

Received May 3, 2018, accepted May 27, 2018, date of publication June 4, 2018, date of current version June 26, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2844088

Unsticking the Wi-Fi Client: Smarter Decisions Using a Software Defined Wireless Solution

JOSE SALDANA^{ID}¹, (Member, IEEE), RUBÉN MUNILLA¹, SALIM ERYIGIT², OMER TOPAL²,
JOSÉ RUIZ-MAS^{ID}¹, JULIÁN FERNÁNDEZ-NAVAJAS^{ID}¹, AND LUIS SEQUEIRA³

¹Aragon Institute of Engineering Research (I3A), EINA, University of Zaragoza, 50018 Zaragoza, Spain

²Department of Research Engineering, AirTies Wireless Networks, 34394 Istanbul, Turkey

³Department of Informatics, King's College London, London WC2R 2LS, U.K.

Corresponding author: Jose Saldana (jsaldana@unizar.es)

This work was supported in part by EU H2020 Wi-5 under Grant 644262, in part by TISFIBE through the Government of Spain under Grant TIN2015-64770-R, and in part by the European Social Fund and Government of Aragon under Grant Research Group T31_17R.

ABSTRACT This paper presents a novel software-defined wireless network architecture that integrates coordination mechanisms to enhance the capabilities of a set of central managed Wi-Fi access points (APs). The global architecture is presented in detail, where the handoff mechanism is integrated with a set of active and passive monitoring tools and other functionalities, resulting in a solution that is able to provide smart functionalities using low-cost commercial APs. The framework includes a central controller that has all the information available, and is therefore able to make smart decisions about the assignment of clients to APs. This avoids the problem of the “sticky client” that remains connected to the original AP it is associated with, rather than moving to a nearby AP, which would be a better choice. Two different test scenarios are used to compare a proactive and a reactive handoff mechanism in realistic conditions, with different walking speeds. The results illustrate the advantage of the proactive handoff, as it is more scalable and allows a better integration with other functionalities such as load balancing. The delay incurred by the handoff between APs in different channels is measured with three wireless devices, using five values for the inter-beacon time, proving that fast and seamless handoffs are possible in the scenario. The paper shows that these advanced functionalities, usually available in proprietary solutions, can also be achieved using off-the-shelf equipment.

INDEX TERMS Enterprise WLAN, seamless handoff, SDN, SDWN.

I. INTRODUCTION

During the last few years we have witnessed an important increase in the use of wireless devices (*e.g.*, smartphones, laptops and tablets), and the subsequent deployment of extensive wireless networks in areas such as business centers, airports, malls, campuses or even entire cities. This situation has boosted the emergence of solutions that include a set of coordinated Wi-Fi Access Points (APs from now), usually known as “Enterprise Wi-Fi.” However, these solutions are usually proprietary, closed and costly, which in most of the cases makes them infeasible for many organizations. Some of these solutions also lack the required flexibility, scalability, and dynamism in order to optimize the utilization of Wi-Fi networks and to alleviate spectrum congestion. In this context, the scientific community is looking for proposals for inter-AP coordination, enabling advanced features such as

load balancing, frequency planning or power control, while making use of low-cost hardware and open software.

The clients (stations, STAs from now) that connect to these wireless networks implement their own algorithms for selecting the AP to associate with. These algorithms try to select the best candidate among the different networks they can *see* in the wireless environment. For this task, they rely on the available information, *i.e.*, the power level of the beacons received from each AP in the neighborhood. As a result, these decisions are only based on local knowledge of each STA. Hence, they are totally uncoordinated and result in situations like the so-called “sticky client problem,” a term that is used to describe a STA that remains connected to an AP that it knows, even if the distance has been significantly increased due to the movement of the user [1]. This results in a rate reduction which also harms the rest of the STAs connected

to the AP [2]. Another problem is that the normal 802.11 handoff has a delay of several hundreds of milliseconds [3], which may be unacceptable for real-time applications such as VoIP or online games.

In this context, some works have proposed the adaptation of certain abstractions from Software Defined Networks (SDNs), such as the concept of *flow*, for its use in wireless networks [4], [5]. The integration of the wireless features with SDN is a hot research topic known as SDWN (Software-Defined Wireless Networks). Important modifications and extensions are needed, since SDNs do not capture by themselves all the issues appearing in wireless scenarios (interference, mobility, channels, etc.). Some proposals are in place [6], [7], and this is also the context of the present work, as it has already proven to be an effective way to achieve very fast handoffs, frequency selection, and power control in Wi-Fi networks [8].

In a previous work [9], a study of the reactive seamless handoff and the incurred delays was presented. In this article, we present a novel SDWN architecture that aims to integrate a set of mechanisms that enhance the capabilities of Wi-Fi APs, by means of a better coordination. The contribution of the present work is threefold:

- The presentation of the global architecture, where the handoff mechanism is integrated with a set of monitoring tools and other functionalities, resulting in a solution that is able to provide smart functionalities using low-cost commercial APs.
- The presentation of new monitoring functionalities, both active and passive.
- The evaluation and comparison of two handoff mechanisms (one reactive and other proactive).

The remainder of the paper is as follows: a review of the state of the art is presented in section II, the detailed explanation of our architecture is given in section III, and the implementation of the prototype is provided in section IV. Validation tests and results are detailed in section V. Finally, we conclude the paper with section VI.

II. STATE OF THE ART

A. ARCHITECTURAL PROPOSALS FOR SDWN

The Software-Defined Networking Research Group (SDNRG) of the Internet Research Task Force (IRTF) defined the layers and architecture terminology for SDN systems [10]. Even though the document did not aim to standardize any layer or interface, it provided a reference of the approaches that can be taken when defining SDN architectures. The Open Networking Foundation (ONF) proposed another architecture [11], which is more service-oriented, whereas the one proposed by the IRTF has a more functional view [12]. We will follow the IRTF approach in our proposal.

The architecture proposed by the IRTF (see Figure 1) consists of: an *Application Plane* where applications that define network behavior reside; a *Network Services Abstraction Layer* that provides access from applications of the control, management; the *Control Plane* is responsible for making

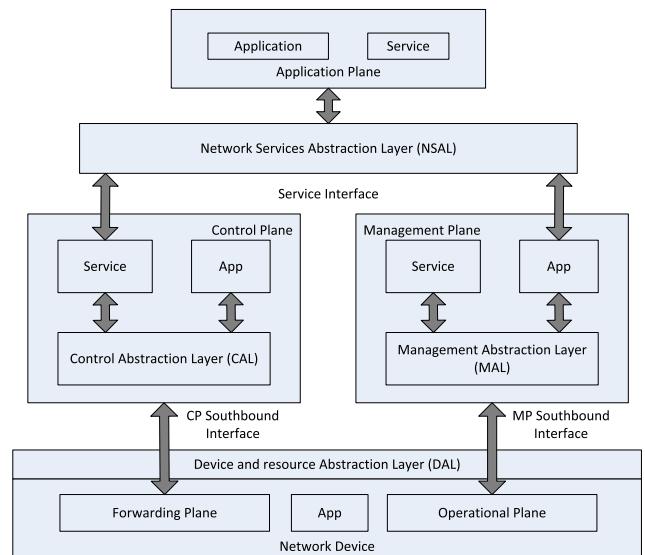


FIGURE 1. SDN Layer Architecture according to RFC 7426.

decisions on how packets should be forwarded by one or more network devices and pushing such decisions down to the network devices for execution; the *Management Plane* is in charge of monitoring, configuring, and maintaining network devices, e.g., making decisions regarding the state of a network device; and the *Device and resource Abstraction Layer* abstracts the resources of the device's forwarding and operational planes to the control and management planes.

Some examples of architectural proposals for SDWN can be found in the literature: in [13], the EmPOWER architecture was proposed, integrating different Radio Access Technologies (RAT), and proposing a set of programming abstractions to model some important aspects of wireless networks. This architecture was also used in [14], where a joint algorithm for mobility management and rate adaptation for multicast communications in 802.11 networks was proposed. In [15], another architecture was proposed aimed at minimizing the packet-level delay violation of a target service; in this case, all the Base Stations (BSs) are configured to use the same MAC address; specific virtual BSs are created for managing each service. All these proposals include a central controller in charge of network management.

B. VIRTUAL ACCESS POINTS

SDWN architectures also need to integrate the capacity to manage the specific issues that appear in wireless scenarios. For this aim, the *Light Virtual Access Point* (LVAP) abstraction was first proposed in [12] as a way for providing flexibility in wireless networks. When a STA connects for the first time, a central controller creates an LVAP for it, and assigns it to a physical AP (the most suitable one for the STA). An LVAP is just a tuple of 4 fields: a MAC address and an SSID to be used to communicate with the STA, the real MAC of the STA and its IP address. Therefore, a physical AP can host a number of LVAPs, and it will use a different one for

communicating with each client, *i.e.*, the same AP can send frames with different source MAC addresses, depending on the destination STA.

The controller can move the LVAP from one physical AP to another, according to the movement of the STA (the LVAP *travels* with its STA). As a consequence, the STA will always *see* the same AP (it is always receiving frames from the same MAC and SSID), so it will not resort to its algorithms for finding other APs. Therefore, a centralized control of the assignment of the STAs to the APs can be performed. It should be noted that this does not require any modification on the STA, which runs standard 802.11.

A distributed solution using LVAPs was introduced in [17], where a protocol for the direct exchange of information between APs was also proposed. One limitation of this proposal is that, due to the absence of a central controller, each AP has to build a list of neighbor APs by itself.

