LEAF: Improving Handoff Flexibility of IEEE 802.11 Networks With an SDN-Based Virtual Access Point Framework

Juan Lucas Vieira, Daniel Mosse, and Diego Passos, *Member, IEEE*

***Abstract*—Mobile devices' popularization has brought several new applications to communication networks. As we move into an increasingly denser scenario, problems such as collisions between transmissions and unbalanced load become more pronounced. Moreover, while station-based handoff is inefficient to reduce these issues, network-wide handover decisions might provide better network resource management. This paper proposes LEAF, an access point virtualization solution based on Software Defined Networking to enable station (STA) handover conducted by the network, based on a global scope. Unlike other solutions in the literature, our proposal fully supports multichannel migrations through the IEEE 802.11h Channel Switch Announcement without restricting the channel utilization by the access points. To demonstrate the feasibility of such an approach, we present experimental data regarding the behavior of several different devices in face of this mechanism. We also evaluate our complete virtualization solution, which reveals that the handoff of STAs did not lead to significant packet losses or delays in STAs' connections, while providing a foundation to improve network's self-management and flexibility, allowing association control and load balancing tasks to be executed on top of our solution.**

***Index Terms*—Channel switch announcement, software defined networking, virtual access points, wireless networks.**

I. Introduction

In TRADITIONAL IEEE 802.11 networks, the station (STA) is responsible for the handoff decision [1]. The handoff is the process in which an STA disassociates from the current Access Point (AP) and associates with a new AP to improve its connection to the network. The handoff decision is usually limited to the analysis of the signal generated by nearby APs as received by the STA radio [1].

This migration is a costly process, as it requires the STA to perform a scan of the wireless medium, followed by the association with a new AP. The time required to execute these steps can lead to the interruption of communication for several seconds. In addition, the variability of the signal perceived by the radio of an STA can lead to sequential reassociations [1], which might severely jeopardize applications that require the timely delivery of packets. Furthermore, characteristics such as the interference between transmitting devices or the current load of the AP are not known by the STA when selecting an AP. These information could be used to improve AP selection decisions aiming at improved resource utilization and interference reduction.

To overcome the mentioned challenges, the main goals in the related literature involve making better AP selection decisions and reducing handoff overhead. Some authors propose changing STAs [2] behavior, which is not desirable as this approach results in interoperability issues with current devices. Another research strand focuses on transferring the handover responsibility to the Wireless Local Area Network (WLAN), similarly to how migrations are handled in cellular networks [3]. In that sense, the AP virtualization concept has been used by several works [4], [5], [6], [7] to allow STAs to be transferred between physical APs when convenient.

Some works also combine the AP virtualization and the SDN paradigm to allow better management of the network, where a central controller connected to the physical APs handles STA handover. In comparison to the STA-based choices, network-driven handoffs enable better resource management and new applications. For example, if AP utilization data is accessible by the control entity, load-balancing can be performed by migrating STAs to distribute network load among the APs. Likewise, if the goal is to reduce the energy consumption of the network, STAs can be migrated to cluster them in fewer APs, allowing the shutdown of unused APs.

However, these solutions often impose channel utilization restrictions to the physical APs [6], [7], [8], [9] — either requiring neighbor APs to operate in different channels or all APs to share the same channel, which can aggravate interference. Other solutions are only able to make handover decisions based on limited-scope information (*e.g.*, signal strength experienced by an AP) exchanged between a few close APs [5], which restricts network-wide optimizations.

In this paper, we extend our previous work [10] on the proposal of a standard-compliant Lightweight AP Virtualization Framework (LEAF) for IEEE 802.11 networks that utilizes the Channel Switch Announcement (CSA) [11] to enable

Manuscript received 25 October 2023; revised 3 May 2024 and 24 July 2024; accepted 3 August 2024. Date of publication 9 August 2024; date of current version 20 December 2024. This study was financed in part by CAPES – Brazil – Finance Code 001, CNPq and FAPERJ. The associate editor coordinating the review of this article and approving it for publication was C. Assi. *(Corresponding author: Juan Lucas Vieira.)*

Juan Lucas Vieira is with the Instituto de Computação, Universidade Federal Fluminense, Niterói 24210346, Brazil (e-mail: juanvieira@midiacom.uff.br).

Daniel Mosse is with the Computer Science Department, University of Pittsburgh, Pittsburgh, PA 15260 USA.

Diego Passos is with the Department of Electronical Engineering, Telecommunications and Computers, Instituto Superior de Engenharia de Lisboa, 1959-007 Lisbon, Portugal, and also with Instituto Politécnico de Lisboa, 1549-020 Lisbon, Portugal.

Digital Object Identifier 10.1109/TNSM.2024.3441390

the handover of STAs between APs operating on different channels. Our proposed solution also includes a controller for the centralized decision of STA handoff, borrowing from the Software Defined Networking (SDN) concept [12].

As contributions of this work, we highlight the demonstration, through multiple experiments using real hardware, that the proposed solution is, in fact, valid and that it has a significantly faster handover time compared to a deauthentication-based scheme. In addition, we provide an analysis of CSA efficiency and the behavior of mobile devices upon receiving a beacon with the CSA. We also provide an implementation[1] of LEAF, composed by an AP-side agent and the control entity.

In comparison to our previous work, this extended version includes an updated version of the proposed protocol that checks for STA connectivity and that broadcasts its new location after migration. We also added several new experiments regarding the impact of our solution in terms of packet loss, outage time, and throughput to extensively evaluate our solution. Also, we added a discussion section to highlight scenarios that might benefit from our solution and the encountered challenges. We also expanded the related work section with additional works in the literature.

The remainder of this document is organized as follows: Section II reviews concepts related to our proposal. Section III presents related work. Section IV describes our proposed solution. Section V covers the evaluation of the proposal. Section VI discusses the applications, limitations, and future work. Section VII presents the conclusion.

II. Background

The IEEE 802.11h amendment [11] was created to avoid co-channel operation between radars and WLANs operating at the 5 GHz band, as well as to satisfy other regulatory requirements. This extension provides several spectrum management services, such as Transmit Power Control (TPC) and Dynamic Frequency Selection (DFS) that permit, for example, the specification of a maximum transmission power for a channel or to discontinue operation in a specific channel after a radar using the same frequency is detected.

In our proposal, we use the CSA mechanism, part of the DFS service, whose primary purpose is to assist channel switching after a radar detection [11]. In an infrastructure Basic Service Set (BSS), the CSA allows warning associated STAs that the AP's channel will change. From the moment the AP first announces a channel switch until the actual frequency change, the AP may also block transmissions from STAs.

The CSA Information Element (IE) is the structure used to communicate the channel switch. This element can be added to beacons, probe responses or action frames transmitted by APs. STAs operating in managed mode, however, shall not transmit the CSA IE. As can be seen in Figure 1, the format of the CSA IE consists of five pieces of information:

- the *Element ID* field identifies the type of the IE, which, in the case of a CSA, is represented by the value 37;
- the *Length* field specifies the number of following octets that are part of the IE, which is 3 octets in this example;

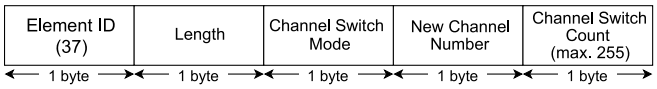


Fig. 1. Format of a CSA IE in an IEEE 802.11 beacon (adapted from [11]).

- the *Channel Switch Mode* field indicates whether the associated STAs' can continue to transmit data during the period of channel transition announcement;
- the *New Channel Number* field indicates the number of the new channel to which the radio will transfer; and
- the *Channel Switch Count* field indicates the number of Target Beacon Transmission Times (TBTTs) until the channel change is performed.

If an AP performs a channel switch operation without using the CSA mechanism, associated STAs would need first to detect the unavailability of the AP, re-scan the wireless medium to find out which channel the AP in question has switched to and later perform the reassociation and reauthentication processes. This can cause considerable delay [1], jeopardizing applications that demand timely transmission of packets (*e.g.*, streaming, VoIP). The use of the CSA can prevent STAs from repeating this reassociation procedure with the AP, reducing the delay of this operation significantly. Besides, the inclusion of the CSA IE in beacon frames enables handoffs between APs operating on separate channels, as explained in Section IV.

III. Related Work

In this section, we present works that propose virtualization solutions or network-driven handoff to circumvent shortcomings of traditional STA-based handoff. Table I presents an overview of the presented proposals.

