FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Slicing-aware Flying Communications Network

João Cristiano Mourão Rodrigues

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Dr. Rui Campos

July 28, 2022

Abstract

With the increasing usage of communications devices, network slicing emerged in 5G networks as a key component to enable the use of multiple communications services with different performance requirements on top of a shared physical network infrastructure. At the same time, flying networks, composed of Unmanned Aerial Vehicles (UAVs), emerged as an adequate communications solution to provide on-demand wireless coverage and communications resources anywhere, anytime. Still, most of the solutions lie on best-effort flying networks.

This Dissertation developed and validated a slicing-aware flying network, composed of UAVs that carry Wi-Fi Access Points, able to provide network slices at target geographical areas. The proposed solution allows minimizing the amount of communications resources used by the flying network, namely the number of UAVs and channel bandwidth, while defining suitable UAV positions.

For that purpose, we propose a novel Slicing-aware Resource Allocation Framework (SARAF) able to dynamically change multiple network configurations on-demand so that a slicing-aware configuration is achieved, while also allowing to collect and process performance metrics for monitoring the Quality of Service (QoS) offered by the network slices made available. In its current version, SARAF employs a slicing-aware state of the art algorithm, called SLICER, which is used to define a suitable placement and allocation of the communications resources in the proposed slicing-aware flying network. Still, SARAF can run together with other state of the art algorithms, representing a contribution to the community for achieving reconfigurable, autonomous networks.

The performance evaluation carried out using a testbed allowed to validate the proposed slicing-aware flying network, which takes advantage of SARAF and SLICER for defining the placement and allocation of communications resources. The resulting flying network meets the requirements associated with multiple slices, while using the minimum amount of communications resources.

Resumo

Com o crescente uso de dispositivos de comunicações, o fatiamento de rede surgiu nas redes 5G como um componente chave para permitir o uso de vários serviços de comunicações com diferentes requisitos de desempenho sobre uma infraestrutura de rede física compartilhada. Ao mesmo tempo, as redes voadoras, compostas por Veículos Aéreos Não Tripulados (UAVs), surgiram como uma solução de comunicações adequada para fornecer cobertura sem fios a pedido e recursos de comunicações em qualquer lugar, a qualquer hora. Ainda assim, a maioria das soluções assenta em redes voadoras *best-effort*.

Esta Dissertação desenvolveu e validou uma rede voadora sensível ao fatiamento de rede, composta por UAVs que transportam pontos de acesso Wi-Fi, capaz de fornecer fatias de rede em áreas geográficas alvo. A solução proposta permite minimizar a quantidade de recursos de comunicações utilizados pela rede voadora, nomeadamente o número de UAVs e a largura de banda do canal, ao mesmo tempo que define posições adequadas para os UAVs.

Para isso, propomos uma nova metodologia de alocação de recursos sensível ao fatiamento (SARAF) capaz de alterar dinamicamente várias configurações de rede a pedido para que uma configuração de fatiamento de rede seja alcançada, além de permitir recolher e processar métricas de desempenho para monitorização da Qualidade de Serviço (QoS) oferecida pelas fatias de rede disponibilizados. Na sua versão atual, o SARAF emprega um algoritmo do estado da arte sensível ao fatiamento de rede, chamado SLICER, que é usado para definir um posicionamento e alocação adequados dos recursos de comunicações na rede voadora sensível ao fatiamento proposta. Ainda assim, o SARAF pode ser executado em conjunto com outros algoritmos do estado da arte, representando uma contribuição para a comunidade na obtenção de redes autónomas reconfiguráveis.

A avaliação de desempenho realizada usando uma *testbed* permitiu validar a proposta de rede voadora sensível ao fatiamento de rede. A rede voadora resultante é capaz de cumprir os requisitos associados a múltiplas fatias de rede, usando a quantidade mínima de recursos de comunicações.

Acknowledgments

Ser-me-ia impossível não dirigir uma palavra de imensa gratidão ao Dr. Rui Campos e ao Eng. André Coelho. O vosso entusiasmo como orientadores, reflexo de um profissionalismo exemplar, permitiu a descoberta de uma vocação que julguei acomodada. Bem hajam pelas extensas horas dedicadas a este trabalho, tamanha foi a disponibilidade.

E se falo em vocação, falo em ti, Paulinha.

João Cristiano Mourão Rodrigues

Contents

List of Figures

List of Tables

Acronyms

AP	Access Point
API	Application Programming Interface
BLER	Block Error Ratio
CCDF	Complementary Cumulative Distribution Functions
CDF	Cumulative Distribution Functions
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DHCP	Dynamic Host Configuration Protocol
DSCP	Differentiated Services Code Point
FAP	Flying Access Point
GS	Guard Interval
LVAP	Light Virtual Access Point
MCS	Modulation and Coding Scheme
NIC	Network Interface Card
ONOS	Open Network Operating System
OS	Operating System
PDR	Packet Delivery Ratio
PLR	Packet Loss Ratio
QoS	Quality of Service
RSSI	Receiver Signal Strength Indicator
RTT	Round Trip Time
SARAF	Slicing-aware Resource Allocation Framework
SDN	Software Defined Networking
SFTP	SSH File Transfer Protocol
SLA	Service Level Agreement
SNR	Signal-to-Noise Ratio
SSH	Secure Shell
SSID	Service Set Identifier
STA	Station
TID	Traffic Identifier
UAV	Unmanned Aerial Vehicle
UAV-BS	Unmanned Aerial Vehicle Base Station
UCI	Unified Configuration Interface
UE	User Equipment

Chapter 1

Introduction

1.1 Context

In the last years, we have faced an enormous increase in the usage of communications devices with heterogeneous network performance requirements, including low latency and high throughput. To address this challenge, new wireless communications solutions are needed to improve the usage of available communications resources. Within this context, network slicing emerged in 5G networks as a key component to enable the use of multiple services with different performance requirements on top of a shared physical network infrastructure.

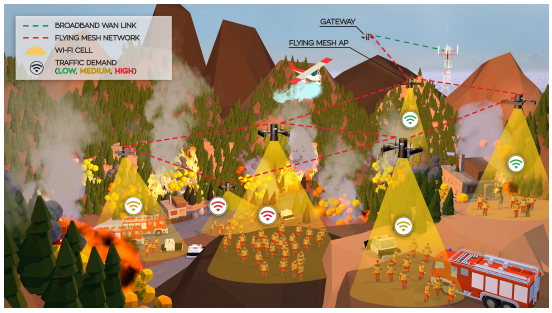
Network slicing is a technique that consists in the virtual and physical division of network resources according to target requirements, providing logical networks to achieve the maximum performance possible, while ensuring the Quality of Service (QoS) levels (e.g., throughput and delay) demanded by those services using the finite network infrastructure resources available. According to the International Telecommunication Union (ITU-T), network slices are considered as logical network partitions composed of multiple virtual resources, isolated and equipped with programmable control and data plane functions [1].

1.2 Motivation and Problem

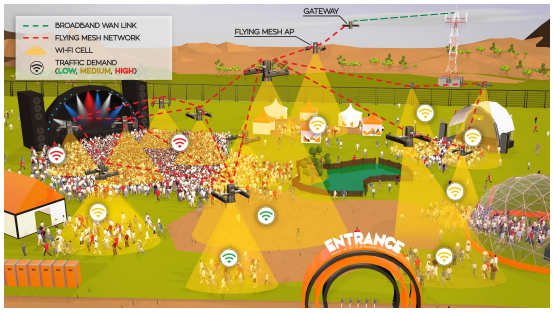
Network slicing can be an important technique to meet target performance requirements in Wi-Fi networks. A QoS-aware approach is already available with the IEEE 802.11e amendment, but it only supports four QoS classes that aggregate flows of the same type (e.g., video), not meeting different QoS guarantees for each flow independently.

When there is a need to reinforce the coverage and capacity of existing networks or a fixed network infrastructure is not available, flying networks, composed of Unmanned Aerial Vehicles (UAVs) carrying communications nodes, emerged as a solution to offer wireless connectivity anywhere, anytime. Still, existing solutions typically aim at maximizing the aggregate performance achieved, but are not able to meet heterogeneous QoS levels demanded by different communications services offered to ground users.

In disaster management scenarios (c.f., Fig. 1.1a), a best-effort wireless coverage service may be sufficient to serve the victims, but the first responders demand reliable wireless connectivity. At crowded events (c.f., Fig. 1.1b), the spectators use applications that go from text messages to high-definition video streaming. The problem lies in designing a shared flying network that uses the minimum amount of communications resources, including the number of UAVs carrying Wi-Fi Access Points (APs), while meeting the QoS levels demanded by different communications services used in certain geographical areas.



(a) Flying network providing wireless connectivity to first responders in a disaster management scenario.



(b) Flying network providing wireless connectivity to spectators in a music festival.

Figure 1.1: Flying network providing wireless connectivity in two scenarios. Reprinted from [2].

1.3 Objectives

In this Dissertation, the main objective was to develop and validate a slicing-aware flying network able to provide network slices with targeted QoS levels at given geographical areas for a given density of ground users over time. These slices were created on top of Wi-Fi APs, carried by UAVs, sharing the network resources available. The challenge consisted in dynamically creating and managing the network slices to meet the users' QoS, while improving the overall utilization of the physical network resources available in the access network [3].

The following specific objectives were pursued:

- Design of a slicing-aware flying network, composed of APs suitable to be carried by UAVs, each providing different network slices;
- Development and implementation of a modular Slicing-aware Resource Allocation Framework to dynamically configure multiple network slices in different APs connected to a centralized controller, in order to enable slicing-aware flying networks;
- Development of a prototype of the proposed slicing-aware flying network;
- Validation and performance evaluation of the proposed solution, using the developed prototype.

1.4 Contributions

The main potential contributions of this Dissertation are three-fold:

- Design and development of a slicing-aware flying network able to provide wireless coverage and meet the QoS levels associated with multiple network slices made available in a given geographical areas;
- The Slicing-aware Resource Allocation Framework (SARAF) capable of defining a suitable slicing-aware configuration in flying networks, while also allowing to collect and process performance metrics for monitoring the QoS offered by the network slices configured;
- Experimental validation and evaluation of the proposed slicing-aware flying network.

A scientific paper including SARAF and SLICER as well as the evaluation results obtained in this work is in preparation for submission to an international journal.

1.5 Document Structure

This document is structured as follows. In Chapter 2, the state of the art is presented, including the main concepts and related works within the scope of this Dissertation. In Chapter 3, the developed solution is described, including its design and implementation. In Chapter 4, the performance evaluation, including the experimental setup and performance results obtained, is discussed. In Chapter 5, the main conclusions achieved and directions for future work are presented.

Chapter 2

State of the Art

This chapter presents existing approaches to design wireless networks able to provide wireless connectivity on-demand, including flying networks, which motivate the solution proposed by this Dissertation. Section 2.1 presents slicing-aware flying network algorithms. Section 2.2 presents concepts on the Software Defined Networking (SDN) paradigm that supported the developed framework proposed by this Dissertation. The chapter ends with a discussion on all the presented solutions and introduces the need for a framework able to integrate state of the art slicing-aware algorithms in real-world flying networks.

2.1 Slicing-aware Flying Networks Algorithms

2.1.1 On-Demand Density-Aware UAV Base Station 3D Placement for Arbitrarily Distributed Users With Guaranteed Data Rates

Lai et al. presented a density-aware placement algorithm to maximize the number of served users, subject to the constraint of the minimum required data rates per user [4]. A polynomial-time deployment approach was proposed to minimize the number of UAVs acting as Base Stations (UAV-BSs) for various user densities. From the power saving point of view, the minimum transmission power UAV-BSs was considered to cover all user equipments (UEs).

The proposed density-aware 3D UAV-BS Placement algorithm aims at maximizing the number of users covered while granting the minimum required data rates per user. In a polynomial-time UAV-BS deployment approach is proposed to minimize the number of UAV-BSs for different scenarios regarding different ground user densities, taking into consideration the minimum transmit power needed to meet the requirements for all User Equipments (UEs) [4]. The focus was on defining the horizontal deployment location for a UAV. The Density-Aware 3D Placement algorithm.

The objective is to maximize the number of served UEs and meet its required QoS, while not exceeding the available data rate provided by each UAV. A list of candidate UEs is collected considered to an appropriate coverage radius definition on different coverage areas, taking into account transmission power efficiency (c.f., Fig. 2.1).

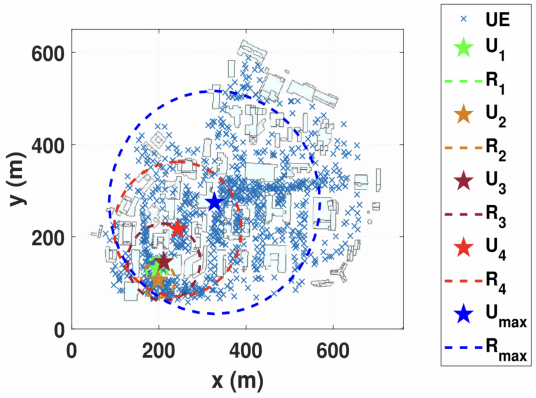


Figure 2.1: Results for location and coverage calculated by the algorithm proposed in [4], considering different data rate constraints, $s_1 = 4$ Mbit/s, $s_2 = 2$ Mbit/s, $s_3 = 1$ Mbit/s, and $s_4 = 500$ kbit/s, where. s_{max} is aimed to maximize number of served UEs and $\lambda(t) = 0.0018(UEs/\mathrm{m}^2)$. Reprinted from [4].

The developed algorithm was able to define the placement of UAV-BSs that maximize the served users in polynomial time. In the simulations, four UEs demanded data rates associated with full HD video streaming, online gaming, web surfing, and VoIP call were considered to validate the proposed method [4].

According to the numerical results presented, the proposed on-demand placement algorithm can support multiple users with different user densities, while improving the transmission power efficiency of a UAV-BS by about 29%.

Other related research works include interference management considering multiple UAV-BSs and the associated deployment strategies [4].

2.1.2 SLICER

In order to achieve slicing-aware flying networks, Coelho et al. proposed SLICER [5], an algorithm enabling the placement and allocation of communications resources. Considering a set of potential UAVs, the SLICER algorithm allows to determine the minimum number of UAVs to actually use, their 3D positions and the communications resources' allocation for each slice. SLICER initially discretizes the 3D space into smaller cuboids, where each UAV is placed. After that, it computes the Signal-to-Noise Ratio (SNR) for the wireless links available between each potential UAV and the ground users. The solution for the placement and allocation of communications resources is determined by solving an optimization problem, in which the objective function aims at minimizing the cost of deploying a slicing-aware flying network, considering that each UAV has a given activation cost. The activation cost of each UAV may be defined according to multiple criteria, such as the cost of the hardware carried on board and the operating cost of the UAV. Finally, SLICER assigns the wireless channels that minimize the bandwidth used and reconfigures the flying network accordingly, presented in Figure 2.2.

