Resource Allocation and Management Techniques for Network Slicing in WiFi Networks

Matías Richart*†, Javier Baliosian*†, Joan Serrat† and Juan-Luis Gorricho†
*University of the Republic, Uruguay, †Universitat Politècnica de Catalunya, Spain
{mrichart, javierba}@fing.edu.uy, serrat@tsc.upc.edu, juanluis@entel.upc.edu

***Abstract*—Network slicing has recently been proposed as one of the main enablers for 5G networks; it is bound to cope with the increasing and heterogeneous performance requirements of these systems. To "slice" a network is to partition a shared physical network into several self-contained logical pieces (slices) that can be tailored to offer different functional or performance requirements. Moreover, a defining characteristic of the slicing paradigm is to provide resource isolation as well as efficient use of resources. In this context, the thesis described in this paper contributes to the problem of slicing WiFi networks by proposing a solution to the problem of enforcing and controlling slices in WiFi Access Points. The focus of the research is on a variant of network slicing called *QoS Slicing*, in which slices have specific performance requirements. In this document, we describe the two main contributions of our research, a resource allocation mechanism to assign resources to slices, and a solution to enforce and control slices with performance requirements in WiFi Access Points.**

I. Introduction

In the last thirty years, wireless communications have been established as a commodity and have become essential in the evolution of telecommunications. The number of devices with wireless capabilities has increased dramatically, and wireless access has become the predominant way of connecting to the Internet. This aspect makes wireless networks to bear ever-increasing amounts of traffic, from web browsing to video streaming and voice calls. Even more, in the last years, there has been a trend of increasing heterogeneity in wireless access networks with new types of communications, diverse access technologies, and various offered services. In 5G, this heterogeneity increases, requiring new management and control techniques to cope with this diversity. In particular, 5G wireless networks need to provide service to a variety of different applications and use-cases employing for this purpose, diverse technologies, and equipment.

However, it is not necessary to cope with all these requirements at the same time, everywhere, and for all users and applications. Some applications may need high data rates but not a reduced latency, or some may need extreme latency guarantees with no requirement on data rates. Hence, the network infrastructure needs to deal with a diversity of traffic patterns, service requirements, and devices capabilities. This diversity is one of the new and different characteristics of 5G, which was not considered on 4G and previous systems.

Therefore, the *Network Slicing* concept has been proposed as a paradigm to enable future 5G networks to support all those various requirements. With slicing, the shared physical network infrastructure is partitioned into multiple self-contained logical pieces (called *slices*) with customized functions established to meet specific network characteristics and requirements. Furthermore, slicing allows providing better resource isolation as well as increased efficiency of resource usage. Within this paradigm, each slice is seen as a dynamic and on-demand end-to-end virtual network that allows infrastructure operators to allocate resources specifically tailored for the service provided by the *tenants*. The tenants have complete control over those isolated resources, and they use them to satisfy their client's demands.

As can be appreciated, network slicing encompasses a variety of different functional and performance requirements and is applied to the entire network. It engenders an enormous number of challenges, mainly associated with the virtualization and allocation of network resources. To implement network slicing in the Radio Access Network (RAN), specific mechanisms to allocate wireless physical resources to the slices are needed. Our research focuses on the design and implementation of resource allocation policies and mechanisms for the wireless edge of the network. In particular, the work presented here focuses on the IEEE 802.11 (WiFi) technology, for which slicing has not been thoroughly studied, despite its doubtless relevance. Notably, given that 5G networks are designed to be multi-technology, and they are being deployed to work with already existing infrastructures. In this context, available WiFi infrastructure, which is massively deployed, can be used to leverage 5G capabilities. The research summarized in this paper intends to overcome this limitation by proposing a solution to implement network slicing in WiFi Access Points (AP). Within this technology, the proposed solution concentrates on slices that demand performance requirements.

A. Objectives

The main goal of the thesis summarized here is to design and develop resource allocation techniques and mechanisms to implement Network Slicing in the context of WiFi access networks. It is expected that the developed mechanisms allocate resources efficiently, that can guarantee different performance requirements of the slices, and that maintain the isolation of the slices in terms of performance.