A. Station-Side Virtualization

In [2] the authors propose a virtualization approach on STAs to enable them to be connected to multiple APs at the same time. Despite the performance gains reported in their work, STAs must be modified to comply with the proposal. Therefore, many current Wi-Fi devices would not benefit from the throughput gains, since they would not be able to connect to multiple APs simultaneously. Thus, station-side or protocol modifications are not feasible or desirable in many scenarios.

B. Network-Only Modification

Another strand of research focuses on modifying the behavior of WLANs without requiring changes to the STAs. Murty et al. [13] propose a solution to enhance the performance in dense scenarios. In their solution, APs hide their Service Set Identifiers (SSIDs), requiring STAs to send probe requests during their active scanning phase. Each probe request received by the APs is sent to a controller, which selects the AP that will reply to the probe and provide network connection to the STA based on free air time and expected transmission rate estimates. The controller can also handoff STAs by making the source AP send a disassociation frame to the STA and the destination AP to reply to the upcoming

[1] Available at: https://github.com/juanlucasvieira/LEAF.

TABLE I
COMPARISON BETWEEN RELATED WORK AND THE PROPOSED SOLUTION

Ref.	Description	SC	NH	VAP	CB	MS	Main Limitation
[2]	Minimizes STA handoff overhead by virtualizing the STA WNIC, allowing it to be connected to multiple APs.	Yes	Yes	No	Yes	No	Requires station-side modifications
[13]	Increases the performance of dense WLANs by handing over stations and estimating free air time and transmission rate.	No	Yes	No	Yes	No	No multichannel support
[6]	Proposes an architecture to enable seamless RSSI-based station mobility through inter-AP control message exchange.	No	Yes	Yes	No	No	No multichannel support
[4]	Proposes a framework to ease the development and employment of network services by providing link abstractions at the edge.	No	Yes	Yes	Yes	No	No multichannel support
[14]	Provides an architecture that extends the SDN paradigm to improve QoS, station mobility and security in dense WLANs.	No	Yes	Yes	Yes	No	No multichannel support
[8]	Improves the mobility and flexibility of Wi-Fi networks by deploying VAPs and virtual SDN controllers on-the-fly.	No	No	Yes	Yes	No	No multichannel support
[15]	Proposes a scheme to enable handover between APs based on RSSI and AP load metrics, utilizing SDN and NVF technologies.	No	Yes	Yes	Yes	No	No multichannel support
[7]	Proposes a SDN-based proactive handover solution based on multiple network metrics, such as current load, RSSI and STA mobility.	No	Yes	Yes	Yes	No	No multichannel support
[16]	Proposes a handover framework that monitors multiple metrics and a machine learning algorithm utilizing these metrics.	No	Yes	Yes	Yes	No	No multichannel support
[5]	Proposes the Multichannel Virtual Access Point concept to enable the handover of stations between APs operating in different frequencies.	No	Yes	Yes	No	Yes	No central controller
[17]	Proposes an architecture in which MAC data processing and management frame creation are held by VAPs deployed in the cloud.	No	Yes	Yes	Yes	Yes[a]	Higher delays for MAC frame processing
[9]	Aims at improving mobility and QoE by proposing an architecture in which a single BSS is replicated throughout all APs.	No	Yes	Yes[b]	Yes	Yes[c]	Nearby APs must operate in distinct channels
[18]	Proposes an architecture that enables programmability and seamless handover using VAPs to improve Wi-Fi network flexibility.	No	Yes	Yes	Yes	Yes[d]	Limited multichannel support when migrating VAPs.
[19]	Proposes an architecture that integrates coordination mechanisms to enhance the capabilities of a set of central managed Wi-Fi APs.	No	Yes	Yes	Yes	Yes	Reactive handoff delay increases with the number of STAs.
Our Work	Enables seamless multichannel handoff initiated by the network, and allows applications to define their own handover decisions.	No	Yes	Yes	Yes	Yes	Limited number of VAPs held by one physical AP

Legend - *SC*: Station Changes required, *NH*: Network-initiated Handoff, *VAP*: Virtualized AP, *CB*: Controller based, *MS*: Multichannel Support.
[a] Only cites CSA to enable station handover between APs operating in different channels. [b] Here we considered the single BSS to be a single a VAP, since the BSS information is replicated in all physical APs. [c] Multichannel support is restricted because nearby APs can not share channels.
[d] Work seems to have limited multichannel support when migrating VAPs.

probe request. Despite the reported throughput gains, the STAs undergo a disruption time of around 1.5 seconds during the handoff for load balancing purposes, since the STA still needs to scan the medium and re-associate with the target AP.

C. Access Point Virtualization

Several works utilize the concept of AP virtualization to bring improvements for WLANs. Authors in [6] propose a solution to permit seamless mobility of STAs. In their proposal, each STA has its own Virtual Access Point (VAP), physical APs can host multiple VAPs, and when an STA moves, its VAP should be moved together to a nearby AP, in a way that the association between the station and the VAP is maintained. When the Received Signal Strength Indicator (RSSI) of the STA on the current AP is weaker than the signal received by another AP in the network, control messages are exchanged between APs to transfer the VAP information. Despite not requiring modifications to the STAs, the solution requires that every AP operates in the same channel, which could aggravate interference problems in dense scenarios.

Suresh et al. [4] propose Odin, an SDN framework for enterprise WLANs that utilizes AP virtualization to simplify the development of applications and abstract changes in the link between an STA and the network. The authors highlight that their framework supports seamless mobility, since the infrastructure is able to handoff STAs without requiring re-associations. Similarly to other above-mentioned proposals, Odin utilizes RSSI as a metric to start the handoff procedure and does not require any STA-sided modifications.

Ethanol [14] extends the SDN paradigm to enable global control of Quality of Service (QoS), STA mobility, and security of dense WLANs. Their controller manages APs, VAPs and current link state, and supports the control OpenFlow switches, while an agent, executed on modified commodity wireless routers, allows wireless link control and QoS definition in the wired ports. Although their control model involves AP virtualization, the work does not detail the virtualization process or if their VAPs are migrated between physical APs. Also, the authors highlight that only part of the functionalities was implemented due to time restrictions and hardware and software constraints. This absence of further exploration in their prototype might obscure some limitations of their work.

To improve mobility and ergonomics of Wi-Fi networks, Stiti et al. [8] have developed a solution that allows STAs to deploy VAPs and virtual SDN controllers on the fly. The authors highlight that, even though multiple VAPs share the same physical AP, their solution provides complete isolation between instances, allowing them to operate independently in the same Wireless Network Interface Card (WNIC). Despite the claimed improvements in scalability, security and flexibility, they do not benchmark the proposed architecture to evaluate the real benefits of the solution in handover scenarios.

Gilani et al. [15] propose an handover scheme based on SDN and Network Function Virtualization (NFV) technologies. In their work, association decisions are made based on AP load and RSSI metrics and the support for seamless approach is achieved by having adjacent APs to share the same VAP and the same channel. However, this approach can lead to serious interference problems, since the same frequency will be utilized by the neighboring APs. Also, having the same VAP in more than one AP can lead to duplicated frames if the STA is in the connectivity are of more than one AP.

Zeljkovic et al. [7] propose an SDN-based solution that uses VAPs to allow the handover of STAs. Rather than wait for the deterioration of the network, their solution estimates the performance of the nodes and hands stations off proactively. The handover decision is based on several metrics, such as RSSI, traffic load, AP capacity and STA mobility. Their evaluation shows a reduction in the number of handovers and higher average throughput in comparison to a reactive algorithm that triggers the handover when the RSSI drops below a threshold and a handover algorithm that keeps an STA associated with the AP with the highest RSSI. A similar handover strategy is used in their following work [16], where they also propose a machine learning approach that uses RSSI, AP load, distance and association information to proactively decide when to migrate an STA.

In dense scenarios, better performance can be achieved when nearby APs operate in orthogonal channels, due to the reduction of collisions caused by simultaneous transmissions in overlapping frequencies. However, none of the aforementioned works address the use of APs operating on separate channels, which is a limiting factor in such scenarios: either all APs would have to operate in the same channel, aggravating the problems of the shared medium, or the handover of STAs would have to be restricted to a subset of target physical APs.