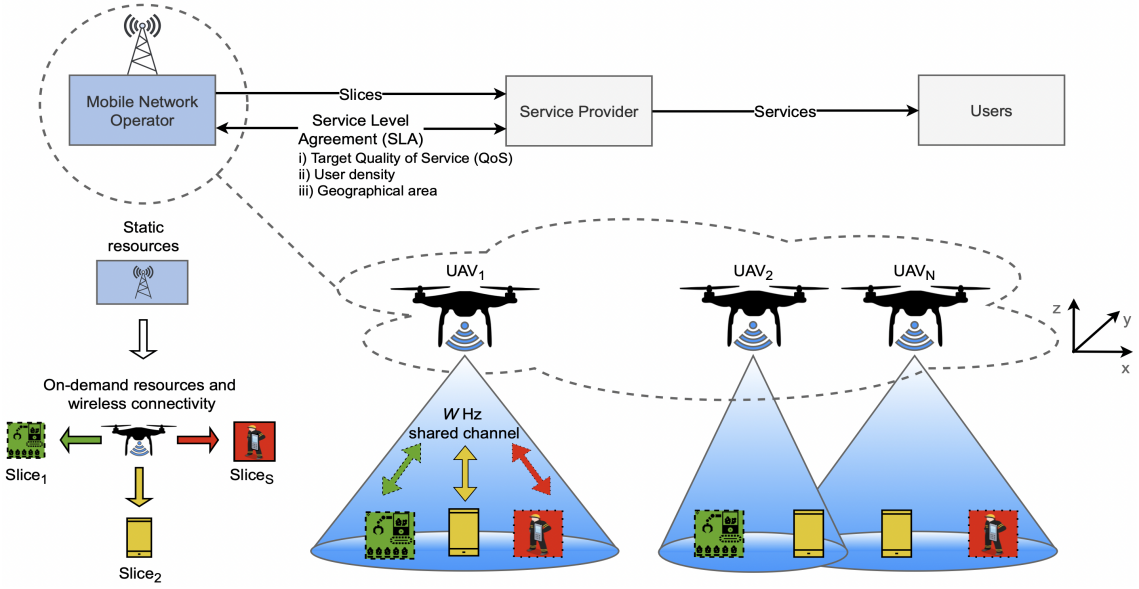


Figure 2.2: Flying network enabling $|S|$ slices at different geographical areas. Reprinted from [5].

2.1.3 Single AP slicing approach

For Wi-Fi networks, many slicing-aware solutions were proposed, mostly based on time division to access resources, in order to accomplish the target performance, isolation between slices, and a dynamic resource allocation according to the slices' requirements [6].

Zehl et al. [7] propose a solution based on time division through different Service Set Identifiers (SSIDs). A first practical approach was considered to perform slicing through different SSIDs on each physical AP, where a given slices' size, based on an assigned time percentage, is created for traffic isolation. This was defined as the first step to perform a functional solution to start with.

An simple initial state of the art approach based on Airtime Division was taken: to create two slices, while considering the need for a high QoS slice with traffic isolation. To show the effectiveness of slicing, a stream of a high-quality video was being generated by a home device, associated with one slice with 75% Airtime on the Home Network, as depicted in Figure 2.3.

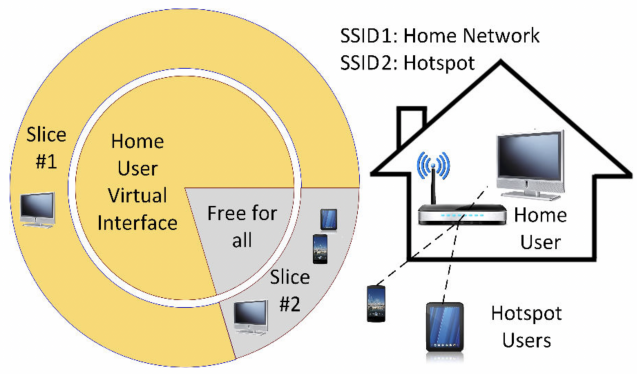


Figure 2.3: Hotspot Slicer: Slicing virtualized Home Wi-Fi networks. Reprinted from [7].

To implement this solution, an adaptation on the Linux Wi-Fi SoftMAC module (software module running on the host system) was done, more specifically on the ath9k driver module. This driver module is responsible for maintaining software queues for each link and each Traffic Identifier (TID). An adaptation of the ath9k driver power saving implementation was performed to pause the traffic per link by steering the ath9k software queues, which is illustrated in Figure 2.4.

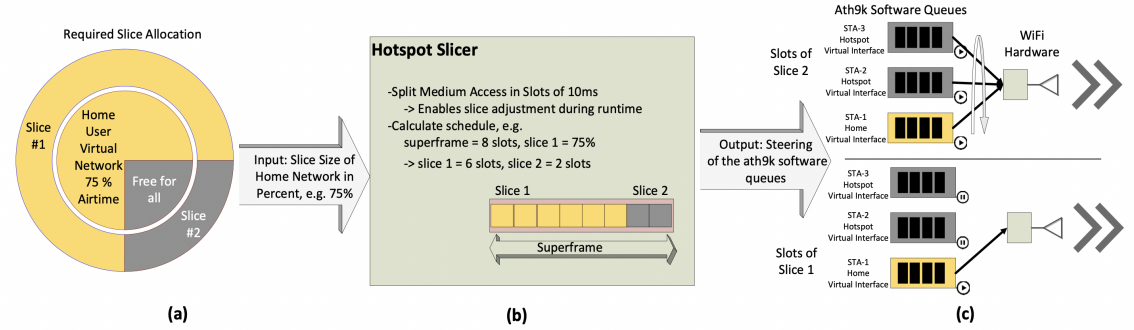


Figure 2.4: Hotspot slicer implementation, where: (a) slice size is defined in percentage of needed air-time for the home network (slice 1); (b) medium access is partitioned within small slots, forming a super frame to enable fine-tuning, in which HotSpot Slicer calculates the needed slots for slicing the home network; (c) slicing is achieved by pausing the ath9k software queues directly before the data is delivered to the hardware. Reprinted from [7].

With this approach, a judder and artifact free streaming was accomplished thanks to the slicing technique employed, as shown in Figure 2.5.

Other research works are worthwhile being investigated further, such as a plan to utilize the MAC slicer together with ResFi [8] to enable traffic separation of neighboring APs to solve hidden node problems.

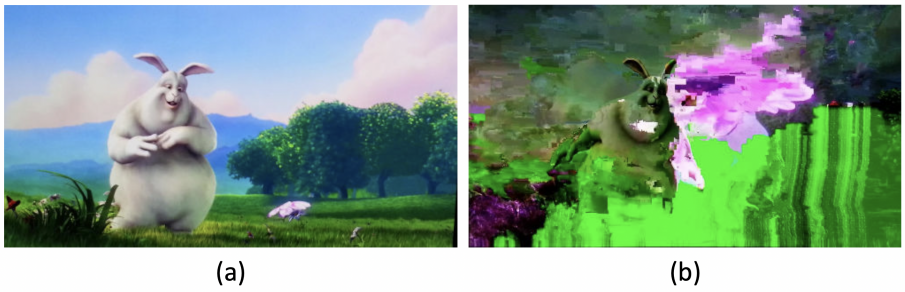


Figure 2.5: Screenshot of the home device when (a) MAC layer slicing is applied, and (b) MAC slicing is disabled (baseline). Reprinted from [7].

2.2 Software Defined Networking (SDN)

Software Defined Networking (SDN) introduced the concept of network programmability providing both a standard protocol to program network devices [9], and a standardized vision of network devices. These concepts strongly facilitate the deployment of network slicing introducing both

flexibility and dynamicity in the configuration of the data plane [10]. Scano et al. [10] presents two implementation examples: the first considers a single SDN controller configuring a shared physical infrastructure to create network slices, where each slice behaves as a conventional (non SDN) network and the second, considers a Hypervisor for creating network slices.

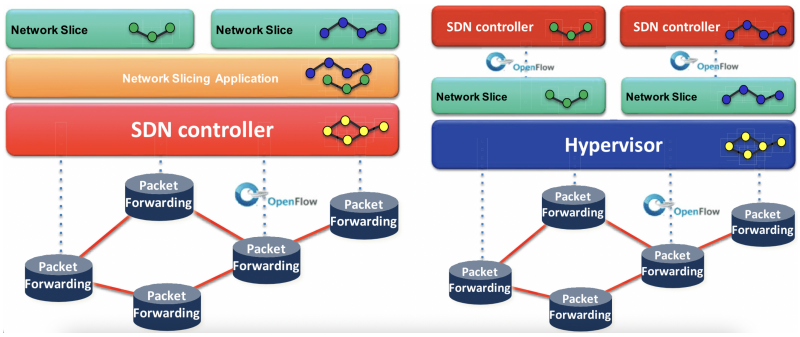


Figure 2.6: Illustrative shared physical infrastructure where the network slices are respectively created by means a single SDN controller (on the left) and a dedicated SDN controller (on the right). Reprinted from [10].

2.2.1 Multiple APs slicing approach

A more robust and mature solution is also presented in the literature: to perform spectrum slicing. In this approach, Odin enables an SDN-based solution that provides features for the network needs (channel selection, load balancing and wireless troubleshooting) with the possibility for programmability of Wi-Fi networks on top of low-cost APs hardware. Odin grants the possibility of decoupling the control from the data plane, allowing a logically centralized network state. This enables control and management through an abstraction of the IEEE 802.11 protocol, while implementing a software-defined Wi-Fi network architecture based on Light Virtual Access Points (LVAPs) [11].

The LVAP is a per-client AP that simplifies the management of client associations, authentication, handovers, and unified slicing of the network. It enables a port-per-source view of Wi-Fi networks similar to wired networks. As such, it remains orthogonal, but complementary to approaches based on physical layer virtualization and RF spectrum slicing [12]. LVAPs are hosted on the agent, and their assignment to agents is controlled by the controller, as depicted in Figure 2.7.

Once a client connects to a specific AP, using a given SSID, it is assigned to an existing slice associated with the correspondent SSID. The client becomes integrated with its LVAP on the slice. The client management is handled by the operating applications in this slice and the controller ensures that an application is only presented with a view of the network corresponding to its slice. Moreover, the control decisions make applications do not have any visibility aside its correspondent LVAP, ensuring logical isolation control between slices. The block diagram presented in Figure 2.8 describes the LVAP assignment process.

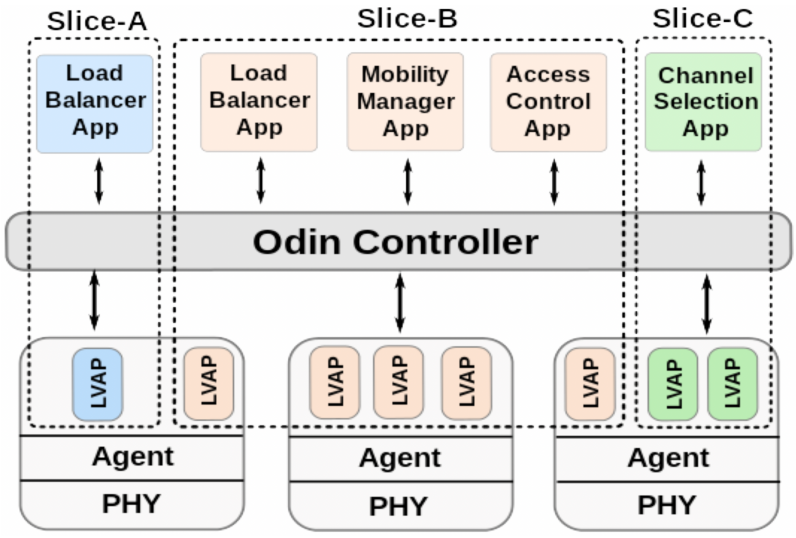


Figure 2.7: Odin applications operate upon a view of LVAPs and physical APs in their respective slices. Reprinted from [11].

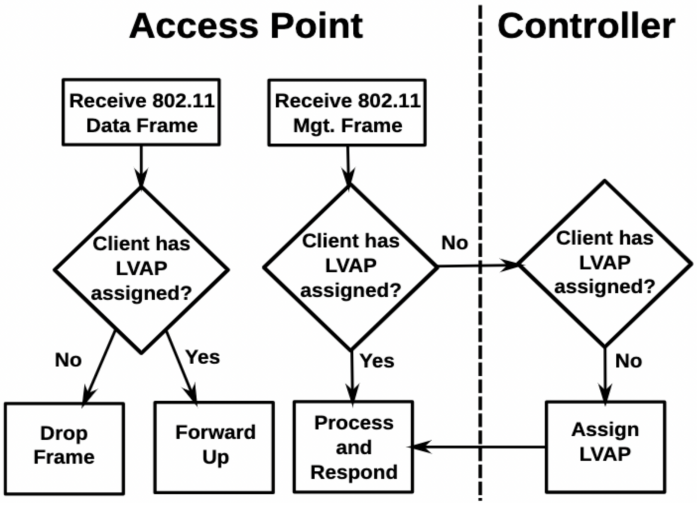


Figure 2.8: Processing path for Wi-Fi frames: Agents invoke a controller for handling management frames. Reprinted from [11].

The Controller uses Open-Flow for Odin specific functionality, such as tracking client IP addresses to be attached to their respective LVAPs. A configuration file allows to assign agents to slices, set SSIDs per slice and associate network applications to run on each slice. The controller uses a TCP-based control channel to invoke the Odin protocol commands on the agents. The controller allows to define states *per-slice*, granting an isolation with a closed view of their respective slice and its associated clients, their LVAPs, and physical APs.

Complementing this, the Agent implements the Wi-Fi split-MAC together with the controller, hosts LVAPs, and collects statistics on a per-frame and host basis. A notification is generated to the Controller whenever a frame that matches a per-frame event subscription registered by a particular

application is received, as described on [11].

Load-balancing typically requires control of clients' attachment points to the network or the ability to trigger the handover of clients between Wi-Fi access points. Without load-balancing, the client is assigned to the first AP that receives the association request. With load-balancing, each LVAP is placed on the physical AP that has the highest Receiver Signal Strength Indicator (RSSI) and does not exceed the client load. Results on the load-balancing feature are depicted in Figure 2.9, where approximately 50% of the clients were able to transmit around 20 MB of data with load balancing enabled compared to 15% without load-balancing. With more APs and proper load balancing this unfairness can be alleviated.