The solutions already presented in [13] and [15] also consider the use of LVAPs. Another LVAP-based solution including a central controller and a set of low-cost *OpenWrt* (<https://openwrt.org/>) APs was presented in [8], using two Southbound protocols that facilitate communication between the controller and the APs: OpenFlow, in charge of controlling the internal switch of the AP, and a new protocol called Odin, that takes care of the wireless part. The central manager is an SDN controller that combines both functionalities, and is also in charge of creating the LVAP for each STA that sends an association request. The controller also hosts a number of algorithms for radio resource management that run as applications on top of it.

C. FRAMEWORKS PROVIDING FAST HANDOFFS

The authors of [18] proposed a scheme based on cognitive management of WLAN, integrating an SDN-based distribution system, which allows checking the resource availability in order to select the best AP. It also includes a QoS-aware handover and proactive forwarding table updates of SDN-based distribution system. It is based on 802.11r, and a modification of the STA is required.

In [19] a channel scanning scheme for 802.11 WLANs was presented. Each AP was equipped with two wireless network interfaces, one of which is dedicated to broadcast beacons in other channels, thus avoiding the need of the STAs to perform active scans when they need a handoff.

In [20] a proactive scanning scheme (KHAT) is incorporated in the STA, including a Kalman filter to predict channel fading; in that moment, the STA alternates between scanning and normal periods, in order to find a better AP. The results were measured in terms of TCP throughput.

In M-SDN [21], the delay required by host-initiated handovers is reduced by means of a handover preparation performing casting to potential target APs in parallel. However, the handover is reactive as it is triggered by the detection of a signal reduction in the initial AP.

In [22], a multipath-transmission SDWN architecture (MP-SDWN) is proposed. It uses MVAP (Multi-connection

Virtual APs) in order to form a single virtual AP over different physical APs.

Some other frameworks that jointly consider SDWN and a virtual abstraction of the underlying network are CloudMAC [7], Anyfi [23] and Odin [8]. In CloudMAC and Anyfi all the wireless frames are passed to an entity called *Access Controller*, whereas in Odin they are translated into Ethernet frames by the AP, as it usually happens in Wi-Fi networks in which the APs are connected to each other, and to the Internet, by a wired network. However, Odin presents two main limitations: firstly, it assumes that all the APs operate in the same channel, which makes it impossible to perform channel planning. Secondly, there is a scalability problem, as broadcast beacons cannot be employed: the AP has to send unicast beacons with a specific MAC to each STA.

These two limitations of Odin were alleviated in [9], where a new handoff mechanism was studied, including the possibility of using different channels. In addition, the beacon rate was reduced, except during a short period after the handoff, thus improving the scalability of the system. In this work, we go a step further, by providing new monitoring functionalities, and comparing different handoff mechanisms, namely reactive and proactive.

In Table 1 we compare the different frameworks we have cited, summarizing their main characteristics:

- if the solution considers APs in different channels;
- if other RATs are considered in addition to 802.11;
- if the solution requires modifications in the STA;
- which entity triggers the handoff;
- the approach taken for managing mobility;
- the parameters used for mobility management: in many cases, the frameworks are able to base their decisions on averaged historical data;
- the kind of virtualization;
- the handoff delay: in many cases, the handoff is not evaluated in terms of delay, but in terms of its effect on the achieved throughput.

Note that the first column (*sticky client*) refers to standard 802.11 in which no special solution for fast handoff is used.

III. PROPOSED SDWN ARCHITECTURE

The name of our proposal is Wi-5, which stands for *What to do With the Wi-Fi Wild West*.¹ The main aim of the proposed architecture is to provide advanced functionalities for coordination in a wireless scenario, with the following design objectives:

- The solution has to work with low-cost APs, *i.e.*, the ones usually deployed by operators in the households of their customers.
- The use of different channels in the APs should be possible.
- It should not require any change in the user terminal.
- The users can move at walking speed.

¹H2020 Wi-5 project, <http://www.wi5.eu/>

TABLE 1. Characteristics of frameworks providing fast handoffs and/or virtualization in 802.11 WLANs.

	Sticky client	Cloud MAC [7]	Anyfi [23]	KHAT [20]	EmPO-WER [13]	Nakauchi [15]	M-SDN [21]	MP-SDWN [22]	Odin [8]	Wi-5
Multichannel	yes	yes	not reported	yes	not reported	yes	yes	no	no	yes
Multi RAT	-	no	no	no	yes	no	no	no	no	no
Requires changes in the STA	no	no	no	yes	no	no	no	no	no	no
Who triggers the handoff	STA	not reported	controller	STA	controller	controller	controller	controller	controller	controller
Mobility management	reactive	-	reactive	proactive	reactive (proactive also possible)	tested manually	reactive	reactive	reactive	reactive and proactive
Parameters used for mobility management	RSSI	-	not reported	RSSI	RSSI, loss rate. Others are possible	AP load. Others are possible	RSSI	RSSI. Others are possible	RSSI, rate, noise, number of packets	RSSI, AP load. Others are possible
Virtualization	no	virtual WLAN cards	Service-specific VAP	no	LVAPs	Service-specific BS	no	MVAPs	LVAPs	LVAPs
Handoff delay	1-2 s	not reported	not reported	not reported	not reported	65 ms	1.5 s	not reported	not reported	50 - 75 ms

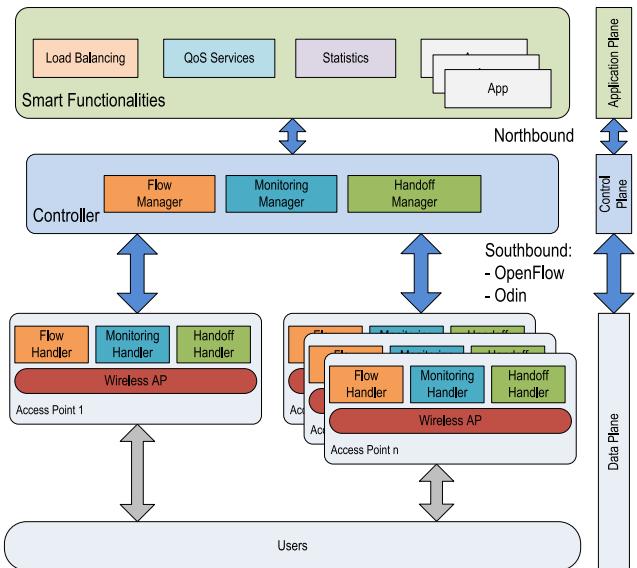
- The use of services with real-time constraints (*e.g.*, VoIP, online gaming) should be possible.

Fast and seamless handovers are a must in this kind of solutions, because it is very convenient to be able to dynamically redistribute the STAs between the APs, and this should happen without degrading the experience of the end user. For example, whenever a load balancing decision is made, or when the user is walking, the system should be able to hand off the STA in a few milliseconds.

The key feature of any SDN architecture is the *control plane*, responsible for defining and enforcing spectrum management and utilization policies. The *data plane* is in charge of moving the information packets, generated or received by the STAs, to their correct destination, using the Wi-Fi APs and the wired connection to the Internet.

Figure 2 shows the proposed architecture, which consists of two main elements: the Wi-5 *controller* and a set of Wi-5 APs. A separation between *control* and *data* planes has been implemented. Several applications (*smart functionalities* in the *application plane*) can run on top of the *controller*, interacting with it through the Northbound interface, in order to manage (by means of the Southbound protocols) the information exchange in the *data plane*.

The APs are equipped with a set of functionalities that help to optimize the utilization of the network resources. This includes a set of monitoring tools, which provide timely information about the wireless spectrum, allowing a set of resource management algorithms to enhance the QoE experienced by the users. They allow the controller to manage the

**FIGURE 2.** Architecture of the solution.

mobility, and to select the best moment for the handoff of a STA between APs.

This architecture enables an abstraction of the underlying technology for applications and services, which will make it able to interact with other technologies (*e.g.*, 4G) in the future.

Each of the elements of the controller has a corresponding one in the managed APs:

- the *flow manager* and the *flow handler* are in charge of the flows associated to the users. It allows the

management of the virtual switches via OpenFlow, the LVAPs and therefore the identification and (re-) association of the STA;

- the *monitoring manager* and *monitoring handler* continuously gather relevant information about the wireless environment (scanning) as *e.g.*, rate, frame size, power, interference levels, air time consumption, etc.;
- the *handoff manager* and the *handoff handler* are in charge of generating the required actions to re-allocate the user terminals. They allow the handoff between APs (even in different channels) without affecting IP and upper levels. For that aim, they coordinate the generation of Channel Switch Announcement (CSA) beacons when required.

One example of the joint action of the handlers would be a seamless handover caused by a load balancing decision: the *monitoring manager* decides that a handoff to another AP is required in order to grant a good QoS for a STA. It instructs the *handoff manager* to perform the handoff, and the *flow manager* to make the required adjustments in the affected control elements (LVAP, virtual switches through OpenFlow, etc.).