Po-Fi [18] is an architecture to provide flexibility and programmability to WLANs, which is achieved by using a protocol-independent technology in which packets are handled according to match-instruction forwarding rules. The VAP concept is used to enable novel services, such as seamless mobility. In this use case, physical APs forward association frames to a central controller that decides which AP should host the VAP. Seamless handover is achieved based on measuring the signal strength of the current and the candidate APs. As the STA moves far from its current AP and closer to the candidate AP, events are sent to the controller to make the VAP migration decision. Similarly to our work, Po-Fi also follows a per-station VAP, which is migrated when an STA must be handed over to another AP. However, the authors do not address if seamless mobility of STAs can be achieved between APs operating in different channels in their proposal.

D. Multichannel Support

Berezin et al. [5] seek to solve the above-mentioned channel limitations by introducing Multichannel VAPs, which uses the CSA mechanism and inter-AP message exchange. In their work, STA migration occurs when an AP detects that the STA's signal is below a threshold. Then, the AP sends scan requests to neighbor APs, which change their channels temporarily to the STA's channel and reply to the requests with the perceived RSSI of the STA. Afterward, the current AP migrates the VAP to the physical AP with the highest reported signal and makes the STA change its frequency to the channel of the new AP. Despite allowing STA handover between APs operating in different channels, the lack of centralized control and global view of the network limits the scope of the handover decisions, which are solely based on the RSSI.

CloudMAC [17] is an OpenFlow based architecture in which VAPs are deployed in a cloud environment. In the solution, physical APs only forward MAC frames, whereas tasks such as management frame creation and MAC data processing are held by VAPs in a virtual machine. The authors also briefly mention the possibility of using CSA for the AP exchange process, without further exploring the utilization of this mechanism in their solution.

Our work shares some similarities to BigAP [9], like using the CSA IE to enable multichannel STA handover and employing a controller to coordinate the migration process. However, that work considers the utilization of a single BSS throughout the APs of the whole network, requiring nearby APs to operate on separate channels to avoid problems inherent to the BSS replication (*e.g.*, duplicates due to two APs receiving a frame from the STA). This requirement might lead to performance degradation if other unmanaged nearby networks heavily utilize channels that must be used by one of the managed APs to cover a certain area. BigAP also requires APs with two WNICs, one to serve as an AP and another to monitor the wireless activity of nearby devices, which increases the cost of deployment of their solution and can be a limiting factor for low-budget networks scenarios.

Two WNICs are also required in the SDN framework proposed by Saldana et al. [19] to enable channel selection, association control, and load-balancing functionalities. One interface is responsible for exchanging data with STAs and monitoring in-channel statistics, while the other is used for off-channel monitoring. The work also supports two seamless handoff approaches. The reactive approach triggers an event on the controller when a number of RSSI samples of the STA is below a threshold, with a hysteresis time to avoid ping-pong. Then, the controller requests APs to listen to packets of the STA, which reply with the scan results to the controller, that decides which AP should host the VAP. In the proactive approach, the controller periodically asks APs to scan for STAs. Using the RSSI information reported by the APs, an algorithm finds better-suited APs for the STAs. Despite not requiring any modification to STAs and supporting multichannel handoff, the fact that the work requires each AP to have two WNICs increases deployment costs.

Despite the many efforts to reduce the handover overhead and increase the performance of WLANs through the virtualization of network components, there is still need for a lightweight solution that is able to operate in commodity hardware and fully support multichannel handovers, without significant restrictions in frequency utilization. Also, delivering global view and control improves network flexibility,

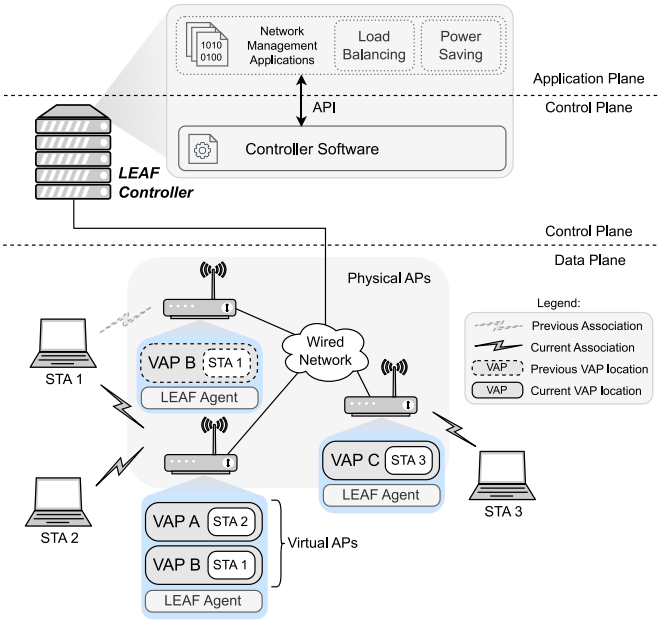


Fig. 2. Overview of the LEAF architecture. Each physical AP executes the LEAF agent and can host multiple VAPs. The controller manages the physical and virtual APs and exposes an API to network management applications.

allowing the employment of different handover algorithms that rely on various network state information.

IV. A Lightweight Virtualization Framework

Our work proposes a solution that allows the STA handover to be initiated by the network infrastructure, while being totally transparent from the STA's point of view. For this, we rely on the AP virtualization and SDNs paradigms. Besides, by using the CSA as an enabling mechanism, our solution supports handover between APs operating on different channels. This section provides architectural aspects of the proposal and implementation details of the developed prototype.

A. Architecture

As shown in Figure 2, the architecture of LEAF consists of three main components: (i) VAPs that encapsulate association information of their corresponding STA; (ii) physical APs running a LEAF agent; and (iii) the LEAF controller.

The concept of AP virtualization can be found in the literature to represent different ideas for an entity that performs tasks or assumes the role of an AP. In our proposal, a *VAP* consists of a data structure that encapsulates the information of a BSS and its associated STA, allowing the portability and transfer of this information between different physical APs. Different from traditional IEEE 802.11 networks, in our proposal the BSS contained in a VAP accepts the association of at most one STA. Therefore, each connected STA has their own VAP. This allows VAP migrations to be performed without interfering with other STAs connected to the network. Each VAP has a specific identifier that is used as the MAC address — instead of the real MAC of the AP interface — and the BSSID, both advertised in its BSS frames.

A *physical AP* is an infrastructure device that hosts one or more VAPs, connecting multiple STAs wirelessly to the network, and that executes an agent software that handles commands received by the controller. The *LEAF controller* is a central entity — connected to the physical APs through a wired network — capable of requesting state and usage information of the APs and their associated STAs. It is responsible for orchestrating the process of VAP migration by exchanging control messages with physical APs and can also rollback VAP migrations in case of request failures or communication timeout. Additionally, the controller exposes an Application Programming Interface (API) that provides functions and AP-state information to external applications.

It is outside the scope of this work to decide when the migration process should be triggered (*i.e.*, the controller does not implement a handover decision algorithm). Instead, the main purpose of this proposal is to improve the flexibility of handoff decisions by establishing a data plane (physical APs) management interface that provides abstractions for applications executing on top of the controller at the control plane. This approach allows VAPs — and, by extension, STAs — to be migrated based on distinct decision algorithms, implemented by these applications using the functions provided by our framework. The objectives of these algorithms may include support for reducing energy consumption of the network or balancing load among physical APs and among channels.

B. VAP Handover

The VAP handover procedure begins when the controller receives a request through its API. The process consists of the exchange of control messages between the controller and the source and destination APs — which might be operating at the same or different channels — regarding the information of the target VAP and its associated STA.

As previously stated, the CSA is a key mechanism to allow inter-channel migration. During the handover process involving two APs operating in different channels, the source AP — *i.e.*, the one to which the STA is currently associated — must attach the CSA IE in the beacons of the VAP to be transferred. The *New Channel Number* is set to the channel of the destination AP. The goal of this approach is prompting the STA to change its radio frequency to match the channel of the new physical AP which will host the VAP after the migration is complete, maintaining the association.

