to 3900 ms in UE communication, given the HM process. Therefore, the 5G network alone is deficient in performing the HM and meeting the LLC requirement. For this reason, **S**licing **H**andover Manag**E**ment **M**echanism (SHEM) was introduced to proactively select the target gNB/slice, taking into account the available resources as the LLC requirement of the application. The evaluation results show that SHEM reduces the HM latency by approximately 3700 ms and achieves 73.5% effectiveness in meeting the LLC requirement of the application.

Resumen

Uno de los principales objetivos de las redes de comunicaciones móviles de Quinta Generación (5G) es soportar aplicaciones de baja latencia de hasta 1ms de Extremo a Extremo (E2E). Para hacer posible las comunicaciones de baja latencia (LLC), 5G ha adoptado soluciones como el Network Slicing (NS) junto con la densificación de los puntos de acceso (también llamados gNB). El NS permite la creación de redes lógicas personalizadas (slices) en los gNB en función de los requisitos de la calidad de servicio (QoS) de una o varias aplicaciones. Por otro lado, la densificación de gNBs/slices aumenta la cobertura y la capacidad de la red. Aunque estas dos soluciones permiten una mejor gestión de los recursos, genera frecuentes cambios de gNB/slice (Gestión de Traspaso - HM) debido a la movilidad de los equipos de usuario (UE). En consecuencia, esta HM en la red 5G tiene dos dificultades para habilitar la LLC. En primer lugar, la incertidumbre a la hora de satisfacer los requisitos de QoS de la aplicación en el gNB/slice de destino, dada la disponibilidad desconocida de recursos en los gNB/slices. En segundo lugar, la interrupción de hasta 3900 ms en la comunicación del UE, dado el proceso de HM. Por lo tanto, la red 5G por sí sola es deficiente para realizar el HM y cumplir con el requisito de LLC. Por esta razón, SHEM -**S**licing **H**andover Manag**E**ment **M**echanism- fue introducido para seleccionar proactivamente el gNB/slice de destino, teniendo en cuenta los recursos disponibles como el requisito de LLC de la aplicación. Los resultados de la evaluación muestran que SHEM reduce la latencia de HM en aproximadamente 3700 ms y logra un 73,5% de efectividad en el cumplimiento del requisito de LLC de la aplicación.

Content

List of Figures VIII

List of Tables X

List of Abbreviations XI

1 Introduction 1

1.1 Problem Statement 1

1.2 Objectives 3

1.2.1 General 3

1.2.2 Specifics 3

1.3 Contributions 3

1.4 Document Structure 4

2 Background and Related Work 5

2.1 Background 5

2.1.1 Software-Define Networking 5

2.1.2 Network Functions Virtualization Management And Orchestration 7
2.1.3 Fifth-Generation of Mobile Communication Networks 7
2.1.3.1 Low-Latency Communications 11
2.1.3.2 Handover Management 11
2.1.3.3 Network Slicing 13
2.2 Related Works 16
2.2.1 Resource Allocation 16
2.2.2 Proactive Service Replication 17
2.2.3 Network Virtualization 18
2.2.4 Conclusions 20

3 Mechanism Based on Network Slicing for Low-Latency Communication Applications 21
3.1 Motivation Scenario 21
3.2 Definition of Slicing Handover Management Mechanism 22
3.3 Algorithm of Slicing Handover Management Mechanism 25

4 Implementation of Mechanism 29
4.1 5G Architecture Overview Together With Mechanism Prototype . . . 29
4.1.1 Architecture Components 30
4.1.1.1 Infrastructure Layer 30
4.1.1.2 Virtualization Layer 31

4.1.1.3 Management And Orchestration Layer 32

4.2 Implementation Requirements 33

4.2.1 Mobility Requirements 33

4.2.2 Network Slicing Requirements 33

4.3 Tools for Implementation 34

4.3.1 FreeRadius 36

4.3.2 Ryu 37

4.3.3 Mininet-WiFi 38

4.3.4 Simulator for Urban Mobility 39

4.4 Handover Management Process 41

5 Evaluation of Mechanism 45

5.1 Testbed 45

5.2 Results 46

5.2.1 Test 1: Network Slicing Check 46

5.2.2 Test 2: Handover Management Latency 46

5.2.3 Test 3: Verification of Meeting the Application Latency Requirement in the Handover Management 48

5.3 Final Remarks 50

6 Conclusions and Future Works 51

6.1 Conclusions 51

6.2 Future Works 52

Bibliography **53**

Appendices **63**

A GitHub Repository 64

B Draft Article 65

C Recommendations 66

List of Figures

2.1 SDN Overview 6

2.2 NFV Overview 8

2.3 5G usage scenarios 9

2.4 SBA 5G 10

2.5 HM Overview 13

2.6 Intra-Slice and Inter-Slice HM 14

2.7 3GPP Solution Groups for NS implementation 14

2.8 Non-3GPP Solution Groups for NS Implementation 15

3.1 Overview of the Motivation Scenario 22

3.2 SHEM Overview 23

4.1 Overview of the Motivation Scenario Architecture 30

4.2 Interaction between tools 35

4.3 Screenshot of the road network built at SUMO 40

4.4 802.11i RSNA establishment 42

4.5 SHEM operation 43

4.6 Inter-slice HM procedure 44
4.7 Intra-slice HM procedure 44

5.1 Test 1: Verification of BW assigned to each slice 47
5.2 Test 2: HM Latency 48
5.3 Test 3: Effectiveness percentage of latency meeting in HM 49

List of Tables

2.1 Related works 20

3.1 QoS requirement of applications 22

4.1 FreeRadius configuration files 37

4.2 Mininet-WiFi configuration files 39

4.3 SUMO configuration files 40

5.1 Testbed parameters 46

5.2 HM duration parameters 47

5.3 gNB/slice target ideal for meeting the E2E latency requirement of the application run by vehicle 49

A.1 GitHub Repository Structure 64

List of Abbreviations

5G	Fifth-Generation
AAA-S	Authorization, Authentication, and Accounting Server
AF	Application Function
AMF	Access and Mobility Management Function
AP	Access Points
API	Application Programming Interfaces
AR	Augmented Reality
AUSF	Authentication Server Function
BW	Bandwidth
CN	Network Core
CP	Control Plane
CRIU	Checkpoint-Restore in User Space
DiffServ	Service Differentiation
DMM	Distributed Mobility Management
DN	Data Network
DRL	Deep Reinforcement Learning

LIST OF ABBREVIATIONS

E2E	End-to-End
EAP	Extensible Authentication Protocol
eMBB	enhanced Mobile Broadband
FT	Fast Transition
HM	Handover Management
HostAPD	Host Access Point Daemon
IEEE	Institute of Electrical and Electronics Engineers
LLC	Low-Latency Communication
LLC	Low-Latency Communications
LVAP	Light Virtual AP
MAC	Media Access Control
MANO	Management and Orchestration
MEC	Mobile Edge Computing
mMTC	massive Machine-type Communications
NaaS	Network-as-a-Service
NEF	Network Exposure Function
NF	Network Function
NFV	Network Function Virtualization
NFVI	Network Functions Virtualization Infrastructure
NFVO	Network Functions Virtualization Orchestrator
NRF	Network Repository Function
NSSAAF	Network Slice Specific Authentication and Authorization Function

LIST OF ABBREVIATIONS

NSSF	Network Slice Selection Function
P-GW	Packet Data Network Gateway
PCF	Policy Control Function
POF	Protocol-Oblivious Forwarding
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RAM	Random Access Memory
RAN	Radio Access Network
RAT	Radio Access Technology
RL	Reinforcement Learning
RRM	Radio Resource Management
RSNA	Robust Security Network Association
RSSI	Received Signal Strength Indicator
RSSI	Received Signal Strength Indicator
SBA	Service-Based Architecture
SHEM	**S**licing **H**andover Manag**E**ment **M**echanism
SMF	Session Management Function
SUMO	Simulator for Urban MObility
UDM	Unified Data Management
UE	User Equipment
UPF	User Plane Functions
URC	Ultra-Reliable Communications

URLLC	Ultra-Reliable and Low-Latency Communications
V2I	Vehicle to Infrastructure
V2N	Vehicle-to-Network
V2X	Vehicle-to-Everything
VAP	Virtual AP
VIM	Virtualized Infrastructure Manager
VNF	Virtualized Network Function
VNFM	Virtual Network Functions Manager
WLAN	Wireless Local Area Network
WPA2	WiFi Protected Access 2

Chapter 1

Introduction

This chapter defines the focus of the undergraduate work. Section 1.1 presents the problem statement. Section 1.2 describes the research objectives for the development of this undergraduate work. Section 1.3 presents the research contributions. And Section 1.4 presents the structure of this document.

1.1 Problem Statement

The Fifth-Generation (5G) networks support emerging application requirements that demand seamless handovers to satisfy Low-Latency Communication (LLC) requirement [1]. These networks deploy numerous Access Points (AP) to improve network resource utilization and enhance the Quality of Service (QoS) expected by mobile users [1, 2]. The process that handles the (dis)connection of a device when it moves between APs is named Handover Management (HM). However, device mobility and network density generates long handover delays (e.g., delays higher than 150 ms [3]) that degrades the network performance and diminish QoS [4]. These handover delays constitutes a limitation for LLC use cases as Augmented Reality (AR), where latency requirement is of 10 ms since the long lag between images can cause user disorientation [5–8]. Therefore, 5G needs to optimize HM aiming to meet LLC requirement and guarantee application connectivity.

In 5G, WiFi will play a key role, since it represents a more affordable, faster, and reliable communication alternative[1] to other wireless technologies such as WiMax and Satellite. Meanwhile, WiFi addresses compliance with LLC requirement in HM through three standards: Service Differentiation (DiffServ - 802.11e standard), Radio Resource Management (RRM - 802.11k standard), and Fast Transition (FT - 802.11r). DiffServ classifies network traffic into four service classes (Background, Best Effort, Video, and Voice) to give different access times to the medium, without reducing the handover delay [10, 11]. RRM simplifies the proactive search for the destination AP by creating a list of available channels from neighboring APs [12]. This way, RRM reduces the handover delay in the discovery phase up to 120 ms [3]. FT allows the AP to store the encryption keys of all network APs [13]. Thus, the devices diminish the authentication delay and achieve a minimum handover delay of 50 ms. Although WiFi still lacks mechanisms that optimize HM to improve network resources management and meet LLC requirement.

Recent research in HM proposes mechanisms such as Resource Allocation, Proactive Service Replication, and Network Virtualization to meet LLC requirement. Resource Allocation [14–17] operates by the reservation of available network resources based on competing for application demands (e.g., link bandwidth (BW) and buffer space in APs). Nevertheless, all demands are impossible to meet, since some applications may receive fewer network resources, increasing the latency in LLC. Proactive Service Replication [18–21] operates by application instances[2] deployment in nearby APs before handover using Mobile Edge Computing (MEC) cloud capabilities (processing and storage). Nonetheless, application instances must be continuously updated, resulting in inadequate use of network resources (besides the storage occupied by the instances) and hence the degradation of overall network performance [18, 19]. Network Virtualization [22–26] works by custom network slices creation with Virtual AP (VAP), where each slice has dedicated resources according to the QoS require-

---

[1] Furthermore, it is estimated that by 2023, there will be 628 million global public Wi-Fi hotspots, 4X more than in 2018 [9].

[2] Virtual machines or Dockers (>10 MB) that can store both application and connection information [18].

ments of one or more applications. However, the slices creation requires the resource reallocation in each AP, generating in a high downtime[3] (>500 ms) that fail LLC requirement. In conclusion, the previous mechanisms by HM evidence the difficulty of meeting LLC requirement in 5G.

This under graduation work deal with the following question:

How to meet LLC requirement in the HM process at 5G?

1.2 Objectives

1.2.1 General

Propose a HM mechanism to meet LLC application requirement in a 5G network.

1.2.2 Specifics

- Introduce a mechanism based on Network Slicing to meet the LLC application requirement in a 5G network.
- Implement a prototype of the proposed mechanism.
- Evaluate the prototype built in terms of latency.

1.3 Contributions

The following work is framed in the research line of Telecommunication Advanced Services of the Telematic Department. Contributions are listed below:

---

[3] Disconnection time caused by the delay of resource reallocation.

- A HM mechanism for meeting LLC application requirement in a 5G network.
- A simulated prototype that implements the HM mechanism.
- An evaluation of the performance of the HM mechanism for a 5G network.

1.4 Document Structure

This document is divided into the chapters described below.

- Chapter 1 presents the **Introduction** containing problem statement, objectives, research contributions and the structure of this document.
- Chapter 2 presents the **Background** with concepts are essential to understand the development of this work. And the **Related Works** that describe research close to that of this work are described.
- Chapter 3 introduce the **Mechanism Based on Network Slicing for Low-Latency Communication Applications**.
- Chapter 4 presents the **Implementation of Mechanism** of SHEM prototype. For this, it defines the architecture of the 5G network based on NS, the implementation scenario, the tools used, and the operation of the SHEM prototype in the HM of the 5G network.
- Chapter 5 presents the **Evaluation of Mechanism** of SHEM prototype. For this evaluation, it defined the evaluation metrics, the evaluation scenarios and the analysis of the results.
- Chapter 6 presents the **Conclusions** and **Future Works**. A main conclusion of the work performed and its implications is provided.

Chapter 2

Background and Related Work

This chapter present the background and related works of this proposal. The background includes: First, an overview of SDN. Second, a review of NFV. Third, describe 5G and the relevance of Low-Latency Communications, Handover Management, and Network Slicing.

2.1 Background

2.1.1 Software-Define Networking

Software-Define Networking (SDN) is a rising networking paradigm originated from McKeown's seminal work [27]. The central concept of SDN is decoupling the control plane, and data plane in the network layer, compared with traditional networks where control and forwarding functionalities are coupled within the switches [28]. Precisely through the centralized controller, SDN handles network flows and can deploy various network applications (Application Plane) such as NS, Load Balancing and Handover Management (see Figure 2.1). Therefore, the switches only need to implement flow processing functions (e.g., forwarding, dropping) based on the rules set by the controller. The default communication protocol between the controller

SDN and switches is OpenFlow. OpenFlow standardizes the switch flow table formats so that each flow is processed according to the corresponding actions defined in the table [29]. Therefore, the advantages of SDN are: network programmability and flexibility, and robustness [29, 30]. Considering these advantages, SDN has been successfully adopted in wired networks and recently is shifted to wireless networks [20, 23].

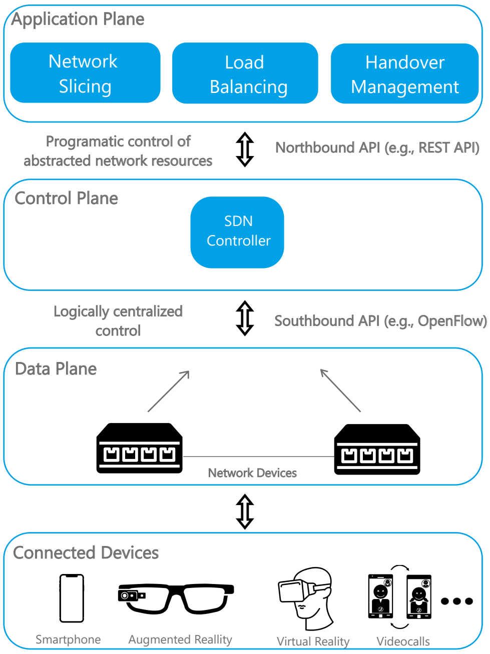


Figure 2.1: SDN Overview

The adoption of SDN in wireless networks has been more complicated, given that communication is changing and unpredictable through a shared medium due to interference, mobility, and dynamic device associations [14]. Additionally, in wireless networks, OpenFlow lacks a secure channel for communication with mobile devices [30]. This way, OpenFlow introduces new challenges to SDN network management and performance. Although OpenFlow allows using the header fields (layer 2, layer 3 and layer 4) of the flows to adjust parameters such as transmission power or transmission rate that improves the communication quality [14]. The use of these header fields introduces fine-grained and application-specific transmission control opportunities, e.g., for seamless handover [31].