IV. ARCHITECTURE IMPLEMENTATION

The open source framework presented in [8] has been taken as a basis for building a prototype that integrates new functionalities²:

- The fact of using a number of different channels introduces a new degree of complexity, making it necessary to resort to the CSA element in the 802.11 beacons (as done in [17]) to make the STAs associated to a virtual AP move to a specific channel during handover.
- Different scanning mechanisms, based on the use of a second wireless interface, have been included. This interface (a low cost Wi-Fi dongle) is able to measure different network parameters (*e.g.*, signal, noise, rate, etc.) of each STA.
- Different handoff mechanisms have been implemented as individual applications (smart functionalities). In addition to the original reactive handoff, a proactive one has been implemented (they will be compared in next sections). Different metrics can be used to trigger these handoffs [24].
- The use of LVAPs requires the creation of a virtual MAC for each STA. This makes it necessary to use unicast beacons, instead of the usually employed multicast ones.
- For the sake of scalability, the inter-beacon time has been modified, and two values are used: one during normal operation, and another one after a channel switch.

The implementation is illustrated in Figure 3. Following the SDN approach, *control* and *data* planes have been separated. *OpenvSwitch*³ is installed in the Wi-5 APs, making

²All the software components of the proposed solution are available at <https://github.com/Wi5>

³Open vSwitch, <http://openvswitch.org>

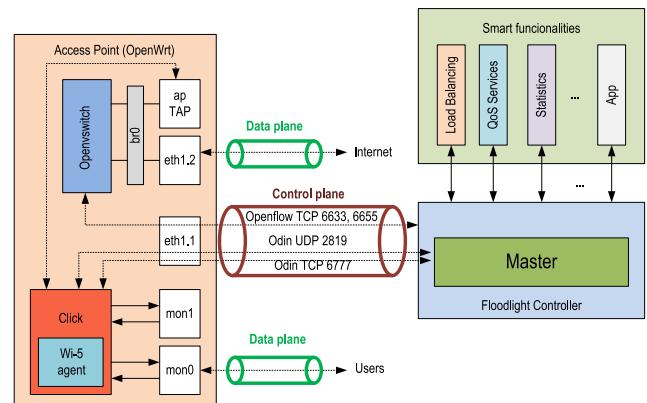


FIGURE 3. Implementation of the solution.

their internal switch behave as an OpenFlow switch managed by Floodlight Controller.⁴ In addition, *Click Modular Router* [25], with a specific module (Odin agent [8]) is run in the AP, allowing the interaction with the controller to directly manage the traffic.

OpenFlow protocol is used in a special way, *i.e.*, the rules added by the controller to these switches do not respond to new flows: they are added whenever a STA is associated to an AP, and removed when it is de-associated. Other rules are needed to make it possible for DHCP traffic to arrive to the controller.

A. MONITORING FUNCTIONALITIES

In order to observe the correct performance of the smart functionalities, a set of monitoring tools has been included in the APs and the controller. The accuracy of the gathered data is fundamental to make correct network control and resource management decisions: channel selection, power control, load balancing, etc.

The internal wireless interface of the AP is set to monitor mode (*mon0*), so it captures all the frames in its channel (we will use the term *internal* monitoring for this). The frames go to the Odin agent, which treats them according to their nature:

- Data frames targeted to one of the LVAPs hosted by the physical AP are sent to a Linux TAP interface called *ap*.
- Control frames (*e.g.*, association requests) are forwarded to the controller using the Odin connection.
- Other frames in the same channel, but not corresponding to the AP, are just taken into account for monitoring purposes, and discarded.

The auxiliary wireless interface (*mon1*) is used for off-channel monitoring (*i.e.*, in other channels -*external* monitoring-), following the requests of the controller.

1) PASSIVE MONITORING

The monitoring process obtains all the information of each frame received by the AP, available in the Radiotap⁵ header

⁴Floodlight SDN Controller, <http://www.projectfloodlight.org/floodlight>

⁵Radiotap, <http://www.radiotap.org>

(for instance: *rate*, *noise*, *signal*, *channel* and *packet length*). Once properly averaged, this information permits functionalities such as:

- Measuring airtime usage by each of the STAs associated to each AP (using the main interface), and the airtime consumed by non-Wi-5 APs and STAs (using the auxiliary interface).
- Performing a smart AP allocation when a new STA tries to join the network.
- Triggering reactive functionalities: *e.g.*, a reconfiguration of a certain AP's channel, etc.

To support the proactive functionalities in the controller (Figure 4 a), every smart application can cycle between sleeping and performing some activity. When the controller needs some information, it requests the corresponding statistics. The AP sends them and resets all the counters.

In the reactive model (Figure 4 b), the smart functionalities use a content-based publish-subscribe mechanism to be aware of certain events. For that aim, applications register subscriptions with the controller, which subsequently registers them in the APs. When an AP receives a frame, it performs a check to see if any of the collected statistics corresponding to the frame matches any of the subscriptions that have been registered with it. If there is a match, the AP sends to the controller a message, indicating the source address of the frame that

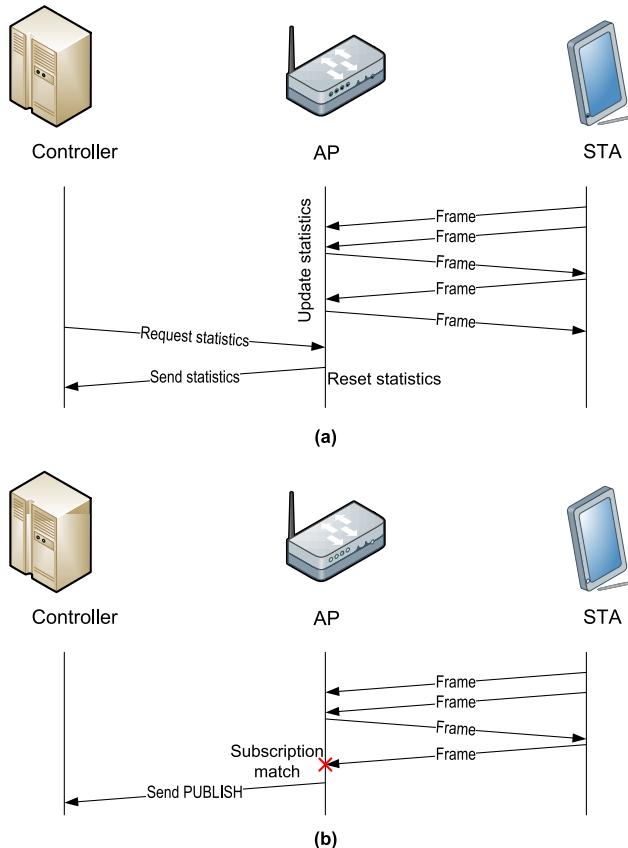


FIGURE 4. Monitoring for a) proactive; b) reactive applications.

triggered the notification. The controller then forwards the event to the corresponding application(s).

In Figure 5, we present the output of an application able to capture the status of the AP; in Figure 6, the internal statistics (using the wireless interface of the AP *mon0*) are shown; and in Figure 7, we include the external statistics (obtained with the auxiliary interface *mon1*) of the different STAs connected to all the APs in the system. All these statistics are available in the controller, which can use them as an input for its optimization algorithms.

```
[DemoStatistics] =====Internal statistics=====
[DemoStatistics] Agent [0]: /192.168.1.13
[DemoStatistics] Txpower: 10 dBm
[DemoStatistics] Channel: 1
[DemoStatistics] Last heard: 389 ms ago
[DemoStatistics]
[DemoStatistics] Agent [1]: /192.168.1.15
[DemoStatistics] Txpower: 10 dBm
[DemoStatistics] Channel: 6
[DemoStatistics] Last heard: 1893 ms ago
[DemoStatistics]
[DemoStatistics] Agent [2]: /192.168.1.14
[DemoStatistics] Txpower: 10 dBm
[DemoStatistics] Channel: 6
[DemoStatistics] Last heard: 142 ms ago
```

FIGURE 5. Monitoring: output an application using passive monitoring for obtaining detailed information of the status of the AP.

```
[DemoStatistics] =====
[DemoStatistics] <<<<< Rx statistics >>>>>
Uplink station   MAC: 30:07:4D:9E:8E:C8
IP: 192.168.2.215
num packets: 4175
avg rate: 51261.7964072 kbps
avg signal: -30.5969219499 dBm
avg length: 295.96239521 bytes
air time: 213.734666667 ms
init time: 1519733522.191784021 sec
end time: 1519733574.866460499 sec
[DemoStatistics] <<<<< Tx statistics >>>>>
Downlink station MAC: 30:07:4D:9E:8E:C8
IP: 192.168.2.215
num packets: 3849
avg rate: 54000 kbps
avg signal: 10 dBm
avg length: 702.74564822 bytes
air time: 400.721185185 ms
init time: 1519733525.415637841 sec
end time: 1519733574.344232353 sec
```

FIGURE 6. Monitoring: output an application using passive monitoring for obtaining detailed information of the STAs connected to an AP.

2) ACTIVE MONITORING

In some cases, it is necessary to resort to active monitoring, *i.e.*, sending some special frames just for monitoring purposes. As an example, in order to perform an optimal channel assignment, it would be convenient to have accurate information about the path-loss between Wi-5 APs. The objective is to build a matrix reflecting the “distance in dBs” between each pair of Wi-5 APs. For that aim, beacons with a special name are sent from one of the APs, while the rest of the APs are listening to them. As the transmit power is known, the path-loss between the transmitter and other receiver APs can be obtained. By rotating the transmitter role among all the APs, the complete path-loss matrix can be formed.

```
[DemoStatistics]
Select agent [0-2]: 0
Select channel to scan [1-11]: 2
Select time to scan (msec): 2000
[DemoStatistics] <<< Scanning in channel 2 >>>>
Station MAC: 76:A0:26:C1:E1:AA
    num packets: 1
    avg rate: 1000 kbps
    avg signal: -70 dBm
    avg length: 126 bytes
    air time: 1.008 ms
    init time: 1519733747.339689262 sec
    end time: 1519733747.339826638 sec
    AP of client: unknown
    Channel of AP: unknown
    Code: non-Wi-5 STA

Station MAC: DC:EF:09:E6:9C:DB
    num packets: 16
    avg rate: 1000 kbps
    avg signal: -54.1879014765 dBm
    avg length: 162 bytes
    air time: 20.736 ms
    init time: 1519733746.736829563 sec
    end time: 1519733748.477789793 sec
    AP of client: unknown
    Channel of AP: unknown
    Code: non-Wi-5 STA
```

FIGURE 7. Monitoring: output an application using passive monitoring for obtaining detailed information of the wireless environment.