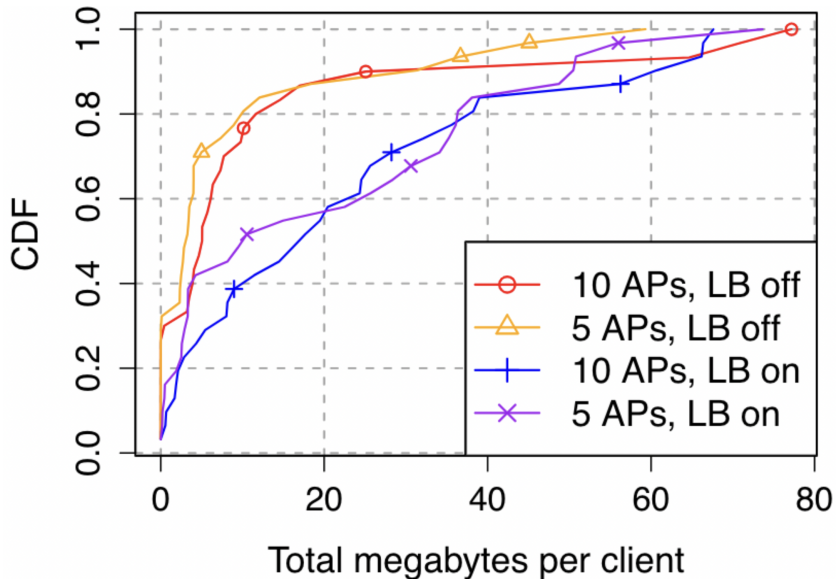


Figure 2.9: Total throughput per client with and without load-balancing represented by a Cumulative Distribution Function. Reprinted from [11].

2.2.2 Network Slicing using SDN

Scano, et al. propose a framework to deploy a set of network slices using a single SDN controller [10]. Deployed slices guarantee isolation in terms of connectivity and performance. The proposed solution was tested on an emulated SDN testbed using the Open Network Operating System (ONOS) project [13] controller.

SDN decouples control and data planes and moves the control to a centralized controller. Such controller implements the network operating system and provides a library of Application Programming Interfaces (APIs) that can be used for simplifying the development of networking applications. Specifically, using SDN two different network slicing approaches can be considered. In the first approach each slice provides to the end-users a traditional network (e.g., enforcing specific SLAs). In the second approach each slice provides to the end-users an SDN network, which

is configurable through an SDN controller. Each slice can be created with different properties depending on the tenant requirements and includes an SDN controller that can be used to configure the resources (i.e., devices and interfaces) belonging to the specific slice.

The developed QoSlicing app utilizes OpenFlow meters and transmission queues on the emulated switches interfaces to implement performance isolation. In each device, all the traffic belonging to a specific slice and directed on the same output port is processed by a single OpenFlow meter. Each meter can include up to two thresholds (i.e., OpenFlow bands): when two threshold are configured, the first one is used to change the Differentiated Services Code Point (DSCP) value of packets exceeding the configures rate, while the second is used to discard all exceeding packets; when a single threshold is configured, exceeding traffic is discarded.

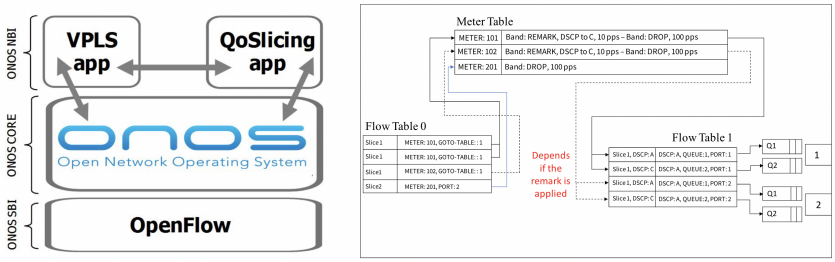


Figure 2.10: Architecture of the proposed solution at the SDN controller and of the pipeline, meter table and transmission queues on the data plane devices. Reprinted from [10].

In particular, Fig. 2.10 reports the pipeline of a switch traversed by two slices: SLICE1 and SLICE2. A traffic profile including meters and output port queues has been applied for SLICE1, while a QoS profile including only meters has been used for SLICE2. For SLICE1 two bands are used within each meter. A remark band set the value of the DSCP field to the value C when a first bandwidth threshold is overcome; then packets are dropped only if a second higher threshold is overcome. When SLICE1 packets are remarked, following rules in Table 1, they are directed to Q2 on the output ports associated with a queue with lower priority. As such, the traffic of SLICE2 is preserved in terms of both total available bandwidth and achievable latency.

An experimental evaluation has been conducted and on an emulated testbed shows that both bitrate and latency are guarateed when two slices share a common link.

2.3 Discussion

Considering all the presented solutions, there are already state of the art algorithms to determine the necessary resources and their positioning in slicing-aware flying networks, such as SLICER. However, there is no agile way to employ the resulting slicing-aware network configurations in real-world flying networks. In addition, SLICER and QoSlicing application were only evaluated in a simulation environment and theoretically, respectively, so it is important to create a framework that allows to integrate state of the art algorithms in real-world flying networks and promote their validation and experimental evaluation.

Chapter 3

Developed solution

In this chapter, the developed solution is presented based on the knowledge gathered and discussed in the previous chapter. In Section 3.1, a generic system design is presented to accomplish a solution for static and dynamic network slicing. In Section 3.2, the dynamic slicing-aware flying network proposed by this Dissertation is presented. In Section 3.3, a modular Slicing-aware Resource Allocation Framework (SARAF) is presented as the main contribution to dynamically configure multiple FAPs as well to collect information about the clients connected to each FAP, in order to enable slicing-aware flying networks. In Section 3.4, the system implementation is detailed.

3.1 Overview

The problem addressed in this Dissertation is the design and implementation of a slicing-aware flying network solution able to meet target coverage and QoS levels, while improving the usage of its overall network resources. Motivated by the state of the art works, a solution is proposed herein. It allows to define a suitable channel assignment approach, minimizing the overall bandwidth used and suitable UAV positions to guarantee the users' targeted QoS and coverage requirements by means of a novel Slicing-aware Resource Allocation Framework (SARAF).

The generic system designed lies on a centralized Flying Access Point (FAP) Controller that computes and allocates the network resources associated with a given number of wireless channels made available by FAPs, which serve ground users with different targeted QoS requirements. This FAP Controller communicates with each FAP, configuring the channel bandwidth needed by each network slice and defining the UAV positions to meet the targeted wireless coverage, taking into account minimum SNR values to be ensured, number of wireless channels available, and the QoS levels to be offered to the ground users. This model, depicted in Figure 3.1, considers two frequency bands for the Network Interface Cards (NICs) on-board each FAP, operating at 2.4 GHz and 5 GHz.

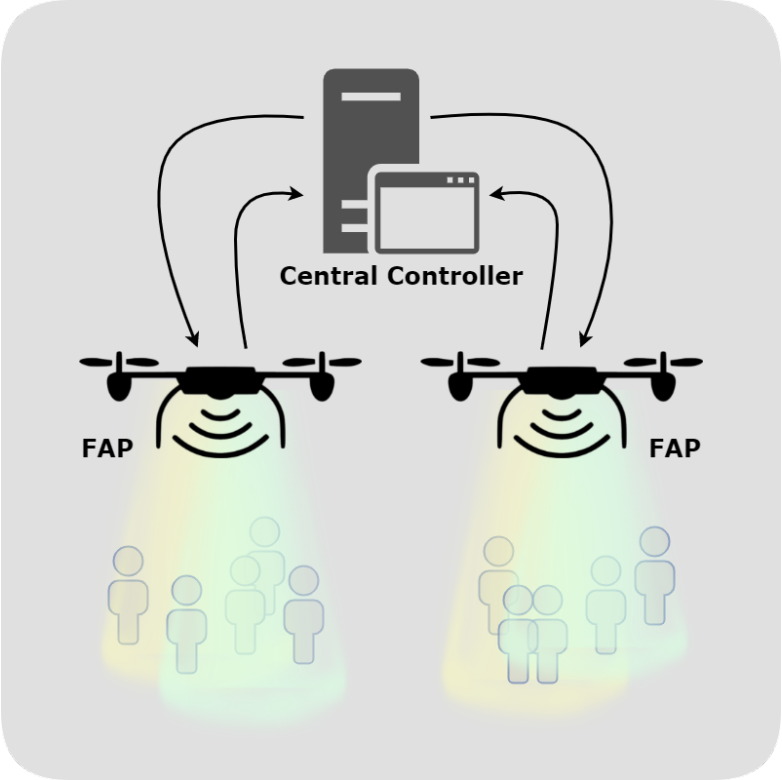


Figure 3.1: System elements composing the proposed solution, considering that each color represents a network slice type associated with a given wireless channel bandwidth.

Two main steps were defined: 1) static network slicing approach; 2) dynamic network slicing approach that computes and allocates the network resources dynamically to accomplish a slicing-aware flying network.

3.1.0.1 Static Network Slicing

Two wireless channels operating at 2.4 GHz and 5 GHz frequency bands, were considered, in order to achieve a baseline solution that employs an independent network providing static resources for each network slice. For that purpose, two NICs working on different frequency bands were used, in order to ensure resource isolation. In this approach, the channel bandwidth was statically computed and configured on each NIC based on the targeted QoS requirements. The system elements composing the proposed solution are depicted in Figure 3.2.

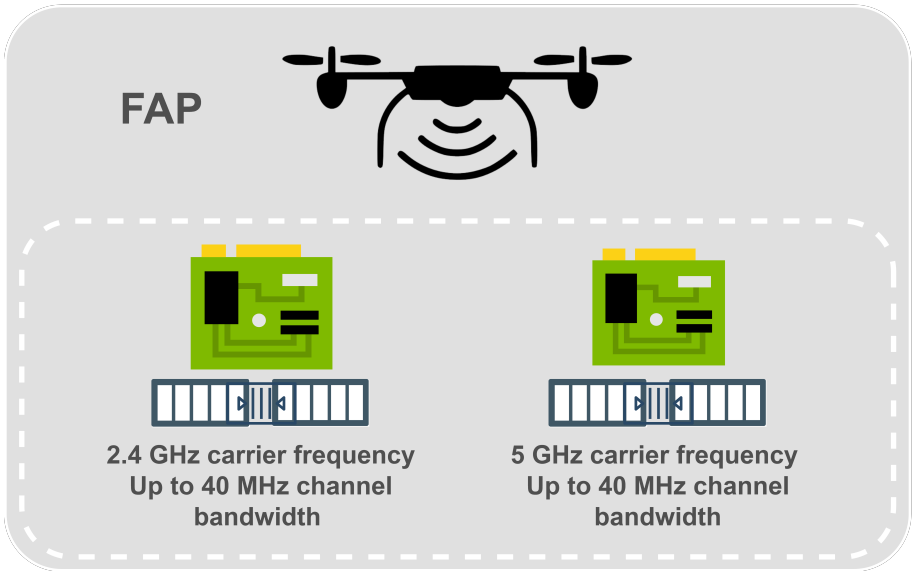


Figure 3.2: System elements composing the proposed solution with two NICs without a FAP Controller.

3.1.0.2 Dynamic Network Slicing

In order to achieve a solution for different and variable networking scenarios, a dynamic solution was designed. It takes advantage of a state of the art algorithm, called SLICER [5], able to dynamically configure the channel bandwidth to be used by each NIC, considering target QoS requirements and users' position information provided as input.

Considering this approach, a Single Board Computer, acting as a FAP Controller, orchestrates the NICs and configures targeted channel bandwidth values, as presented in Figure 3.3.

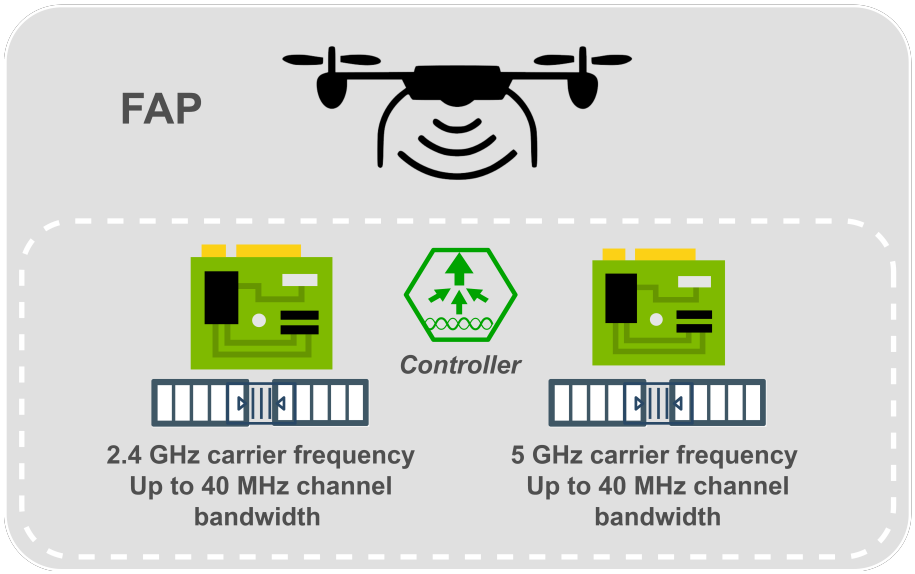


Figure 3.3: System elements composing the proposed solution with two NICs with a FAP Controller.

3.2 Slicing-aware Flying Network

The dynamic slicing-aware flying network proposed by this Dissertation is built upon the SLICER algorithm. The centralized computing associated with the FAP controller allows to achieve an overall control of the slicing-aware flying network. With this solution, the total network resources are orchestrated by a FAP Controller exchanging information with FAPs and computing the channel bandwidth to be autonomously configured on the NICs carried the FAPs, which provide network slices. This solution aims at meeting the QoS levels demanded by users' services.

Network Slicing-aware Algorithm. To implement the proposed solution, the SLICER algorithm, proposed in [5], is used in order to define suitable UAV positions and minimum channel bandwidth to be configured on the NICs. Conceptually, the proposed algorithm divides the ground area where the ground users are located into subareas and considers the targeted QoS requirements associated with the ground users, including maximum throughput and minimum average delay. Then, it to computes the minimum SNR values to be ensured for the wireless links established between each UAV and the ground users served. The minimum SNR is defined so that target data rate values, associated with the Modulation and Coding Scheme (MCS) indexes used, are ensured for the wireless links established. For that purpose, SNR thresholds are considered based on the equation presented in Figure 3.4, which was derived from experimental results obtained. To design the solution, a linear regression that considers the relation between SNR and the throughput obtained for different distances between a client and the FAP, was considered. Different values should be obtained for different wireless communications technologies and network configurations used by the flying network. With this data, a dynamic reconfiguration over time of the bandwidth to be allocated to each subarea is performed, as presented in Figure 3.5, combined with UAV position changes to guarantee the users' targeted QoS, as depicted in Figures 3.6. The SLICER algorithm is presented in Algorithm 1.

Table 3.1 presents the main notation used to formulate the problem addressed by SLICER.

Table 3.1: Main notation used to formulate the problem addressed.