The multichannel migration procedure is performed based on the steps described in Figure 3:

1) The controller requests from AP1 the status information regarding one of its VAPs and the STA connected to it;
2) The controller sends the received VAP information to AP2, which creates a VAP with the same configuration of the original VAP;
3) The controller sends an STA addition command to AP2 including the STA association information in the newly created VAP;
4) The controller sends a start CSA command to AP1, inducing it to announce a channel switch to the channel of AP2;

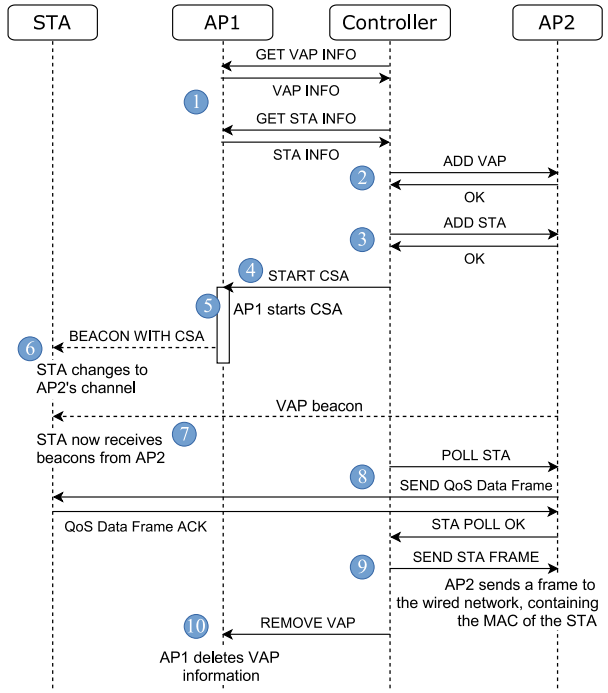


Fig. 3. Steps of the multichannel virtual access point migration procedure.

5) AP1 starts including the CSA IE in the VAP beacons;
6) The STA receives the VAP beacon with the CSA and switches to the channel of AP2;
7) The STA starts using AP2, without disassociating from its VAP;
8) The controller sends a *POLL STA* request to AP2, to verify that the STA is connected to the new AP. Then, AP2 sends a QoS data frame to the STA. Upon receiving an acknowledgment from the STA, AP2 reports the connectivity success to the controller.
9) AP2 sends a frame to the wired network containing the STA MAC address, broadcasts the new STA location.
10) AP1 deletes the VAP information, upon receiving a removal command from the controller.

Note that when both source and destination APs share the same channel, it is unnecessary to induce a channel change of the STA. Therefore, steps 4, 5, and 6 are not required in migrations within the same channel.

Regarding the network initialization, each AP starts with a vacant VAP to which an STA might associate. When an STA associates with this VAP, the AP instantiates another empty VAP to allow the association of other STAs. This process is repeated until the limit of VAPs that can be held by an AP is achieved (please refer to Section VI for more details).

In case that an STA decides to handoff to another physical AP (*e.g.*, due to mobility reasons or signal strength variability), LEAF might allow the STA to associate with another vacant VAP or might block the handoff by restricting association requests from STAs that were already associated.

As stated previously, we do not address when a migration should occur nor which physical AP would be appropriate to serve a given STA. We assume those tasks will be the responsibilities of other applications running on the controller. However, if the new physical AP cannot reach an STA after its VAP migration — which is checked using the STA poll procedure in step 8 — due to the STA mobility or poor link quality, a VAP migration rollback is executed. In fact, if an error is returned by the AP that will receive the VAP through steps 2 to 8, the rollback procedure is performed, which consists in the controller requesting the deletion of the VAP at the target AP, and skipping all the following steps of the migration process. Since the removal of the VAP at the previous physical AP is only performed if the whole process has been successful, the VAP information at the previous physical AP that the STA was associated with is maintained. Thus, the STA can continue to use the previous AP normally.

C. Implementation Details

We have developed an implementation to better assess the feasibility of our solution. In this section, we detail the implementation aspects of the developed software.

To maintain compatibility with existing mobile devices, we aimed at developing our solution without relying on changes to the STA. APs, in comparison, are more feasible to adapt since they are elements of network infrastructure. However, hardware modifications to these devices are also undesirable. Therefore, another goal was to deploy our solution on commercially available WNICs. To that end, our proposed solution retains full compatibility with the IEEE 802.11 standard.

To fulfill these goals, we decided to use the HostAPD 2.7 source code as a base for our AP implementation. HostAPD [20] is an open-source software that allows WNICs to be used as APs. Its code is highly portable, being supported by multiple platforms and embedded devices, such as OpenWRT-supported APs. In our proposed solution, the VAPs are implemented as multiple BSSs over the same WNIC, created and managed by HostAPD. However, the HostAPD, as is, does not provide all the functionality required by our solution.

To support VAP migration, changes were made to the base CSA functionality so that only beacons of the VAP to be migrated include the CSA element — instead of announcing the channel switch to STAs in other VAPs —, avoiding unwanted changes in other VAPs within the physical AP. We also modified HostAPD to allow arbitrary registration of STA data (*e.g.*, MAC address, supported rates, and other capabilities) by the controller at an AP — which would typically be acquired by the AP during the association procedure — so that no STA association is required after switching to a new physical AP. We also extended HostAPD's API to implement the necessary functionality of our solution (*e.g.*, STA's data registration) and allow the management of the VAPs by the controller.

For the control plane, we have developed a Java-based application to handle VAP migrations and manage registered APs and their STAs, through the exchange of synchronous and asynchronous messages. The controller provides a REST-based API that allows applications to send requests to the control plane and retrieve data such as the number of STAs currently

TABLE II
STA-RELATED DATA ACCESSIBLE FROM THE APPLICATION PLANE

Information	Description
RX Packets	Number of packets received by an STA.
TX Packets	Number of packets transmitted by an STA.
RX Bytes	Number of bytes received by an STA.
TX Bytes	Number of bytes transmitted by an STA.
Supported Rates	Data rates supported by the STA.
Inactive Time	Time for which the STA is inactive.
Signal	Signal strength between the STA and the AP.
RX Rate	Link bit rate used for data reception.
TX Rate	Link bit rate used for data transmission.
Connected Time	Time the STA is connected to the AP.

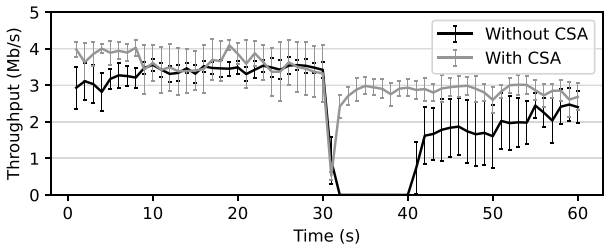


Fig. 4. How the throughput measured by an STA is affected by the AP switching channels with and without the CSA mechanism. The confidence interval is based on 10 executions with a confidence level of 95%.

being served by each AP of the data plane. Table II presents several STA-related data available to the application plane.

With these data, a load-balancing application can, for example, derive the total load of a physical AP by combining the data from each served STA. Based on that, it can eventually handover STAs to better distribute the network load.

V. EVALUATION

We start the evaluation with an analysis of the IEEE 802.11 CSA mechanism, since it is crucial for the multichannel support in our solution. Then, we evaluate our proposed solution through multiple experiments.

A. CSA Performance Assessment

To investigate the impact of the CSA mechanism, we held an experiment to compare the performance of a channel change by an AP with and without advertising it to the associated STAs. For this analysis, we created a simple topology in which a laptop running HostAPD acts as an AP. The laptop also runs a DHCP server to assign IP addresses to a Raspberry 3 Model B running Kali Linux 2018.4 with Nexmon acting as an STA. The *iperf* [21] tool was used to measure TCP throughput between the STA and the AP, which used the IEEE 802.11b mode. In this test, the AP performed a channel switch after 30 seconds. In the first scenario, the CSA mechanism was used with a *Channel Switch Count* of 5 TBTTs. In the second scenario, the AP simply changed its channel without warning the STA.

The obtained results are shown in Figure 4. In the scenario without CSA the STA had to detect the absence of the AP in the original channel, perform a new scan to find out the new channel and, finally, request a reassociation with the AP. All that process caused the network throughput to remain at zero for almost 8 seconds, followed by a slow recovery, as can be seen in Figure 4 (likely due to the congestion control algorithm of TCP severely decreasing its congestion window due to the burst of losses during the disconnection period). With CSA, even though the connection manifested a momentary decrease in throughput during the channel switch of the STA, the downtime was significantly lower, as well as its effect on TCP compared to the previous scenario.

This result emphasizes that the CSA mechanism considerably reduces the delay that an STA requires to find the AP in the new operating channel. Consequently, when applied to the handover of STAs between APs with different channels, the inclusion of CSA IE might also reduce the connection reestablishment delay after the STA handover process.