This information can then be fed to an algorithm that calculates an optimal channel allocation [26].

In Figure 8, we show the scheme of the active monitoring application, where 4 APs have been used. It should be noted that this operation can be done without disrupting the ongoing connections, as it is performed using the auxiliary interface.

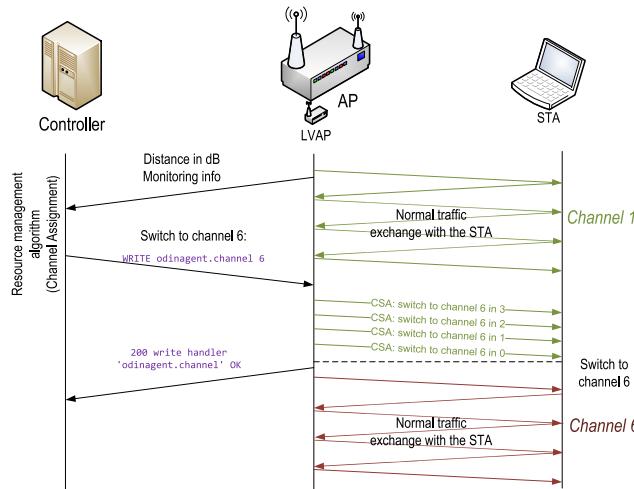


FIGURE 8. Active monitoring scheme.

In Figure 9 we provide the output of the active monitoring application, i.e., the “path-loss matrix” built after sending beacons between the APs. The information of the “distance in dB” between the 4 APs can then be used in order to subsequently run e.g., a channel assignment algorithm [26].

B. SEAMLESS HANDOFF

As aforementioned, seamless handoff is a crucial part of the proposed architecture. Not only is it used when the user

```
[ChannelAssignment] === MATRIX OF PATHLOSS (dB) ===
192.168.1.13 ----- 67.95 dB 65.52 dB 53.97 dB
192.168.1.15 64.68 dB ----- 60 dB 52.14 dB
192.168.1.14 60 dB 50 dB ----- 40 dB
192.168.1.10 54.31 dB 60 dB 41.02 dB -----
[ChannelAssignment] =====
Optimization terminated.
[ChannelAssignment] ====== CHANNEL ASSIGNMENTS ======
1 11 6 11
```

FIGURE 9. Monitoring: output an application using active monitoring for calculating the path loss matrix between 4 APs.

walks and changes its AP, but it is also necessary in many of the smart functionalities. For example, if a load balancing algorithm is run, it can decide to move a STA from one AP to another, requiring a seamless handoff.

1) REACTIVE HANDOFF

A reactive seamless handoff was proposed and measured in [9]. This smart functionality has now been improved by the inclusion of new conditions for the trigger: a subscription is set in the AP, so it sends a message to the controller if a number of consecutive frames (5 by default) are under a power threshold (*e.g.*, -56 dBm), meaning that the STA is moving away from the AP. There is a time to reset the trigger (1 s by default), and a hysteresis time (15 s by default) to avoid ping-pong effect. As an example, the condition for sending the message to the controller could be: “5 packets below -56 dBm have been received from a STA in the last 1 s, and a previous handoff of this STA has not happened in the last 15 s.”

When the controller receives the message, it sends a *Scan Request* to the APs in the neighborhood. For a short period of time (1 s by default), all neighbor APs switch their auxiliary interfaces to the channel of the origin AP and listen to packets originated by the STA. The APs that are able to hear packets of the STA, send a *Scan Response* message to the controller, which then selects the best suited AP and moves the LVAP to it.

The handoff also requires the use of beacons with the CSA element enabled, telling the STA to move to the channel of the new AP (see the detailed scheme in [9]). It should be noted that, from the point of view of the STA, this is just seen as a channel switch within the same AP, *i.e.*, the STA does not know that it has been moved from one AP to another, as the MAC address of the AP and the SSID remain the same (they are those of the LVAP). Therefore, there is no re-association nor IP address modification, so ongoing communications are only interrupted briefly due to the channel switching.

As broadcast beacons cannot be used with LVAPs, the use of two different inter-beacon times was proposed in [9] so as to improve the scalability: a value of inter-beacon time is used normally, but a burst of beacons, with a shorter time, is sent after the handoff.

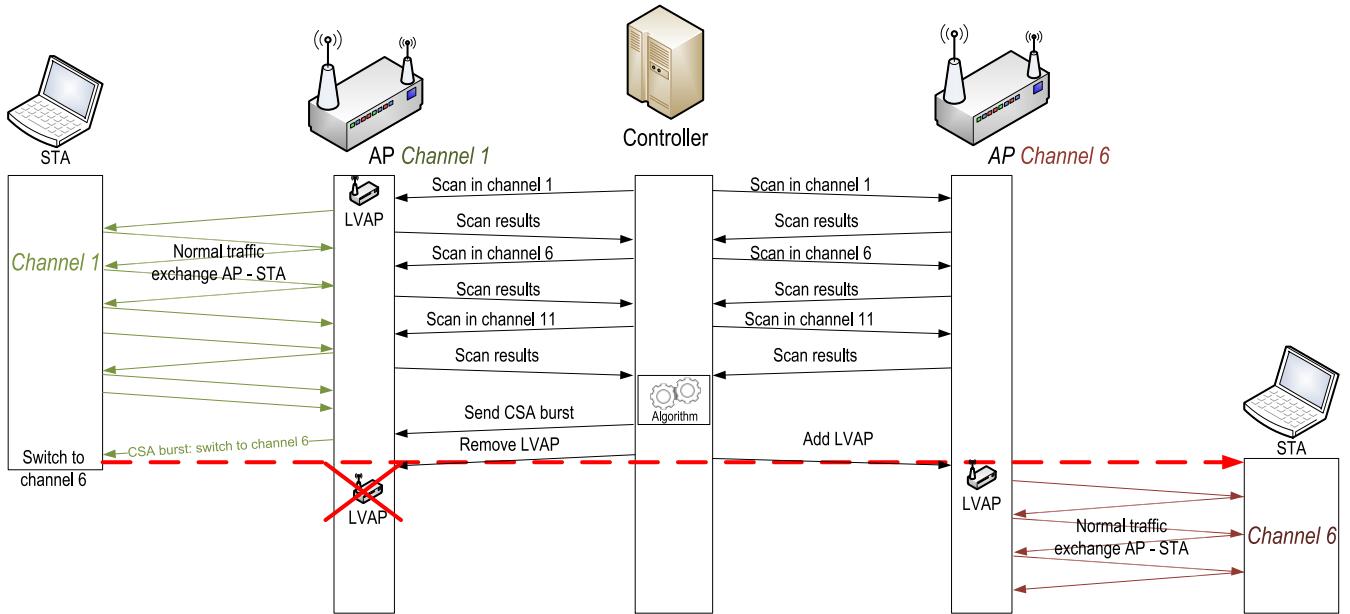


FIGURE 10. Scheme of the proactive handoff.

In the case of a handoff between APs in the same channel (as it happened in [8]), the CSA beacons are not required, so the procedure becomes simpler. In addition, the time required by the wireless card of the STA to switch its channel is null, so the handoff can be even faster.

2) PROACTIVE HANDOFF

In the last subsection we have explained the mechanism for performing reactive handoffs that are triggered by the origin AP, *i.e.*, the one to which the STA is associated. However, another option is to define a proactive smart functionality in the controller, following a *scan - gather results - make assignment decision* loop. If this loop is fast enough, it will perform a sort of fast load balancing, which can at the same time be in charge of mobility management, *i.e.*, assigning the STA to the best suited AP.

As shown in Figure 10, the controller asks all the APs to scan for STAs they can hear, using their auxiliary interfaces for a short period of time (200 ms by default on each channel). After that, it is able to build a matrix of the STAs that can be heard by each AP, and their respective RSSI values.

In order to keep track of some historical data to be used for the decisions, a *weighted RSSI* (*wRSSI*) is defined. Its value is updated using the average power level measured in the last iteration. $RSSI_{i,j}$ stands for the average value of the RSSI of STA_i at AP_j, measured in milliwatts.