2.1.2 Network Functions Virtualization Management And Orchestration

In non-virtualized networks, Network Functions (NFs) is a combination of vendor-specific software and hardware, often referred to as network nodes or network elements. NFs Virtualization (NFV[1]) represents a step forward for the diverse stakeholders in the telecommunication environment. NFV decouples the entire classes of network node functions (i.e., NFs) from the network hardware to create virtualized network services such as routing, load balancers, and firewalls [33]. NFs traditionally use dedicated hardware as gateways that are now implemented in software running on general-purpose hardware. Therefore, NFV provides an architecture with the ability to customize and instantiate the Virtualized NFs (VNFs), reducing dependence on proprietary hardware, facilitating network scalability, management of network resources, and and create personalized network slices (see Subsection 2.1.3.3) [20, 33, 34].

Figure 2.2 shows the high-level overview of NFV, where virtual resources result from the abstraction of physical resources (e.g., processing, storage, and network devices) through a virtualization layer. Thus, NFV can be instantiated on dockers or virtual machines. Additionally, NFV Management and Orchestration (MANO) layer provides the management and orchestration of the lifecycle of physical and software resources, VNFs, and network services, that enables the infrastructure virtualization [35]. With NFV enables more dynamic networks and awareness of the specific needs of the network. Furthermore, combining NFV with SDN could optimize the configuration and slicing of the network resources based on the QoS requirements of the flows [23, 36].

2.1.3 Fifth-Generation of Mobile Communication Networks

The Fifth-Generation of mobile communication networks (5G), introduced in 3GPP-Release 15 [37]. 5G is designed to support emerging QoS requirements of applica-

---

[1] NFV is a network architecture concept standardized by ETSI [32].

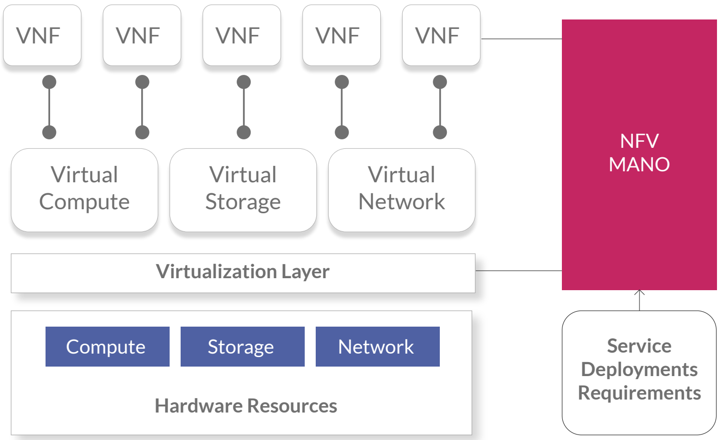


Figure 2.2: NFV Overview

tions that demand peak data rates ($\sim$ 20 Gbit/s), high-reliability communication (error rates from 10^{-5} to 10^{-9}), and intensive computing and real-time data processing (latency of 1 to 10ms) [7, 38]. 5G combines the above requirements into three usage scenarios (see Figure 2.3): enhanced Mobile Broadband (eMBB), massive Machine-type Communications (mMTC), and Ultra-Reliable and Low-Latency Communications (URLLC) [39, 40]. Since the above scenarios and requirements are an unexplored region in 3G/4G networks, 5G introduces improvements such as increased network coverage and capacity by network densification [12]. Network densification is the use of different wireless access technologies (such as WiFi) with various coverage sizes and topologies. In this way, it reduces the coverage area of each gNB and increases spatial-spectral efficiency, offering more capacity and data rate [41]. Therefore, 5G proposes to meet the strict QoS requirements of applications by increasing: network coverage, network capacity, and network densification [41].

In addition to increasing network coverage, network capacity and network densification, 5G adopts NS through SDN/NFV virtualization [42]. Thus, 5G introduces a Service-Based Architecture (SBA) that is independent of 4G LTE networks [37]. 5G SBA includes the Radio Access Network (RAN), 5G Core Network (CN), and User Equipment (UE). RAN is a collection of interconnected gNBs linked to the CN and provides coverage in a specific area based on Radio Access Technologies (RATs) [43]. CN is the composition of various VNFs, with a defined separation between Control

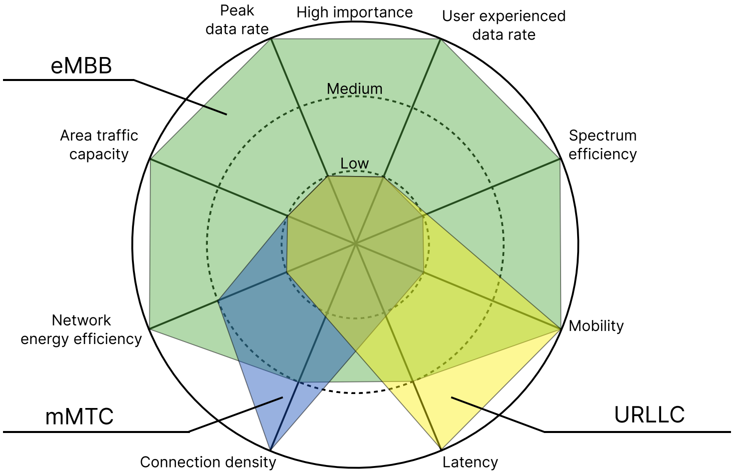


Figure 2.3: 5G usage scenarios

Plane functions (CP) and User Plane Functions (UPF), as shown in Figure 2.4. The CP is itself a forwarding path to exchange information for service operation. While, UPFs play a critical role in the data transfer process, providing the interconnection point between the UE and the Data Network (DN). On the other hand, CP functions represent all the signaling used to support the functions that set and maintain the UPFs. Signaling refers to the exchange of information to enable, however, not to provide the E2E communication service itself. Below is a summary of each of the VNFs of 5G SBA.

The CP functions correspond to:

- Access and Mobility Management Function (AMF) provides UE-based authentication, authorization, mobility management, etc.
- Authentication Server Function (AUSF) stores data for UE authentication.
- Unified Data Management (UDM) stores UE subscription data.
- Session Management Function (SMF) is responsible for session management and also selects and controls the UPF for data transfer.
- Policy Control Function (PCF) can instruct different routing policies.

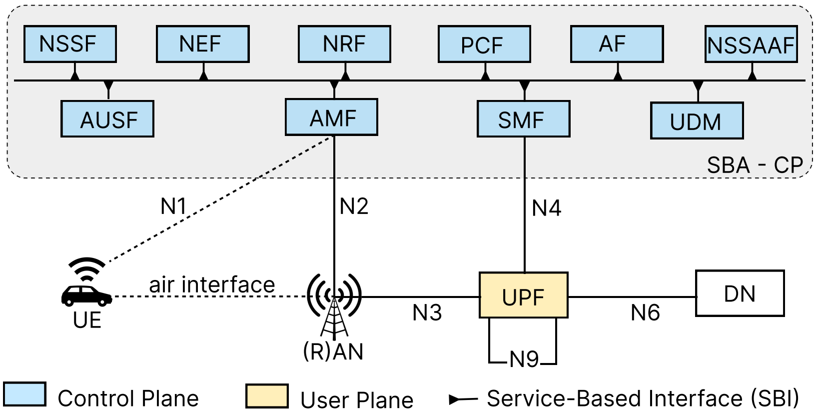


Figure 2.4: SBA 5G

To specifically address the cloud-native design and the paradigm shift from an entity-based network (4G) to a function-based network, 5G introduces the following NFs.

- Network Repository Function (NRF) provides registration and discovery functionality, allowing NFs to discover mutually and communicate through open Application Programming Interfaces (APIs), in contrast to LTE, which uses predefined interfaces between elements. For example, the AMF service exposes information regarding mobility-related events and relevant statistics to other NFs.
- Network Exposure Function (NEF) provides the means to securely collect, store and expose the services and capabilities of 3GPP network functions (e.g., to third parties or between NFs).
- Application Function (AF) represents any additional CP functions that may be required, e.g., to implement network fragmentation.

While to enable NS, 5G introduces the following CP functions.

- Network Slice Selection Function (NSSF) helps with the selection of Network Slice instances and AMFs that will serve a particular UE.

- Network Slice Specific Authentication and Authorization Function (NSSAAF) enables support for Network Specific Authentication and Authorization according to specified with an Authorization, Authentication, and Accounting Server (AAA-S).

2.1.3.1 Low-Latency Communications

URLLC comprises Low-Latency Communications (LLC) and Ultra-Reliable Communications (URC) [44, 45]. URLLC is essential for enabling mission-critical 5G applications, such as AR and remote driving[2]. The QoS requirements of these applications implicate error rate up to an 1 - 10^{-5} (URC) and E2E[3] latency up to 1 ms (LLC) [46, 47]. Research such as [38, 44] shows that compliance with the URC is simpler than LLC. Compliance with the URC requirement would be possible through network densification, interface diversity, selective channel transmission, or temporary storage. In comparison, compliance with the LLC requirement is complex, given the loss of connection caused by HM and the untimely reallocation of resources that degrades the QoS application requirements. Given this complexity and the gaps exposed in the Section 2.2 - Related Works, this undergraduate work is centered on meeting the LLC requirement.

2.1.3.2 Handover Management

5G proposes increasing network coverage and capacity by network densification to meet the requirements of the three usage scenarios. This increase implies that the UE can be located in the area of more than one gNB available to connect to and manage a handover. Nevertheless, the UE can increase the handover probability, resulting in the interruption of the UE-gNB connection. Furthermore, the increased interruptions generate i) degradation of the QoS provided to the UE and ii) the associated signaling overhead, decreasing or even canceling the gains from network densification [41]. Therefore, 5G needs an HM capable of meeting UE QoS regardless of network conditions or UE mobility.

---

[2] Remote driving belongs to the group of Vehicle-to-Everything (V2X) applications.

[3] E2E means End-to-End

In 5G, HM consists of a three-phase process (see Figure 2.5): preparation, execution, and completion [48, 49].

- **Preparation phase:** HM starts with sending the measurement reports[4] from the UE to the connected gNB (source gNB). Based on the measurement reports, the source gNB chooses and sends the handover request to the target gNB. According to the resources and admission control of the target gNB, the handover request of the UE can be accepted or rejected. With the acceptance of the handover request, the source gNB sends the handover action message to the target gNB to the UE.
- **Execution phase:** Once the UE receives the handover action, the UE synchronizes the gNB change. To avoid data loss while the switch is in progress, the source gNB redirects the data to a buffer in the target gNB. When the UE has been successfully switched (associated) to the target gNB, the data in the buffer is sent to the UE. Thus, this phase is over, and the completion phase begins.
- **Completion phase:** The target gNB requests the AMF to update the downlink data path to the UE to complete the handover. Subsequently, the AMF reconfigures the data path in itself and the UPF. On UPF reconfiguration, the UPF informs the source gNB to release radio and CN resources related to the UE. Moreover, thus, the handover is completed. As an additional note, AMF reconfiguration may trigger additional procedures such as AAA.

According to [12, 50], in RATs such as WiFi, the HM time exceeds 1500 ms, whereas the preparation phase (discovery of neighboring gNBs) can take up to 90% of the HM time. Apart from that, if the RAT implements robust authentication procedures such as WiFi Protected Access 2 (WPA2), the HM time will increase by up to 500 ms [50]. Therefore, to meet the QoS requirements of LLC, it is necessary to improve each of the HM phases, especially the preparation phase.

---

[4] Measurement reports configured by the target gNB and includes as a minimum the Received Signal Strength Indicator (RSSI) of the neighboring gNBs.

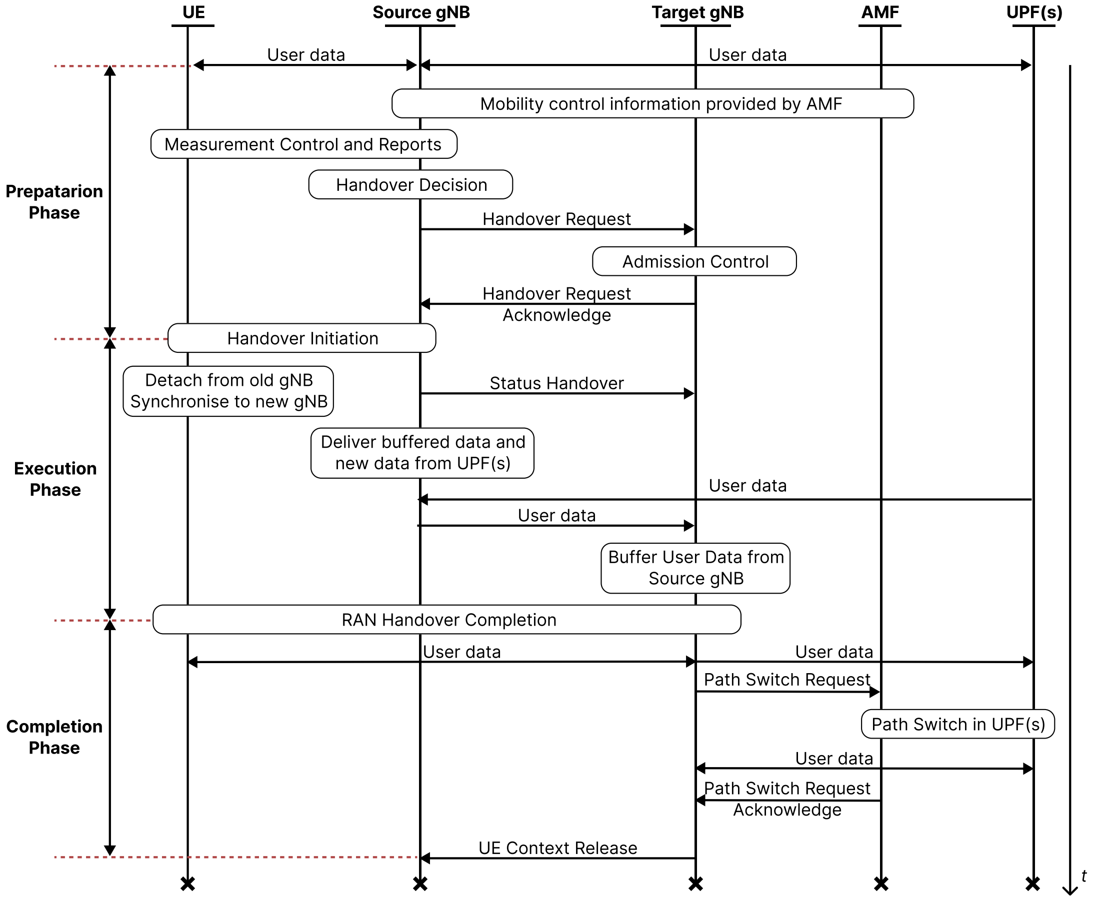


Figure 2.5: HM Overview

Additionally, in a network implementing NS, there are two types of HM, intra-slice, and inter-slice (see Figure 2.6). The intra-slice HM indicates the change of slice without changing the gNB. Inter-slice HM indicates the connection change to a slice in another gNB [42, 49, 51].