Symbol	Definition
UAV_i	Unmanned Aerial Vehicle i
GU_a	Ground User a
N	Set of UAVs available in the flying network
S	Set of network slices provided by the flying network
A	Geographical area covered by the flying network
A^s	Geographical area where slice s is available
t_k	Period during which the allocated resources are static for all network slices
C	Cuboid representing the 3D space within which the UAVs can be positioned
P_i	3D Position of UAV_i
P_a	3D Position of GU_a
A^s_m	Rectangular fixed-size geographical subarea associated to slice s
M^s	Set of subareas A^s_m
$r^s_{i,m}$	Number of wireless channels provided by UAV_i to subarea A^s_m
W_i	Total channel bandwidth available at UAV_i
$c^s_{i,m}$	Bidirectional network capacity provided by an wireless channel of UAV_i to subarea A^s_m
$SNR_{i,a}$	Signal-to-noise ratio on the wireless link between UAV_i and the Ground User at P_a

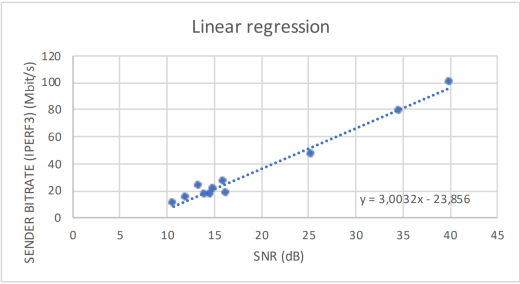


Figure 3.4: 2.4 GHz throughput linear regression.

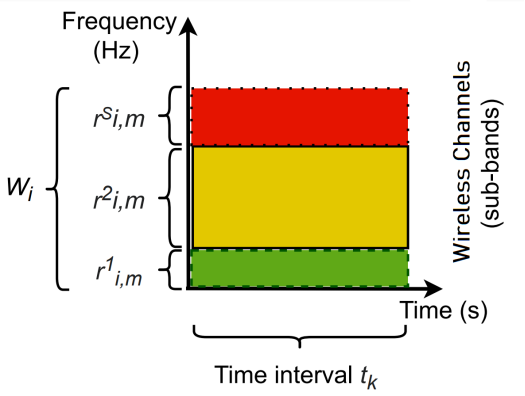
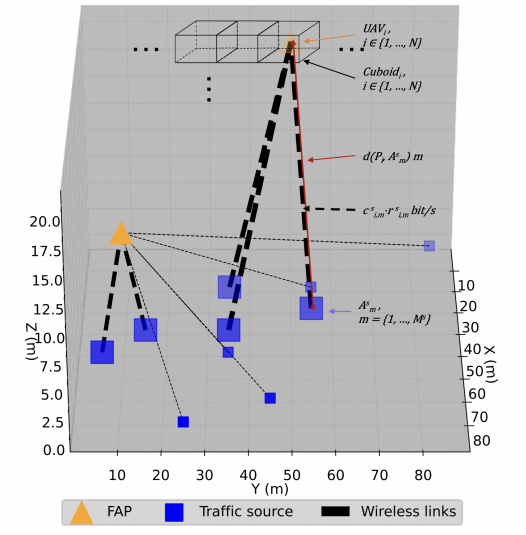
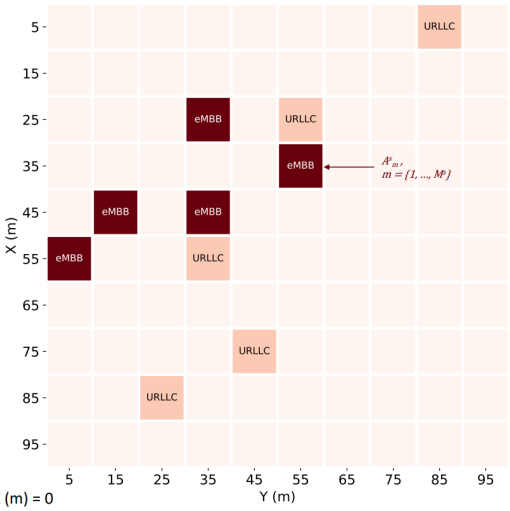


Figure 3.5: Allocation of the wireless channels available at UAV_i to $|S|$ *slices*,s $\in \{1,...,S\}$. Each slice s is provided in subarea A^s_m, with $m \in \{1,...,M^s\}$, during the time interval t_k. Reprinted from [5].



(a) Division of the cuboid C into N smaller cuboids (solid black lines), which are candidate positions for placing the UAVs acting as FAPs. The orange triangles represent the UAVs selected, while the blue squares represent the traffic sources. The wireless links are depicted by the dashed black lines, which are as large as their capacity.



(b) Division of the ground area A into subareas. The light and dark brown squares represent the subareas where the network slices are available. The remaining area is provided with a best-effort wireless coverage service.

Figure 3.6: Flying network enabling on-demand coverage-aware network slices. Reprinted from [10].

Algorithm 1 – SLICER algorithm

Discretize cuboid C into N cuboids centered at P_i
Compute $SNR_{i,a}$ for the wireless link available between each potential UAV_i and the GU_a at P_a
Compute the network capacity $c_{i,a}$ provided by a wireless channel with bandwidth B to the GU_a at P_a
Solve an optimization problem using a state of the art solver
Assign the wireless channels that minimize the bandwidth used
Reconfigure the flying network accordingly

3.3 Slicing-aware Resource Allocation Framework (SARAF)

A modular Slicing-aware Resource Allocation Framework (SARAF) was developed to dynamically configure multiple FAPs connected to a centralized controller, in order to enable slicing-aware flying networks. SARAF uses Paramiko, a high-level pure-Python API implementation of the SSHv2 protocol [14], enabling that a centralized controller connected to a set of FAPs is able to dynamically change multiple network configurations, including IP addresses, FAP's interfaces status, SSIDs, channel number and channel bandwidth. SARAF is compliant with the SDN paradigm, ensuring data and control plane separation between the controller and each FAP.

SARAF was designed to be generic and applicable in different real-world flying networks, in

order to guarantee network slices according to user requirements and locations. It grants a dynamic adaptation to any scenario, enabling on-demand network configuration, while also allowing to collect and process performance metrics for monitoring the QoS offered by the slices. In its current version, SARAF uses SLICER to define a suitable placement and allocation of the communications resources, including UAV positions and channel bandwidth, but it can be used with other algorithms from the state of the art. SARAF represents a contribution to the community, enabling the agile validation and evaluation of solutions for reconfigurable flying networks.

The source code of SARAF is available in [15].

Network functions were developed to change the configuration of each interface, including carrier frequency and channel bandwidth, and collect all the given information on the current state of the interfaces. They are detailed in what follows.

3.3.1 Collect stations info

A network function, was developed to collect information about the clients connected each FAP, including IP address, MAC address, signal power level, and physical data rate based on the MCSs automatically selected (c.f., SARAF function 1).

3.3.2 PING tests

PING tests were included in SARAF to collect the average Round Trip Time (RTT) (c.f., SARAF function 2). The PING duration was specified in the integrated configurations file. All the PING iterations were also collected on a log file to be used for the analysis of the results.

3.3.3 iPerf3 tests

iPerf3 tests are used by SARAF to assess the throughput and Packet Delivery Ratio (PDR) (c.f., SARAF function 2). They were configured on reverse mode, since the iPerf3 server was running on each client.

The following metrics were collected in each test:

- DEVICE NUMBER
- DEVICE MAC ADDRESS
- DEVICE IP ADDRESS
- DEVICE SIGNAL (NOT CONSIDERING NOISE) (OPENWRT)
- TX BITRATE (OPENWRT)
- RX BITRATE (OPENWRT)
- SENDER BITRATE (IPERF3)
- SENDER JITTER (IPERF3)
- SENDER PACKET LOSS RATIO (IPERF3)
- RECEIVER BITRATE (IPERF3)
- RECEIVER JITTER (IPERF3) - RECEIVER PACKET LOSS RATIO (IPERF3)
- RTT (PING)

- RADIO CHANNEL NUMBER (OPENWRT)
- RADIO CHANNEL BANDWIDTH (OPENWRT)
- RADIO CENTER FREQUENCY (OPENWRT)

3.3.4 Configurations file

A configuration CSV file was created to define the testing parameters: the type of action was defined by means of two options:

- **0**: To monitor the QoS offered by the network slices.
- **1**: To reconfigure the network according to the slicing-aware configuration computed.

Type **0** includes the interface radio number, as well as the radio channel number and channel bandwidth to be configured. Type **1** includes the length of the iPerf3 test in seconds, the bitrate in bits/s, and the PING test duration and interval in seconds for sending each packet.

3.4 System Implementation

The designed system is composed of the following elements:

- **Flying Access Point (FAP)**: TP-Link TL-WR902AC v3 (Figure 3.8) was chosen considering its low weight, its small dimensions, its low energy consumption, and its cost-effectiveness and availability, which make it as the most suitable AP model option to be used in a flying network.
- **Controller**: Raspberry Pi 4B, which is a central node that runs SARAF and processes performance metrics and monitoring data.
- **Clients**: Five Raspberry Pis, each with a Panda Wireless N600 USB NIC, that act as Wi-Fi Stations (STAs) that use the network slices provided by the FAPs.

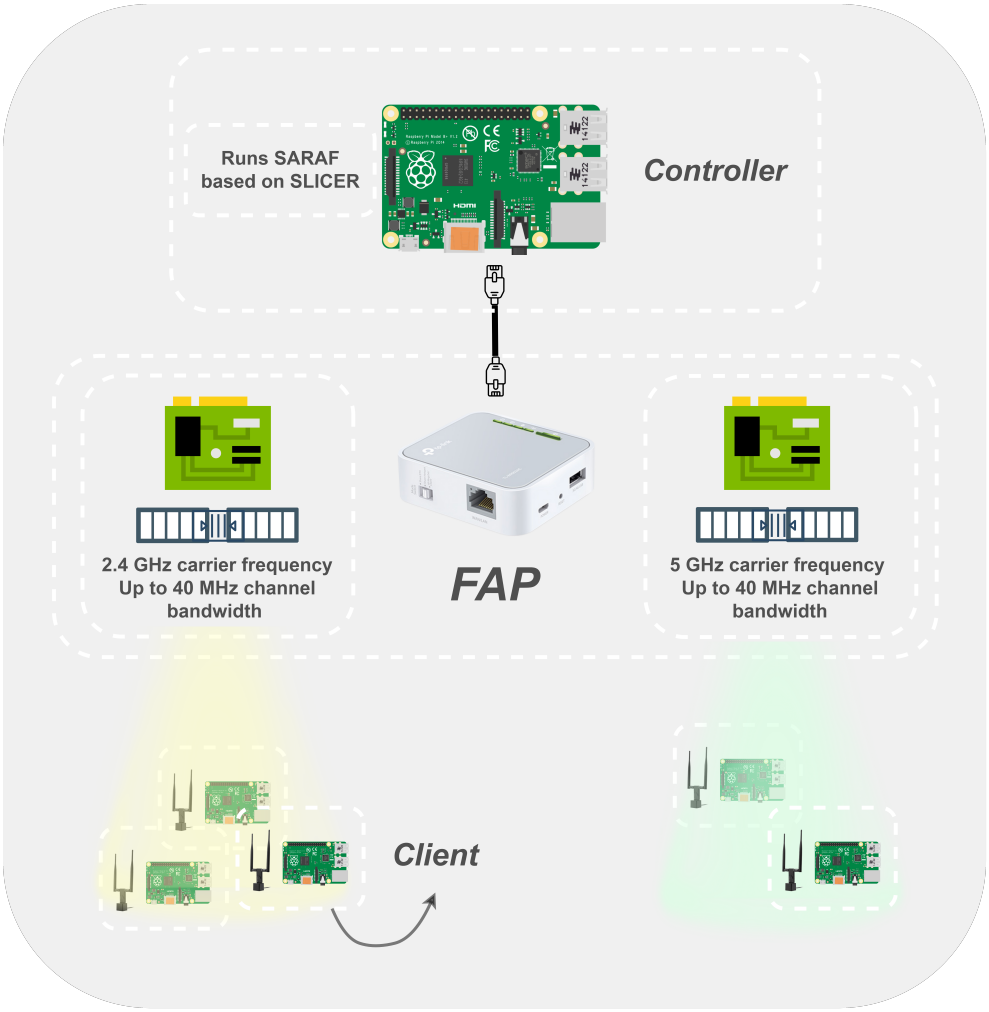


Figure 3.7: Testbed elements.

3.4.1 Setup

With the remote access enabled, the wireless interfaces were configured on the system files and using UCI's command-line utility via SSH. The wireless channels and channel bandwidth were configured according to the networking scenario under evaluation.

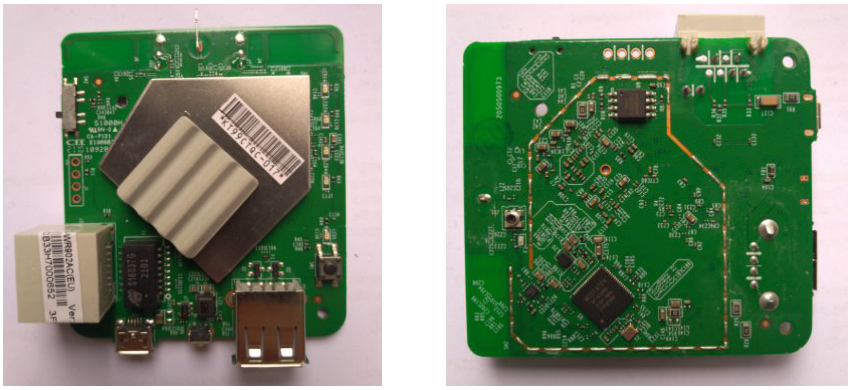
3.4.1.1 Remote access configuration

The DHCP server was disabled on each FAP and defined on the controller connected via Ethernet, as presented in the Figure 3.7. Remote access via SSH and SFTP was configured to grant external access to the FAP system files.

3.4.1.2 Flying Access Point (FAP)

The chosen FAP, TP-Link TL-WR902AC v3, is a Dual-Band AP to be carried by each UAV that operates at 2.4 GHz (IEEE 802.11n) and 5 GHz (IEEE 802.11ac) frequency bands, by means of

two wireless hardware NICs (*phy0* and *phy1*), and supports up to 4 SSIDs in 2.4 GHz and up to 4 SSIDs in 5 GHz. SARAF was developed to autonomously reconfigure the channel bandwidth to be used by each NIC over time. 14 available wireless channels are available in 2.4 GHz (ranging from 2412 MHz to 2484 MHz) with a transmission power of 24.0 dBm and 25 available wireless channels are available in 5 GHz (ranging from 5180 MHz to 5845 MHz) with a transmission power of 12.0 dBm.



(a) Top view. (b) Bottom view.

Figure 3.8: TL-WR902AC v3 PCB. Source: [16]

The OpenWrt Operating System (OS) [17] was used. OpenWrt is a lightweight OS that grants a way to free from the application selection and configuration provided by the vendor, allowing to configure network devices, such as Access Points, through the use of packages that suit the desired application. For developers, OpenWrt is a suitable OS to build an application without having to build a complete firmware around it; for users this means the ability for full customization [17] [16]. The configurations of the wireless interfaces (2.4 GHz and 5 GHz) was done through the Unified Configuration Interface (UCI) [18], including remote access, network interfaces and wireless settings. The FAP was power supplied by a battery with a total capacity of 4000 mAh. The FAP was configured on bridge mode, acting as a DHCP client.