The specific objectives proposed for this research were:

1) To study the resource model possibilities available in the context of WiFi technology as well as to analyze possible models for the slices' requests of resources in WiFi.

978-1-7281-4973-8/20/$31.00

2) To design, develop, and evaluate enforcement and control mechanisms for the allocation of radio resources to slices in WiFi.
3) To focus the design, development, and evaluation of the proposed mechanisms in terms of efficiency, quality of service guarantees, and isolation.

B. Contributions and Publications

This thesis contributes to the area of resource management and control in wireless networks. Specifically, it proposes new approaches for dynamic resource allocation in WiFi networks for achieving Quality of Service Slicing. The main contributions are:

- A survey on wireless slicing approaches. It contributes with a comprehensive review of existent work and with a discussion of current limitations and open research problems [1].
- A resource model and allocation mechanism to partition and allocate the transmission time (airtime) in WiFi Access Points [2], [3].
- A dynamic allocation mechanism for QoS-oriented Slicing in WiFi that: implements the resource allocation in the wireless hardware, guarantees isolation, and ensures the slice performance requirements are met [4].

In this document, we briefly present the two last contributions. In Section II, we describe our slicing mechanism based on airtime sharing in WiFi APs. Then, in Section III, we briefly describe our QoS Slicing model, which includes the formulation and solution of a dynamic resource allocation problem to achieve efficient resource sharing. Finally, we present some concluding remarks in Section IV.

II. An Airtime Allocation Mechanism for WiFi

In this section, we present our mechanism for slicing a WiFi Access Point (AP) by considering the transmission time (airtime) as the resource to share. Airtime sharing implies distributing the transmission time, assigning a fraction to each slice. Achieving airtime sharing and implementing the allocation of airtime to different slices is a complex task mainly because the airtime consumed by a given communication depends on its transmission rate and the retransmissions made by the MAC layer. Then, the actual airtime consumed by a transmission must be estimated before the transmission and can only be accurately computed after the transmission completes.

To implement airtime allocation to deploy slices in a WiFi AP, we propose a queuing structure and a scheduling mechanism inspired on the work from [5] for airtime fairness among clients of a WiFi AP. Our proposal adapts and extends this airtime scheduling mechanism to be used for network slicing. We call our mechanism *Adaptive Time-Excess Round Robin* (ATERR). ATERR is envisioned to be deployed onto the APs of a WiFi network, replacing the queuing and scheduling algorithms of the AP. This proposal overcomes some of the limitations of existing works ([6], [7]), which need to manage and modify low-level EDCA parameters from the WiFi driver, such as the *Contention Window* and the *Transmission Opportunity*.

To manage the sharing of resources, we consider that our mechanism receives the requested resource share of a slice s

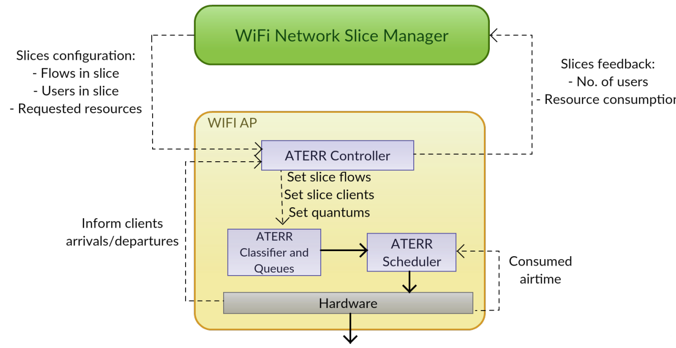

Fig. 1. ATERR Architecture.

based on three parameters: p_s, K_s, and W_s. A slice requests a share of the total airtime, as a quantified value between 0 and 1, identified by p_s ($p_s \in (0, 1]$), and represents the resources to be allocated to the slice. Also, to provide flexibility, it is possible to define a *tolerance* K_s, which measures the possible maximum deviation from the expected resource share. W_s is a time window over which the airtime allocation is computed and where the resource share plus the deviation must be guaranteed. Hence, it is required that in a time window of size W_s the slice s receives, on average, a ratio of resources between ($p_s - K_s$, $p_s + K_s$).