2.1.3.3 Network Slicing

Network Slicing (NS) is one key feature to address the QoS requirements of the usage scenarios that coexist in 5G [42]. To this end, NS leverages the decoupling introduced by SDN and NFV to virtualize the network infrastructure [52, 53]. Thus, NS allows the management (i.e., creating, modifying, and deleting) of logical networks

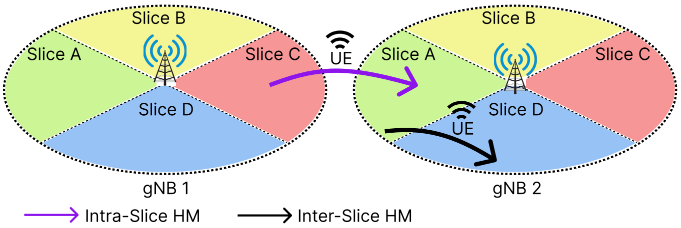


Figure 2.6: Intra-Slice and Inter-Slice HM

(slices) customized according to the QoS requirements of one or more applications over a common infrastructure. Thus, each slice has a collection of dedicated and shared virtual resources. As evidenced in Figures 2.7 and 2.8, this resource collection can include multiple parts of the network such as UEs, RAN, and CN. Therefore, NS provides a Network-as-a-Service (NaaS) model, which can flexibly allocate and reallocate resources according to the QoS requirements of one or more applications. For all the above reasons, NS is a key solution to meet the QoS requirements of diverse and complex usage scenarios such as LLC.

To implement NS in 5G, 3GPP defines three solution groups (Figure 2.7), where each group has a different distribution of the CN, i.e., UPFs and CP functions [54]:

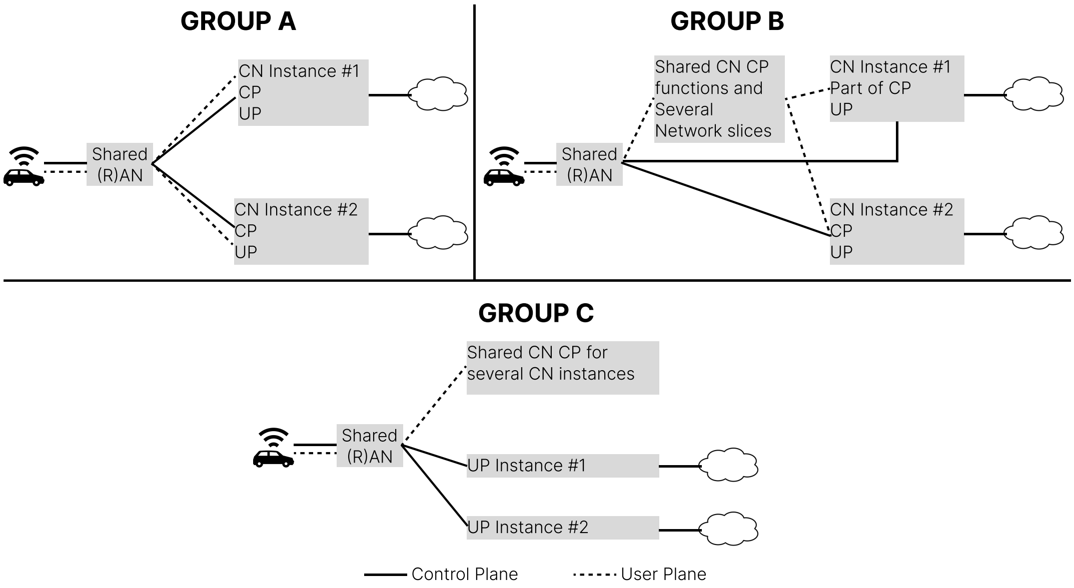


Figure 2.7: 3GPP Solution Groups for NS implementation

- Group A: All slices share the RAN, while the CN are dedicated for each slice.

- Group B: All slices assume a common RAN, while UPF and CP functions, one part is common and another part is dedicated.
- Group C: All slices share RAN and CP functions, while UPF are dedicated.

Since the previous three groups discarded slicing in the RAN, in [55], the authors propose three different solutions for the NS implementation (Figure 2.8).

- CN-only slicing: Each slice has a dedicated CN, while the RAN is shared between all the slice.
- RAN-only slicing: All slices share the CN, while NS is implemented in the RAN hardware.
- CN-RAN slicing: This slicing is E2E since each slice has a dedicated portion of the RAN and CN.

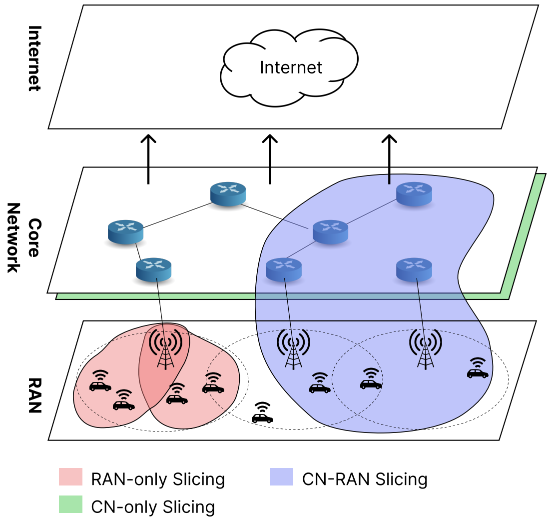


Figure 2.8: Non-3GPP Solution Groups for NS Implementation

From the above NS implementation solutions, this undergraduate work considers CN-RAN slicing together with Group B. This combination of solutions is appropriate for implementing slices such as LLC. Furthermore, this combination provides E2E customization in both RAN resources and CN functions. Thus, allowing flexibility in the management of the slices and optimization of processes given the possibility of sharing or dedicating the CN functions.

2.2 Related Works

This section presents a review of related work according to HM that focus on reducing handover latency and meeting applications QoS requirements. These works were classified into three subsections. Subsection 2.2.1 presents the works that Allocate Resources by reserving available network resources. Subsection 2.2.2 presents the works that perform Proactive Service Replication by deploying application instances on nearby APs. Subsection 2.2.3 presents work that implements Network Virtualization to create customized network slices. Finally, this section summarizes the related works in Table 2.1, exposing their differences with with the proposed HM mechanism. From here, the observations on these works are derived and highlight what is needed for the proposed HM mechanism to meet the objectives proposed in Section 1.2.

2.2.1 Resource Allocation

In [14], the authors designed an HM scheme based on BW reservation policies sensitive to the traffic class. This scheme reserves the BW in each AP according to service classes (e.g., Best Effort, Background, Video, and Voice) and handover processes. However, if the BW is unused, it must remain available. In this way, the handover traffic always has the necessary BW and achieves handovers without data loss with a minimum delay of 60 ms. This scheme has the following limitations: i) low scalability of the network due to the disuse of network resources, and ii) high handover delay for LLC requirements, because of the insufficient traffic classes, the lack of efficient resource management.

In [15], a dynamic QoS based IP HM procedure was proposed to handle the application-centric mobility management. Such QoS based handover process ensures the required quality level for the on-going connections according to the policies enforced by the SDN controller. This HM procedure uses SDN to identify (before handover) the appropriate route to provide the required bit rate according to the applications QoS. At the same time, SDN allows the IP address to be maintained to avoid disconnection during the handover. In this manner, this proposal omits the association phase and reduces the delay up to 50 ms. However, the handover delay is still excessive for LLC requirements which needs handover delay less than 10 ms.

In [16], the authors presented a proactive approach to radio channel assignment in conjunction with HM. This approach determines the channel queue and channel occupancy time in AP. For when a handover occurs, the approach selects the objective AP with the lowest channel occupancy. Thus, this approach avoids handover blocking. Although it can generate critical delays for LLC-type applications as a result of receiving traffic flows without differentiating whether it is delay-sensitive or not.

In [17], the authors proposed a Machine Learning based method to find an optimal handover mechanism. This method allows us to predict whether the handover that is going to happen will maintain the throughput, optimizing resource allocation between APs. However, to maintain an algorithm with predictive levels of acts, at first, this proposal needs data to train this algorithm. Therefore, there will be quite an amount of wrong decisions about handover prediction causing QoS degradation.

2.2.2 Proactive Service Replication

In [18], the researchers propose the proactive replication of stateless application instances in neighbouring AP. This proposal maintains and updates instances in the neighbouring AP with application and connection data. When the handover occurs, the instance must update less amount of data (>10 MB). Although devices perform handovers without data loss, there is a downtime of more than 500 ms. Therefore, this proposal violates the LLC requirements.

In [19], the authors optimized the proactive copying of application connection information through dockers in the neighbouring APs. This proposal uses mobility prediction algorithms to minimize containers in neighboring APs. This way, this work achieves an excellent rate of 97.5% for seamless handover (with at least 4 APs). However, this proposal ignores the evaluation of handover delay. At the same time, the high waste of network resources used in dockers, makes the present solution inefficient to meet the rigorous LLC requirements.

In [20], researchers introduced the full-state application migration mechanism based on a predefined path. The mechanism uses Checkpoint-Restore in User Space (CRIU) to save the executing application state in a container before handover. Subsequently, the container is copied to the destination AP, and the application is restarted according to the CRIU checkpoint. Although this mechanism conserves all application data, it has a downtime (>1000ms) that impairs the continuity of applications with LLC requirements.

In [21], the authors proposed a vehicular MEC architecture instead of simply offloading LTE infrastructure. Routing all the packets with the MEC network achieves Vehicle to Infrastructure (V2I) communications with very low packet delay (10 - 30ms). Also, this architecture provides seamless handover with Distributed Mobility Management (DMM) in the MEC network. Nevertheless, in order to achieve seamless handovers with low delay, this architecture makes use of requests on servers close to the user. When the servers for the services are far away from the vehicle, any request outside the MEC network will have adverse effects on the seamless handover and delay times.

2.2.3 Network Virtualization

In [22], the authors propose BYON to create network slices with dedicated resources according to a set of QoS requirements. BYON has an SDN controller to configure each slice in an additional AP interface. Furthermore, the SDN controller enables APs to store flows to avoid packet loss during handover. BYON achieves handovers without packet loss in less than 65 ms. However, BYON has high handover delay that degrades LLC requirements. Furthermore, it is few scalable, given the difficulty of adding the necessary interfaces in each AP.

In [23], the authors propose ADE2WiNFV to provide NaaS, i.e., to offer custom network slices according to a set of QoS requirements. ADE2WiNFV combines SDN and NFV to virtualize/assign APs, network resources and NFs, and thus offer independent network slices. Additionally, ADE2WiNFV implements Protocol-Oblivious Forwarding (POF) to route flows to their corresponding VAP, even when the handover occurs. In this manner, ADE2WiNFV meets the applications QoS with a minimum handover delay of 220 ms. However, ADE2WiNFV has the following disadvantages: i) excessive handover delay compared with LLC requirements, ii) lack of resource reallocation in physical APs (given the handovers of the devices), and iii) high downtime (>500 ms). To sump up, ADE2WiNFV degrades the Qos requirements as LLC.

In [24], the authors present Odin to introduce the concept of Light Virtual AP (LVAP) based on SDN. LVAP gives the illusion that each device has its own AP. For when the handover occurs, the SDN controller only has to change (in LVAP) the registry of the linked AP. Thus, the devices skip the authentication phase and reduce the handover delay up to 1 ms. However, a more significan number of devices considerably increases the handover delay due to rising control traffic. Therefore, although Odin achieves delays according to the LLC requirements, it needs to improve its handover mechanism.

In [25], the researchers proposed an open enterprise WiFi solution based on virtual APs, managed by a central Wireless Local Area Network (WLAN) controller. It allows seamless handovers between APs in different channels, maintaining the QoS of real-time services. This is achieved by omitting the discovery and authentication phases of the handover. The scheme assigns each device a VAP and each AP an additional interface. Furthermore through an SDN controller, the virtual APs employ the additional AP interface to discover the available channel (for handover) in the neighbouring APs, and authenticate the device with the discovered channel. In this way, The device thinks it's still in the same AP, reducing the handover delay up to 22 ms. However, this virtual APs scheme degrades compliance with the LLC requirements due to the high handover delay.

In [26], researchers propose a new architecture for LTE and WiFi networks to achieve low latency. This solution use SDN and NFV to create LVAPs. In order to meet low latency, they use a Packet Data Network Gateway (P-GW)[5] that serves to download and extract the data to a Wireless Access Gateway. The LVAP decreases the handover latency using the same BSSID with all the LVAPs, making the device

---

[5] P-GW allows traffic mapping from LTE to WiFi, and a Wireless Access Gateway interacts with the user as an LVAP

think it remains in the same network. However, the network ignores the available resources in the destination AP when it triggers a handover. This may generate latency in case there are many users or few resources in the destination LVAP.

2.2.4 Conclusions

Table 2.1 presents the summary of related works to HM. First, Resource Allocation presents problems like limited network resources. Second, Proactive Service Replication shows a lacks in inadequate utilization of network resources, and finally, Network Virtualization increases delays at slice/instance creations. Finally, the Table 2.1 shows that achieving seamless handovers alone is deficient in meeting LLC requirement. Therefore, proposed HM mechanism proposes timely HM and seamless handovers (based on SDN/NFV) to meet LLC requirement.

Work	Type	HM		Seamless Handover	LLC	SDN	NFV
		Proactive	Reactive				
[14]	RA		✓	✓			
[15]	RA	✓		✓		✓	
[16]	RA	✓				✓	
[17]	RA	✓		✓			
[18]	PSR		✓	✓			
[19]	PSR	✓		✓			
[20]	PSR	✓		✓		✓	✓
[21]	PSR	✓		✓			
[22]	NV	✓		✓			
[23]	NV	✓				✓	✓
[24]	NV		✓	✓	✓	✓	
[25]	NV		✓	✓		✓	
[26]	NV		✓	✓		✓	✓
Proposed HM mechanism		✓		✓	✓	✓	✓

RA: Resource Allocation - PSR: Proactive Service Replication - NV: Network Virtualization

Table 2.1: Related works

Chapter 3

Mechanism Based on Network Slicing for Low-Latency Communication Applications

This chapter presents the design of the HM Mechanism in NS-based 5G networks, which from now on will be referred to as SHEM - **S**licing **H**andover Manag**E**ment **M**echanism. The design has three sections. Section 3.1 presents the motivation scenario. Section 3.2 introduces an SHEM overview. Section 3.3 presents the built Mechanism of SHEM.

3.1 Motivation Scenario

The SHEM design was based on the motivational scenario presented in Figure 3.1, which is an LLC mobility scenario in a 5G network based on NS. Specifically, the scenario is of remote driving, where the UEs (vehicles) are controlled by a Vehicle-to-Network (V2N) application through the gNBs. Therefore, the network has three applications on remote servers, two applications are V2N, and one application is for general purposes. For optimal performance, these applications demand QoS requirements, abbreviated in Table 3.1 [47, 56]. To meet those requirements, the network implements a slice for each application. In this way, each slice in each gNB (gNB-

/slice) has sufficient resources (for a given number of vehicles) to meet the QoS requirements of the corresponding application. However, the HM of the vehicle between gNBs threatens the fulfillment of the latency requirement (Subsection 2.1.3.2). For this reason, SHEM is introduced into the network to perform HM, meeting the latency requirement.