3.4.2 Controller

The Controller, implemented by means of a Raspberry Pi 4B single-board computer, runs SARAF and was configured to handle DHCP requests on the network, being responsible for configuring each FAP and its interfaces, and collecting and processing performance metrics, after launching and gathering the results of the network tests. The controller was also responsible for handling the network's external access, allowing the possibility to update the configuration files that define the tests' variables. The controller was power supplied by a battery with a total capacity of 4000 mAh.

3.4.3 Clients

The five Raspberry Pis, each with a dual-band wireless USB adapter Panda Wireless N600, were configured as IEEE 802.11 STAs. The Raspberry Pis were power supplied by a battery with a total capacity of 4000 mAh.

They were configured with Raspberry Pi OS Lite. The following system service file was included to run an iPerf3 server on a defined port automatically:

```
[Unit]
Description=iPerf3 server
After=syslog.target network.target auditd.service

[Service]
ExecStart=/usr/bin/iPerf3 -s -p 5002

[Install]
WantedBy=multi-user.target
```

In this illustrative configuration, port 5002 is associated with one of the Raspberry Pis. The remaining were given different port numbers, ranging from 5003 to 5006.

3.5 Summary

This chapter presented the developed solution based on the knowledge gathered and discussed in the previous chapter. A generic system design was proposed to accomplish a dynamic slicing-aware flying network. A modular Slicing-aware Resource Allocation Framework (SARAF) was also presented as the main contribution to dynamically configure multiple FAPs and collect information about the clients connected to each FAP, enabling slicing-aware flying networks. This chapter also detailed the system implementation.

The performance evaluation of the proposed solution in different networking scenarios is presented in Chapter 4.

Chapter 4

Performance Evaluation

In this chapter, the developed solution is presented based on the knowledge gathered and discussed in the previous chapter. Section 4.1 presents the experimental setup allowed to validate the robustness and modularity of SARAF, while collecting and processing network performance metrics for performance evaluation when SLICER is used. Section 4.2 presents the results of the experimental tests carried out.

4.1 Experimental Setup

For validating SARAF and evaluating the SLICER algorithm experimentally [5], multiple tests on different scenarios were performed. They allowed to validate the robustness and modularity of SARAF, while collecting and processing network performance metrics for performance evaluation when SLICER is used.

A set of five clients and five evaluation scenarios were considered. The clients were associated with a slice characterized by a given traffic demand of 4 or 20 Mbit/s. At least two clients were randomly associated to each slice. The clients' position distribution was also randomly assigned on a defined ground area, considering the physical space available.

(a) FAP point of view during the single client test.

(b) Sports complex upper view. Source: [19]

(c) Client point of view.

(d) Baseline testing.

Figure 4.1: Sports complex photos.

At first, tests were performed on a sports complex (c.f., Fig. 4.1a, Fig. 4.1b and Fig. 4.1c) using only one client at different distances, ranging from 5 to 60 meters away from the FAP, in order to determine the linear regression equations associated with the relation between the SNR and throughput values obtained in practice for the network configuration employed (c.f., Fig. B.1 and B.2, presented in Appendix B). The obtained linear regressions are considered by the SLICER algorithm. The Wi-Fi parameters that characterize the NICs acting as FAPs are presented in Table 4.1. As an example, for the first scenario, the parameters presented in Table 4.1 resulted from the SLICER output (FAP position and NICs' bandwidth) considering the randomly assigned clients' position distribution and slice assignment. The graphical output for the same scenario is presented in Table 4.2. The parameters regarding the remaining scenarios are presented in Appendix C.

Table 4.1: FAP characteristics.

IEEE 802.11 standard	IEEE 802.11n
Guard Interval (GS)	800 ns
Channel bandwidth	20 or 40 MHz
Number of antennas	2
Maximum physical data rate	750 Mbit/s
Frequency bands	2.4 GHz & 5 GHz
Transmission power	12.0 dBm (5180 MHz) & 24.0 dBm (2412 MHz)
Noise power	-95 dBm

An example for the FAP and client distribution at this venue is presented in Fig. 4.3b.

Due to the lack of perfect omnidirectionality that characterizes both antennas of the AP model used in practice, the tests could not be carried out considering a 360º client dispersion at the

Table 4.2: Scenario 1 configuration.

Client	Slice	Throughput (Mbit/s)	Bandwidth (MHz)	X Position (m)	Y Position (m)
FAP				30	29
Client 1	Slice 2	20	40	26	32
Client 2	Slice 2	20	40	37	37
Client 3	Slice 2	20	40	2	22
Client 4	Slice 1	4	40	7	18
Client 5	Slice 1	4	40	14	35

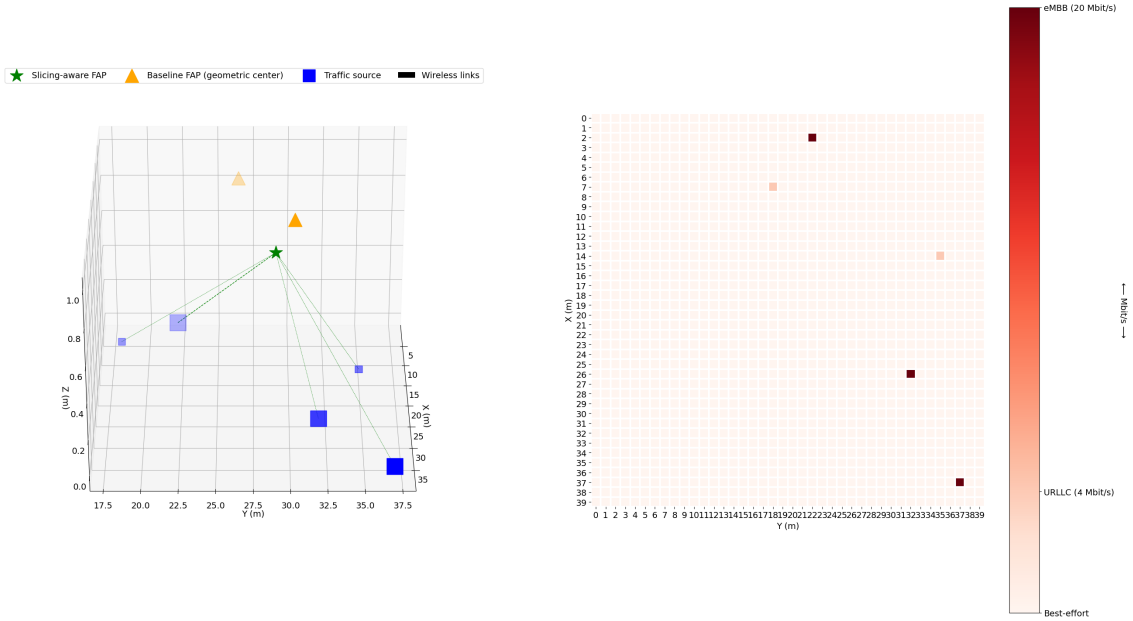


Figure 4.2: Illustrative example of SLICER output for the FAP placement, considering a set of static clients on the ground, which were randomly positioned and assigned to two network slices that make up Scenario 1. Source: SLICER output.

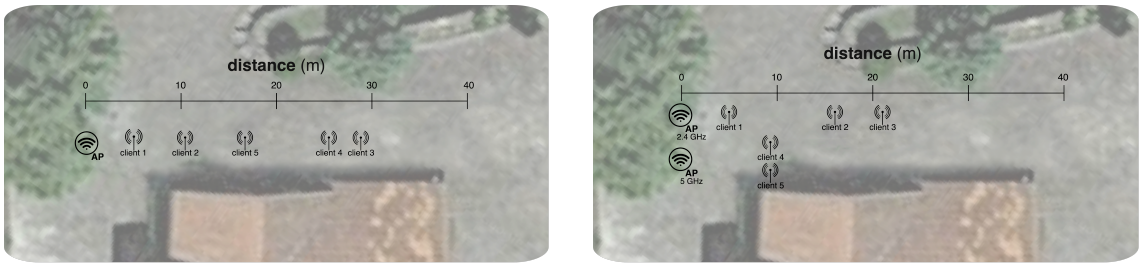
venue. To overcome this limitation, the Euclidean distance between each client and the FAP was calculated, and the clients were placed inline, so that the same Euclidean distance in each scenario was achieved, while ensuring radio line of sight between all clients and the FAPs. Fig. 4.4 illustrates the spatial reconfiguration of the networking scenarios presented in Fig. 4.3, considering the same Euclidean distances between each client and the associated FAP.

(a) Illustrative example of FAP and clients' positions defined by SLICER.

(b) Illustrative example of FAP and clients' positions for the Baseline.

Figure 4.3: llustrative example regarding both the FAP and clients positions distribution on the sports complex. Adaptation from: [20].

Taking into account the output of SLICER for the FAP position and channel bandwidth allocation, SARAF was used to assign the wireless radio resources accordingly, as well as to collect and process the network performance metrics associated with the wireless links established with all clients. The tests considering all clients were performed at the same time window for each scenario.



(a) Illustrative example of FAP and clients' positions defined by SLICER.

(b) Illustrative example of FAP and clients' positions for the Baseline.

Figure 4.4: Reconfiguration of the networking scenarios presented in Fig. 4.5, in order to ensure radio line of sight between all clients and the FAPs. (Adapted from Google Maps).

4.2 Performance Evaluation

4.2.1 Methodology

This section presents the results of the experimental tests carried out in five scenarios made up of random positions for five clients in a 40 m x 40 m area. For each scenario, each client was randomly associated with a throughput requirement of 4 Mbit/s or 20 Mbit/s, with at least two clients for each value. The set of clients with the same throughput requirement composes the same slice.

Two solutions for the placement and allocation of communications resources were evaluated: SLICER and Baseline. In the former, the FAP positions and channel bandwidth were computed by the SLICER algorithm run. For all scenarios, SLICER resulted in only one FAP, which was able to ensure the throughput requirements of all clients. The Baseline solution employs a FAP for each slice, considering each slice as an independent network with isolated resources, while employing the same exact clients' position and throughput requirements as SLICER. Each FAP was placed in the geometric center of the clients associated with the slice it serves.

For each scenario two different channel bandwidth combinations were considered:

- 40 MHz channel bandwidth for both the 2.4 GHz and 5 GHz NICs (each NIC serves a single slice);
- Exact same channel bandwidth defined by SLICER for each NIC in each scenario.

For each scenario, the following tests were carried out (c.f., Fig. 4.5):

- **An iPerf3 test to collect the throughput and packet loss ratio results, considering the following conditions:**
  - UDP traffic exchanged between the FAP and each client;
  - Packet size of 1470 bytes;
  - Target bandwidth randomly set to 4 Mbit/s or 20 Mbit/s, for each client;
  - *Reverse mode* parameter of iPerf3 set, to define uplink direction for the data traffic generated (from the client to the FAP);
  - Traffic generation during 10 s;
  - Six iterations.
- **A PING test to collect the RTT:**
  - Time interval between packets, set to 0.1 ms;
  - Six iterations.

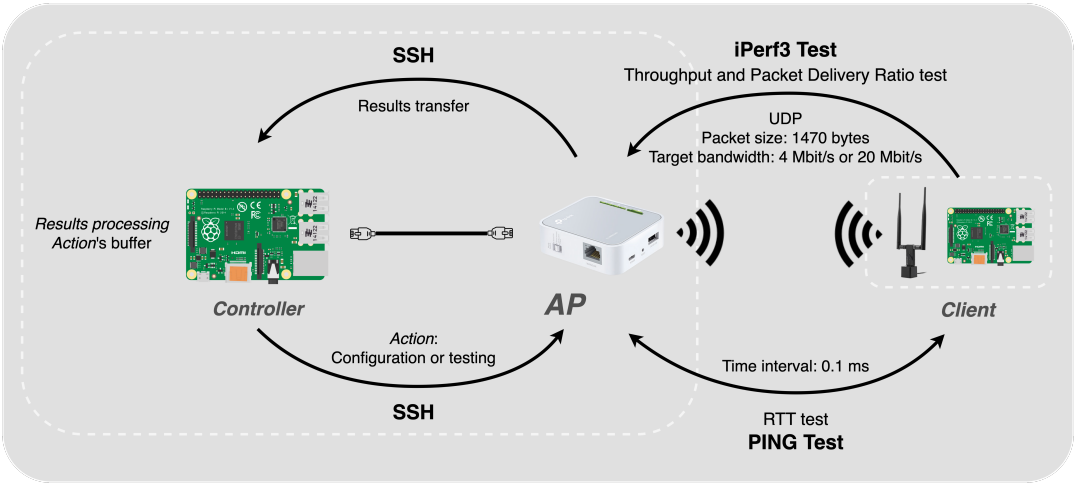
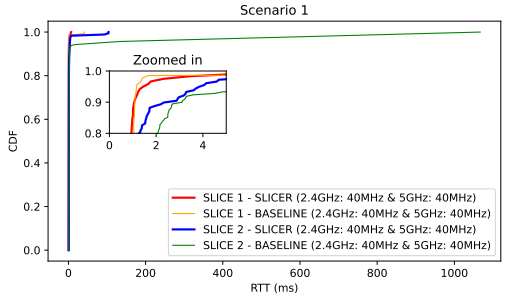


Figure 4.5: Testbed elements.

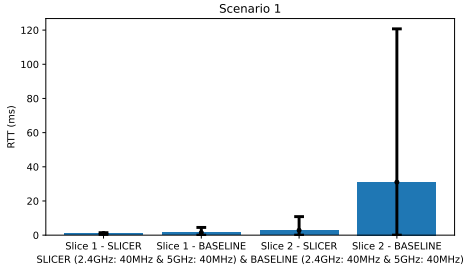
4.2.2 Results analysis

The obtained results are presented by means of Complementary Cumulative Distribution Functions (CCDFs) for the throughput and Packet Delivery Ratio (PDR). CDFs are used to represent the measured RTT. The CDF $F(x)$ gives the percentage of samples with a delay lower than or equal to x, while the CCDF $F'(x)$ gives the percentage of samples with throughput and PDR greater than x. The metrics were collected during the field testing of the previously presented scenarios.

For the first scenario, SLICER uses the maximum channel bandwidth possible for the network interfaces used (40 MHz), as the Baseline solution. However, SLICER uses only one FAP. SLICER is able to ensure the required clients' throughput (c.f., Fig. D.1) and PDR (c.f., Fig. D.3)) while using the same channel bandwidth of the Baseline. The Slicer RTT (c.f., Fig. 4.6) is lower on Slice 2, when compared with the Baseline.



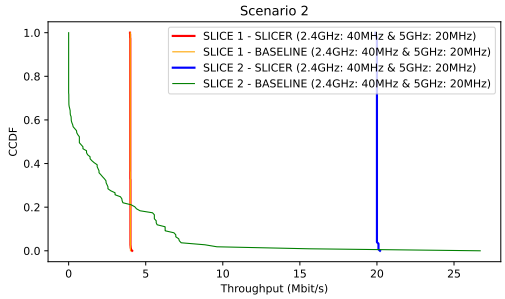
(a) SLICER and Baseline RTT CDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.