A. ATERR Architecture

The proposed airtime allocation mechanism can be separated into three different parts (see Figure 1): the *Controller*, the *Classifier and Queues*, and the *Scheduler*, which are all implemented as part of the WiFi device. The Controller is in charge of managing the flows and clients on each slice and on configuring the required resources to each slice. The Classifier identifies the current traffic flows and assigns a queue to each flow while the Scheduler decides which queue to provide service to guarantee each slice requirements.

The ATERR Controller is local to the AP and is in charge of communicating with the *Network Slice Manager* (a global manager with a general view of the entire network). It receives the requests of deploying new slices in the WiFi AP with a set of configuration parameters.

ATERR maintains a queue for each client's traffic within a slice. Hence, we have a queue per client and per slice. This means that if a client participates in three slices, three queues are created in the system for such a client, one per slice. In Figure 2, we show in detail the queuing structure within the ATERR Classifier and Queues. The implemented Classifier fills the different queues assigning each traffic flow to its corresponding queue following the slice's description received from the Controller.

B. ATERR Scheduling

The previously described queues are serviced following a round-robin scheduling approach, which depends on a given *quantum* of time. The quantum is a configurable parameter that controls how much airtime is allocated to each queue in one round. When a packet is dequeued and transmitted, the difference between the airtime consumed by the packet and the

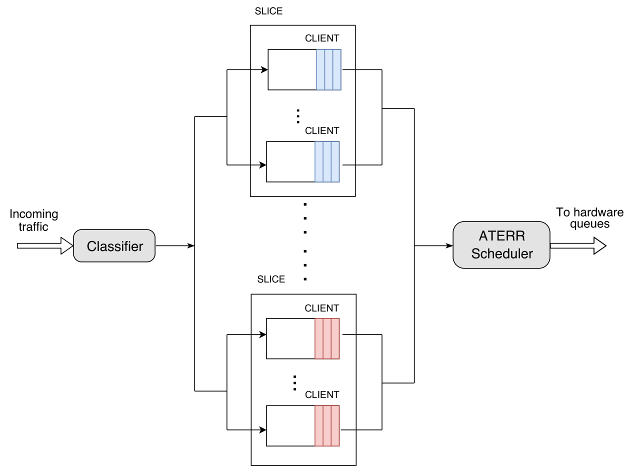

Fig. 2. ATERR Queuing Structure with the Classifier and the Scheduler.

quantum is kept in a variable called *time-excess*. Packets are dequeued while a negative time-excess remains, and when it reaches a positive value, no more packets are dequeued, and the algorithm moves to the next queue in a round-robin manner. On every new round, each queue's time-excess is updated with the previous time-excess value minus its quantum. This grants a direct control over the airtime used by each flow, regardless of the packet sizes or the transmission rate. However, since the MAC layer may perform packet-aggregation and packet retransmissions after a packet is dequeued, the actual airtime consumed is unknown beforehand. Then, the time-excess of a queue needs to be updated after each packet transmission. This may cause an allocated airtime "excess" in one round.

Given that the slice requests are expressed in ratios of airtime to be allocated, the quantum of each queue is dynamically computed based on the requested airtime and the system state. Therefore, each queue's quantum size needs to be recalculated every time a queue is created or removed (i.e., when a client connects or disconnects) on the AP. In [3], we develop a method for this quantum computation.

C. Analysis of the Solution

In [3] we devise a theoretical analysis of the mechanism, showing the parameters that influence the airtime allocation, and providing the necessary tools to guarantee a requested airtime share. In particular, we obtain a lower bound for the allocation window (W_s) needed to guarantee the required airtime share to any slice with an error lower than K_s. The obtained bound is a function of the number of queues (slices and clients) in the AP (N) and the number of clients in the slice (N_s):

$$W_s \geq \frac{T_{max}}{K_s p_s} \left(p_s \hat{N} + N_s + \sqrt{(p_s \hat{N} + N_s)^2 + (K_s p_s \hat{N})^2} \right) - N T_{max} \quad (1)$$

where $\hat{N} = N - 2N_s$.