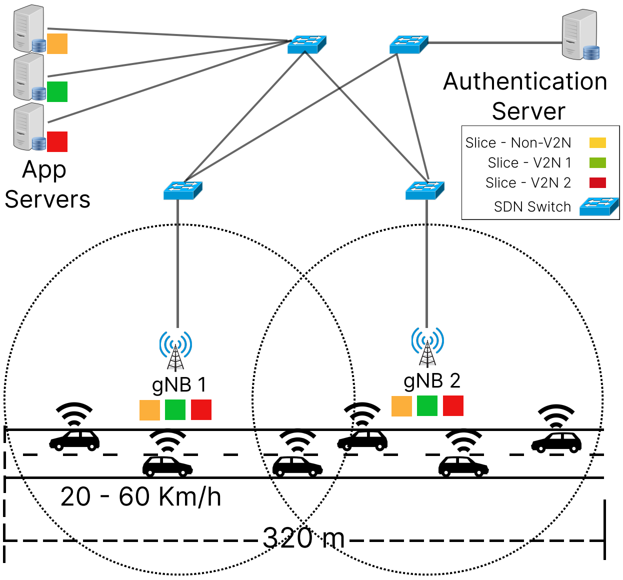


Figure 3.1: Overview of the Motivation Scenario

APP \ QoS Requirement	Latency E2E [ms]	BW x Vehicle [Mbps]
Non-V2N application	50	10
V2N Application - 1	20	25
V2N Application - 2	10	35

Table 3.1: QoS requirement of applications

3.2 Definition of Slicing Handover Management Mechanism

According to the HM presented in Section 2.1.3.2, **S**licing **H**andover Manag**E**ment **M**echanism (SHEM) addresses the preparation phase since it represents most of the latency of the HM. For this, SHEM determines the neighboring gNB/slices, taking advantage of the coverage range of the beacons being larger than the service region, i.e., the region that provides the expected QoS [57]. In this way, SHEM determine the best target gNB/slice proactively (before entering the service region), considering the 5G network conditions and the QoS requirements of the vehicle application. For this purpose, SHEM includes three modules (see Figure 3.2): Monitoring, Evaluator, and Actuator.

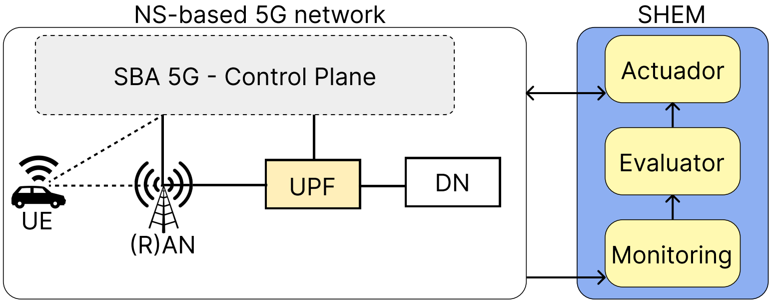


Figure 3.2: SHEM Overview

Below only the objective of each module is explained, since SHEM modules can be built using different technologies from Heuristics to Machine Learning (The built of SHEM is presented in Section 3.3).

- **Monitoring module** has the function of determining the status of the conditions of both the 5G network and the vehicle application. For this purpose, the Monitoring module collects data[1] from the vehicle and the gNBs/slices (both source and neighboring). The collected data are according to the following criteria: concerning the mobility of the vehicle, the QoS of the slice, the QoS of the application executed by the vehicle, and the resource availability of the slice. Although the above criteria have more than one variable, SHEM represents the criteria with the following variables:

    1. Criterion concerning vehicle mobility: corresponds to the Received Signal Strength Indicator (RSSI) variation between vehicle and gNB/slice. The RSSI is a quantitative variable of real type. In this way, SHEM can deduce the proximity between the vehicle and the gNB/slices, and initiate the HM before losing communication with the source gNB/slice.

---

[1] The data delivery to the Monitoring module is independent of SHEM operation since data can be delivered through different routes, e.g., through gNBs, SDN controller, or UEs (see Chapter 4).

2. Criterion concerning the QoS of the application executed by the vehicle: Corresponds to the latency requirement of the application executed by the vehicle. This latency is generally represented quantitatively by integers.
3. Criterion concerning the QoS of the slice: Corresponds to the latency between the vehicle and the application server. This latency is generally represented quantitatively by natural numbers. This variable is relevant since SHEM can determine the appropriate target gNB/slice to meet the latency requirement of the application executed by the vehicle (Criterion number 2).
4. Criterion concerning the use of slice resources: Corresponds to the variation of vehicles connected to the gNB/slice. This variation is represented by natural numbers. In this way, SHEM can determine the resource availability at each gNB/slice and initiate the selection of the target gNB/-slice when the source gNB/slice has a high number of vehicles, i.e., low resource availability.

Finally, when the Monitoring module collects the data from the gNB/slices (origin and neighboring) according to the above criteria, it delivers them to the Evaluator module. Additionally, the Actuator module also sends the vehicle connection information, i.e., the Media Access Control (MAC) address of the vehicle, the MAC address of the source gNB/slice, the vehicle identifier, and the latency requirement of the executed application.

- **Evaluator module** has the objective of determining the appropriate target gNB/slice for the HM of the vehicle. For this purpose, the Evaluator module analyzes the data sent by the Monitoring module. In this way, the Evaluator module must determine two targets gNB/slice. One the gNB/slice ensures meeting the latency required by the application executed by the vehicle. And, although the other gNB/slice does not meet the latency requirement, it does ensure the availability of resources. Subsequently, the Evaluator module delivers these two target gNB/slices to the Actuator module.
- **Actuator module** has the function of initiating the HM of the vehicle to the target gNB/slice. For this, the Actuator module receives the two target

gNB/slice options and initiates the HM toward the gNB/slice, aware of the latency requirement. If the previous gNB/slice is empty (i.e., they are non-existent), the Actuator module initiates the HM to the gNB/slice for general-purpose applications. If, in both cases, the target gNB/slice is non-existent, therefore, the vehicle maintains the connection to the source gNB/slice.

After getting to know the modules that integrate SHEM, Section 3.3 presents the feasible technology to building it.

3.3 Algorithm of Slicing Handover Management Mechanism

In the literature, the heuristic algorithm (a technique designed to solve a specific problem) seems way faster and a more feasible solution than algorithms such as Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) [58].

This section presents the SHEM mechanism built using heuristic programming with Python (The implementation of SHEM mechanism in the NS-based 5G network is presented in the Chapter 4). Thus, the Python script called SHEM_mechanism is presented in the GitHub repository exposed in Appendix A. Next, the algorithms of the SHEM mechanism are presented.

Where:

- j: gNB ID, k: slice ID, and v: vehicle ID.
- n: Number of neighboring $\text{gNB}_j/\text{slice}_k$.
- $[js, ks]$: Referring to **s**ource gNB/slice.
- $[jt, kt]$: Referring to **t**arget gNB/slice.
- $[jtL, ktL]$: Referring to **t**arget gNB/slice that meets the **l**atency requirement of the vehicle.
- th: Referring to **th**reshold.
- $Lat_{\{x\}}$: Latency value referring to x.
- $Load_{\{x\}}$: Load value referring to x. Number of connected vehicles.

- $mac_{\{x\}}$: MAC address referring to x.
- $[RSSI_{t-1}, RSSI_{t}]_{j,k}$: Variation RSSI between vehicle and gNB/slice j,k.
- $[Load_{t-1}, Load_{t}]_{j,k}$: Variation of vehicles connected to the gNB/slice j,k.

Algorithm 1 presents the Monitoring module. The Monitoring module receives the data (s(t)) from the source and neighboring gNB/slices and the connection information (c) from the vehicle. Then, if the vehicle has changed gNB/slice, the Monitoring module calculates the HM times -The HM times depends on the RAT used since the HM process can be different between RATs, in this work, the HM times are explained in Section 4.4-. Subsequently, the Monitoring module sends s(t) and c to the Evaluator module.

Algorithm 1: Monitoring Module - SHEM

```
Require: $Load_{th}$,$RSSI_{th}$
/* Monitoring module                                              */
Receives the c and s(t);
/* s(t)=[$RSSI_{j,k}$(t-1), $RSSI_{j,k}(t)$, $Lat_k$, $Load_{j,k}$(t-1), $Load_{j,k}(t)]_n$   */
/* c=$mac_v$, $mac_{js,ks}$, $id_v$, $Lat_v$                          */
if vehicle changed gNB/slice connected then
    calculate HM time;
end
Evaluator(s(t),c,$Load_{th}$,$RSSI_{th}$);
```

Algorithm 2 presents the Evaluator module. From s(t) and c, the Evaluator module checks if the vehicle is under the RSSI threshold (line 2-10) or the current slice has exceeded the load threshold (line 11-16). Being below the RSSI threshold leads to performing an inter-slice; however, it must be verified if the vehicle is approaching that target gNB/slice. On the other hand, exceeding the load threshold may lead to an inter-slice or intra-slice. Subsequently, the Evaluator module determines two target gNB/slice for the HM. One gNB/slice meets the latency requirement of the vehicle application, while the other gNB/slice does not. To determine the gNB/slice that does not meet the latency requirement, the Evaluator module selects only the gNB/slice with the lowest load. While to determine the gNB/slice that meets that latency requirement, the Evaluator module adds a condition that the latency required by the vehicle application is less than or equal to the latency offered by the gNB/slice. Subsequently, the Evaluator module sends these two targets gNB/slice together with the MAC address of the vehicle to the Actuator module.

Algorithm 2: Evaluator Module - SHEM

```
/* Evaluator Module */
Function Evaluator($s(t)$,$c$,$Load_{th}$,$RSSI_{th}$):
    /* HM for degraded RSSI */
    if $RSSI_{js,ks}(t) \leqslant RSSI_{th}$ then
        if Vehicle moves away from the $gNB_{js}/slice_{ks}$ then
            foreach $gNB_j/slice_k \in s(t)$ do
                /* Inter-Slice HM. Considers only the gNBs
                   different than $gNB_{js}$ */
                if $gNB_{jt}! = gNB_{js}$ then
                    Find $gNB_{jt}/slice_{kt}$ with lowest load;
                    Find $gNB_{jtL}/slice_{ktL}$ with lowest load, considering
                      $Lat_v \leqslant Lat_k$;
                end
            end
        end
    else if $Load_{js,ks}(t) > Load_{th}$ then
        /* HM for excess vehicles */
        /* Determine the $gNB_{jt}/slice_{kt}$ and $gNB_{jtL}/slice_{ktL}$ */
        foreach $gNB_j/slice_k \in s(t)$ do
            /* Intra-Slice and Inter-Slice HM. Considers alls the
               gNBs/slices */
            Find $gNB_{jt}/slice_{kt}$ with lowest load;
            Find $gNB_{jtL}/slice_{ktL}$ with lowest load, considering $Lat_v \leqslant Lat_k$;
        end
    else
        /* Retain current $gNB_{js}/slice_{ks}$ */
        $gNB_{jt}/slice_{kt} = None$;
        $gNB_{jtL}/slice_{ktL} = None$;
    end
    Actuator($gNB_{js}/slice_{ks}$,$gNB_{jt}/slice_{kt}$,$gNB_{jtL}/slice_{ktL}$,$mac_v$);
    return None;
```

Algorithm 3 presents the Actuator module. The Actuator module initially verifies that the target gNB/slice that meets the latency requirement of the vehicle application is different from none and the source gNB/slice. If the above is true, the Actuator module initiates the vehicle's HM to this gNB/slice. Otherwise, the Actuator module initiates the HM to the other target gNB/slice, as long as it differs from none. In case of omitting the HM initiation, the vehicle maintains the connection with the source gNB/slice.

Algorithm 3: Actuator Module - SHEM

```
/* Actuator Module                                                    */
Function Actuator($gNB_{js}/slice_{ks}$,$gNB_{jt}/slice_{kt}$,$gNB_{jtL}/slice_{ktL}$,$mac_v$):
    if ($gNB_{jtL}/slice_{ktL}$!=$gNB_{js}/slice_{ks}$)$\wedge$($gNB_{jtL}/slice_{ktL}$!=None) then
        HM of $mac_v$ to $gNB_{jtL}/slice_{ktL}$;
    else if $gNB_{jt}/slice_{kt}$!=None then
        HM of $mac_v$ to $gNB_{jt}/slice_{kt}$;
    else
        Retain current $gNB_{js}/slice_{ks}$
    end
return None;
```

Chapter 4

Implementation of Mechanism

This chapter presents the implementation of SHEM prototype in a 5G network based on NS. The implementation is presented in four sections. Section 4.1 presents an overview of the 5G network architecture together with SHEM prototype. Section 4.2 presents the requirements for the SHEM prototype implementation in the 5G network. Section 4.3 presents the tools used to implement the SHEM prototype in the NS-based 5G network. Section 4.4 presents the HM of SHEM prototype in the NS-based 5G network.

4.1 5G Architecture Overview Together With Mechanism Prototype

This section proposes the high-level 5G network architecture based on the motivation scenario of the section 3.1, that defines the architecture components and relationship with SHEM prototype. The architecture is based on the three-layer model (infrastructure layer, virtualization layer, and service layer) plus the cross-cutting MANO layer [52, 53]. The above layers are aligned with the SBA 5G and its key enablers, i.e., SDN and NFV, as described in Chapter 2.

To avoid confusion, it is emphasized that the full implementation of the 5G architecture is difficult, as there is no platform or tool that integrates all the layers of the 5G architecture. Therefore, Section 4.3 presents the tools that allow to collectively implement the 5G network to execute the SHEM prototype.

4.1.1 Architecture Components

The three layers of the high-level architecture and the corresponding components are illustrated in Figure 4.1 and discussed briefly below.

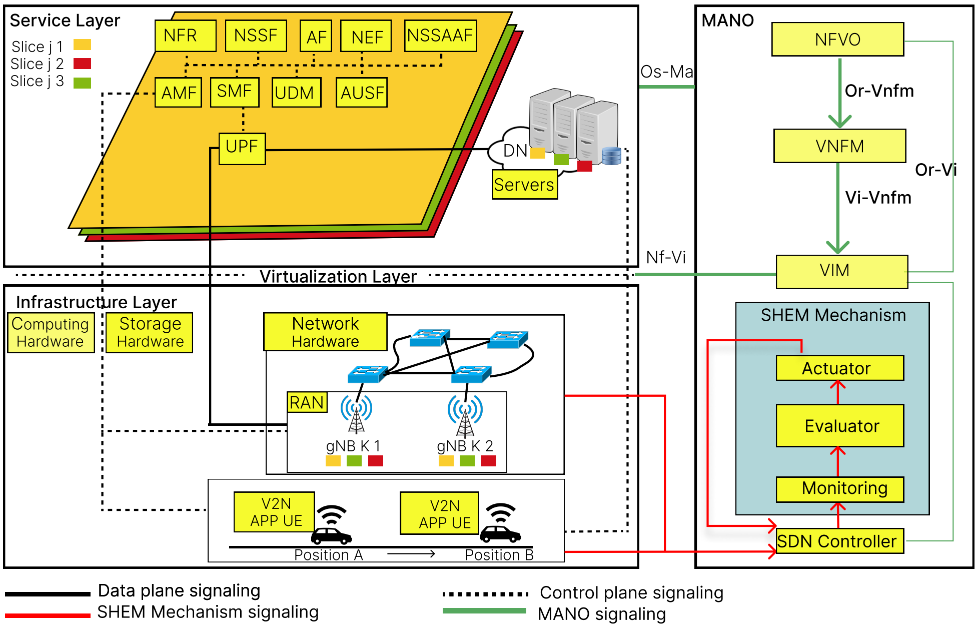


Figure 4.1: Overview of the Motivation Scenario Architecture

4.1.1.1 Infrastructure Layer

The infrastructure layer comprises the entire physical infrastructure, including computing, storage and network hardware. Therefore, the infrastructure layer comprises the following elements: UEs, RAN nodes, MEC servers and the transport network.

- UEs are vehicles controlled remotely through a V2N application hosted on remote MEC Servers. In addition, the vehicles must have a wireless interface to connect to the RAN nodes to communicate with the V2N application.