B. CSA Station Behavior

The IEEE 802.11 standard specifies several rules related to the transmission of management frames with the CSA IE. However, it does not state that an STA must necessarily change its frequency to the advertised channel upon receiving frames with the CSA IE. According to the standard, STAs might prefer to handoff to another BSS instead of following the AP to the new advertised channel.

In the context of the proposed solution, such flexibility in the reaction to a CSA could lead to a disassociation of the STA during the migration of a VAP, increasing the time of disconnection. Therefore, we held an analysis to investigate the behavior of mobile devices made by popular manufacturers. Our goal was to verify if these devices follow the AP in the channel switch when they receive the CSA IE and if they respect the transmission constraint imposed by the AP during the announcement period (when the Channel Switch Mode is set, as explained in Section II).

In this scenario, we used a laptop running Ubuntu 18.04.02 LTS, HostAPD, and a DHCP server. The *ping* command was utilized to generate traffic between the AP and the devices. The AP changed its channel between channels 1 and 6 of the 2.4 GHz band using the CSA mechanism. A higher *Channel Switch Count* number of 200 TBTTs was used to highlight the STA's blocking behavior — since, with a very small number, the transmission restraint might not be perceived — and to avoid the case in which the STA was not aware of the channel switch due to losses of beacons containing the CSA IE.

Table III presents the results obtained with several computers and laptops as STAs — using different hardware and software —, while Table IV displays results for smartphones and tablets of popular brands. In both cases, the *Reassoc.* column highlights if the STA had to reassociate with the AP — after, possibly, re-scanning the shared medium — once channel switching was triggered. The *Blocks TX* column presents which devices restrained their transmissions upon receiving a CSA IE with the *Channel Switch Mode* field equal to 1, measured by observing if any data frame was transmitted by the devices during the CSA procedure.

None of the analyzed devices preferred to associate with another BSS. However, they showed different behaviors

TABLE III
CSA RECEPTION BEHAVIOR OF DIFFERENT COMPUTERS AND LAPTOPS

Device	WNIC	OS / Driver	Reas-soc.	Blocks TX
Acer Aspire E1-471-6404	Qualcomm AR9485	Ubuntu 18.04.2 / ath9k	No	No
Dell Inspiron i14-5448-B30	Intel AC 7265	Ubuntu 18.04.2 / iwlwifi	No	Yes
		Windows 10 (1809) / netwtw04.sys	Yes	Yes
Dell Inspiron i15-7572-A30S	Qualcomm QCA6174	Ubuntu 18.04.1 / ath10k_pci	No	Yes
		Windows 10 (1803) / qcamain10x64.sys	Yes	No
Dell Inspiron 13-7359	Intel AC 3165	Ubuntu 18.04.02 / iwlwifi	No	Yes
		Windows 10 (1809) / netwtw04.sys	Yes	Yes
Raspberry Pi 3 Model B	Broadcom BCM43438	Kali 2018.4 w/ Nexmon / brcmfmac	No	No
		Raspbian 9 / brcmfmac	No	No

TABLE IV
CSA RECEPTION BEHAVIOR OF DIFFERENT SMARTPHONES AND TABLETS

Device	OS	Reassoc.	Blocks TX
Apple iPad Air (A1566)	iOS 12.3.1	No	Yes
Samsung SM-G973F	Android 9	No	Yes
Samsung SM-P585M	Android 8.1	Yes	No
Motorola XT1600	Android 7.1.1	Yes	No
Samsung SM-G925F	Android 7.0	No	No
LG E612F	Android 4.1.2	Yes	No
Samsung GT-P6800	Android 4.1.2	Yes	Yes

concerning the compliance with the block-transmission command and the need to reassociate with the original AP. In Table III, it can be seen that three mobile devices requested a reassociation with the AP when running Windows. However, it is interesting to note that in the Ubuntu environment, these same devices did not require reassociation and continued to send packets as soon as they switched channels. This shows that potentially incompatible devices may only require software updates to support CSA without requiring physical modifications to their hardware. Furthermore, in some situations, the device performed a reassociation (value Yes) and also respected the transmission constraint command (value Yes). This behavior implies that STAs that recognize the CSA mechanism may still request a reassociation to the AP after channel switching.

Despite providing new functionality for IEEE 802.11 networks, it is essential to assess the quantitative impacts of our proposal. Therefore, in the following sections we present experiments that evaluate several aspects of our solution.

C. VAP Overhead

Unlike a traditional IEEE 802.11 network where a BSS can contain multiple associated STAs, our proposal specifies that each VAP accepts only one STA, allowing each STA to have its own VAP and VAP migrations to occur at any time without interrupting the connection of other STAs associated with the AP. Therefore, physical APs must host multiple VAPs, and since each VAP originates the creation of a BSS, the AP WNIC must host multiple BSSs at the same time.

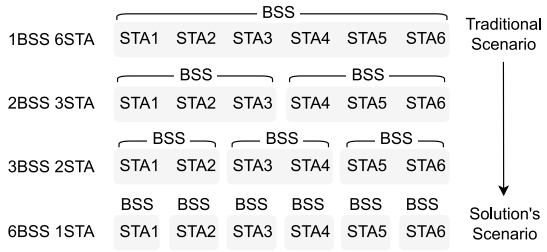


Fig. 5. BSS distribution scenarios for the evaluation of the overhead caused by multiple BSSs in a single physical AP.

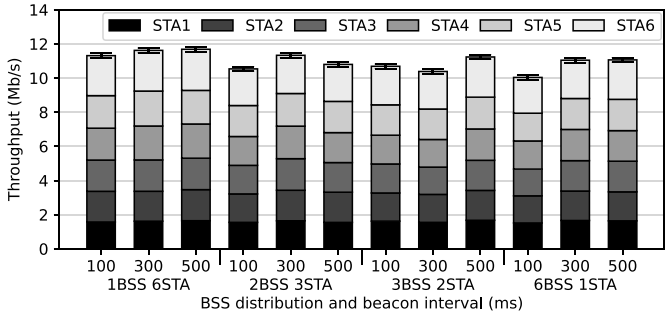


Fig. 6. Comparison of the average throughput of each STA. The confidence interval is based on a confidence level of 95%.

The inclusion of extra BSSs to be run by the WNIC increases the resource utilization of the shared medium and the AP since more management frames need to be processed or transmitted (*e.g.*, rather than sending a single beacon to all associated STAs, a beacon will be sent for each BSS and, hence, for each associated STA). An experiment with multiple STAs was held to analyze the performance impact of increasing the number of BSSs hosted by an AP.

For this experiment, we used 6 Raspberry Pi 3 Model B as STAs and an Ubuntu laptop with an Atheros AR9485 WNIC as an AP. All of them were operating in IEEE 802.11g mode. First, we started with all six STAs associated to only one BSS. Then, in each step of the experiment, the number of BSSs was increased such that the number of associated STAs would remain equally divided among them (*i.e.*, each BSS would contain the same number of STAs), as shown in Figure 5. With the goal of analyzing possible benefits in reducing the amount of beacons sent by each BSS of the AP in a period of time, we also varied the time interval between beacon transmissions for each step of the experiment. The *iperf* tool was used to generate uplink TCP traffic from the STAs to the network at the maximum bandwidth available. The Network Time Protocol (NTP) was used to synchronize the clocks of STAs and the AP.

In this experiment, each step was repeated 60 times and, at each run, the throughput experienced by the STAs was recorded for 50 seconds. Then, the average of the observed values was calculated. Figure 6 shows the average and aggregate throughput experienced by each STA for each configuration.

Comparing the configurations with beacon intervals of 100 ms, one can observe that the aggregated throughput decreased when the number of BSSs increased, which would corroborate the idea that performance degrades with the

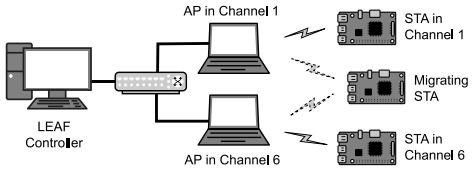


Fig. 7. Scenario used for RTT evaluation of the proposed solution, composed of three STAs, two physical APs and one controller connected by a switch.