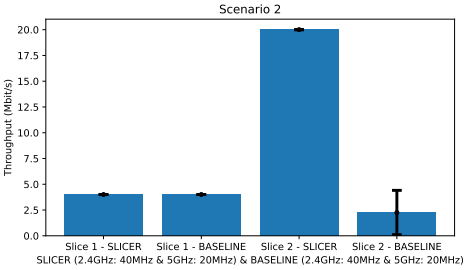
(b) SLICER and Baseline average RTT values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

Figure 4.6: Scenario 1 RTT results.

In the second scenario, the Baseline is configured with 40 MHz on both network interfaces. In this scenario, it can be observed that SLICER and the Baseline meet the throughput and PDR (c.f., Fig D.6) requirements, while SLICER does not use the maximum channel bandwidth available (c.f., Fig. 4.7). For the RTT (c.f., Fig. D.5), SLICER has a slightly higher average value and standard deviation when compared with the Baseline.



(a) SLICER and Baseline throughput CCDF, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for both 5 GHz NICs.



(b) SLICER and Baseline average throughput values, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for both 5 GHz NICs.

Figure 4.7: Scenario 2 throughput results.

In the fourth scenario, SLICER and the Baseline meet the throughput (c.f., Fig. D.10) and PDR (c.f., Fig. D.12) requirements, while SLICER uses 40 MHz on both network interfaces used. Therefore, when comparing the SLICER results with the Baseline using the same channel bandwidth, the SLICER solution grants the throughput requirements while the Baseline solution does not. The RTT achieved when using SLICER (c.f., Fig. D.11) has a higher average value and standard deviation for Slice 1 when compared with the Baseline. The results regarding all the scenarios are presented in Appendix D.

4.2.3 Discussion

In all scenarios, SLICER provides improved performance, considering throughput, RTT and PDR, when compared with the Baseline for the same amount of resources. However, a higher RTT on SLICER's slices is noticed in scenarios 2, 3 and 4. The increase was up to 1 ms at 80% of the performed tests for a single slice, which can be considered a negligible value. Different factors may be used to justify the RTT values obtained, including:

- Propagation time
- Queuing time
- Processing time
- Transmission time

Considering that all clients were placed at a distance up to 34 meters from the AP, the propagation time impact is negligible for the higher RTT obtained with SLICER: propagation delay is in the order of microseconds.

On the other hand, the queuing time also does not have a sufficient impact on the RTT results, considering that the available channel capacity is enough to carry the traffic offered by each client. In addition, SLICER and Baseline NICs have the same number of connected clients and have the same queue buffer size.

When it comes to the processing time, the CPU usage on each AP was collected on all the tests and is presented in Appendix E. It can be concluded that the RTT is not affected by the CPU usage on each AP, since the average value was approximately 10 % in the tests for all scenarios.

The main factor that justifies the increased RTT associated with SLICER is the transmission time. On SLICER, clients with different throughput requirements (4 Mbit/s or 20 Mbit/s) can be served by the same AP's NIC. Yet, since the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) assigns the same transmission time for all clients so that fairness in the medium access is ensured, clients with a lower SNR value require a longer time to transmit the same amount of information. This occurs because a lower SNR value induces the auto-rate mechanism to select an MCS index with a lower data rate. This justifies the increased RTT results obtained when using SLICER, especially for the clients offering 20 Mbit/s. In addition, since SLICER induces only a minimum MCS index with a data rate high enough to meet the required throughput for each client,

this may result in an increased transmission time when compared with the Baseline. In fact, the Baseline solution places a FAP in the geometric center of the positions of the clients associated with each slice. This allows maximizing the SNR offered to all clients and induces higher MCS indexes, which enable reduced transmission time.

Considering the experimental results obtained, it can be concluded that SLICER meets the requirements associated with multiple slices while using multiple slices with the minimum amount of communications resources. In fact, for all the scenarios considered, SLICER used a single FAP, reducing in half the total of FAPs needed to assure the targeted QoS. The higher RTT measured in some scenarios when using SLICER is negligible when compared with the Baseline.

Chapter 5

Conclusion

In this Dissertation, Network Slicing was employed to improve the overall network performance in a flying network where a dynamic allocation of the network resources over time is required to meet the QoS levels demanded by multiple ground users. Most of the research works presented in the literature are focused on best-effort solutions. This motivated a solution that considers a suitable channel assignment approach for minimizing the overall bandwidth used, while defining suitable UAV positions to guarantee the users' targeted QoS and coverage by means of a slicing-aware flying network.

We proposed the design and implementation of a slicing-aware flying network solution able to guarantee the users' targeted QoS and coverage requirements by means of a novel Slicing-aware Resource Allocation Framework (SARAF). SARAF is a modular framework able to dynamically change multiple network configurations on-demand in real-world flying networks. SARAF enables a slicing-aware network configuration, while allowing to collect and process performance metrics for monitoring the QoS offered by the slices made available. It takes advantage of a state of the art algorithm, called SLICER, for defining a suitable placement and allocation of the communications resources available. Still, SARAF is suitable to be integrated with other algorithms from the state of the art, which represents a contribution to the community.

Considering the experimental results obtained, it can be concluded that the proposed slicing-aware flying network meets the requirements associated with multiple slices while using the minimum amount of communications resources. In fact, for all the scenarios considered, the slicing-aware approach employed uses a single FAP, reducing in half the total of FAPs needed to assure the targeted QoS when compared with a baseline solution.

Concluding, all the objectives of this Dissertation were successfully achieved.

In future work, there are improvements to the proposed solution that can be considered, including:

- Integrate a function for autonomous control of the UAV positioning using SARAF;
- Improve the solution to take into account the non-omnidirectionality of the antennas used;
- Ensure system scalability by means of NIC virtualization.

Appendix A

SARAF functions

This appendix presents the network functions of SARAF that were implemented for configuring and collecting information about the status of the FAP interfaces.

SARAF 1 – Collect stations info

```
Read configurations file
Open an SSH connection
if action == changeHtMode then
    Read radioNumber
    Read htMode
    Execute UCI's command via Paramiko to set htMode on radioNumber
end if
if action == changeChannelNumber then
    Read radioNumber
    Read channelNumber
    Execute UCI's command via Paramiko to set channelNumber on radioNumber
end if
if action == getRadioOpenWrtInfo then
    Read radioNumber
    Get channelNumber
    Get channelWidth
    Get channelCenterFrequency
    Return values
end if
if action == getStationsOpenWrtInfo then
    while stationNumber ≤ totalStationsNumber do
        Get stationNumber
        Get MACaddress
        Get IPaddress
        Get signalDb
        Get txBitrate
        Get rxBitrate
    end while
end if
Reload radioNumber
```

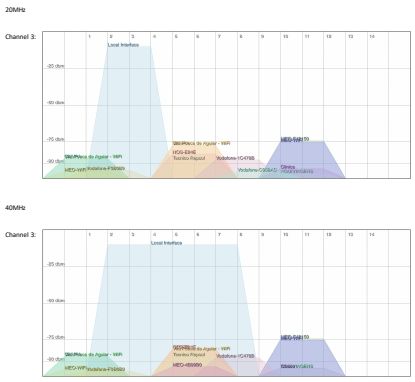
SARAF 2 – Assess throughput, PDR and RTT metrics

Open an SSH connection
if action == *iperf3Test* **then**
Execute UCI's command via Paramiko to perform a iPerf3 test on a specified *ip*, *UDPBandwidth* and *testDuration*.
if *Port* != *None* **then**
if *requestedJSON* == *True* **then**
Execute UCI's command via Paramiko to perform a iPerf3 test on specified *Port* with a JSON output.
else
Execute UCI's command via Paramiko to perform a iPerf3 test on specified *Port*.
end if
end if
else
if *requestedJSON* == *True* **then**
Execute UCI's command via Paramiko to perform a iPerf3 test with a JSON output.
else
Execute UCI's command via Paramiko to perform a iPerf3 test.
end if
Save *stdout* on a *log* file
Get *throughput*
Get *jitter*
Get *packetLossRatio*
Return *metrics*
end if
if action == *pingTest* **then**
Execute UCI's command via Paramiko to perform a PING test on a specified *ip*, *timeInterval* and *testDuration*.
Save *stdout* on a *log* file
Return RTT average value
end if

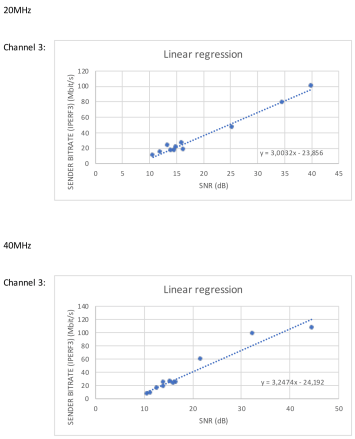
Appendix B

Experimental wireless channel characterization

This appendix presents the obtained linear regressions for the relation between experimental SNR and throughput obtained, which were considered by the SLICER algorithm.

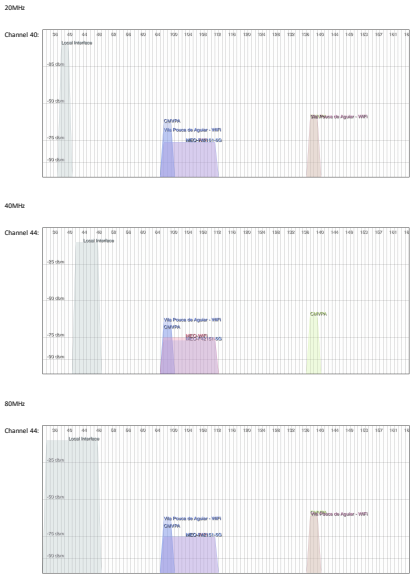


(a) 2.4 GHz wireless channel analysis. Source: OpenWrt platform.

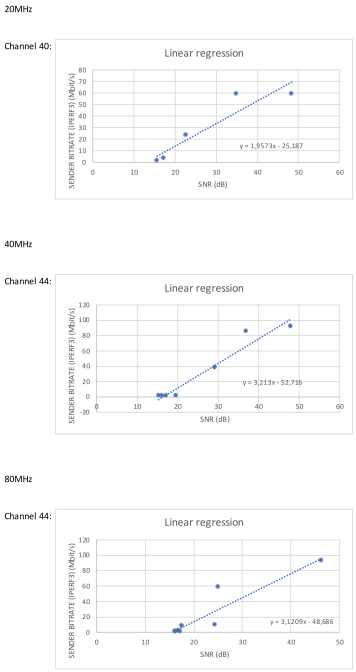


(b) Throughput vs. SNR linear regression using different channel bandwidth values.

Figure B.1: Experimental wireless channel characterization at 2.4 GHz frequency band.



(a) 5 GHz wireless channel analysis. Source: OpenWrt platform.



(b) Throughput linear regression using different channel bandwidth values.

Figure B.2: Experimental wireless channel characterization at 5GHz frequency band.

Appendix C

Networking scenarios

This appendix presents the networking scenarios considered in the performance evaluation carried out, including the FAP positions and channel bandwidth computed by SLICER.

Table C.1: Scenario 1 configuration.

Client	Slice	Throughput (Mbit/s)	Bandwidth (MHz)	X Position (m)	Y Position (m)
FAP				30	29
Client 1	Slice 2	20	40	26	32
Client 2	Slice 2	20	40	37	37
Client 3	Slice 2	20	40	2	22
Client 4	Slice 1	4	40	7	18
Client 5	Slice 1	4	40	14	35

Table C.2: Scenario 2 configuration.

Client	Slice	Throughput (Mbit/s)	Bandwidth (MHz)	X Position (m)	Y Position (m)
FAP				40	26
Client 1	Slice 1	4	20	6	30
Client 2	Slice 1	4	20	34	20
Client 3	Slice 1	4	40	26	24
Client 4	Slice 2	20	40	35	17
Client 5	Slice 2	20	40	34	32

Table C.3: Scenario 3 configuration.

Client	Slice	Throughput (Mbit/s)	Bandwidth (MHz)	X Position (m)	Y Position (m)
FAP				39	24
Client 1	Slice 2	20	40	38	25
Client 2	Slice 1	4	40	30	37
Client 3	Slice 1	4	40	26	35
Client 4	Slice 2	20	40	15	37
Client 5	Slice 1	4	40	24	1

Table C.4: Scenario 4 configuration.

Client	Slice	Throughput (Mbit/s)	Bandwidth (MHz)	X Position (m)	Y Position (m)
FAP				28	17
Client 1	Slice 1	4	20	6	28
Client 2	Slice 1	4	20	29	33
Client 3	Slice 1	4	20	19	9
Client 4	Slice 2	20	40	34	34
Client 5	Slice 2	20	40	33	17

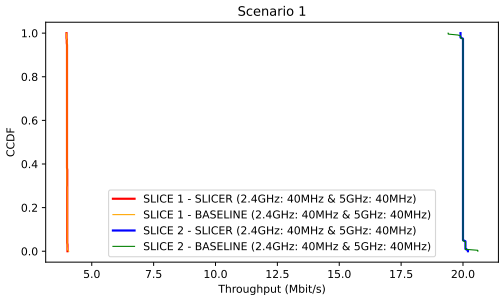
Table C.5: Scenario 5 configuration.

Client	Slice	Throughput (Mbit/s)	Bandwidth (MHz)	X Position (m)	Y Position (m)
FAP				25	16
Client 1	Slice 2	20	40	2	14
Client 2	Slice 2	20	40	14	27
Client 3	Slice 1	4	40	28	0
Client 4	Slice 1	4	40	17	25
Client 5	Slice 2	20	40	24	32

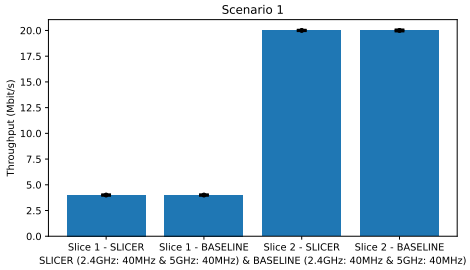
Appendix D

Results

This appendix presents the results of the experimental tests carried out.

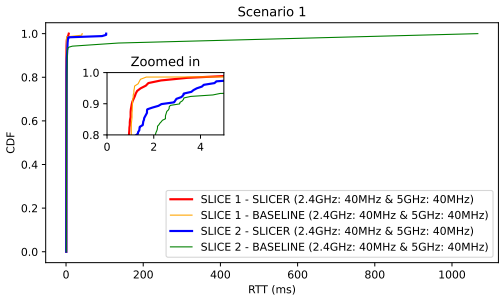


(a) SLICER and Baseline throughput CCDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

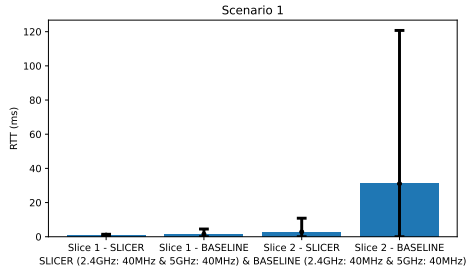


(b) SLICER and Baseline average throughput values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

Figure D.1: Scenario 1 throughput results.