This result is significant for the negotiation of slice requirements with a tenant, and to implement an access control mechanism for new clients or slices to an AP. Given the directly proportional relationship between the number of slices and clients per slice in the system and the size of the allocation window, it is expected that the slicing mechanism limits the number of simultaneous clients to assure the slice's requirements. This can be achieved by a client access control mechanism that can reject new clients' connections to the AP if the current slices' allocations can not be achieved with the requested window sizes. Moreover, the slicing mechanism can reject or renegotiate new slice requirements if, with the current system status (current slices' allocations and the number of clients per slice), the new slice requirement cannot be fulfilled.

III. Quality of Service Slicing for WiFi

The airtime resource allocation proposal from the previous section allocates a fixed amount of resources only based on the requested ratio of each slice, which, given the characteristics of the wireless medium, makes it impossible to guarantee any performance metric. In other words, it does not consider the possibility of adapting the resource allocation to the different slices based on the achieved throughput or delay of the slice's traffic.

Therefore, we propose a solution to implement QoS Slicing in WiFi APs, which seeks to provide a minimum guaranteed transmission bit rate and guaranteed bounded queuing delay at the AP. The complexity of this approach resides in the variability of the needed airtime to achieve a given performance because of the variability in the capacity of the wireless channel. Then, to be able to implement slices that can guarantee some performance requirements, it is necessary to dynamically allocate airtime to slices based on the current channel conditions.

In this regard, we follow the approach of *opportunistic scheduling*, where the scheduler takes advantage of channel capacities or local system information to decide the best transmission opportunity. This approach has been thoroughly studied to provide QoS guarantees [8]; however, all of the existing works have concentrated on theoretical proposals or cellular technologies, but, to the best of our knowledge, they have overlooked WiFi.

A. Qos Slicing System Model

We devise the problem of guaranteeing a minimum bit rate to each client of a slice jointly with providing an upper bound on the queuing delay, as a dynamic resource allocation problem that can be optimized. For this allocation problem, we assume that the traffic arrival rate to the different slices and clients is an unknown stochastic process. We also assume that the AP is operating in a stochastic environment, which we consider unpredictable and uncontrollable. In particular, given the medium access control of WiFi, we consider the existence of interference and congestion in the wireless medium as well as the possibility of packet collisions.

In this regard, the approach followed is to reuse the idea of airtime allocation from the ATERR algorithm. In ATERR, all the possible variations of the unknown environment are accounted in the airtime, providing an exact measurement of the time consumed in a transmission. Hence, all the unknown parameters that may affect our system are contemplated in the allocated airtime. We also propose using the ATERR mechanism but with a fixed quantum size to obtain a time-slotted system. In other words, the idea is that each time the scheduling algorithm assigns a transmission opportunity to a client, as much data as possible is transmitted to that client for the duration of the quantum. In WiFi, the transmissions are made in frames, and queues also store frames of data. Hence, it is very likely that the size of the quantum does not precisely match a given number of frames. The use of the ATERR algorithm thus becomes relevant, since the additional time consumed in one assignment is decremented from the next one.

Finally, given the previous system model, we consider the following parameters for a given AP:

- Let $\mathcal{S}$ be the set of slices instantiated in the AP.
- Let us consider that the slice $s \in \mathcal{S}$ serves a set of clients $\mathcal{N}_s$. We identify a client $n \in \mathcal{N}_s$ with the couple (n, s).
- K_s is the minimum bit rate requirement, which must be guaranteed to every client of the slice s.
- $A_{n,s}(t)$ is the arrival rate of the client n of slice s in time slot t.
- $\mathbb{C}_{n,s}(t)$ is the channel capacity between the AP and the client n of slice s in time slot t.