- RAN nodes represent the gNBs, optionally, these nodes could be SDN-enabled. Although these nodes can support any RAT, the RAT chosen was WiFi. From the WiFi amendments, 802.11g, 802.11i, and 802.11r were considered. In summary, 802.11g enables a BW of up to 54 Mbps between the gNBs, and vehicles [59]. The 802.11i enables Robust Security Network Association (RSNA) using WPA2 and 802.1X authorization framework [60–62]. And 802.11r (or FT) allows to reduce the WPA2/802.1X authentication process [63].
- MEC servers. MEC is an emerging technology with the main idea of implementing content-oriented intelligence. MEC brings content, NFs and resources closer to the end user, extending the conventional data center to the edge of the network [64]. MEC by locating closer to where data is generated and consumed, enables improvements such as high BW, ultra-low latency, and real-time RAN location awareness and information. Thus, these improvements provide cloud computing capabilities to host the V2N application, supporting your QoS requirements [65].
- The transport network interconnects the RAN nodes with the CN and DN (MEC Server) using SDN devices such as switches and routers.

4.1.1.2 Virtualization Layer

The virtualization layer creates an abstract view of the infrastructure layer resources and provides these resources/resource pools to the service layer for use. The virtualization layer not only virtualizes network resources such as RAN but can also virtualize compute and storage resources for the service layer. In this way, the virtualization layer can divide and isolate the virtual resources into several subgroups and assign each of the resource subgroups to a virtual network or network segment. Creating these network segments is also possible with the MANO cross-layer management explained below.

4.1.1.3 Management And Orchestration Layer

The Management ANd Orchestration (MANO) layer performs all management, coordination, and automation tasks specific to virtualization [66]. This layer includes NFV Orchestrator (NFVO), Virtualized Infrastructure Manager (VIM) and the VNFs Manager (VNFM). In addition, this layer integrates the SHEM prototype (proposed in Chapter 3) through the SDN controller.

- NFVO. This is a central management entity responsible for orchestrating the resources used concerning the infrastructure layer and the virtualization layer. It is also responsible for orchestrating network services, i.e., the functions deployed at the service layer.
- VNFM. Performs configuration and lifecycle management of VNFs in your domain.
- VIM. It helps manage NFV Infrastructure (NFVI) resources, i.e., infrastructure layer resources.
- SHEM prototype. It is implemented in the MANO layer because it has faster and more direct communication with the SDN controller. In this way, the SDN controller sends the input variables (network and vehicle statistics) required by the SHEM in less time. Analogously, SHEM can send the output variable (target slice to perform the handover) to the SDN controller, according to the design proposed in Sections 3.2 and 3.3.
- SDN controller. Through the SDN controller, traffic routes (of transport network) are established and can be automatically reconfigured to manage traffic engineering requirements (and network resources) or to react to possible network failures and changing conditions (e.g., HM).

4.2 Implementation Requirements

To implement the NS-based 5G network architecture presented in the previous section, it is clear that all network layers must be emulated or simulated. However, the 5G network implementation presents two sets of implicit requirements that must be considered. These two groups are the mobility requirements and the NS requirements.

4.2.1 Mobility Requirements

To implement mobility and vehicle-RAN interaction, network 5G must support the following requirements:

1. Simulation of a vehicular system that allows to configure parameters such as:
    - Speed
    - Address
    - Position
    - Trajectory
2. Emulation of gNB and vehicles that allows to configure the following parameters:
    - Position
    - 802.11g, 802.11i, and 802.11r amendments.
3. Implementation of an authentication server that supports the 802.1X framework.

4.2.2 Network Slicing Requirements

As mentioned in Subsection 2.1.3.3, this degree work considers NS at both the service and infrastructure layers. In the implementation, however, NS at the infrastructure layer is only realized in the RAN. Therefore, the implemented 5G network must satisfy the following requirements to support NS.

1. Deploy the service layer functions (described in Subection 2.1.3) that enable mobility management and NS in the 5G network.
2. Abstraction of network resources.
3. Establish MEC servers with QoS requirements such as latency and BW for the corresponding V2N application.
4. SDN-based switching capability required for network routing configuration.
5. Orchestration and visibility of the SDN network.

4.3 Tools for Implementation

Based on the architecture in Figure 4.1 and the implementation requirements in the previous section, each layer of the architecture is described below and in Figure 4.2 with the tools that implement the SHEM prototype in the NS-based 5G network.

- Service layer is implemented through FreeRadius[67] and Ryu[68]. FreeRadius is an 802.1X authentication server using the RADIUS[1] protocol, which manages the access and use of network resources. Therefore, FreeRadius reproduces the behavior of NSSAAF, NSSF, UDM, and AUSF, that collectively manage access to slice resources [37, 69]. On the other hand, Ryu implements the VNFs (AMF, SMF, and UPF) related to mobility management. Thus, only the 5G CN VNFs required for the manage access to slice resources were implemented.
- The virtualization layer is implicitly implemented by FreeRadius and Ryu, i.e., these tools internally configure the computational and storage resources needed for the VNFs of the service layer. With respect to network resources, these are virtualized by Mininet-WiFi and Ryu.

---

[1] Remote Authentication Dial-In User Service

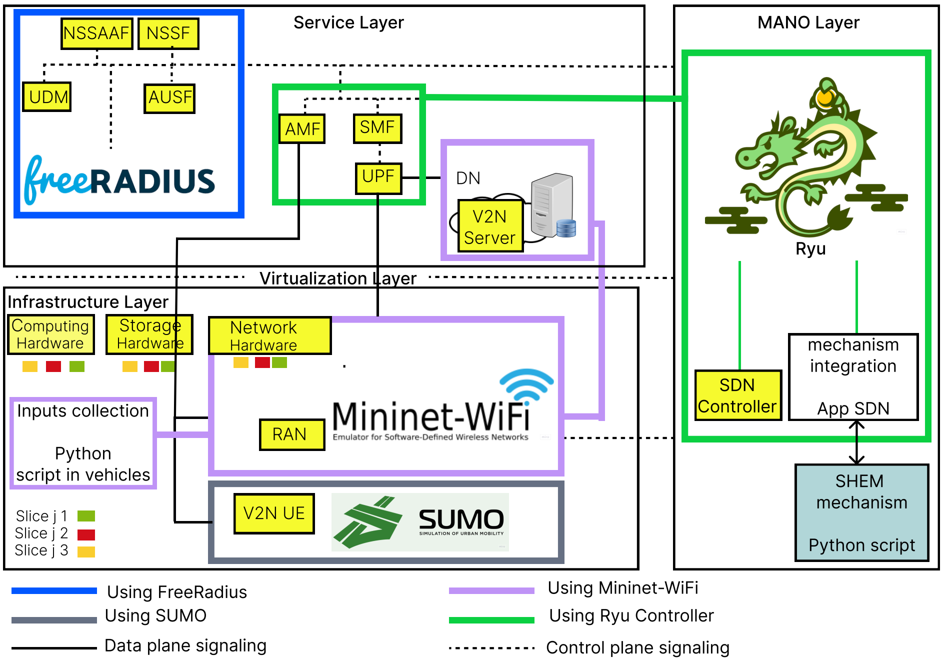


Figure 4.2: Interaction between tools

- MANO layer integrates the SHEM prototype and Ryu. Although this layer includes the entities NFVO, VNFM, and VIM, their implementations are null, since the scope of this work, only HM in terms of latency without slice resource management was addressed. However, these entities could be implemented through tools such as OSM[70]. In addition, the implementation of SHEM prototype was realized through a Python script (called SHEM_mechanism) explained in Section 3.3. Finally, the communication between Ryu and SHEM prototype was done through the SDN application called mechanism_integration (Python script). The scripts are presented in the GitHub repository exposed in Appendix A.

- The infrastructure layer is implemented thanks to Mininet-WiFi[71], and Simulator for Urban MObility (SUMO)[72]. Mininet-WiFi emulates all the network infrastructure, i.e., wireless stations (vehicles), access points (gNBs), MEC servers (V2N servers), switches, and links, together with 802.11g/i/r amendments. Meanwhile, SUMO simulates the traffic of the vehicles emulated by Mininet-WiFi. In addition, Mininet-WiFi must connect with the service layer and the MANO layer, i.e., with FreeRadius and Ryu.

Regarding the NS in the RAN, this is performed by Mininet-WiFi using the tools Host Access Point Daemon (HostAPD)[2] and Basic OpenFlow User Space Software Switch (BOFUSS)[74]. HostAPD allows to emulation gNBs with multiple virtual interfaces (virtual gNBs), where each slice corresponds to a WiFi network over the virtual gNB. Subsequently, BOFFUS configures the BW corresponding to each slice according to Table 3.1.

Additionally, the collection of the necessary inputs for the SHEM prototype must be done in this infrastructure layer. This collection is performed by each vehicle through a Python script called inputs_collection and exposed in the GitHub repository of Appendix A. Subsequent to the collection, the vehicle sends the inputs to the SHEM prototype through the SDN controller.

4.3.1 FreeRadius

FreeRadius is an open-source AAA-S written in Python that implements 802.1X authentication using RADIUS protocol [75]. 802.1X is an IEEE standard for port-based network access control. Port-based network access control allows a network administrator to restrict the use of gNBs (ports) to allow only authenticated and authorized UEs to communicate. The implementation of 802.1X authentication requires three essential components: i) Supplicant, corresponds to a software client running on the UEs (vehicle); ii) Authenticator, corresponds to the gNB; and iii) Authentication Server, corresponds to the RADIUS server, in this case, FreeRadius. Moreover, for the exchange of authentication information between the supplicant and the authentication server, Extensible Authentication Protocol (EAP) is used [62]. Thus, the authenticator is only a proxy that enables the communication between the supplicant and the authentication server.

To enable 802.1X authentication in 5G network, the Supplicants, Authenticators, and the Authentication Server, i.e., the vehicles, gNBs, and FreeRadius, were confi-

---

[2] HostAPD (Host Access Point Daemon)[73] is a user-space AP software capable of emulating the IEEE 802.11 standard through the conversion of normal network interface cards into APs and authentication servers.

gured. The configuration of the vehicles and gNBs has been realized by Mininet-WiFi (see Subsection 4.3.3). On the other hand, to configure FreeRadius, it was necessary to modify the files summarized in Table 4.1. Additionally, these files are presented in the GitHub repository exposed in Appendix A.

File	Description
freeradius/clients.conf	Used to define the authenticators (i.e. gNBs) with their password and IP address.
freeradius/mods-config/files/authorize	Used to define the supplicants (i.e. vehicles) along with their password. In addition, this file is used for both authorization and authentication.
freeradius/sites-available/default	It is a virtual server corresponding to the authentication server, which manages all requests by default. Therefore, this server receives and redirects the EAP authentication request to the inner-tunnel server.
freeradius/sites-available/inner-tunnel	It is a virtual server that manages the EAP requests. In addition, this server determines if the password sent by the supplicant is correct.
freeradius/mods-vailable/eap	It defines the EAP method to be used. It also determines the virtual server that will manage the EAP requests. The server determined is ”inner-tunnel”. And the EAP method used is TTLS [76].
freeradius/radiusd.conf	This file enables the authentication logs. By default, it is the main FreeRadius file, where it defines the location of the configuration files (to be executed) of the virtual servers, modules (e.g. EAP), and authenticators.

Table 4.1: FreeRadius configuration files

4.3.2 Ryu

Ryu is an open-source framework written in Python. Ryu provides software components with well-defined REST APIs that make it easy for developers to create new network management and control applications and abstract, orchestrate, and visualize network resources. Ryu supports various protocols for managing network devices, such as OpenFlow, Netconf, and OF-config. Concerning OpenFlow, Ryu supports the following versions: 1.0, 1.2, 1.3, 1.4, and 1.5.

Ryu to 5G network helps the mobility management through two Ryu applications, called simple_switch_13 and mechanism_integration. The simple simple_switch_13 application allows the switches (with OpenFlow 1.3) emulated by Mininet-WiFi to enable the routing of network flows. Regarding the mechanism_integration application, it allows communication between Ryu and the SHEM mechanism. Finally, these Ryu applications are presented in the GitHub repository exposed in Appendix A.

4.3.3 Mininet-WiFi

This is a branch of the OpenFlow-SDN telecom network emulator called Mininet [77]. Mininet-WiFi extends the functionality of Mininet by adding wireless stations (vehicles) and virtualized APs (gNBs) based on the standard Linux wireless driver and the 80211 hwsim wireless simulation driver. This means that Mininet-WiFi has added support for the 802.11 protocol in a Mininet network scenario.

To emulate the infrastructure layer presented in Section 3.1, it is through the script created in Python called infrastructure_layer.py (see the GitHub repository exposed in Appendix A) that contains the instructions to configure to Mininet-WiFi. This script creates the entire infrastructure layer, i.e., servers, switches, ethernet links, gNBs, and vehicles. It also configures 802.11g/i/r amendments, OpenFlow rules, routing, and IP addressing. In addition, it enables integration with Ryu and FreeRadius. The integration with Ryu is done internally by Mininet-WiFi through the IP address and port where Ryu is located. The integration with FreeRadius is more complex and is explained below.

Integration with FreeRadius is through the 802.11i amendment since it enables WPA2 and 802.1X. In this way, vehicles (Supplicants) and gNBs (Authenticators) can generate and store the keys for authentication with FreeRadius. To configure the WPA2/802.1X protocol in the gNBs and vehicles, Mininet-WiFi integrates the HostAPD and wpa_supplicant[3] tools. HostAPD allows Mininet-WiFi to gene-

---

[3] wpa_supplicant[78] emulates the IEEE 802.1X supplicant. It implements WPA key negotiation with a WPA authenticator (gNB) and authentication server. In addition, it enables binding between the authenticator and supplicant.

rate a daemon to each gNB to emulate 802.11g/i/r amendments together with the WPA2/802.1X configuration for communication with FreeRadius (authentication server). In addition, HostAPD also allows the configuration of multiple WiFi interfaces on each gNB and, thus, enables NS in the RAN. Concerning wpa_supplicant, it generates a daemon for each vehicle to enable the WPA2/802.1X protocol. In addition, wpa_supplicant supports gNB handovers and IEEE 802.11 authentication/association between the vehicle and gNB. The daemon configuration files for both the vehicles and the gNBs are presented in the GitHub repository exposed in Appendix A.

In summary, Table 4.2 presents the files used to emulate the infrastructure layer.

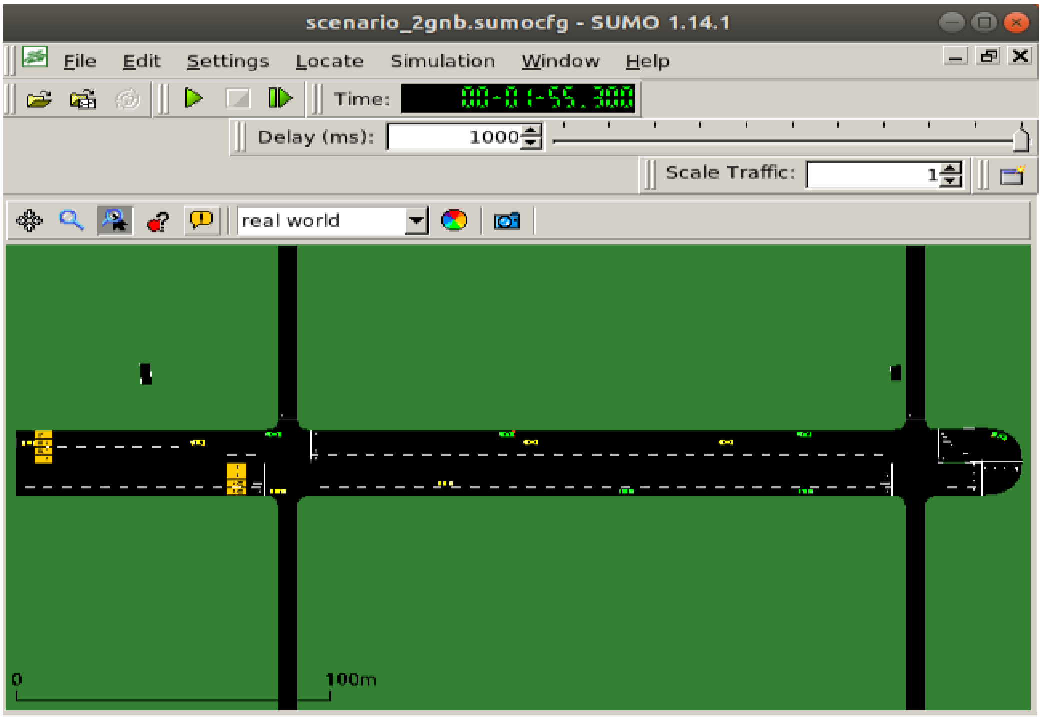
File	**Description**
infrastructure_layer.py	Configure Mininet-WiFi to emulate the infrastructure layer.
file.staconf	Used by wpa_supplicant to enable WPA2/802.1X protocol on vehicles.
file.apconf	Used by HostAPD to configure 802.11g/i/r amendments and WPA2/802.1X protocol on gNBs.