This is the matrix with the *weighted* values of RSSI:

$$\begin{pmatrix} wRSSI_{1,1} & wRSSI_{1,2} & wRSSI_{1,3} & \dots & wRSSI_{1,k} \\ wRSSI_{2,1} & wRSSI_{2,2} & wRSSI_{2,3} & \dots & wRSSI_{2,k} \\ wRSSI_{3,1} & wRSSI_{3,2} & wRSSI_{3,3} & \dots & wRSSI_{3,k} \\ \dots & \dots & \dots & \dots & \dots \\ wRSSI_{m,1} & wRSSI_{m,2} & wRSSI_{m,3} & \dots & wRSSI_{m,k} \end{pmatrix} \quad (1)$$

On each iteration, the new value is calculated this way:

$$wRSSI_{i,j}^{N+1} = \alpha \cdot RSSI_{i,j}^N + (1 - \alpha) \cdot wRSSI_{i,j}^N \quad (2)$$

Where $RSSI_{i,j}^N$ stands for the average value measured in the last iteration (*i.e.*, the average power with which AP_i receives the frames from STA_j at the N^{th} iteration), $wRSSI_{i,j}^N$ represents the N^{th} value of the weighted RSSI, and $0 < \alpha < 1$ is the smoothing factor.

The value of α modifies the behavior of the algorithm: if it is low (*e.g.*, 0.2), the algorithm is slow and gives more importance to the history, *i.e.*, it takes time to make the handoff. If it is set to a high value *e.g.*, 0.8, the STA is moved fast between APs, as the importance of the most recent observation increases. In subsequent sections we will present some tests aimed at empirically finding an optimal value for this parameter.

It should be noted that the values of *wRSSI* are stored for all STA-AP pairs (not only for the AP where the STA is associated). The initial value is -99.9 until the AP receives the first frame from a STA.

Using this information, an algorithm is executed to optimally assign each STA to the best suited AP. For example, an application may decide that a handover is needed when all of the following conditions hold: *a*) there is an AP with a better RSSI than the current one; *b*) a hysteresis time has passed; and *c*) a power level threshold is reached.

If the new STA assignment requires an AP re-allocation, the handoff mechanism is performed in a similar way to the reactive one, *i.e.*, a burst of CSA beacons is sent to the STA, and then the LVAP is moved just after it. This results in a fast and seamless handoff. From that moment (red dash line in Figure 10), the STA will start working in the new channel.

V. EVALUATION

In this section we present an evaluation of the functionalities just presented. We will first describe the scenarios used for the evaluation. Then, we will present some qualitative results aimed at illustrating the advantages of the proposed architecture. Next, a series of tests comparing the handoff mechanisms, including their advantages and drawbacks will be included. Finally, some quantitative measurements will be presented.

A. TEST ENVIRONMENTS

Two different scenarios have been used for the tests:

- AirTies's Mecidiyekoy Test House (AirTies from now), a two floor apartment in Istanbul, which provides a controlled environment that allows automated and repeatable tests by the use of robots with wheels, which move through predefined paths carrying wireless devices on them. APs can be placed in four different locations. Its layout is given in Figure 11 a (only the top floor has been used), and a robot carrying two mobile handsets can be seen in Figure 11 b.
- A corridor between different labs in a building of University of Zaragoza, Spain (Unizar from now), see Figure 12. This is a scenario with about 15 interfering APs. The tests are performed by a walking person carrying a wireless device. As it is a more aggressive scenario, the repeatability of the tests is not granted, so we will only use it for obtaining qualitative results.

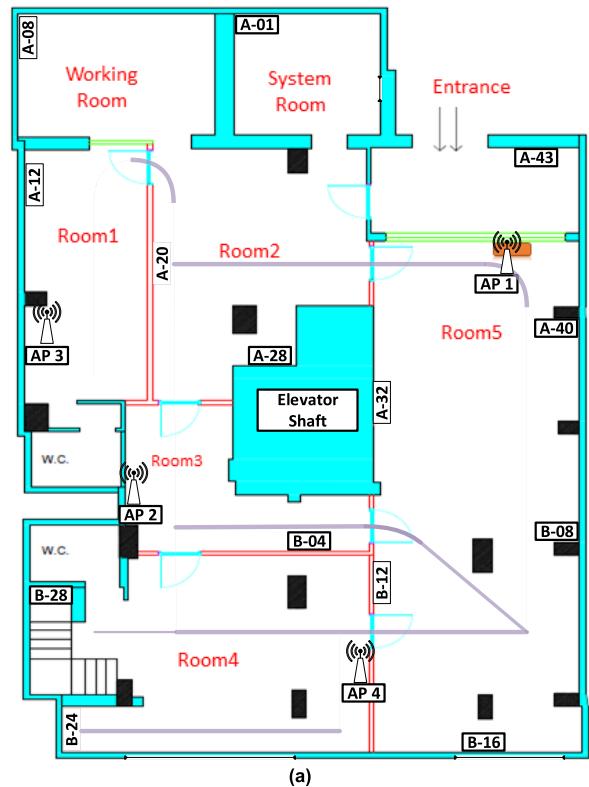
We will use one scenario or another depending on the kind of tests to be performed. Both setups include a number of off-the-shelf APs (Netgear R6100 or TP-Link 1750 Archer with OpenWrt 15.05). They use an auxiliary USB TP-LINK TL-WN722N wireless card. The APs are configured in different channels in the 2.4 GHz band; the controller runs Debian 8.2 (Linux kernel 3.16.0.4); a DHCP server is also included.

The network scheme used in both test scenarios is shown in Figure 13. Control and data planes are separated. The traffic is sent from the moving STA to a server, located in a VM in a computer that also hosts the Wi-Fi controller. Flows of UDP packets with a constant rate are used for the tests, so as to estimate the handover time as a function of the number of packets that do not arrive to their destination.

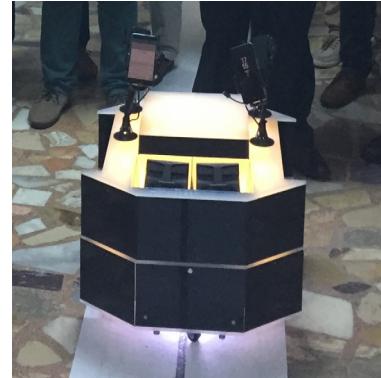
B. QUALITATIVE RESULTS

In order to illustrate the difference between state-of-the-art handoff and our proposals, we have run some experiments in Unizar using a Wi-Fi network with 3 APs, with a user carrying a laptop moving within the coverage of the network. The STA runs Debian 9.1 workstation (Linux kernel 3.16.0.4). The wireless card of the STA is a LINKSYS WUSB54GC dongle.

Figure 14 a shows a conventional handover experiment: the laptop first connects to AP15, as it is the closest one when the user starts walking (see Figure 12) down the corridor (the power received in the AP is represented by the blue line



(a)



(b)

FIGURE 11. Scenarios for the tests: a) AirTies test house; b) AirTies test robot.

“monitor AP15”). As the user keeps on moving, the received signal from AP15 becomes weaker and the signal received from AP14 becomes stronger (green line “monitor AP14”) as it is closest to the client. However, the client does not initiate the handover until it loses connectivity with AP15 (around $t = 50$ s), at which point the service is interrupted as proven by the sharp drop in the measured throughput. At this moment, when the client initiates the handover procedure by scanning from closer APs, it detects that AP13’s signal (red line “monitor AP13”) is stronger than that of AP14, and connects to it. In other words, it has ignored AP14 although it was the closest AP during a significant time. This is an example of the “sticky client” effect, i.e., the STA prefers to

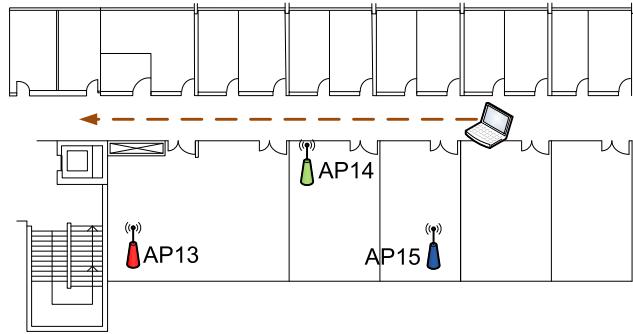


FIGURE 12. Scenarios for the tests: Unizar lab.

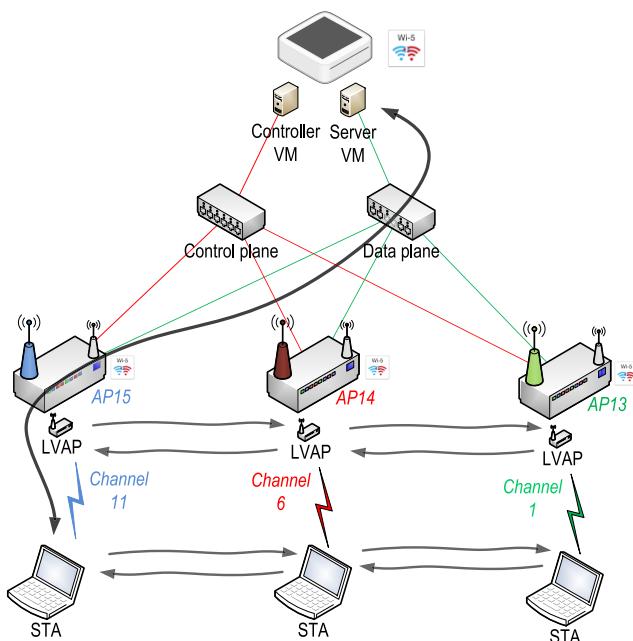


FIGURE 13. Network scheme used in the tests.

stay in a known AP than to explore new options. It is obvious that the conventional handover does not always guarantee seamless transition when a wireless client is mobile.