1) Bit Rate Modelling: The bit rate obtained by a client mostly depends on two factors: the channel capacity and the amount of time the AP transmits to that client. Therefore, we have that the bit rate to a client n of slice s, in time slot t, is given by:

$$R_{n,s}(t) = \mathbb{C}_{n,s}(t) \times x_{n,s}(t) \tag{2}$$

where $x_{n,s}(t) \in \{0, 1\}$ represents if a transmission to that client is assigned. Nevertheless, $R_{n,s}(t)$ is a random process, because the channel capacity varies randomly, depending on several factors. Accordingly, for our optimization problem formulation, we consider the expected time average of the bit rate:

$$\overline{R}_{n,s} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{\mathbb{C}_{n,s}(\tau) x_{n,s}(\tau)\} \tag{3}$$

2) Delay Modelling: As a delay guarantee, we propose to consider an upper bound on the delay of every packet. Then, our proposed model manages two decision parameters (on each time slot) to guarantee that the delay of each packet is below a given threshold: (1) select the next queue to schedule for transmission; (2) drop packets from the head of the queues (the number of packets to drop is part of the decision). Therefore, considering the possibility of dropping packets, the queues' dynamics can be expressed as:

$$Q_{n,s}(t+1) = [Q_{n,s}(t) - R_{n,s}(t) - D_{n,s}(t)]^+ + A_{n,s}(t) \tag{4}$$

where $[\cdot]^+ = \max\{\cdot, 0\}$.

To achieve bounded delay guarantees, we consider *ϵ-persistent service queues* [9] into the problem. These queues are virtual, and they do not represent real network queues, but add new constraints to the problem to assure delay guarantees. These new virtual queues are defined by the following update equation for each client n of every slice s:

$$Z_{n,s}(t+1) = \begin{cases} [Z_{n,s}(t) + \epsilon_{n,s} - R_{n,s}(t) - D_{n,s}(t)]^+ & \text{if } Q_{n,s}(t) > 0 \\ [Z_{n,s}(t) - D_{n,s}(t) - R^{max}_{n,s}]^+ & \text{if } Q_{n,s}(t) = 0 \end{cases} \tag{5}$$

where $\epsilon_{n,s}$ are pre-defined constants.

In [10] we show that bounded delay is guaranteed by any control algorithm that maintains the size of both queues $Q_{n,s}(t)$ and $Z_{n,s}(t)$ bounded by finite maximums. Intuitively, the ϵ-persistent service queue allows having a virtual queue that always has "incoming traffic," so bounding its length, jointly with the real data queues, permits to have an upper bound on the delay. Then, with this approach, the guaranteed bounded delay problem is transformed into a problem of bounding queues. Therefore, the QoS Slicing formulation we are developing needs to include this requirement of queue bounds to bound the delay.

B. Problem Formulation

Considering $\overline{u}_{n,s}$ as the expected average throughput, we define the following fair utility function, which approximates proportional fairness:

$$\phi(\overline{\boldsymbol{u}}) = \sum_{s\in\mathcal{S}} \sum_{n\in\mathcal{N}_s} \log(1 + \omega \overline{u}_{n,s}). \tag{6}$$

for some constant $\omega > 0$ and where $\overline{\boldsymbol{u}}$ is the vector of average throughputs.

Then, we can formulate a *stochastic optimization problem* that maximizes the average expected total throughput subject to the bit rate and queue stability constraints:

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{D}}{\text{maximize}} \quad & \phi(\overline{\boldsymbol{u}}) && (7) \\ \text{subject to} \quad & \overline{R}_{n,s} \geq K_s, && (8) \\ & \overline{Q}_{n,s} < \infty, && (9) \\ & \overline{Z}_{n,s} < \infty, && (10) \\ & 0 \leq u_{n,s}(t) \leq A^{max}_{n,s}, && (11) \\ & x_{n,s}(t) \in \{0, 1\}, && (12) \\ & \sum_{s\in\mathcal{S}} \sum_{n\in\mathcal{N}_s} x_{n,s}(t) \leq 1, && (13) \\ & 0 \leq D_{n,s}(t) \leq D^{max}_{n,s}. && (14) \end{aligned}$$

In this optimization problem, the objective is to find the transmission airtime assignments $x_{n,s}(t)$ and the dropping decisions $D_{n,s}(t)$ to maximize the total average expected throughput.

Constraint (8) considers the minimum average expected bit rate K_s of each slice. Constraints (9) and (10) are the stability conditions for the packet and the ϵ-persistent service queues, respectively.