Table 4.2: Mininet-WiFi configuration files

4.3.4 Simulator for Urban Mobility

Simulator for Urban MObility (SUMO) is an open-source, microscopic, multi-modal traffic simulation package written in Python. SUMO allows the simulation of traffic demand (set of individual vehicles) moving through a given road network. On the other hand, SUMO is microscopic, i.e., each vehicle is explicitly modeled, has its route, and moves individually through the network. Thus, the simulations are deterministic by default, but several options exist to introduce randomness. In this way, SUMO can support a variety of traffic management situations, such as congestion and road closures.

Figure 4.3 shows the road network built for 5G network. This road network is 320 meters long, with 12 vehicles with speeds between 20 and 60 km/h. Table 4.3 shows the files needed to build the road network. In addition, these files are shown in the GitHub repository exposed in Appendix A.



Time: 00-01-55.300

Delay (ms): 1000

Scale Traffic: 1

Figure 4.3: Screenshot of the road network built at SUMO

File	Description
File.net.xml	Builds the traffic network with the descriptions of the streets or highways, with their corresponding lanes, intersections and connections.
File.rou.xml	Define the routes to be followed by each simulated vehicle, as well as the characteristics of the vehicle following each route.
File.add.xml	This file is optional, and contains definitions such as traffic lights, bus stops, among others.
File.settings.xml	This file is optional, and sets the parameters of the graphical user interface.
File.sumocfg	This is the main file that groups all the previous configuration files.

Table 4.3: SUMO configuration files

4.4 Handover Management Process

Previous to the HM process, the emulated 5G network must realize the 802.11i RSNA establishment between the vehicle and the gNB/slice. Subsequently, the vehicle can realize the HM together with SHEM prototype. The HM can be inter-slice or intra-slice. To continue, first the 802.11i RSNA establishment is detailed, and later the inter-slice and intra-slice HM are detailed.

Figure 4.4 shows the 802.11i RSNA establishment that starts with the preparation phase. The preparation phase starts when the vehicle sends the probe request to the neighboring gNBs/slices. With the probe responses, the vehicle chooses the gNB/slice to connect to. Here, the preparation phase ends, and the execution phase starts. In the execution phase, the vehicle initially realizes open authentication and association with the gNB/slice. Thus the vehicle is authenticated and associated with the gNB/slice. However, even access to the gNB/slice continues blocked, until meeting the set of security capabilities of the 802.11i amendment. This set comprises 802.1X authentication and 4-Way Handshake and Group Handshake key generation and caching. Therefore, when the Group Handshake terminates, the 802.1X port is deblocked, allowing the vehicle to access the gNB/slice resources. Finally, the vehicle informs the target gNB/slice that the 802.11i RSNA establishment has been successful, thus ending the execution phase and initiating the completion phase. In the completion phase, the gNB/slice requests the AMF to update information on mobility, session (in SMF), and routing (in UPF) of the RSNA establishment performed between the vehicle and the gNB/slice. Consequently, the 802.11i RSNA establishment is completed.

Regarding inter-slice and intra-slice HM, these are supported by SHEM prototype. The purpose of SHEM is to improve the preparation phase, proactively selecting (previous to the loss of communication with the source gNB/slice) the best target gNB/slice, considering the 5G network conditions and the QoS requirements of the application. For this purpose, SHEM includes three modules: Monitoring, Evaluator, and Actuator, defined in Section 3.2. Figure 4.5 evidences the operation of the modules of SHEM. In summary, the preparation phase starts when the vehicle finds

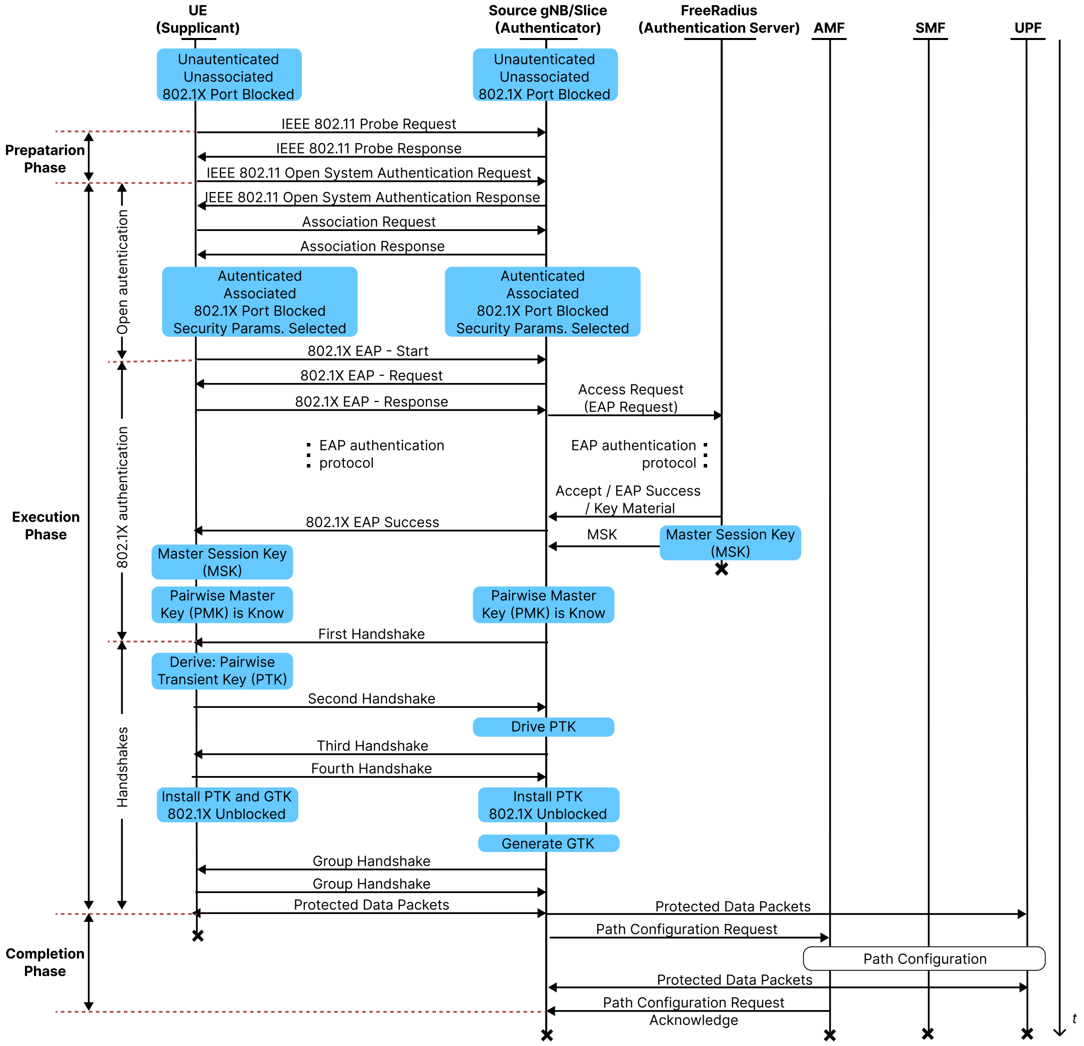


Figure 4.4: 802.11i RSNA establishment

the neighboring gNBs through beacon[4] capture. Having identified the neighboring gNBs, the vehicle collects and sends the SHEM inputs (defined in Section 3.2) to the Ryu application called mechanism_integration. Thus, this Ryu application sends these inputs to the Monitoring module, where it verifies and delivers the inputs to the Evaluator module. Evalutor module determines two options for the handover for the vehicle, i.e., whether the vehicle should execute: i) a HM to the gNB/slice that ensures meeting the latency required by the application executed by the vehicle; ii) a HM to the gNB/slice that ensures meeting the availability of resources.

---

[4] Beacons are frames transmitted periodically by gNBs, with the purpose of informing vehicles about nearby gNBs along with the channel status [79].

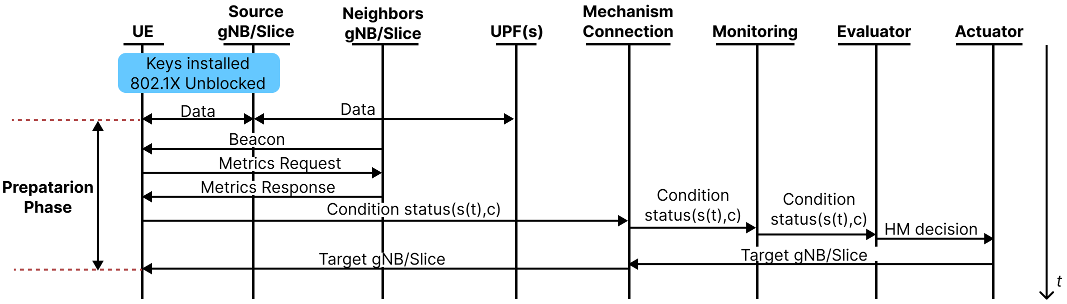


Figure 4.5: SHEM operation

Subsequently, these two options are sent to the Actuator module, which initiates the HM (to the execution phase), prioritizing the gNB/slice that meets the latency requirement of the vehicle. In case neither of the two options exists, the vehicle maintains the connection with the source gNB/slice.

Regarding the execution and completion phases, firstly, the inter-slice HM is explained, and secondly, the intra-slice HM is explained.

Figure 4.6 shows the execution and completion phases of the inter-slice HM. The execution phase employs the cached keys to omit 802.1X authentication between the vehicle and the target gNB/slice. Thus, the execution phase starts with the 4-Way Handshake process and the Group Handshake. Referring to the completion phase, the target gNB/slice performs two actions: i) inform the source gNB/slice to disassociate the vehicle to liberate the resources; ii) request the AMF to update information on mobility, session (in SMF), and routing (in UPF) of the HM realized between the vehicle and the target gNB/slice.

On the other hand, intra-slice HM in the execution phase omits 802.1X authentication and Handshakes (see Figure 4.7). This is because the HM is within the source gNB. Thus the 802.11r amendment can derive the complete set of 802.11i security capabilities previous to reassociation. Hence, the 802.11r amendment introduces a fast reassociation that incorporates handshakes into open authentication (also known as FT authentication) [63].

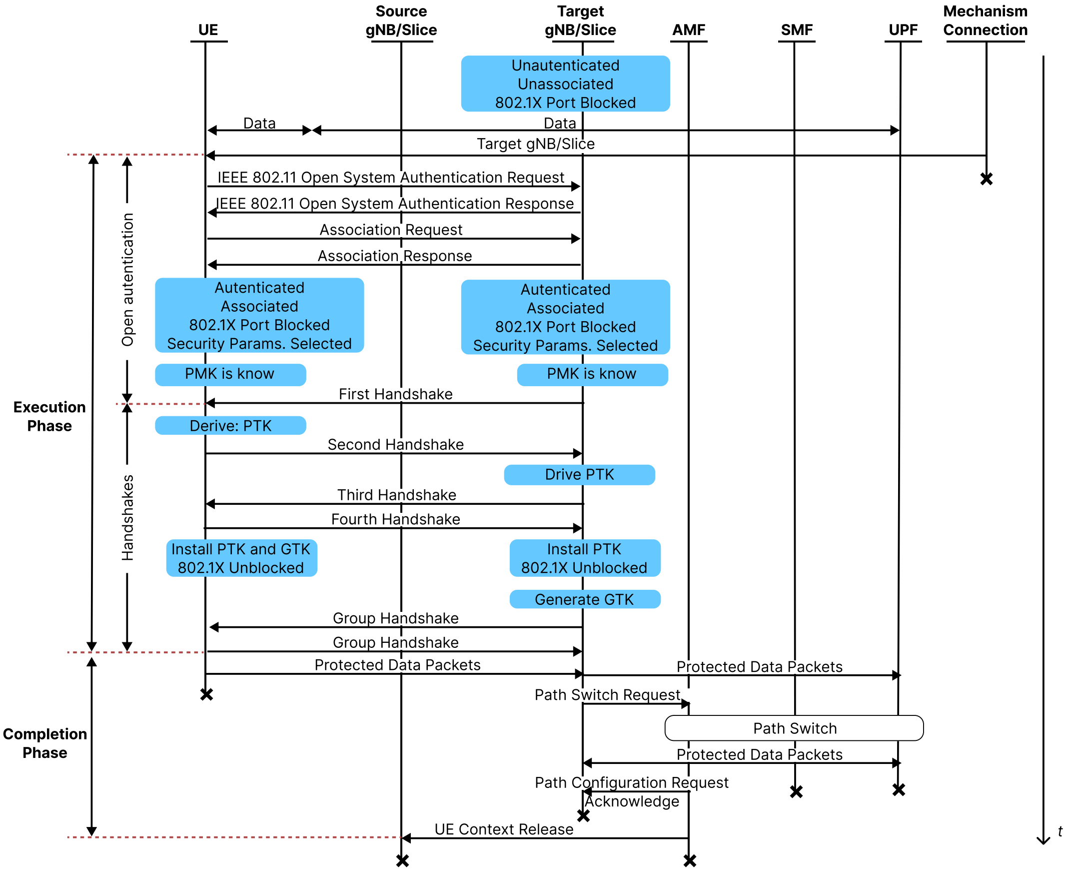


Figure 4.6: Inter-slice HM procedure

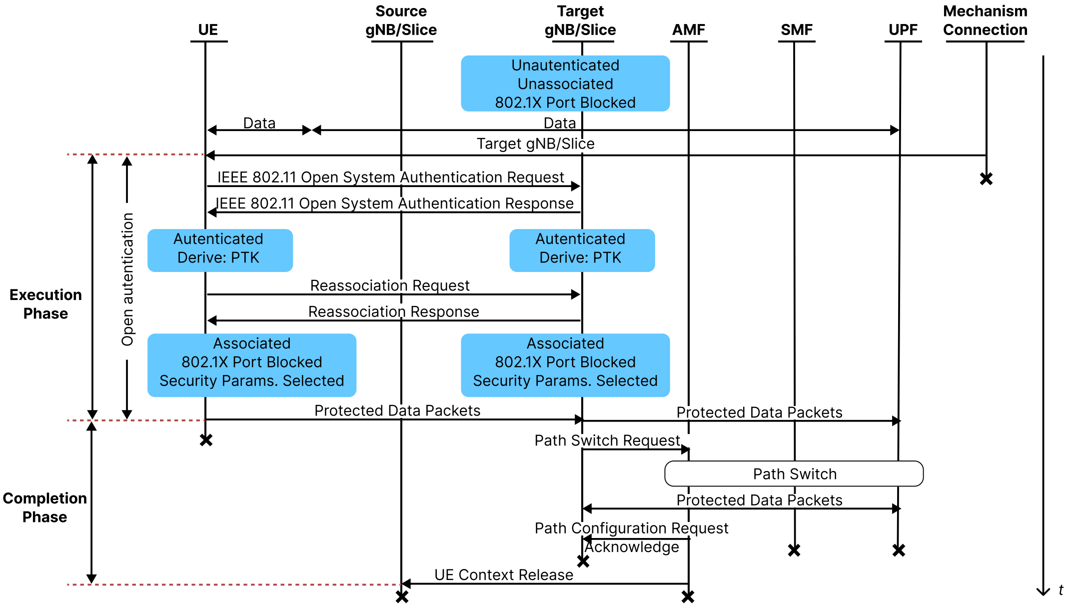


Figure 4.7: Intra-slice HM procedure

Chapter 5

Evaluation of Mechanism

This chapter presents the evaluation of the SHEM prototype implemented in the NS-based 5G network. First, the testbench is described. Second, the results obtained are presented. Thirdly, the results obtained are analyzed.