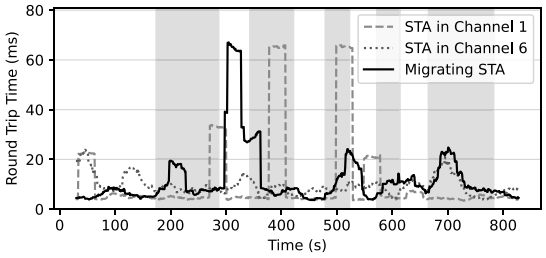


Fig. 8. RTT perceived by each STA during the experiment. A moving average with a sliding window of 30 points was used to smooth the data variation. The VAP is migrated between physical APs ten times. The greyed out area represents the periods in which the VAP was hosted by the AP in channel 6.

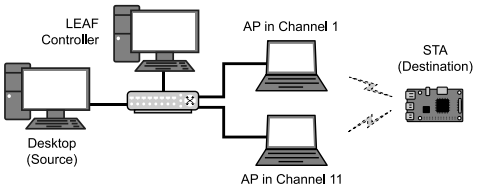


Fig. 9. Topology of throughput, outage time and packet loss experiments.

number of BSSs per AP. However, with a beacon interval of 300 ms, for instance, this behavior is not observed since the throughput in the last configuration is higher than in the third one, even with the double of BSSs being hosted by the AP.

For the first and the last BSS configurations, increasing the transmission interval of beacons resulted in a throughput increase. Still, the same behavior is not observed for the second and third configurations. Also, despite STA6 achieving slightly higher throughput, the results show that the achieved throughput for each BSS configuration is somewhat fairly divided between the six STAs, with a Jain's fairness index [22] no worse than 0.98.

As a conclusion of this experiment, although the first configuration achieved slightly higher throughput, the increase in the number of BSSs did not cause significant performance degradation. The experiment attests that our solution is capable of achieving throughput rates comparable to commonly used WLAN scenarios while providing greater flexibility for the management of IEEE 802.11 networks. For denser scenarios, with a greater number of STAs per physical AP, a higher beacon interval can be used to mitigate the beacon overhead.

D. Handover Benchmark

The STA handover phase performed by the network is a key point of the proposed solution. In this section, we present experiments that measure VAP migration performance in relation to four different metrics: Round Trip Time (RTT), throughput, outage time, and packet loss.

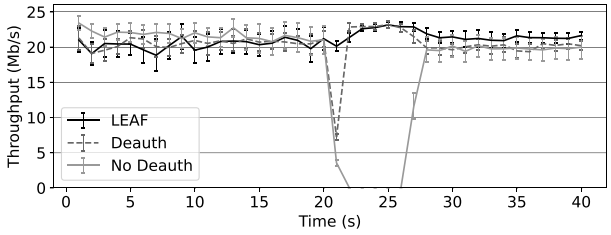
1) Round Trip Time: The handover of STAs might delay and disrupt network connections. In time-based applications, such as Voice over IP (VoIP) and video conference, these latency issues may severely deteriorate the Quality of Experience (QoE) of users. Therefore, we begin the handover evaluation investigating the impact of multiple VAP migrations toward the RTT of an ongoing connection.

For the evaluation of the RTT achieved during the migration of a VAP, we utilized three Raspberry Pi 3 Model B as STAs, two laptops — running instances of the LEAF agent, based on HostAPD code — acting as physical APs on channels 1 and 6, and a desktop — running LEAF's controller software — serving as a controller, as shown in Figure 7. Two of the STAs were used as performance baselines of each channel. They were associated to VAPs operating in each of the two physical APs. The third STA and its respective VAP was transferred between the two physical APs ten times during the experiment. The migrations were started arbitrarily by sending migration requests to the APs, without relying on a decision mechanism. All STAs ran the *ping* command to measure the RTT of the communication between the STA and its VAP. We used a *Channel Switch Count* number of 15 TBTTs.

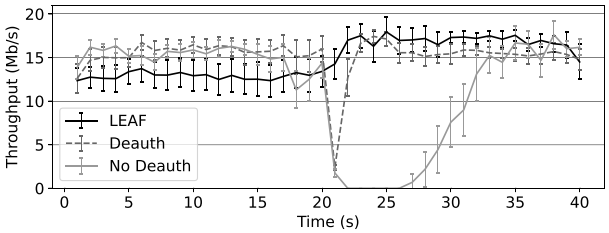
The results in Figure 8 show that a peak in the RTT of the migrating STA connection occurs around 300 s when the STA changes to channel 1. However, the stationary STA in channel 1 experiences a similar behavior around 400 and 500 s, which denotes a performance degradation in channel 1. Thus, throughout the experiment, the migrating STA experiences RTTs that are compatible with the RTTs of the stationary STAs and the performance of each channel. Therefore, the handoff of an STA does not introduce significant communication delays, confirming that the proposed solution can, in fact, perform a multichannel handoff without interrupting the connection.

2) Throughput, Outage Time and Packet Loss: In addition to quantifying the impact of the VAP migration process, in the following experiments, we compared the proposed solution to two other handover scenarios. In the first scenario, the AP to which the STA is currently associated broadcasts a deauthentication frame. In the second case, the current AP is turned off without transmitting any warning to the STA. In both cases the STA is forced to associate with the second available AP. Throughout the section, we referred to the prior scenario as *Deauth*, while the second was called *No Deauth*.

For all the scenarios mentioned above, we have used the same network, as illustrated in Figure 9: the experiment topology comprises two laptops with Atheros WNICs running the LEAF agent acting as a physical AP operating in the IEEE 802.11g mode on channels 1 and 11 of the 2.4 GHz band. An Ubuntu PC was used to execute the LEAF controller. We also utilized a desktop to generate traffic in the network. All these devices were connected to each other through a switch. Finally, a Raspberry Pi 3 Model B was used as an STA to where traffic is being sent. Clock synchronization was done using NTP to start the experiments at the source and the destination at a specified time. To generate and capture the transmitted traffic we used respectively the *iperf* and *tcpdump* tools. We repeated



(a) UDP



(b) TCP

Fig. 10. Average UDP and TCP throughput experienced by the station in three different handoff scenarios. The confidence interval is based on 30 executions with a confidence level of 95%.

the experiment using both TCP and UDP as transport protocols for the injected traffic. In both cases, *iperf* was configured to generate traffic at rate of 30 Mb/s at the application layer. In this analysis, the experiment was repeated 30 times and each run lasted 40 seconds. The controller was configured to migrate the STA and deactivate the BSS running on the AP in channel 1 after 20 seconds of execution.

Figure 10 shows the average throughput experienced by the STA for each handoff scenario. One can observe that there was no significant throughput drop during the migration period (*i.e.*, no valleys can be seen around 20 seconds of execution) when using the proposed solution. Moreover, it is interesting to note that after migration, TCP throughput increased. This behavior can be explained by the fact that channel 11 was less busy than channel 1, and thus TCP was able to increase its packet transmission rate. Before the migration, the average throughput for TCP is below the other two compared scenarios. Since each scenario was executed at different times, the variability in the medium utilization lead to a lower average throughput. This behavior is not observable in the UDP results. For the *Deauth* scenario, a noticeable throughput drop occurs after 20 seconds of execution for both TCP and UDP. Finally, in the *No Deauth* scenario, the throughput for both protocols drops to zero during several seconds, with a slow recovery of the TCP throughput due to its congestion control.

Following the throughput analysis, we also addressed the connection outage time caused by the handover. The outage period refers to the amount of time during which the STA is not able to receive or transmit data because it is still changing its channel (in the scenario of the proposed solution) or it is not yet associated with the other available AP after disassociating from the first AP (in the other two scenarios). In our experiment, the outage time was measured as the delay between the last TCP/UDP packet received through the AP in channel 1 until the first packet received through the AP in

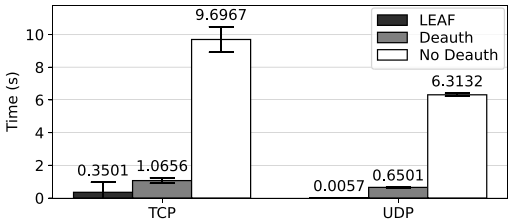


Fig. 11. Average outage times experienced by the STA in three handoff scenarios. The 95% confidence intervals are based on 30 executions.

channel 11. The *tshark* tool was used to process the traffic exchanged between source and destination. Figure 11 shows the average outage time for TCP and UDP traffic.