C. Proposed Solution

Our proposal consists of solving the previous stochastic problem by applying the *Lyapunov Optimization Theory* described in [11]. This method allows us to build a new deterministic problem, which provides an approximate solution to the original one. Even more, such a solution can be made arbitrarily close to the optimal one, but with a trade-off on how constraints are fulfilled. Briefly, the approach consists of representing the time-average constraints (like the one on (8)) as virtual queues and formulating the optimization problem with queue stability constraints. Then, a new optimization is formulated with the joint objective of minimizing the queues variation and maximizing the utility function. The balance between these two objectives is controlled by an adjustable parameter V. As a consequence, a solution is obtained which, at every slot t, resolves an

Algorithm 1: QoS Slicing Scheduler Pseudocode.

function *Scheduler()* **is**
    output: Scheduling and drops on each slot t
    while *true* **do**
        foreach $s \in \mathcal{S}, n \in \mathcal{N}_f$ **do**
            $\mathbb{C}_{n,s} \leftarrow getCapacity(n,s)$;
        end
        $\boldsymbol{B} = \mathbb{C} * (\boldsymbol{G} + \boldsymbol{Q} + \boldsymbol{Z})$;
        $queue \leftarrow GetQueue(queueList, \arg\max \boldsymbol{B})$;
        while $queue.excess < 0$ **do**
            $airtime \leftarrow transmitPacket(queue)$;
            $queue.excess \leftarrow queue.excess + airtime$
        end
        $queue.excess \leftarrow queue.excess - QUANTUM$;
        foreach $s \in \mathcal{S}, n \in \mathcal{N}_f$ **do**
            if $Q_{n,s} + Z_{n,s} > Y_{n,s}$ **then**
                $dropPackets(n, s, D^{max}_{n,s})$;
            end
            Update virtual queues.
        end
    end
end

optimization problem and finds the airtime allocations $x_{n,s}(t)$ that must be assigned to each client of every slice. It also finds the necessary packet drops at each queue $D_{n,s}(t)$. Note that the random channel capacities of each client and the queue backlogs on time slot t act as constants in the optimization problem.

As we have already discussed, this task of assigning transmission opportunities to the different clients is performed by the *Scheduler* of the AP. Hence, based on the previous analysis, we develop a scheduling algorithm that implements the proposed solution. Given our system model, the optimization that must be solved on each time slot t corresponds to finding the client with the maximum product of channel capacity and sum of queues' backlogs: $\mathbb{C}_{n,s}(t)(G_{n,s}(t) + Q_{n,s}(t) + Z_{n,s}(t))$ where $G_{n,s}(t)$ is the virtual queue for the bit rate constraint. As can be seen in Algorithm 1, each iteration of the algorithm corresponds to a time slot where the traffic queue of the client (n, s) that maximizes this value is assigned for transmission. Then, packets are dequeued and transmitted until the quantum is totally consumed following the ATERR approach. After the transmission has ended, each client's queue is checked to decide if any packet drops are necessary. The comparison $Q_{n,s} + Z_{n,s} > Y_{n,s}$ also follows from the optimization problem where $Y_{n,s}$ is a virtual queue of the problem. Finally, the virtual queues for each client and slice are updated, considering the packets sent and dropped.

In [10], we provide proof that this solution satisfies all constraints and that the obtained utility differs from the target utility by no more than B/V where B is a constant. On the other hand, the bound over the time average queues' backlogs increases linearly with V. Then, the solution utility can be made as close to the optimal utility as V is increased but with a trade-off on the satisfaction of the constraints. In the case of our QoS Slicing problem, this trade-off translates into a compromise between the optimal utility achieved and the satisfaction of the bit rate and delay guarantees.

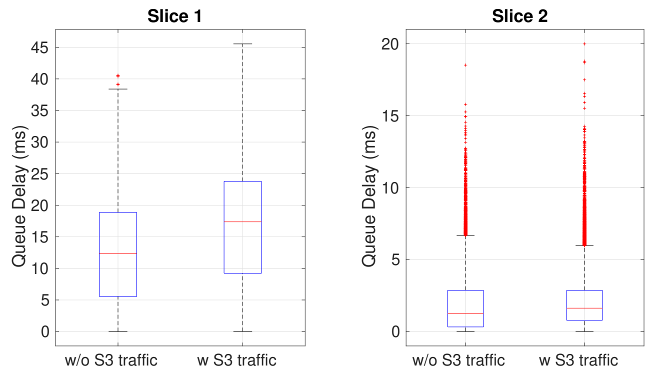

Fig. 3. Delay for Slices 1 and 2 with and without Traffic on Slice 3.