5.1 Testbed

The SHEM prototype was analytically evaluated to verify the latency in HM in the emulated 5G network. For this evaluation, the 5G network was configured with the implementation tools (presented in Section 4.3) on a VirtualBox[1] VM VirtualBox[80] with Ubuntu 18.04.6 LTS, Linux Kernel 5.8.18, 7GB RAM, and quad-core Intel i5 12600K. To realize the evaluation, initially the parameters of both the 5G network and the SHEM prototype were configured, according to Table 5.1. Subsequently, the evaluation was realized according to three tests. The first test checks the NS implementation, corroborating the bandwidth of each slice. The second test verifies the contribution of SHEM to the latency reduction in the HM of the 5G network. Moreover, the third test verifies the compliance of the latency requirement of V2N and Non-V2N applications after the HM.

---

[1] VirutalBox[80] is an open source software that allows virtualizing multiple VMs inside the native operating system.

Parameter		**Value**
SHEM	RSSI threshold	-65 dBm
	Slice load threshold	2
Network 5G	# gNBs	2
	# slices x gNB	3
	gNB coverage radius	250 m
	Beacon interval of gNB	50 kus
	Total number of vehicles	12
	# LLC vehicles	6
	# Non-LLC vehicles	6
	Speed of LLC vehicles	60
	Speed of non-LLC vehicles	20-40

Table 5.1: Testbed parameters

5.2 Results

5.2.1 Test 1: Network Slicing Check

This test checks the BW of the gNB slices, configured according to Table 3.1. For this, using the iPerf[81] tool, a vehicle transmitted 100 UDP flows (of 54 Mbps BW) through the WiFi slice interface to the corresponding application server. At the end of the transmission of each flow, iPerf delivers the server report, indicating the BW achieved in the transmission. In total, this test obtained 600 reports, given that the network has two gNBs, and each gNB has three slices. In the reports summarized in Figure 5.1, it is evident that the flow BW was reduced close to the configured BW.

5.2.2 Test 2: Handover Management Latency

To evaluate the contribution of SHEM to reducing the HM latency of the emulated 5G network, the HM process described in Section 4.4 was considered. From there, it is evident that the HM latency integrates the durations of the preparation, execution, and completion phases (Equation 5.1). Regarding the preparation phase, this is performed by SHEM except in the 802.11i RSNA establishment. With respect to the execution phase, it includes the durations of the authentication and open associ-

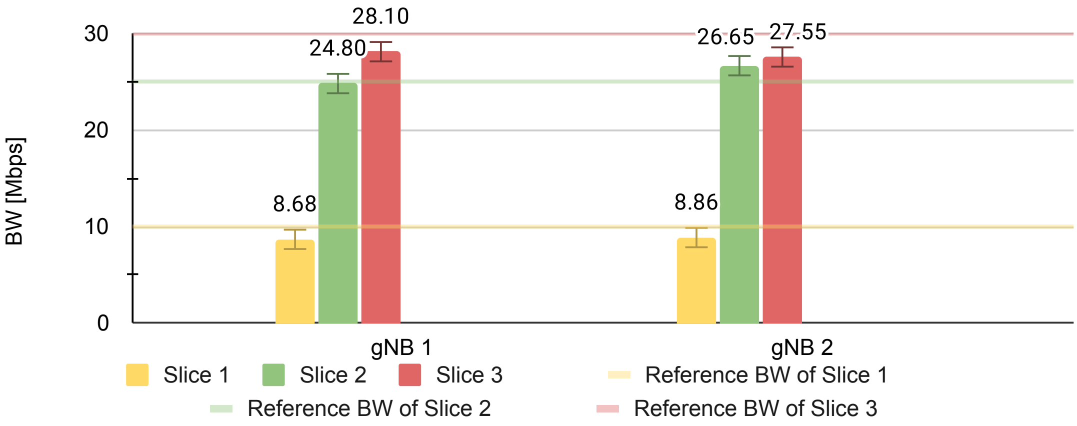


Figure 5.1: Test 1: Verification of BW assigned to each slice

ation, 802.1X authentication, 4-way handshake, and group handshake, as shown in Equation 5.2. Moreover, for intra-slice HM, the durations of 802.1X authentication, 4-way handshake, and group handshake are zero. In summary, Table 5.2 presents the above parameters that make up the HM latency.

$$L_{HM} = T_{prep} + T_{exec} + T_{comp} \tag{5.1}$$

$$T_{exec} = T_{open} + T_{802.1X} + T_{4way} + T_{g\_h} \tag{5.2}$$

Symbol	**Definition**
L_{HM}	HM latency
T_{open}	Duration of open authentication
$T_{802.1X}$	Duration of 802.1X Authentication
T_{4way}	Duration of 4-way handshake
T_{g_h}	Duration of the group handshake
T_{prep}	Duration of the preparation phase
T_{exec}	Duration of the execution phase
T_{comp}	Duration of the completion phase

Table 5.2: HM duration parameters

To obtain the durations of the preparation and execution phases, the wpa_supplicant logs of the vehicle were used. While to obtain the duration of the completion phase, it was obtained directly through timestamps inside the Ryu application called mechanism_integration. In this way, these results were collected in the dataset called test2.csv (see the GitHub repository exposed in Appendix A). Furthermore, with this dataset, it was obtained that SHEM allows omitting the preparation phase in the inter-slice and intra-slice HM (see Figure 5.2). Thus, compared to the 802.11i RSNA establishment, SHEM reduced about 3700 ms of the preparation phase.

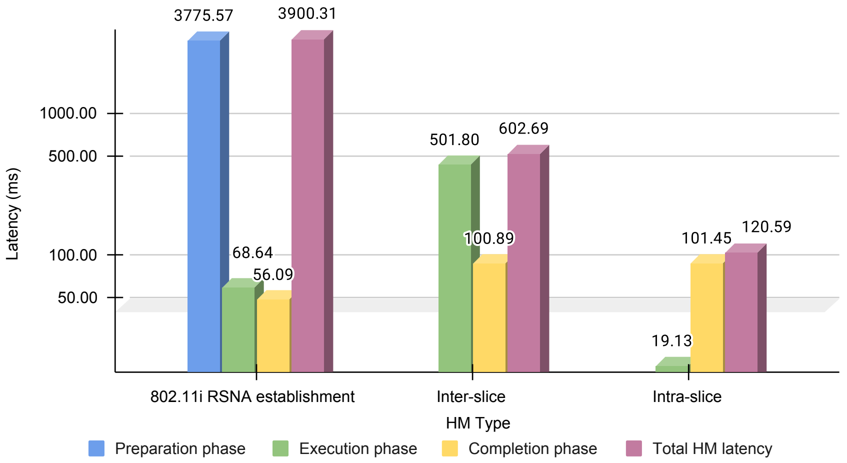


Figure 5.2: Test 2: HM Latency

5.2.3 Test 3: Verification of Meeting the Application Latency Requirement in the Handover Management

This test verifies the effectiveness of SHEM in selecting the target gNB/slice that satisfies the E2E latency requirement of the application executed by the vehicle. For this purpose, Table 3.1 configures both the minimum E2E latency offered by each slice and the maximum E2E latency allowed by each vehicle. Thus, the vehicle E2E latency requirement will be met (effectiveness) as long as the E2E latency of the selected target slice is equal to or less than the E2E latency value required by the vehicle, as shown in Table 5.3. However, this selection is sometimes infeasible since the appropriate destination slice may be overloaded with vehicles. For this reason, the effectiveness of SHEM in selecting the target gNB/slice is tested.

Maximum E2E Latency Allowed by each vehicle	Minimum E2E Latency offered by each slice		
	Slice Non-V2N (50 ms)	Slice V2N-1 (20 ms)	Slice V2N-2 (10 ms)
Vehicle App (50 ms)	✓	✓	✓
Vehicle App (20 ms)		✓	✓
Vehicle App (10 ms)			✓

Table 5.3: gNB/slice target ideal for meeting the E2E latency requirement of the application run by vehicle

To check the effectiveness of SHEM,the inter-slice and intra-slice HMs of the dataset generated in Test 2 are analyzed. In total, there are 844 HMs, 556 inter-slice, and 288 intra-slice. In each type of HM, the effectiveness was checked, obtaining that of the 556 inter-slice HM, 404 (72.7%) were effective. And of the 288 intra-slice HM, 214 (74.3%) were effective. Thus, SHEM obtained an average 73.5% effectiveness rate. Figure 5.3 summarizes the above results obtained by SHEM.

$$\%_{Effectiveness} = \frac{Total\ number\ of\ HM}{Number\ of\ effective\ HM} * 100 \tag{5.3}$$

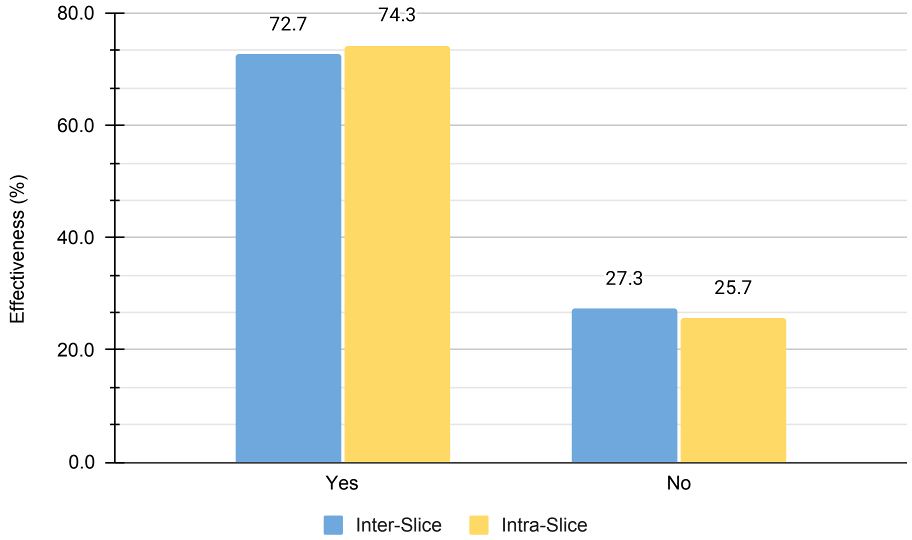


Figure 5.3: Test 3: Effectiveness percentage of latency meeting in HM

5.3 Final Remarks

From the tests, SHEM has two contributions to the HM of the emulated 5G network together with NS. The first contribution is the omission of the HM preparation phase, thus obtaining a reduction of about 3700 ms. The second contribution is the 73.5% effectiveness in determining the target gNB/slice in order to meet the latency requirement of the application executed by the vehicle. These contributions are due to the latency requirement that SHEM considers to determine the target gNB/slice proactively and passively. That is, before channel degradation and without interrupting the communication between the vehicle and the source gNB/slice. Furthermore, SHEM is viable in both inter-slice HM and intra-slice HM. Therefore, SHEM is a solution for HM latency reduction in 5G networks based on NS.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

This work presented the proposed solution to answer the research question: **How to meet LLC requirement in the HM process at 5G?**

A relevant problem in 5G mobile networks is that it interrupts communication with the UE when the UE must switch gNB (HM) due to mobility or QoS degradation. Several works aimed to solve this problem, but they present drawbacks such as the capability of being proactive, omitting the virtualization of resources (NS), and avoiding a seamless HM, i.e., without interrupting the communication between UE and 5G network. Aiming to overcome this problem, in this work, SHEM was proposed, which aims to select the target gNB/slice proactively, passively, and aware of the latency requirement of both the UE and the one offered by the slice.

SHEM is based on a three-module design (monitoring, evaluator, and actuator) that can be implemented and adapted with both heuristic programming and machine learning techniques. Moreover, SHEM can operate in both inter-slice HM and intra-slice HM. In particular, SHEM was implemented using heuristic programming. Where SHEM determines the target gNB/slice according to the latency requirement of the gNB/slice. Subsequently, SHEM proactively initiates the HM through the

SDN Controller.

Therefore, the test results demonstrate that SHEM completely omits the HM preparation phase. Thus, obtaining a significant reduction of the HM latency with good effectiveness in selecting the target gNB/slice.

6.2 Future Works

According to the work done for the development of this project, some ideas for future work are presented below:

- Extend the design of the SHEM to be able to address: i) the execution phase and the completion phase of HM; ii) eMBB and mMTC usage scenarios, as well as emerging 6G usage scenarios; iii) NS in both the NC as in RAN; and iv) others QoS requirements.
- Implement SHEM using other techniques, e.g., machine learning.
- Implement SHEM over a physical 5G network.
- Evaluate the performance of SHEM in terms of resource consumption.

Bibliography

[1] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. C. M. Leung, “Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges,” *IEEE Communications Magazine*, vol. 55, pp. 138–145, Aug 2017.

[2] A. Raschellà, F. Bouhafs, M. Seyedebrahimi, M. Mackay, and Q. Shi, “Quality of Service Oriented Access Point Selection Framework for Large Wi-Fi Networks,” *IEEE Transactions on Network and Service Management*, vol. 14, pp. 441–455, June 2017.

[3] G. Athanasiou, T. Korakis, and L. Tassiulas, “An 802.11k Compliant Framework for Cooperative Handoff in Wireless Networks,” *EURASIP J. Wireless Comm. and Networking*, vol. 2009, 01 2009.

[4] R. A. Addad, D. L. C. Dutra, T. Taleb, M. Bagaa, and H. Flinck, “MIRA!: An SDN-Based Framework for Cross-Domain Fast Migration of Ultra-Low Latency 5G Services,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2018.

[5] C. W. Huawei, “Challenges in Networking to Support Augmented Reality and Virtual Reality,” 2016.

[6] P. Popovski, Stefanović, J. J. Nielsen, E. de Carvalho, M. Angjelichinoski, K. F. Trillingsgaard, and A. Bana, “Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC),” *IEEE Transactions on Communications*, vol. 67, pp. 5783–5801, Aug 2019.

[7] C. Campolo, R. Fontes, A. Molinaro, C. E. Rothenberg, and A. Iera, “Slicing on the Road: Enabling the Automotive Vertical through 5G Network Softwarization,” *Sensors*, vol. 18, no. 12, p. 4435, 2018.

[8] M. Nabiyouni, S. Scerbo, D. A. Bowman, and T. Höllerer, “Relative Effects of Real-world and Virtual-World Latency on an Augmented Reality Training Task: An AR Simulation Experiment,” *Frontiers in ICT*, vol. 3, p. 34, 2017.

[9] “Cisco annual internet Report - Cisco Annual Internet Report (2018–2023) White Paper,” Jan 2022.

[10] “IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements,” *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*, pp. 1–212, Nov 2005.

[11] A. Malik, J. Qadir, B. Ahmad, K.-L. A. Yau, and U. Ullah, “QoS in IEEE 802.11-based Wireless Networks: A Contemporary Review,” *Journal of Network and Computer Applications*, vol. 55, pp. 24 – 46, 2015.