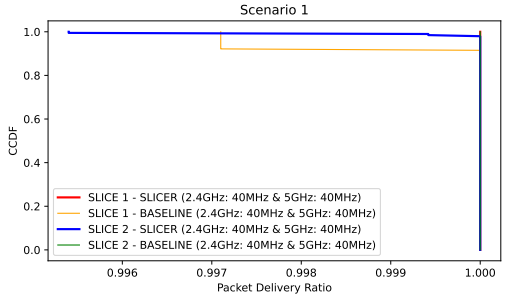


(a) SLICER and Baseline RTT CDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

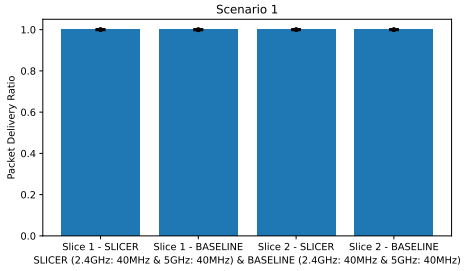


(b) SLICER and Baseline average RTT values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

Figure D.2: Scenario 1 RTT results.

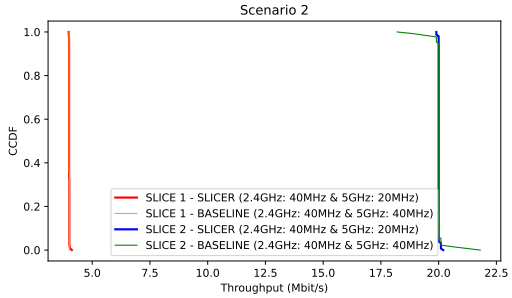


(a) SLICER and Baseline PDR CCDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

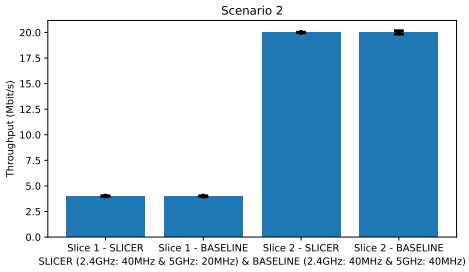


(b) SLICER and Baseline average PDR values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

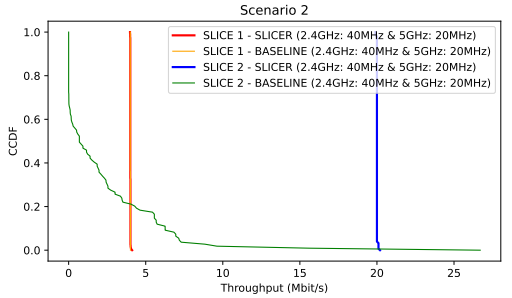
Figure D.3: Scenario 1 PDR results.



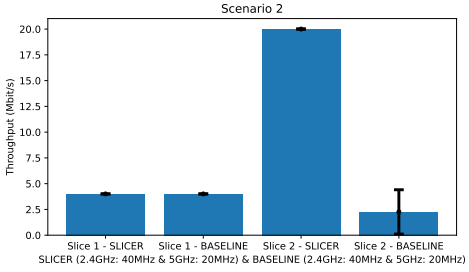
(a) SLICER and Baseline throughput CCDF, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for 5 GHz NIC on SLICER and 40 MHz channel bandwidth for 5 GHz NIC on Baseline.



(b) SLICER and Baseline average throughput values, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for 5 GHz NIC on SLICER and 40 MHz channel bandwidth for 5 GHz NIC on Baseline.

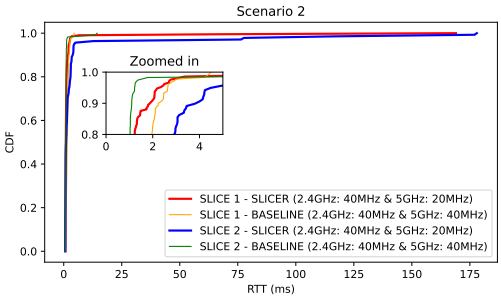


(c) SLICER and Baseline throughput CCDF, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for both 5 GHz NICs.

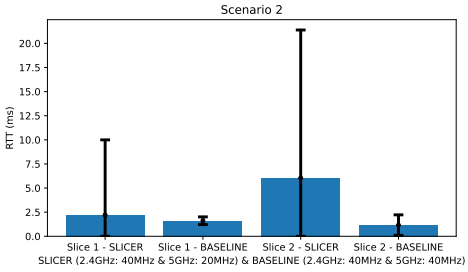


(d) SLICER and Baseline average throughput values, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for both 5 GHz NICs.

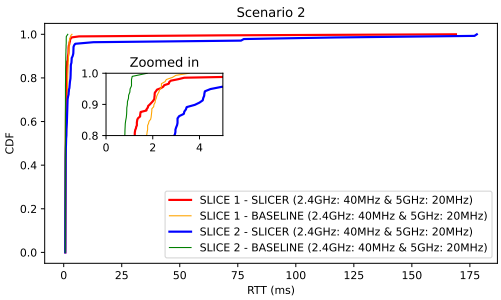
Figure D.4: Scenario 2 throughput results.



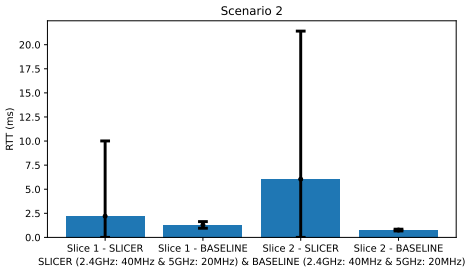
(a) SLICER and Baseline RTT CDF, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for 5 GHz NIC on SLICER and 40 MHz channel bandwidth for 5 GHz NIC on Baseline.



(b) SLICER and Baseline average RTT values, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for 5 GHz NIC on SLICER and 40 MHz channel bandwidth for 5 GHz NIC on Baseline.

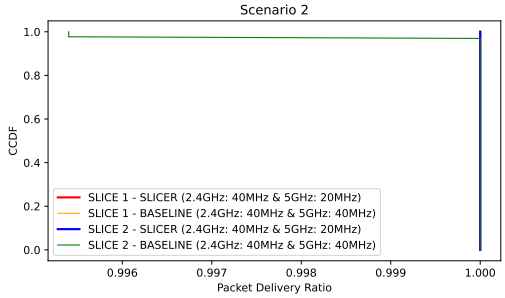


(c) SLICER and Baseline RTT CDF, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for 5 GHz NIC on SLICER and 40 MHz channel bandwidth for 5 GHz NIC on Baseline.

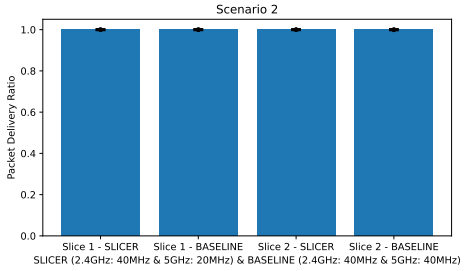


(d) SLICER and Baseline average RTT values, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for 5 GHz NIC on SLICER and 40 MHz channel bandwidth for 5 GHz NIC on Baseline.

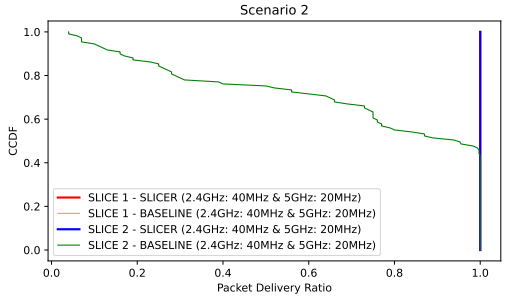
Figure D.5: Scenario 2 RTT results.



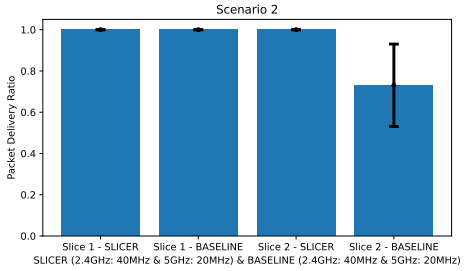
(a) SLICER and Baseline PDR CCDF, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for 5 GHz NIC on SLICER and 40 MHz channel bandwidth for 5 GHz NIC on Baseline.



(b) SLICER and Baseline average PDR values, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for 5 GHz NIC on SLICER and 40 MHz channel bandwidth for 5 GHz NIC on Baseline.

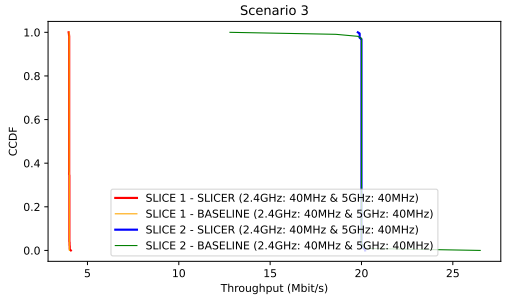


(c) SLICER and Baseline PDR CCDF, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for both 5 GHz NICs.

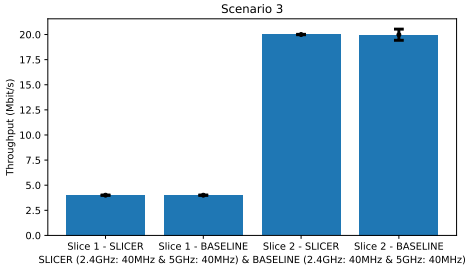


(d) SLICER and Baseline average PDR values, considering 40 MHz channel bandwidth for both 2.4 GHz NICs and 20 MHz channel bandwidth for both 5 GHz NICs.

Figure D.6: Scenario 2 PDR results.

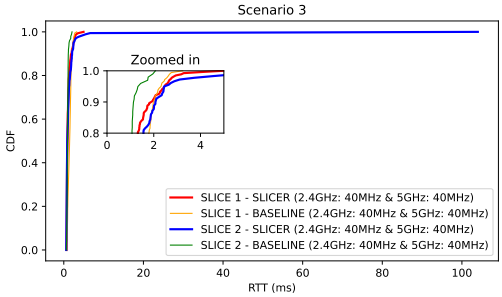


(a) SLICER and Baseline throughput CCDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

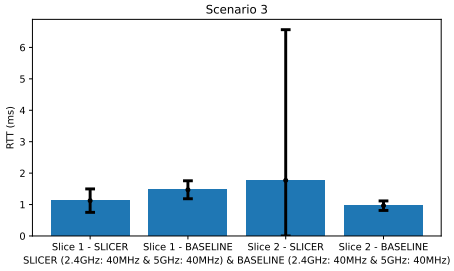


(b) SLICER and Baseline average throughput values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

Figure D.7: Scenario 3 throughput results.

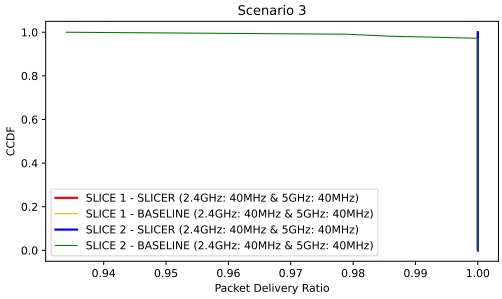


(a) SLICER and Baseline RTT CDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

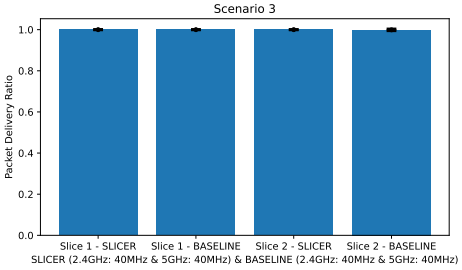


(b) SLICER and Baseline average RTT values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

Figure D.8: Scenario 3 RTT results.

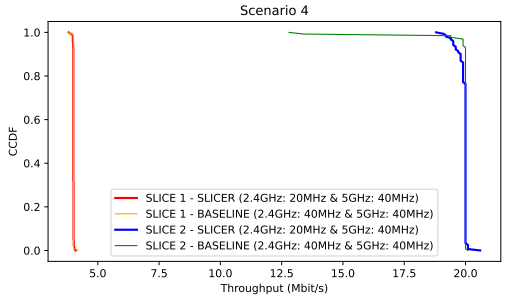


(a) SLICER and Baseline PDR CCDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

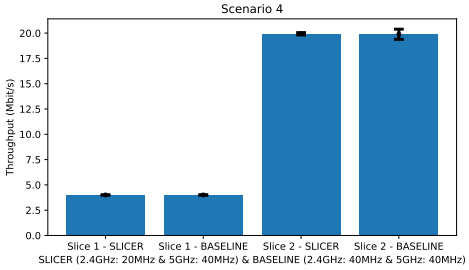


(b) SLICER and Baseline average PDR values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

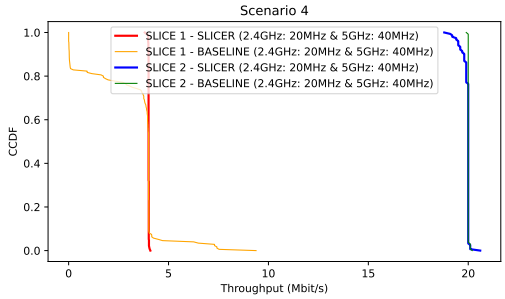
Figure D.9: Scenario 3 PDR results.



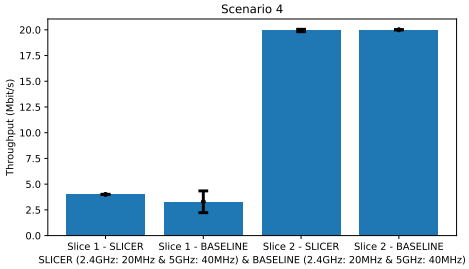
(a) SLICER and Baseline throughput CCDF, considering 20 MHz channel bandwidth for 2.4 GHz NIC on SLICER and 40 MHz channel bandwidth for 2.4 GHz NIC on Baseline and 40 MHz channel bandwidth for both 5 GHz NICs.



(b) SLICER and Baseline average throughput values, considering 20 MHz channel bandwidth for 2.4 GHz NIC on SLICER and 40 MHz channel bandwidth for 2.4 GHz NIC on Baseline, and 40 MHz channel bandwidth for both 5 GHz NICs.