The results show an average outage time of 350 ms for TCP and less than 6 ms for UDP using our solution. In comparison to the *Deauth* approach — which is similar to a typical STA-based handover — this represents a reduction of around 67% in the average outage time for TCP connections and 99% for UDP. Also, not advertising the BSS deactivation increases the average outage time approximately 9 times, as can be noticed by examining the *Deauth* and *No Deauth* scenarios.

Comparing both transport protocols, our proposal achieved a TCP outage time significantly higher than that of UDP. This occurred because in one of the 30 executions the outage time was unexpectedly high, with the TCP traffic taking several seconds to resume after the handover. Although the cause for this is unknown, we investigated this particular result and found that the proposed solution behaved as expected, since the STA received the CSA IE and started receiving frames from the new physical AP without requiring a new association with its VAP. Analyzing the median outage time as an alternative metric reduces the outlier interference in the results. In comparison to the *Deauth* scenario — which had a median outage time of 923 ms —, our solution resulted in a reduction of approximately 97% in the using TCP, with a median outage time of only 23 ms. We should highlight that, even on average — *i.e.*, with the effect of the outlier —, our solution achieved a satisfactory outcome.

For the packet loss analysis we repeated the same methodology used in the two previous experiments. However, we also used a sniffer in the desktop node (*i.e.*, the source of the traffic) to analyze the packet rate at the traffic source. Since both TCP and UDP operate over the Internet Protocol (IP), the identification field in the IP header was used for packet loss computation. If a gap in the values of this field is detected between two consecutive packets, then packets were considered lost.

Figure 12 shows the average percentage of packets lost during the experiment. In all cases the UDP traffic had significantly more packets lost than TCP. This behavior is expected since, unlike UDP, TCP has a congestion control mechanism that reduces its transmission rate when acknowledgments for sent packets are not received. Consequently, the number of lost packets is also reduced. Moreover, one can observe that the proposed solution obtained a lower packet loss rate with UDP, when compared to the *Deauth* and *No Deauth* scenarios. In contrast, our mechanism exhibited a slightly higher amount

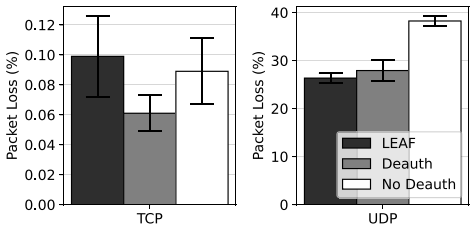
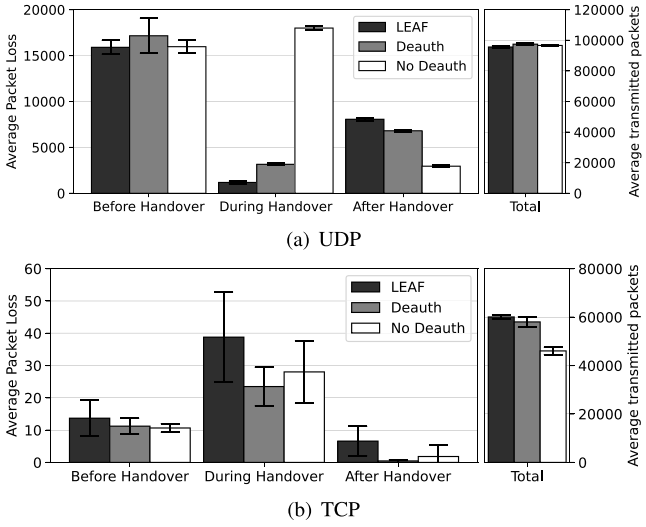


Fig. 12. Average packet loss percentage experienced by the STA through the duration of the experiment (40 seconds). The confidence interval is based on 30 executions with a confidence level of 95%.



(a) UDP

(b) TCP

Fig. 13. Average packet loss experienced by the STA before, during and after the handover and the average number of packets transmitted during the experiment. The 95% confidence intervals are based on 30 executions.

of lost packets for TCP traffic. However, this difference was within the confidence interval of the other two scenarios. Moreover, this direct comparison is somewhat misleading. Notice that this experiment comprises three different stages: before, during and after the handover. Thus, in order to completely understand the behavior of each approach, it is important to scrutinize the losses during each of these stages. Figure 13 presents the average number of lost packets in each stage of the experiment and of transmitted packets.

For UDP, our proposal resulted in fewer packet losses during the handoff phase when compared to other scenarios, exhibiting an average of less than 2%. In comparison, the *Deauth* scenario had 161% more losses. On the other hand, for the *No Deauth* scenario, most packets were lost during the handover period among the three analyzed periods.

With TCP, however, our solution resulted in the highest packet loss rates during the handover. Nevertheless, this adverse result is again explained by TCP's congestion control mechanism. As shown in Figure 14, during the handover process, TCP's transmission rate with our solution was higher than that of the two other scenarios. Consequently, more packets were lost during the brief outage time, but the average packet loss during this phase only accounted for less than 0.1% of total transmitted packets. For the other two scenarios, however, packet transmission rates drop significantly, resulting

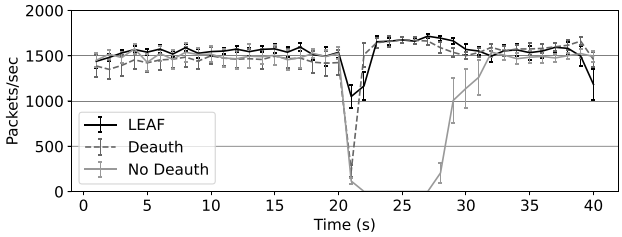


Fig. 14. Average TCP packet rate measured in the source. The confidence interval is based on 30 executions of the experiment.

in fewer losses and fewer transmitted packets. With UDP, conversely, the transmission rate remains constant throughout the execution at around 2675 packets per second (omitted due to size restrictions), and the number of lost packets is proportional to the outage time, which corroborates the relationship TCP's transmission rate and packet loss during the handover.

Our results show that LEAF can be executed in current off-the-shelf devices, significantly reducing the overhead of STA handover and providing a mechanism for migrating STAs, which are important characteristics for its applicability in real-world applications. LEAF can be used for several management tasks, such as association control, load balancing, interference mitigation, and energy saving, as further discussed in Section VI. Different from other approaches in the literature, LEAF fully supports migrations between APs operating in multiple channels — with no restrictions on channel usage — and does not require multiple WNICs per AP.

VI. Discussion and Future Work

LEAF increases the flexibility and management of WLANs through the migration process. Therefore, several scenarios can benefit from the functionality provided by our solution.

A well-known problem of IEEE 802.11 networks is when an STA, in the border of the coverage area of two or more APs, switches continuously from one AP to another due to the variations of signal in a phenomenon known as *ping-pong effect*, which remains a challenge in current devices [23]. The proposed solution could be used to mitigate this problem by moving the VAP along with the STA, avoiding need for the STA to perform handoffs between BSSs.

In dense scenarios, multiple STAs may try to connect to the same closest AP, even if there are other idle APs in the vicinity. The proposed solution can be used to perform load management by quickly migrating STAs to available APs in order to balance the resource utilization of the STAs. The same technique can be applied to mitigate interference between transmitters on the same channel, migrating STAs to APs operating in non-overlapping channels to reduce collisions during transmissions. Also, in periods of low utilization, the energy consumption of a WLAN can be reduced by migrating STAs from sub-utilized APs, concentrating these STAs at fewer APs of the network, which allows the deactivation of the radio of the unused infrastructure devices. From the network administration perspective, LEAF can also be used by administrators to fine-tune the network, ensuring that STAs are

associated with the most suitable AP or tailoring associations for specific scenarios. For example, during conferences or events, administrators might want to migrate STAs from auditorium APs when extra bandwidth is required.

Despite the advantages brought by our solution, we detected limitations during the development and evaluation of our prototype, which might increase the expected packet loss and handoff delay upon migration but do not invalidate the migration process or the benefits provided by the control plane.

Although CSA was included to the base IEEE 802.11 standard in 2007 [24], legacy devices may still not perform the channel switch upon receiving the CSA IE. Nevertheless, LEAF is compatible with these legacy devices, but after migration, they need to re-scan and reassociate with the VAP on the new physical AP. This might result in longer delays and packet losses during the handover when compared to fully CSA-compliant devices.