[12] A. Mukherjee, “Energy Efficiency and Delay in 5G Ultra-Reliable Low-Latency Communications System Architectures,” *IEEE Network*, vol. 32, pp. 55–61, March 2018.

[13] A. A. Tabassam, H. Trsek, S. Heiss, and J. Jasperneite, “Fast and Seamless Handover for Secure Mobile I Applications with 802.11r,” in *2009 IEEE 34th Conference on Local Computer Networks*, pp. 750–757, Oct 2009.

[14] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann, “OpenSDWN: Programmatic Control over Home and Enterprise WiFi,” in *Proceedings of the 1st ACM SIGCOMM symposium on software defined networking research*, pp. 1–12, 2015.

[15] A. Elakkiya and P. Selvaraj, “QoS based IP mobility management scheme for

the next generation SDN-LTE network," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 1355–1360, Jan 2018.

[16] Y. Kirsal, V. V. Paranthaman, and G. Mapp, "Exploring analytical models to maintain Quality-of-Service for Resource Management using a proactive approach in highly mobile environments," in *2018 7th International Conference on Computers Communications and Control (ICCCC)*, pp. 176–182, May 2018.

[17] S. Aghabozorgi, A. Bayati, K. Nguyen, C. Despins, and M. Cheriet, "Toward Predictive Handover Mechanism in Software-Defined Enterprise Wi-Fi Networks," in *2019 IEEE Sustainability through ICT Summit (StICT)*, pp. 1–6, June 2019.

[18] I. Farris, T. Taleb, H. Flinck, and A. Iera, "Providing Ultra-Short Latency to User-Centric 5G Applications at the Mobile Network Edge," *Transactions on Emerging Telecommunications Technologies*, Mar 2017.

[19] I. Farris, T. Taleb, M. Bagaa, and H. Flick, "Optimizing Service Replication for Mobile Delay-sensitive Applications in 5G Edge Network," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2017.

[20] R. A. Addad, D. L. Cadette Dutra, M. Bagaa, T. Taleb, and H. Flinck, "Towards a Fast Service Migration in 5G," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1–6, Oct 2018.

[21] S. Zhou, P. P. Netalkar, Y. Chang, Y. Xu, and J. Chao, "The MEC-Based Architecture Design for Low-Latency and Fast Hand-Off Vehicular Networking," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–7, Aug 2018.

[22] K. Nakauchi, Z. Lei, Y. Shoji, Y. Kitatsuji, M. Ito, and H. Yokota, "Bring Your Own Network — Design and Implementation of a Virtualized WiFi Network," in *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*, pp. 483–488, Jan 2014.

[23] K. Han, S. Li, S. Tang, H. Huang, S. Zhao, G. Fu, and Z. Zhu, "Application-Driven End-to-End Slicing: When Wireless Network Virtualization Orches-

trates With NFV-Based Mobile Edge Computing," *IEEE Access*, vol. 6, pp. 26567–26577, 2018.

[24] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards Programmable Enterprise WLANS with Odin," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, (New York, NY, USA), pp. 115–120, ACM, 2012.

[25] L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, and J. Almodovar, "Building an SDN Enterprise WLAN Based on Virtual APs," *IEEE Communications Letters*, vol. 21, pp. 374–377, Feb 2017.

[26] A. S. Dayem Alfoudi, M. Dighriri, G. M. Lee, R. Pereira, and F. P. Tso, "Traffic management in LTE-WiFi slicing networks," in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pp. 268–273, March 2017.

[27] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[28] F. Estrada-Solano, A. Ordonez, L. Z. Granville, and O. M. C. Rendon, "A Framework for SDN Integrated Management based on a CIM Model and a Vertical Management Plane," *Computer Communications*, vol. 102, pp. 150 – 164, 2017.

[29] J. Chen, B. Liu, H. Zhou, Q. Yu, L. Gui, and X. . Shen, "QoS-Driven Efficient Client Association in High-Density Software-Defined WLAN," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 7372–7383, Aug 2017.

[30] N. Kiran, C. Yin, Y. Hu, Z. Arain, C. Pan, I. Khan, Y. Zhang, and G. M Shafiqur Rahman, "Handover Based on AP Load in Software Defined Wi-Fi Systems," *Journal of Communications and Networks*, vol. 19, January 2017.

[31] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing

using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, p. 106984, 2020.

[32] ETSI, "Network Functions Virtualisation (NFV); Architectural Framework," *ETSI GS NFV 002 V1.2.1*, Dec 2014.

[33] M. Richart, J. Baliosian, J. Serrat, and J. Gorricho, "Resource Slicing in Virtual Wireless Networks: A Survey," *IEEE Transactions on Network and Service Management*, vol. 13, pp. 462–476, Sep. 2016.

[34] A. Devlic, A. Hamidian, D. Liang, M. Eriksson, A. Consoli, and J. Lundstedt, "NESMO: Network Slicing Management and Orchestration Framework," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1202–1208, May 2017.

[35] A. J. Gonzalez, G. Nencioni, A. Kamisiński, B. E. Helvik, and P. E. Heegaard, "Dependability of the NFV Orchestrator: State of the Art and Research Challenges," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3307–3329, 2018.

[36] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an SDN-enabled NFV architecture," *IEEE Communications Magazine*, vol. 53, pp. 187–193, April 2015.

[37] 3rd Generation Partnership Project (3GPP), "System architecture for the 5G System (5GS)," (Release 15), 3GPP TS 23.501 v15.13.0, Mar 2022.

[38] J. J. Nielsen, R. Liu, and P. Popovski, "Ultra-Reliable Low Latency Communication Using Interface Diversity," *IEEE Transactions on Communications*, vol. 66, pp. 1322–1334, March 2018.

[39] I. Vision, "IMT Vision–Framework and overall objectives of the future development of IMT for 2020 and beyond," *ITU-R Recommendation*, p. 2083, 2015.

[40] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. Leung, "Network Slicing Based 5g and Future Mobile Networks: Mobility, Resource Management, and Challenges," *IEEE communications magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[41] N. Akkari and N. Dimitriou, “MOBILITY MANAGEMENT SOLUTIONS FOR 5G NETWORKS: ARCHITECTURE AND SERVICES,” *Computer Networks*, vol. 169, p. 107082, 2020.

[42] International Telecommunication Union (ITU), “Framework of the IMT-2020 network,” ITU-T Recommendation Y.3102, May 2018.

[43] V. S. Pana, O. P. Babalola, and V. Balyan, “5g radio access networks: A survey,” *Array*, vol. 14, p. 100170, Jul 2022.

[44] M. Bennis, M. Debbah, and H. V. Poor, “Ultra-Reliable and Low-Latency Wireless Communication: Tail, Risk, and Scale,” *Proceedings of the IEEE*, vol. 106, pp. 1834–1853, Oct 2018.

[45] J. Anderson, H. Hu, U. Agarwal, C. Lowery, H. Li, and A. Apon, “Performance Considerations of Network Functions Virtualization using Containers,” in *2016 International Conference on Computing, Networking and Communications (ICNC)*, pp. 1–7, Feb 2016.

[46] *Study on Scenarios and Requirements for Next Generation Access Technologies*, no. (Release 15), document 3GPP TR 38.913 V15.0.0, Jun 2018.

[47] 3rd Generation Partnership Project (3GPP), “Study on enhancement of 3GPP support for 5G V2X services,” (Release 15), 3GPP TR 22.886 v15.3.0, Sep 2018.

[48] 3rd Generation Partnership Project (3GPP), “NR; NR and NG-RAN Overall description; Stage-2,” (Release 15), 3GPP TS 38.300 v15.13.0, Jul 2021.

[49] 3rd Generation Partnership Project (3GPP), “Procedures for the 5G System (5GS),” (Release 15), 3GPP TS 23.502 v15.16.0, Jun 2022.

[50] S. Feirer and T. Sauter, “Seamless handover in industrial WLAN using IEEE 802.11k,” in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1234–1239, June 2017.

[51] N. Mouawad, R. Naja, and S. Tohme, “Inter-slice handover management in a V2X slicing environment using bargaining games,” *Wireless Networks*, vol. 26, no. 5, pp. 3883–3903, 2020.

[52] A. Nayak Manjeshwar, P. Jha, A. Karandikar, and P. Chaporkar, “VirtRAN: An SDN/NFV-Based Framework for 5G RAN Slicing,” *Journal of the Indian Institute of Science*, vol. 100, no. 2, pp. 409–434, 2020.

[53] European Telecommunications Standards Institute (ETSI), “Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework,” ETSI GS NFV-EVE 005 v1.1.1, Dec 2015.

[54] 3rd Generation Partnership Project (3GPP), “Study on Architecture for Next Generation System,” (Release 14), 3GPP TR 23.799 v14.0.0, Dec 2016.

[55] X. Zhou, R. Li, T. Chen, and H. Zhang, “Network Slicing as a Service: Enabling Enterprises’ Own Software-Defined Cellular Networks,” *IEEE Communications Magazine*, vol. 54, no. 7, pp. 146–153, 2016.

[56] 3rd Generation Partnership Project (3GPP), “Service requirements for enhanced V2X scenarios,” (Release 15), 3GPP TS 22.186 v15.4.0, Sep 2018.

[57] P. Sun, N. AlJeri, and A. Boukerche, “An Energy-Efficient Proactive Handover Scheme for Vehicular Networks Based on Passive RSU Detection,” *IEEE Transactions on Sustainable Computing*, vol. 5, no. 1, pp. 37–47, 2020.

[58] A. Kammoun, N. Tabbane, G. Diaz, A. Dandoush, and N. Achir, “End-to-End Efficient Heuristic Algorithm for 5G Network Slicing,” in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pp. 386–392, 2018.

[59] “IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band,” *IEEE Std 802.11g-2003 (Amendment to IEEE Std 802.11, 1999 Edn. (Reaff 2003) as amended by IEEE Stds 802.11a-1999, 802.11b-1999, 802.11b-1999/Cor 1-2001, and 802.11d-2001)*, pp. 1–104, 2003.

[60] “IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks -

Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. 1–1076, 2007.

[61] "IEEE Standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements," *IEEE Std 802.11i-2004*, pp. 1–190, 2004.

[62] B. Aboba and P. Calhoun, "RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)." RFC 3579, Sept. 2003.

[63] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 2: Fast basic service set (bss) transition," *IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008)*, pp. 1–126, 2008.

[64] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[65] C. Tselios and G. Tsolis, "On qoe-awareness through virtualized probes in 5g networks," in *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, pp. 159–164, 2016.

[66] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.

[67] "FreeRADIUS." `https://freeradius.org/`.

[68] "Ryu." `https://ryu-sdn.org/`.

[69] 3rd Generation Partnership Project (3GPP), "5G System; Interworking between 5G Network and external Data Networks," (Release 15), 3GPP TS 29.561 v15.7.0, Jun 2021.

[70] "Open Source MANO (OSM)." `https://osm.etsi.org/`.

[71] "Mininet-WiFi." `https://github.com/intrig-unicamp/mininet-wifi`.

[72] "Simulation of Urban MObility (SUMO)." `https://www.eclipse.org/sumo/`.

[73] "Host Access Point Daemon (HostAPD)." `https://w1.fi/hostapd/`.

[74] "Basic OpenFlow Software Switch (BOFUSS)." `https://github.com/CPqD/ofsoftswitch13`.

[75] A. Rubens, C. Rigney, S. Willens, and W. A. Simpson, "Remote Authentication Dial In User Service (RADIUS)." RFC 2865, June 2000.

[76] P. Funk and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)." RFC 5281, Aug. 2008.

[77] "Mininet." `http://mininet.org/`.

[78] "wpa_supplicant." `https://w1.fi/wpa_supplicant/`.

[79] "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," *IEEE Std 802.11-1997*, pp. 1–445, 1997.

[80] "VirtualBox." `https://www.virtualbox.org/`.

[81] "iPerf." `https://iperf.fr/`.

[82] "Mininet-WiFi with Containernet." `https://mininet-wifi.github.io/containernet/`.

[83] C. F. Ugnaya Mendoza, "Implementation and experimental evaluation of Cooperative Awareness Basic Service for V2X Communications," Master's thesis, Universitat Politècnica de Catalunya, 2019.

[84] "IEEE 802.11p." `https://mininet-wifi.github.io/80211p`.

[85] "Mininet-WiFi Discussion Group." `https://groups.google.com/g/mininet-wifi-discuss`.

[86] "Reproducible Research." `https://github.com/ramonfontes/reproducible-research`.

[87] "5G Infrastructure Public Private Partnership (5G PPP)." `https://5g-ppp.eu/`.

[88] "Awesome 5G." `https://github.com/calee0219/awesome-5g`.

Handover Mechanism to Meet Low-Latency Application Requirement in 5G

APPENDICES

Undergraduate Work

Yeison Eduardo Caicedo Garzon

Andres Sebastian Garzon Jimenez

Advisor: Msc. Fulvio Yesid Vivas Cantero
Co-Advisor: PhD. Oscar Mauricio Caicedo Rendón

Department of Telematics
Faculty of Electronic and Telecommunications Engineering
Universidad del Cauca
Popayan, Cauca, 2022

A GitHub Repository

Appendix A presents the contents of the GitHub repository that hosts the implementation and evaluation code of SHEM mechanism. Table A.1 presents the structure of the GitHub repository available at:

`https://github.com/AndresGarzonJ/SHEM_Mechanism`

Folder	**Description**
FreeRadius	This directory contains the files needed to configure the RADIUS server.
SHEM_Mechanism	This directory contains the script of SHEM mechanism.
Mininet-WiFi	This directory contains the files that create the network infrastructure emulated in Mininet-WiFi. In addition, it contains the script that executes the vehicle for the collection and sending of the inputs of SHEM mechanism, and contains the script that allows to perform Test 1.
Ryu	This directory contains the scripts of the SDN applications executed by Ryu.
SUMO	This directory contains the files that allow to create the road infrastructure to simulate the mobility of the vehicles emulated by Mininet-WiFi.
Test_results	This directory contains the test datasets.

Table A.1: GitHub Repository Structure

B Draft Article

Appendix B presents the draft article for publication.

- **Andres S. Garzon, Yeison E. Caicedo**, Fulvio Y. Vivas Cantero and Oscar Mauricio Caicedo Rendon. **SHEM: Handover Mechanism to meet Low-Latency application Requirement in 5G.** Applied Science
  - Status: Draft.
  - Link to article: `https://github.com/YeisonHunt/article_SHEM`

C Recommendations

Appendix C presents recommendations for investigators who wish to continue the improvement of SHEM.

- In the implementation process, it is possible to have version incompatibility between different operating system dependencies. A solution to this is using virtual environments, which allow a customized user space.
- Although SHEM lacks a virtualization platform like OSM with OpenStack, a first step could be to implement containers in Mininet-WiFi with the Containert tool [82].
- For emulation of 5GHz frequencies in Mininet-WiFi, it is recommended to use an Ubuntu Kernel higher than version 5.5. And ideally compile the Kernel in a custom way, as shown in [83, 84].
- To increase the knowledge of Mininet-WiFi usage, the Mininet-WiFi community [85] is an excellent help. This community is very active and allows to share ideas and develop new features of Mininet-WiFi. In addition, [86] is a repository on GitHub that contains research done on Mininet-WiFi that can be reproducible.
- Regarding implementing the 5G core, the 5G Infrastructure Public Private Partnership (5G PPP) website has several projects [87]. Also, [88] has a list of other 5G projects.
- Considering the different versions of the programs. Creating APIs inside containers to serve the request and return the responses seems a good solution for compatibility issues.
- Creating the testbed itself could be a challenge. Therefore, before choosing the set of tools, it is advisable to test the features offered by each tool on a small scale. In this way, the tool may or may not be discarded before the implementation.