D. A Mechanism for Guaranteeing Isolation

An isolation problem appears when more resources than available are needed to satisfy all the slices' performance requests. Given that the proposed approach only works when there exists a feasible solution, we have designed a mechanism to detect and mitigate this situation. Although admission control mechanisms may prevent this from happening when instantiating new clients or slices, the channel conditions of a client might worsen after the initial connection, causing the scheduler to take resources from other slices to provide the agreed QoS.

Our proposal consists of integrating the isolation management to the scheduler, to deal with the isolation issue when a client's channel conditions deteriorate, and more resources than available are necessary. We propose a solution in two stages:

- A *monitoring stage*, by monitoring the evolution of the virtual queues, violations of the required guarantees can be detected.
- An *action stage*, where the clients and the actions to take are chosen to move the system to a stable state.

E. Experimental Evaluation

We evaluate the behavior and performance of the proposed slicing mechanism by implementing it on the MATLAB Simulink software. In the prototype, we model the queue and scheduling operation, the input traffic patterns, as well as the variable channel conditions of the wireless links. The implemented model is available at [12]. The goal of the evaluation is to show how our solution provides the QoS guarantees to slices with different requirements when deployed on a WiFi network. We tested slices with different traffic patterns and different QoS requirements, having clients with different and variable channel conditions.

The simulation scenario is composed of a single WiFi Access Point (AP) and several clients that connect to such AP. We consider three slices deployed at the AP. Slice 1 requires a minimum guaranteed bit rate of 300Kbps for each flow, with a maximum allowed delay (delay bound) of 50ms. Slice 2 requests a guaranteed bit rate of 3Mbps, with a maximum delay of 25ms. Finally, Slice 3 does not require any QoS guarantees and is tailored for bulk background traffic.

We show results with and without traffic in Slice 3 to assess how our solution correctly manages isolation and guarantees

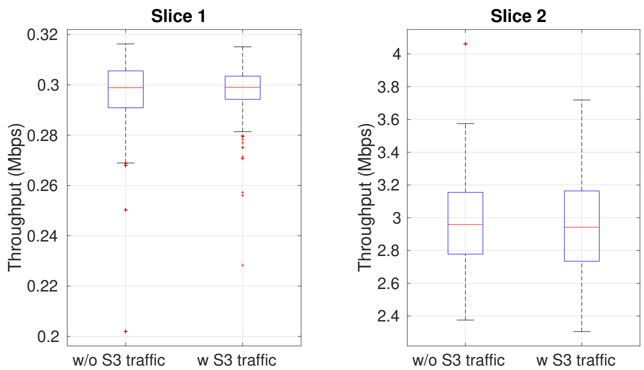

Fig. 4. Throughput for Slices 1 and 2 with and without Traffic on Slice 3.

QoS requirements from Slice 1 and Slice 2. Because of space constraints, we show, for each slice, the highest obtained delay and the lowest throughput (the worst client). In Figures 3 and 4, it can be observed that in both cases (with and without traffic on Slice 3), the QoS requirements of delay and bit rate are always guaranteed. When traffic is generated in Slice 3, there are some variations on the average delay, but the maximum delay is kept below the required bound. More extensive evaluations and results can be found in [10].

IV. Conclusions

Given the complexity of allocating resources in the WiFi technology, mainly because of its medium access mechanism, we develop an approach based on the allocation of the transmission time (airtime). The mechanism is based on considering the airtime as the wireless resource that slices can request, and that can be shared and allocated. Although airtime control has already been considered in WiFi for fairness objectives, it has not yet been considered to implement slicing. Despite its simplicity, we consider this is one of the major contributions of our research, which has already been taken and extended by other researchers. The main advantage of the proposal, in comparison with other works on service differentiation in WiFi, is that it avoids the modification of MAC-layer parameters, and it takes into account the hardware behavior, to avoid queue buildups at lower layers, and allows packet aggregation.