In contrast, when the developed Wi-5 functionalities are activated (Figure 14 b), a different behavior is observed. In this case, a reactive application runs in the controller: around $t = 20$ s, the signal from AP15 drops below the predefined threshold (amber line), which is reported to the controller. The controller then requests the two other APs to scan for the STA for 4 s. According to the results reported by the APs, the signal from AP14 (green line) is the strongest one. Therefore, AP14 is selected and the STAs is handed off to it. The same thing happens around $t = 40$ s. In both cases, the throughput at the STA (black line) is barely affected, i.e., the handover of the client's connection from one AP to the other was performed seamlessly and without interruption of service.

Finally, in Figure 14 c we present the results of a similar experiment, in which the proactive handoff application has been used ($\alpha = 0.8$, hysteresis time = 4s). As explained

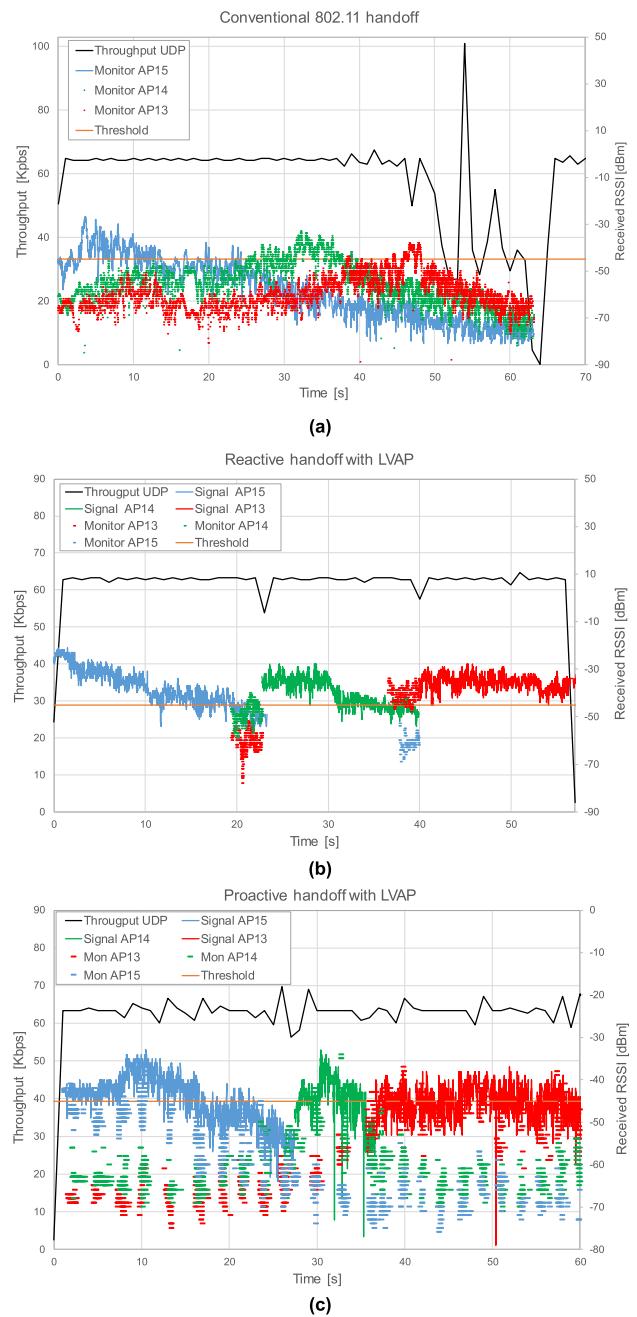


FIGURE 14. Comparison between Wi-Fi handoffs: a) state-of-the-art; b) reactive handoff; c) proactive handoff.

before, this application performs a continuous scanning of the channels, and periodically builds a matrix of STAs and APs. The figure shows that, during the first 25 seconds, AP15 is the best suited one to serve the STA: the samples obtained by the two other APs are always below. However, at $t = 25$ s, the monitoring interface of AP14 starts reporting better RSSI values. After some time, the controller selects this AP and moves the STA to it. A similar process happens at $t = 35$ s. It can be observed that the controller performs a complete scanning every 2-3 seconds, which is enough for giving the best service to a walking user.

It should be remarked that these three experiments have been run using the very same hardware. The only difference is that in the first case the Wi-5 software was not running, and in the two others we activated it, with the corresponding functionalities, showing a better performance which allows the STA to use the best suited AP for it on each moment.

1) LIMITATIONS OF THE REACTIVE HANDOFF

In this subsection we present the results that have been obtained after performing some scalability evaluations with the reactive handoff in the AirTies test scenario (Figure 11a).

Two APs have been used, and a number of STAs are initially very close to and connected to *AP1*, sending TCP traffic (generated using *iperf*)⁶ to a computer connected to the data network. Three STAs (*STA2* and *STA3* used in the experiments are mobile handsets, whereas *STA1* is a notebook computer) are carried by a robot, so they move slowly towards the second AP simultaneously.

The handover delays for individual STAs as well as the overall average (of all involved STAs) for each case are given in Table 2. When there is a single STA, the handover delays are relatively reasonable with an average value of 21.9 ms. On the other hand, an increase in the handover delay is observed when the number of stations increases. The average handover delay with 2 STAs is around 1.5 seconds, whereas it is around 1.8 seconds with 3 STAs. The reason for this is that a number of scanning requests arrive to the AP almost simultaneously, as the AP reports that it is “loosing” the signal of the STAs. The AP has to scan for each MAC address for a period of 1 second, so the rest of the scanning requests are put in a queue. This is the reason why the handoff delay increases with the number of STAs.

TABLE 2. Handover delays for cases with different number of STAs [ms].

	1 STA		2 STAs		3 STAs	
	STA1	STA2	STA1	STA2	STA3	
Average	21.9	17.4	1645.8	112.9	721.8	3137.1
Std. dev.	16.1	27.5	3097.4	259.2	1271.5	4010.9
Overall avg	21.9		1562.5		1847.2	

This limitation does not appear in the proactive handoff scheme, as there is a continuous loop running, despite of the movement of the STAs. Therefore, we can conclude that the proactive handoff mechanism presents some advantages:

- New parameters, in addition to RSSI, can be considered, in order to make smarter algorithms. Some examples could be load balancing based on the traffic of an AP, on the services being run, etc.
- The mobility management functionality can be integrated with load balancing. This avoids potential problems caused by contradictory decisions between load balancing and mobility management.
- Improved scalability, as the scanning time is shared for all STAs, whereas in the reactive handoff the application

⁶iPerf, <https://iperf.fr/>

has to order the scanning for each terminal spending longer time. This can happen if a user is carrying two devices, e.g. using a tablet and carrying a mobile handset in his/her pocket.

- The parameters that are used for making the decision are more “correlated” in time, i.e., they correspond to the same interval for all the STAs.
- Some historical information can also be used, giving specific weights to new and to old data. Therefore, the decision is not just based in the information obtained in the last seconds.

2) OPTIMIZATION OF THE PROACTIVE HANDOFF PARAMETERS

As explained in Section IV.B, the proactive handoff has two main parameters, namely the hysteresis time, which avoids frequent handoffs, and the value of α , which gives a higher or lower weight to the historical data with respect to the last measurement.

In this subsection we present some empirical tests aimed at obtaining the optimal values for these two parameters in a realistic scenario. Therefore, they have been performed at AirTies test house. A tour is as follows (see Figure 11 a): a user starts in front of *AP1* (room 5), goes to *AP4* following the path (room 4), then to *AP2* (room 3), to *AP3* (room 1) and finally back to *AP1* (room 2 and room 5). Therefore, the total number of handoffs should ideally be 4 per tour.

Tests at different speeds have been performed: *a*) a robot at low speed; *b*) a person at walking speed (*normal walk*); and *c*) a person walking fast (*fast walk*). The results for different values of α with a fixed hysteresis time value of 4 seconds are presented in Table 3, including the average number of handoffs per tour (an average of 4 tours), and also the average number of decisions (not all decisions result in a handoff), which depends on the duration of each test. For the given layout and movement pattern, we expect to have 4 handovers each tour, as the STA moves in front of each AP once during a tour.

TABLE 3. Effect of α parameter: average number of handoffs per tour.

	Handoffs $\alpha=1$	Handoffs $\alpha=0.8$	Handoffs $\alpha=0.6$	Number of decisions
Robot	7.75	4.5	4	564
Normal walk	5.75	3.75	2.5	178
Fast walk	5.75	3	2	105

We observe that a value of $\alpha = 0.8$ provides good results for the three speeds, as the average number of handoffs is nearly 4.

Another test battery has been run in order to find the optimal value of the hysteresis time. In Table 4, we show the number of handoffs obtained for $\alpha = 0.8$, when using different values for the hysteresis time. It can be observed that a time of 4 seconds is a good option for this parameter

TABLE 4. Effect of hysteresis time: average number of handoffs per tour.

	Handoffs time = 2s	Handoffs time=4s	Handoffs time=10s	Number of decisions
Robot	7.5	4.5	4	564
Normal walk	7	3.75	3.75	184
Fast walk	7	3	2.75	100

as the average number of handovers is close to 4 for different mobility speeds for that case.

C. QUANTITATIVE RESULTS

This section presents some quantitative measurements of the handoff delay. This parameter is of primary importance in the proposed solution, as it is intended to support real-time services in which disruption can be critical: for example, in a Skype call or in an online game party, being disconnected for 1 or 2 seconds may jeopardize the user's experience.