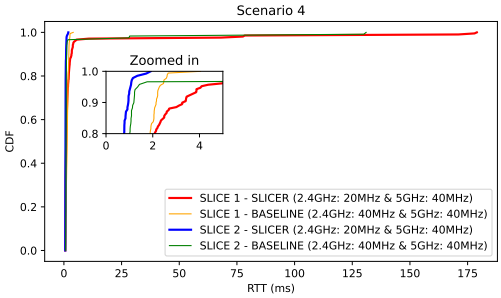


(c) SLICER and Baseline throughput CCDF, considering 20 MHz channel bandwidth for both 2.4 GHz NICs and 40 MHz channel bandwidth for both 5 GHz NICs

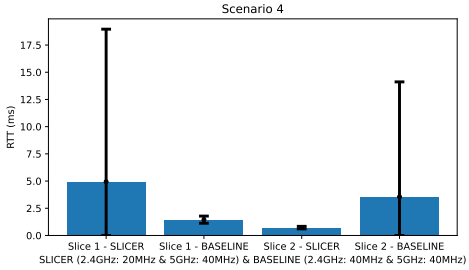


(d) SLICER and Baseline average throughput values, considering 20 MHz channel bandwidth for both 2.4 GHz NICs and 40 MHz channel bandwidth for both 5 GHz NICs

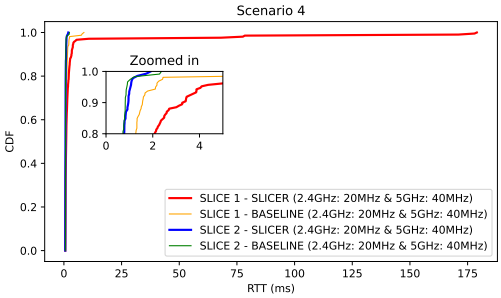
Figure D.10: Scenario 4 throughput results.



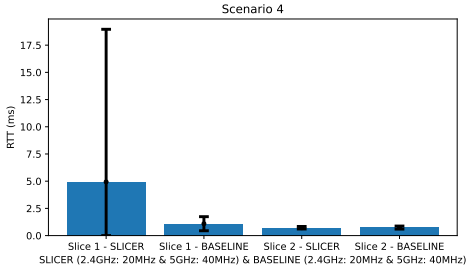
(a) SLICER and Baseline RTT CDF, considering 20 MHz channel bandwidth for 2.4 GHz NIC on SLICER and 40 MHz channel bandwidth for 2.4 GHz NIC on Baseline and 40 MHz channel bandwidth for both 5 GHz NICs.



(b) SLICER and Baseline average RTT values, considering 20 MHz channel bandwidth for 2.4 GHz NIC on SLICER and 40 MHz channel bandwidth for 2.4 GHz NIC on Baseline, and 40 MHz channel bandwidth for both 5 GHz NICs.

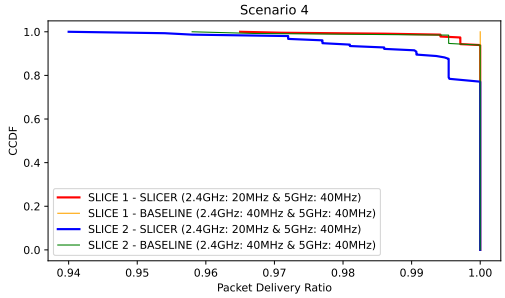


(c) SLICER and Baseline RTT CDF, considering 20 MHz channel bandwidth for both 2.4 GHz NICs and 40 MHz channel bandwidth for both 5 GHz NICs

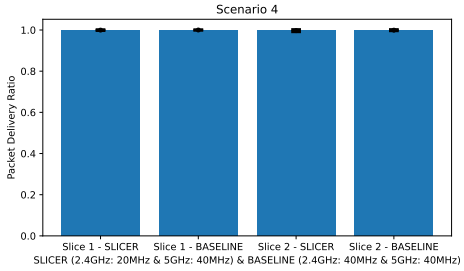


(d) SLICER and Baseline average RTT values, considering 20 MHz channel bandwidth for both 2.4 GHz NICs and 40 MHz channel bandwidth for both 5 GHz NICs

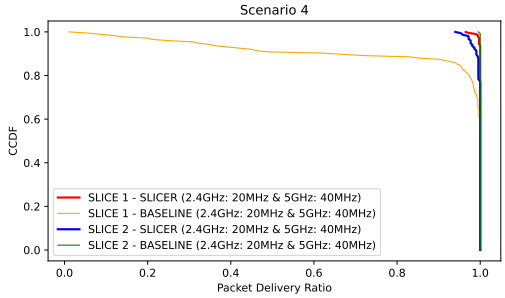
Figure D.11: Scenario 4 RTT results.



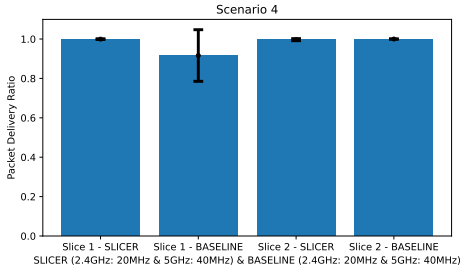
(a) SLICER and Baseline PDR CCDF, considering 20 MHz channel bandwidth for 2.4 GHz NIC on SLICER and 40 MHz channel bandwidth for 2.4 GHz NIC on Baseline and 40 MHz channel bandwidth for both 5 GHz NICs.



(b) SLICER and Baseline average PDR values, considering 20 MHz channel bandwidth for 2.4 GHz NIC on SLICER and 40 MHz channel bandwidth for 2.4 GHz NIC on Baseline, and 40 MHz channel bandwidth for both 5 GHz NICs.

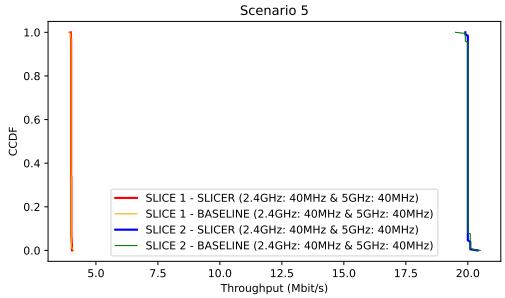


(c) SLICER and Baseline PDR CCDF, considering 20 MHz channel bandwidth for both 2.4 GHz NICs and 40 MHz channel bandwidth for both 5 GHz NICs

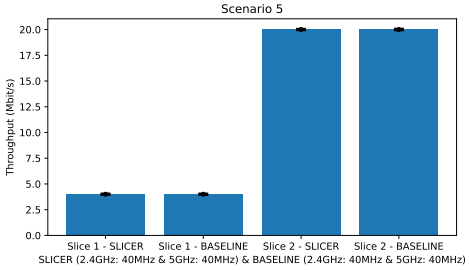


(d) SLICER and Baseline average PDR values, considering 20 MHz channel bandwidth for both 2.4 GHz NICs and 40 MHz channel bandwidth for both 5 GHz NICs

Figure D.12: Scenario 4 PDR results.

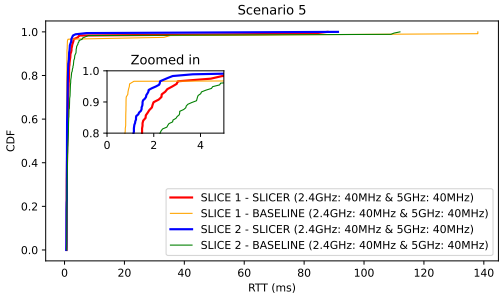


(a) SLICER and Baseline throughput CCDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

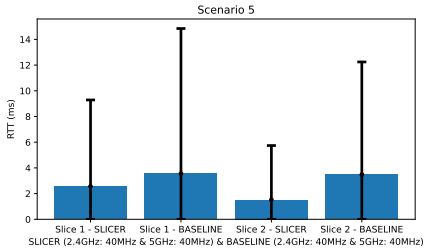


(b) SLICER and Baseline average throughput values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

Figure D.13: Scenario 5 throughput results.

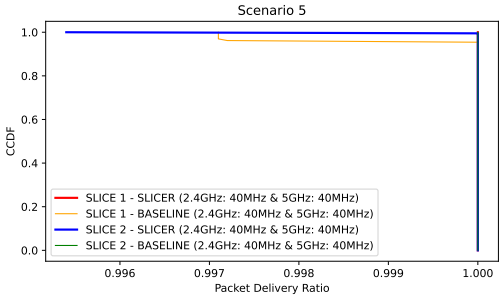


(a) SLICER and Baseline RTT CDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

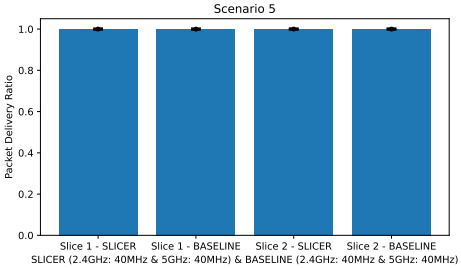


(b) SLICER and Baseline average RTT values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

Figure D.14: Scenario 5 RTT results.



(a) SLICER and Baseline PDR CCDF, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.



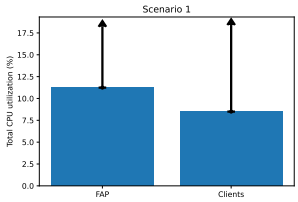
(b) SLICER and Baseline average PDR values, considering 40 MHz channel bandwidth for both 2.4 and 5 GHz NICs.

Figure D.15: Scenario 5 PDR results.

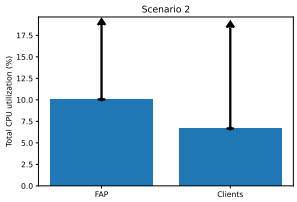
Appendix E

FAP and clients CPU utilization

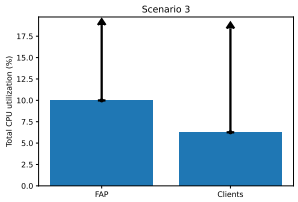
This appendix presents the CPU usage on each AP and correspondent clients collected on all the tests.



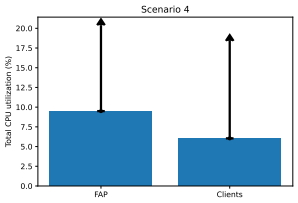
(a) Scenario 1 CPU utilization



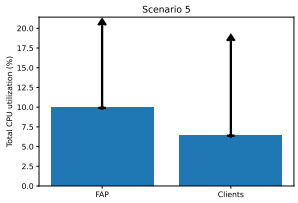
(b) Scenario 2 CPU utilization



(c) Scenario 3 CPU utilization



(d) Scenario 4 CPU utilization



(e) Scenario 5 CPU utilization

Figure E.1: Average and maximum CPU utilization percentage of FAP and clients for all scenarios.

Bibliography

[1] ITU-T N Study Group. *ITU-T Rec. Y.3011 (01/2012) Framework of network virtualization for future networks*. 2012.

[2] *Wise*. Accessed on 27/06/2022. URL: https://wise.inesctec.pt/.

[3] Joan Josep Aleixendri, August Betzler, and Daniel Camps-Mur. "A practical approach to slicing Wi-Fi RANs in future 5G networks". In: vol. 2019-April. Institute of Electrical and Electronics Engineers Inc., Apr. 2019. ISBN: 9781538676462. DOI: 10.1109/WCNC.2019.8885777.

[4] Chuan-Chi Lai, Chun-Ting Chen, and Li-Chun Wang. "On-Demand Density-Aware UAV Base Station 3D Placement for Arbitrarily Distributed Users With Guaranteed Data Rates". In: *IEEE Wireless Communications Letters* 8.3 (2019), pp. 913–916. DOI: 10.1109/LWC.2019.2899599.

[5] André Coelho et al. "Placement and Allocation of Communications Resources in Slicing-aware Flying Networks". In: *2022 17th Wireless On-Demand Network Systems and Services Conference (WONS)*. 2022, pp. 1–8. DOI: 10.23919/WONS54113.2022.9764562.

[6] Matteo Nerini and David Palma. "5G Network Slicing for Wi-Fi Networks". In: (Jan. 2021). URL: http://arxiv.org/abs/2101.12644.

[7] Sven Zehl, Anatolij Zubow, and Adam Wolisz. "Hotspot slicer: Slicing virtualized home Wi-Fi networks for air-time guarantee and traffic isolation". In: Institute of Electrical and Electronics Engineers Inc., July 2017. ISBN: 9781538627228. DOI: 10.1109/WoWMoM.2017.7974329.

[8] Sven Zehl et al. "ResFi: A secure framework for self organized Radio Resource Management in residential WiFi networks". In: *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2016, pp. 1–11. DOI: 10.1109/WoWMoM.2016.7523511.

[9] *OpenFlow Switch Specification Version 1.5.1 (Protocol version 0x06) for information on specification licensing through membership agreements*. 2015. Accessed on 13/05/2022. URL: http://www.opennetworking.org.

[10] D. Scano et al. “Network Slicing in SDN Networks”. In: *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. 2020, pp. 1–4. DOI: 10.1109/ICTON51198.2020.9203184.

[11] Julius Schulz-Zander et al. “Programmatic Orchestration of WiFi Networks”. In: *2014 USENIX Annual Technical Conference (USENIX ATC 14)*. Philadelphia, PA: USENIX Association, June 2014, pp. 347–358. ISBN: 978-1-931971-10-2. URL: https://www.usenix.org/conference/atc14/technical-sessions/presentation/schulz-zandery.

[12] Steven Hong, Jeffrey Mehlman, and Sachin Katti. “Picasso: Flexible RF and Spectrum Slicing”. In: *ACM SIGCOMM Computer Communication Review* 42 (Aug. 2012). DOI: 10.1145/2342356.2342411.

[13] C. Argyropoulos et al. “Control-plane slicing methods in multi-tenant software defined networks”. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2015, pp. 612–618. DOI: 10.1109/INM.2015.7140345.

[14] *Welcome to Paramiko’s documentation!* Jan. 2022. Accessed on 03/06/2022. URL: https://docs.paramiko.org/en/stable.

[15] *Slicing-aware Flying Communications Network - GitLab*. Accessed on 27/06/2022. URL: https://gitlab.com/jcristianorodrigues/slicing-aware-flying-communications-network.

[16] Nat. *TP-link TL-WR902AC v3*. 2022. Accessed on 03/05/2022. URL: https://openwrt.org/toh/tp-link/tl-wr902ac_v3.

[17] Rich Brown. *Welcome to the openwrt project*. 2021. Accessed on 03/06/2022. URL: https://openwrt.org/start.

[18] Jo-Philipp Wich. *The UCI system*. 2021. Accessed on 13/06/2022. URL: https://openwrt.org/docs/guide-user/base-system/uci.

[19] *Sporting Vai Treinar no Complexo Desportivo de Vila pouca de aguiar*. 2017. Accessed on 23/06/2022. URL: https://noticiasdevilareal.com/sporting-vai-treinar-no-complexo-desportivo-de-vila-pouca-de-aguiar/.

[20] *Complexo Desportivo do Chaves - Vila Pouca de Aguiar*. Jan. 2017. Accessed on 17/05/2022. URL: https://www.valentestransmontanos.com/2017/01/complexo-desportivo-do-chaves-vila.html.