As already mentioned, slicing has become an essential part of the current 5G network design. In this context, providing WiFi networks with the ability to implement slicing further facilitates the integration of this technology in the 5G ecosystem. Consequently, we regard our QoS Slicing proposal for WiFi as a significant contribution to the further development of 5G. The problem formulation and the design and implementation of a solution with the Lyapunov Optimization Theory are important contributions. However, the application of this approach to the WiFi technology would not be possible without two other crucial mechanisms developed in this thesis. First, given that scheduling in WiFi is not based on time slots, we incorporate to the QoS scheduler the ATERR mechanism previously developed. This allows us to have a system that approximates a time-slotted solution and also provides feedback on the consumed airtime, which is crucial to calculate the exact channel capacity. Secondly, as the adopted theory does not consider cases when there is no feasible solution (lack of resources), we design and implement a mechanism to detect and correct unfeasible situations. In summary, we contributed with a novel mechanism to implement slices with bit rate and delay constraints in WiFi devices. The mechanism is developed by the application of a known technique but has never been applied to the WiFi technology. This novel application brought new challenges that have been worked out to achieve a comprehensive solution.

As a conclusion, we have accomplished the three objectives formulated for this research: (1) we have selected airtime as the WiFi resource to be partitioned and we have proposed two different models for requests: one based on resources, where slices can request portions of the total airtime available; and one based on performance, where slices can impose requirements on the minimum bit rate and the maximum delay allowable to its flows; (2) for each type of slice requests we have designed, developed and evaluated two different enforcement and control mechanisms; (3) the mechanisms manage resources efficiently since slices with excess traffic can benefit from the unused resources, the isolation between slices is always honored, and the QoS Slicing mechanism offers quality of service guarantees.

Acknowledgment

This work has been supported by the European Commission and the Spanish Government (Fondo Europeo de Desarrollo Regional, FEDER) by means of the EU H2020 NECOS (777067) and ADVICE (TEC2015-71329) projects.

References

[1] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, 2016.

[2] M. Richart, J. Baliosian, J. Serrat, J.-L. Gorricho, R. Agüero, and N. Agoulmine, "Resource allocation for network slicing in wifi access points," in *Network and Service Management (CNSM), 2017 13th International Conference on*. IEEE, 2017, pp. 1–4.

[3] M. Richart, J. Baliosian, J. Serrat, J.-L. Gorricho, and R. Agüero, "Slicing in wifi networks through airtime-based resource allocation," *Journal of Network and Systems Management*, vol. 27, no. 3, pp. 784–814, Jul 2019.

[4] ——, "Guaranteed bit rate slicing in wifi networks," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, Ed., April 2019, pp. 1–8.

[5] T. Høiland-Jørgensen, M. Kazior, D. Täht, P. Hurtig, and A. Brunstrom, "Ending the anomaly: Achieving low latency and airtime fairness in wifi," in *2017 USENIX Annual Technical Conference (USENIX ATC 17)*. Santa Clara, CA: USENIX Association, 2017, pp. 139–151.

[6] K. Nakauchi, Y. Shoji, and N. Nishinaga, "Airtime-based resource control in wireless lans for wireless network virtualization," in *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on*. IEEE, 2012, pp. 166–169.

[7] K. Guo, S. Sanadhya, and T. Woo, "Vifi: virtualizing wlan using commodity hardware," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 3, pp. 41–48, 2015.

[8] A. Asadi and V. Mancuso, "A survey on opportunistic scheduling in wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1671–1688, 2013.

[9] M. J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 1728–1736.

[10] M. Richart, "Resource allocation and management techniques for network slicing in wifi networks," PhD Thesis, Universidad de la República - Universitat Politècnica de Catalunya, Nov. 2019. [Online]. Available: http://hdl.handle.net/2117/173365

[11] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[12] M. Richart, "Qos slicing simulation code for matlab simulink," Apr 2019. [Online]. Available: 10.5281/zenodo.2649134