1) TEST METHODOLOGY

The methodology considers a setup including two APs and a single STA. Two parameters are varied: inter-beacon time and the wireless card. An application is set in the controller, which every 3 seconds hands off the STA between the two APs. The test environment is Unizar lab, *i.e.*, a noisy one.

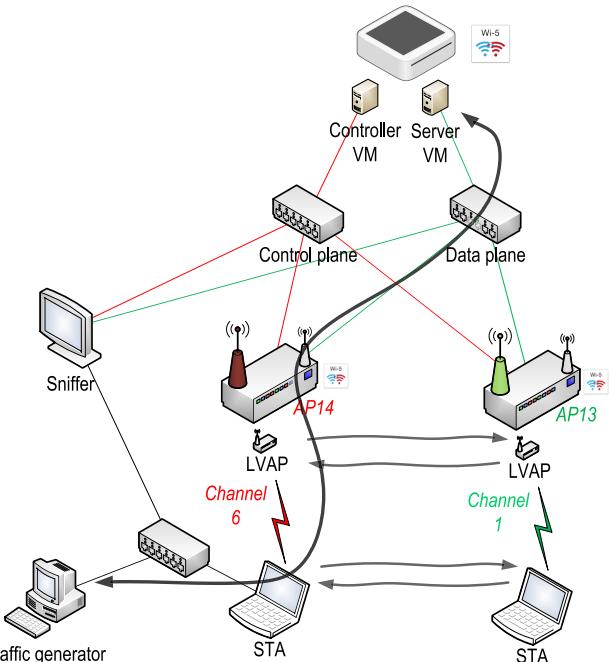
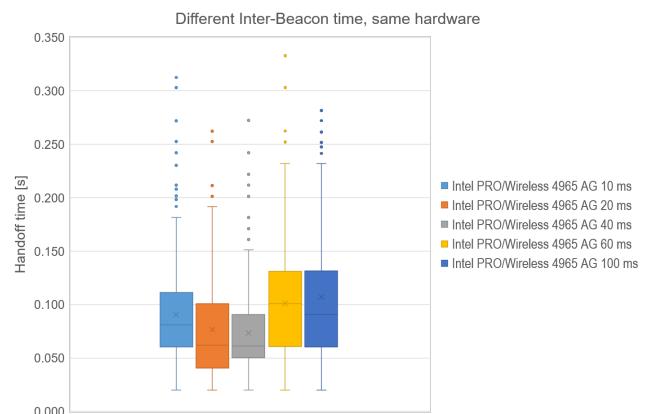
As illustrated in Figure 15, the setup includes a traffic generator using D-ITG [27], connected to the STA with an Ethernet cable. The STA receives this traffic by the Ethernet card, and forwards it to the AP via the 802.11 interface. This allows us to include a sniffer which can capture the traffic in that cable, in addition to that in the control and data planes. Through this setup, we can get a synchronized and combined packet capture, which enables fine grain delay calculations.

A flow of small UDP packets (80 bytes of payload) with an inter-packet time of 10 ms is sent from the generator to the server. After obtaining a number of handoffs, a Perl script obtains the delays and marks the lost packets. Finally, a Python script calculates the delay, as the time between the arrival of the packets before and after the handoff.

2) MODIFICATION OF INTER-BEACON TIME

First, the effect of the modification of inter-beacon time is studied. This parameter is not fixed by the 802.11 specification, so it can be tuned. In our case, a tradeoff appears: on the one hand, if the inter-beacon time is too low, it will result in a low efficiency, as a big amount of air time will just be occupied by the beacons. In our case, the use of LVAPs makes it even more difficult, as broadcast beacons cannot be used (each STA receives beacons from a different virtual MAC address).

In Figure 16, we represent a box and whisker plot obtained after 1,779 handoffs (355 on average per inter-beacon time value) using Intel PRO/Wireless 4965 wireless card. Although some values are higher, the average is in some cases 75 ms, and the vast majority of the handoffs take less

**FIGURE 15.** Test setup for measuring the handoff delay.**FIGURE 16.** Box plot of the handoff delay, using the same hardware (Intel PRO/Wireless 4965) and different values of inter-beacon time.

than 150 ms. Some outlier results can also be observed: as the measurements take place in a noisy environment with more than 15 APs, in some cases the amount of lost packets can be higher.

Curiously, the lowest value of inter-beacon time (10 ms) does not draw the best results. The reason for this can be that the device is receiving too many beacons, so it becomes too busy. A value of 60 ms can be recommended, but also 100 ms (the default value in many commercial APs) can be sufficient in order to provide fast handoffs.

3) EFFECT OF DIFFERENT WIRELESS CARDS

In this subsection, the behavior of three different wireless interfaces is compared. Figure 17 shows the box plot obtained

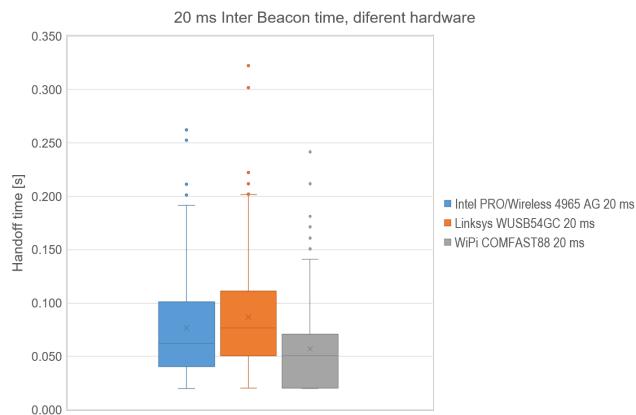


FIGURE 17. Box plot of the handoff delay, using a constant inter-beacon time of 20 ms and different hardware.

with three different low cost wireless devices (\$10 to 15), namely Intel PRO/Wireless 4965, Linksys WUSB54GC and WiPi COMFAST88. A total number of 893 handoffs have been performed (average 297 per wireless interface). It can be observed that the three devices have good behavior, *i.e.*, the average handoff time is always below 200 ms and the median is between 50 and 75 ms.

Finally, Figure 18 shows the results obtained with the same hardware, but using an inter-beacon time of 100 ms. It can be observed that in this case, the WiFi behaves poorly, *i.e.*, although the median and the average are low, some of the handoffs may require up to 500 ms.

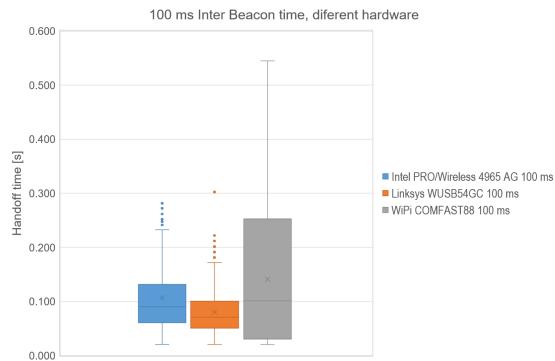


FIGURE 18. Box plot of the handoff delay, using a constant inter-beacon time of 100 ms and different hardware.

All in all, it can be seen that the overall delays are not significant, even for these low-cost wireless cards, and even using an inter-beacon time of 100 ms.

VI. CONCLUSIONS

This article has presented a novel SDWN architecture that aims to integrate a set of mechanisms that enhance the capabilities of Wi-Fi APs, by means of a better coordination. The global architecture has been presented in detail, where the handoff mechanism is integrated with a set of active and passive monitoring tools and other functionalities, resulting

in a solution that is able to provide smart functionalities using low-cost commercial APs.

The framework includes a central controller which has all the information available, and is therefore able to make smart decisions about the assignment of clients to APs. This avoids the problem of the “sticky client,” which remains connected to the original AP it associated with, rather than moving to a nearby one that would be a better choice for it.

Two different test scenarios have been used to compare proactive and reactive handoff mechanisms in realistic conditions, and the advantages of the proactive approach have been highlighted. The inter-channel handoff delay has been measured with three different devices, using five values for the inter-beacon time, proving that fast and seamless handoffs are possible in the scenario, even with low cost off-the-self equipment.

As future work, the improvement of the scalability of the system is being considered, by means of a hierarchy of controllers: a first layer of controllers close to the APs would be in charge of managing the handoffs with low delay, and other layers could perform coordination between a number of controllers. Another possibility is the integration with other technologies as *e.g.*, mobile networks, as required by 5G, as done *e.g.*, in EmPOWER. Finally, better management of the QoS could be achieved by monitoring the services being managed by the APs, and tuning the priorities accordingly.

REFERENCES

- [1] K. Schneider, D. Turgut, and M. Chatterjee, “An experimental study on Layer 2 roaming for 802.11 based WLANs,” in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Espoo, Finland, Jun. 2007, pp. 1–6.
- [2] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, “Performance anomaly of 802.11 b,” in *Proc. INFOCOM*, Mar. 2003, pp. 836–843.
- [3] A. Mishra, M. Shin, and W. Arbaugh, “An empirical analysis of the IEEE 802.11 MAC layer handoff process,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 93–102, Apr. 2003.
- [4] R. Riggio, T. Rasheed, and M. K. Marina, “Poster: Programming software-defined wireless networks,” in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 413–416.
- [5] R. Riggio, T. M. Rasheed, and R. Narayanan, “Virtual network functions orchestration in enterprise WLANs,” in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1220–1225.
- [6] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann, “OpenSDWN: Programmatic control over home and enterprise WiFi,” in *Proc. 1st ACM SIGCOMM Symp. Softw. Defined Netw. Res.*, Jun. 2015, p. 16.
- [7] J. Vestin, P. Dely, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, “CloudMAC: Towards software defined WLANs,” *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 16, no. 4, pp. 42–45, Feb. 2013.
- [8] J. Schulz-Zander, P. L. Suresh, N. Sarrar, A. Feldmann, T. Hühn, and R. Merz, “Programmatic orchestration of WiFi networks,” in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 347–358.
- [9] L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, and J. Almodovar, “Building an SDN enterprise WLAN based on virtual APs,” *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 374–377, Feb. 2017.
- [10] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, *Software-Defined Networking (SDN): Layers and Architecture Terminology*, document RFC 7426, 2015.
- [11] Open Networking Foundation. (Jun. 2014). *SDN Architecture Version 1.0*. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TRSDNARCH_1.0_06062014.pdf

- [12] J. Schultz, R. Szczepanski, K. Haensge, M. Maruschke, N. Bayer, and H. Einsiedler, "OpenGUFU: An extensible graphical user flow interface for an SDN-enabled wireless testbed," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Auton. Secure Comput., Pervasive Intell. Comput.*, Liverpool, U.K., Oct. 2015, pp. 770–776.
- [13] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, "Programming abstractions for software-defined wireless networks," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 2, pp. 146–162, Jun. 2015.
- [14] E. Coronado, R. Riggio, J. Villalón, and A. Garrido, "Joint mobility management and multicast rate adaptation in software-defined enterprise WLANs," *IEEE Trans. Netw. Service Manage.*, to be published. [Online]. Available: <https://ieeexplore.ieee.org/document/8269376/>, doi: 10.1109/TNSM.2018.2798296.
- [15] K. Nakuchi and Y. Shoji, "WiFi network virtualization to control the connectivity of a target service," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 2, pp. 308–319, Jun. 2015.
- [16] Y. Grunenberger and F. Rousseau, "Virtual access points for transparent mobility in wireless LANs," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2010, pp. 1–6.
- [17] M. E. Berezin, F. Rousseau, and A. Duda, "Multichannel virtual access points for seamless handoffs in IEEE 802.11 wireless networks," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, May 2011, pp. 1–5.
- [18] H. Hwang and Y.-T. Kim, "Enhanced fast BSS transition on enterprise WLAN with SDN-based distribution system," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, Tokyo, Japan, Nov. 2017, pp. 1–5.
- [19] J.-P. Jeong, Y. D. Park, and Y.-J. Suh, "An efficient channel scanning scheme with dual-interfaces for seamless handoff in IEEE 802.11 WLANs," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 169–172, Jan. 2018.
- [20] N. Montavont, A. Blanc, R. Navas, T. Kerdoncuff, and G. Castignani, "Handover triggering in IEEE 802.11 networks," in *Proc. IEEE 16th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Boston, MA, USA, Jun. 2015, pp. 1–9.
- [21] C. Chen, Y.-T. Lin, L.-H. Yen, M.-C. Chan, and C.-C. Tseng, "Mobility management for low-latency handover in SDN-based enterprise networks," in *Proc. IEEE WCNC*, Doha, Qatar, Apr. 2016, pp. 1–6.
- [22] C. Xu, W. Jin, G. Zhao, H. Tianfield, S. Yu, and Y. Qu, "A novel multipath-transmission supported software defined wireless network architecture," *IEEE Access*, vol. 5, pp. 2111–2125, 2017.
- [23] Anyfi Networks. (Apr. 2015). *Software-Defined Wireless Networking: Concepts, Principles and Motivations*. [Online]. Available: http://www.anyfinetworks.com/get_resource/anyfi-sdwn-concepts-whitepaper
- [24] V. Mhatre and K. Papagiannaki, "Using smart triggers for improved user performance in 802.11 wireless networks," in *Proc. 4th Int. Conf. Mobile Syst., Appl. Services*, Jun. 2006, pp. 246–259.
- [25] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, 2000.
- [26] M. Seyedebrahimi, F. Bouhafs, A. Raschellà, M. Mackay, and Q. Shi, "SDN-based channel assignment algorithm for interference management in dense Wi-Fi networks," presented at the 25th Eur. Conf. Netw. Commun. (EuCNC), Athens, Greece, Jun. 2016, pp. 128–132.
- [27] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Comput. Netw.*, vol. 56, no. 15, pp. 3531–3547, 2012.



JOSE SALDANA (M'11) was born in San Sebastián, Spain, in 1974. He received the B.S. and M.S. degrees in telecommunications engineering in 1998 and 2008, respectively, and the Ph.D. degree in information technologies from the University of Zaragoza in 2011. He is currently a Senior Post-Doctoral Researcher with the Department of Engineering and Communications. His research interests focus on quality of service in real-time multimedia services, as VoIP and networked online games, traffic optimization and resource management in Wireless LANs.

Dr. Saldana is a member of the Internet Society. He currently serves as an Editor for the IEEE ACCESS, and as an Area Editor for the *KSII Transactions on Internet and Information Services*. For several years, he served on the Organization Committee of the IEEE Consumer Communications and Networking Conference, and in the Technical Program Committee of many other conferences such as IEEE ICC and IEEE GLOBECOM.



RUBÉN MUNILLA was born in Logroño, La Rioja, Spain, in 1983. He received the degree in telecommunications technology and services engineering from the University of Zaragoza (UZ), Spain, in 2017.

He was a Support Technician with the Computer and Telecommunication Service, University of Zaragoza, from 2011 to 2014. From 2014 to 2016, he was a Software Developer and Support Personnel with the Higher Engineering and Architecture School, UZ (EINA). In 2017, he was with the IT Department, Araelec, where he was focused in telecommunications infrastructure. He is currently a Research Engineer with the Department of Electronics Engineering and Communications, EINA, where he was involved in wireless networks, SDN, software development, and deploying test scenarios.



SALIM ERYIGIT received the B.S. degree in industrial engineering and the M.S. and Ph.D. degrees in computer engineering from Boğaziçi University, Istanbul, Turkey, in 2003, 2008, and 2014, respectively.

He is currently a Senior Research and Development Engineer with AirTies Wireless Networks, Istanbul. His current research interests include network optimization, wireless networks, scheduling, green communications, and device-to-device communications.



OMER TOPAL received the B.S. degree in electronic and communication engineering from the Izmir Institute of Technology, Izmir, Turkey, in 2016. He is currently pursuing the M.Sc. degree with the Department of Telecommunication Engineering, Istanbul Technical University, Istanbul, Turkey.

He is currently a Research and Development Engineer with AirTies Wireless Networks, Istanbul. His current research interests include network optimization, wireless networks and simulation modeling of wireless networks.



JOSÉ RUIZ-MAS was born in Lorca, Murcia, Spain, in 1965. He received the Engineering degree in telecommunications from the Universitat Politècnica de Catalunya, Spain, in 1991, and the Ph.D. degree in telecommunications engineering from the University of Zaragoza (UZ) in 2001.

He was a Software Engineer with TAO Open Systems from 1992 to 1994. From 1994 to 2003, he joined the EINA as an Assistant Professor and then he became an Associate Professor. He was the Director of the Telefonica Chair, University of Zaragoza, from 2004 to 2008 and Coordinator of Master in information technology and mobile communications from 2007 to 2009. He is with the Department of Electronics Engineering and Communications, Higher Engineering and Architecture School, UZ. He is a member of the Aragón Institute of Engineering Research (I3A). He has been a co-investigator since 1995 of research grants from EU Research Projects, the Ministry of Science and Technology, the Sanitary Research Funds, and the Government of Aragon (Spain) in the areas of distributed multimedia system and wireless networks. Major industrial and mobile companies in the area of wireless communications also support his work. He is a co-investigator of several research projects supported by companies as Telefónica and Teltronic. His current research activity lies in the area of communication networks with special emphasis on wireless networks, distributed multimedia system, quality of service, and quality of experience.



JULIÁN FERNÁNDEZ-NAVAJAS was born in Alfaro, La Rioja, Spain, in 1969. He received the Engineering degree in telecommunications from the Universidad Politécnica de Valencia, Spain, in 1993, and the Ph.D. degree telecommunications engineering from the University of Zaragoza (UZ) in 2000.

From 1994 to 2002, he joined the EINA as an Assistant Professor and then he became an Associate Professor. He is with the Department of Electronics Engineering and Communications, Higher Engineering and Architecture School, UZ. He is a member of the Aragón Institute of Engineering Research (I3A). He has been a co-investigator since 1995 of research grants from EU Research Projects, the Ministry of Science and Technology, the Sanitary Research Funds, and the Government of Aragón (Spain) in the areas of distributed multimedia system and wireless networks. Major industrial and mobile companies in the area of wireless communications also support his work. He is a co-investigator of several research projects supported by companies as Telefónica. His current research activity lies in the area of communication networks with special emphasis on wireless networks, distributed multimedia system, quality of service, and quality of experience.



LUIS SEQUEIRA received the M.S. degree in digital communications from the University of Costa Rica in 2008, and the M.S. degree in mobile networks and the Ph.D. degree (*cum laude*) in information technologies from the University of Zaragoza in 2012 and 2015, respectively.

His research interests focus on quality of service in multimedia systems, analysis and modeling of access networks and multimedia services. He was also involved in Wi5 project on a SDN-based Wi-Fi architecture developing a coordination platform between access points in order to improve quality of service, reduce radio interference, and optimize Wi-Fi spectrum usage.

He is currently a Research Associate with King's College London, where he is involving in software-defined networking and orchestration of network services.

• • •