Because the virtualization solution consists of an encapsulation of BSSs, the number of concurrent BSSs supported by the WNIC of the devices that act as APs — which varies depending on its model and manufacturer — can also be a limiting factor (since the number of VAPs supported by each physical AP in our solution is directly limited by how many BSSs the interface supports). That limitation can be mitigated by adding multiple WNICs in each physical AP, although that will increase the cost of deployment. Another way to circumvent this limitation is by having the WNIC of the physical APs in monitor mode and making the packet filtering and processing tasks for each VAP at a software level. We intend to further investigate how to overcome this limitation without relying on additional WNICs.

Regarding the beacon generation of the VAPs, since one beacon is generated for each VAP, a signaling overhead is expected, as stated in Section V-C. One could argue that an approach to reduce the number of transmitted beacons is to piggyback the information of these multiple VAPs being held by the same AP in a single beacon. In fact, the IEEE 802.11 [24] standard provides a Multiple BSSID beaconing procedure to advertise all available BSSs managed by an AP in a single beacon. However, our proposal requires the BSSIDs of each VAP to be independent of the current AP and other VAP, so that we can migrate the VAPs (and their BSSIDs) between different devices. In the Multiple BSSID beaconing, the identifiers of piggybacked BSSs are derived from the reference BSS, which creates a dependency between them. Another issue is related to the CSA mechanism. Our approach includes the CSA IE only in the beacon of the VAP that will be migrated. In the Multiple BSSID beaconing, the inclusion of the CSA IE in a beacon could lead to unwanted behavior — such as inducing a channel switch for every STA associated with the BSSs — since a single beacon is transmitted for all BSSs in an AP. Adjusting the Multiple BSSID beaconing to allow the explicit transmission of each BSSID would solve the first issue, while the second can be solved by including the identifier of the BSS that will change its channel in the CSA procedure. However, this would require modifications to the IEEE 802.11 standard.

Besides, in our implementation, the AP periodically transmits beacons for each occupied and available VAP. Because the solution does not indicate which VAP is vacant, a new STA, seeking connection to the network, might attempt to connect to several occupied VAPs before successfully associating with the available VAP. An approach to mitigate this problem is to increase the beaconing interval or even disable the beacon transmission of occupied VAPs. This would make vacant VAPs more apparent during the scan process of nearby STAs, in comparison to the occupied ones, and would also help to reduce the beaconing overhead of VAPs. The IEEE 802.11 [24] standard provides a mechanism to announce a new beacon transmission interval, but this mechanism seems to be exclusive to DMG (Directional Multi-Gigabit) BSSs. Thus, we intend to explore alternatives to enable increasing beacon intervals of occupied VAPs and their synchronization impact in the future.

We also intend to analyze the solution scalability concerning the demands of the application plane (*e.g.*, how often the application plane requests data plane information, which will vary according to the type of application). In addition, we intend to build solutions for load-balancing, mobility management and network power-saving that rely on the proposed architecture, implemented as algorithms of the application plane which communicate with our control plane.

VII. Conclusion

In this work, we presented a lightweight AP virtualization solution which relies on a central controller to enable inter-channel handover of STAs based on a network-wide view rather than the constrained view of the STA.

Our proposal allows the transparent handover of STAs between multiple APs, as the STA remains associated with the same BSS (or VAP) during the handover process. Despite not addressing when the migration should occur, the abstraction layer provided by our controller allows the handover decision to be made based on different metrics, providing flexibility to applications that can implement their own handover and AP selection algorithms on top of our solution. Based on the developed prototype, we carried out experiments to assess the feasibility and performance aspects of the solution. Our results show that our proposal indeed allows seamless migration of STAs between physical APs, with little to no impact in terms of delay and packet losses even when the migration involves a channel switch. As such, we believe that our proposal increases the flexibility of WLANs, allowing network-wide optimization tasks. Besides the enhancements discussed in Section VI, as future research, we plan to study which metrics are relevant to network management purposes and expand LEAF's capabilities to gather and provide those metrics to the application plane.

References

[1] H. Balbi, D. Passos, R. C. Carrano, L. Magalhães, and C. Albuquerque, "A case study of association instability in dense IEEE 802.11 networks," in *Proc. IEEE Symp. Comput. Commun.*, 2019, pp. 1–6.

[2] M. Kawada, M. Tamai, and K. Yasumoto, "A trigger-based dynamic load balancing method for WLANs using virtualized network interfaces," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2013, pp. 1091–1096.

[3] N. Ekiz, T. Salih, S. Kucukoner, and K. Fidanboylu, "An overview of handoff techniques in cellular networks," *Int. J. Inf. Technol.*, vol. 2, no. 3, pp. 132–136, 2005.

[4] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANS with Odin," in *Proc. Workshop Hot Topics Softw. Defin. Netw. (HotSDN)*, 2012, pp. 115–120.

[5] M. E. Berezin, F. Rousseau, and A. Duda, "Multichannel virtual access points for seamless handoffs in IEEE 802.11 wireless networks," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, 2011, pp. 1–5.

[6] Y. Grunenberger and F. Rousseau, "Virtual access points for transparent mobility in wireless LANs," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2010, pp. 1–6.

[7] E. Zeljkovic, J. M. Marquez-Barja, A. Kassler, R. Riggio, and S. Latré, "Proactive access point driven handovers in IEEE 802.11 networks," in *Proc. Int. Conf. Netw. Service Manag.*, 2018, pp. 261–267.

[8] O. Stiti, O. Braham, and G. Pujolle, "Virtual openflow-based SDN Wi-Fi access point," in *Proc. Glob. Inf. Infrastruct. Netw. Symp. (GIIS)*, 2015, pp. 1–3.

[9] A. Zubow, S. Zehl, and A. Wolisz, "BIGAP—Seamless handover in high performance enterprise IEEE 802.11 networks," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, 2016, pp. 445–453.

[10] J. L. Vieira and D. Passos, "An SDN-based access point virtualization solution for multichannel IEEE 802.11 networks," in *Proc. Int. Conf. Netw. Future (NoF)*, 2019, pp. 122–125.

[11] *SO/IEC International Standard—Information Technology—Local and Metropolitan Area Networks—Part 11: Amendment 5: Spectrum and Transmit Power Management Extensions in the 5 GHz Band in Europe*, IEEE Standard 802.11h-2003 (Amendment to IEEE Standard 802.11-1999), Oct. 2003, doi: 10.1109/IEEESTD.2003.94393.

[12] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.

[13] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, "Designing high performance enterprise Wi-Fi networks," in *Proc. NSDI*, vol. 8, 2008, pp. 73–88.

[14] H. Moura, G. V. C. Bessa, M. A. M. Vieira, and D. F. Macedo, "Ethanol: Software defined networking for 802.11 wireless networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, 2015, pp. 388–396.

[15] S. M. M. Gilani, T. Hong, W. Jin, G. Zhao, H. M. Heang, and C. Xu, "Mobility management in IEEE 802.11 WLAN using SDN/NFV technologies," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, pp. 1–14, Apr. 2017.

[16] E. Zeljković, N. Slamnik-Krieštorac, S. Latré, and J. M. Marquez-Barja, "ABRAHAM: Machine learning backed proactive handover algorithm using SDN," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1522–1536, Dec. 2019.

[17] P. Dely, J. Vestin, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, "CloudMAC—An openflow based architecture for 802.11 MAC layer processing in the cloud," in *Proc. IEEE Globecom Workshops*, 2012, pp. 186–191.

[18] Z. Shi, Y. Tian, X. Wang, J. Pan, and X. Zhang, "Po-Fi: Facilitating innovations on WiFi networks with an SDN approach," *Comput. Netw.*, vol. 187, Mar. 2021, Art. no. 107781.

[19] J. Saldana et al., "Unsticking the Wi-Fi client: Smarter decisions using a software defined wireless solution," *IEEE Access*, vol. 6, pp. 30917–30931, 2018.

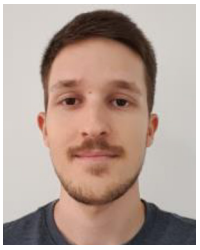
[20] J. Malinen et al. "hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/WPA3/EAP/RADIUS authenticator." hostapd. [Online]. Available: https://w1.fi/hostapd/

[21] "iperf2." 2024. [Online]. Available: https://sourceforge.net/projects/iperf2/

[22] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination," Eastern Res. Lab., Digit. Equip. Corp., Hudson, MA, USA, Rep. TR-301, 1984.

[23] H. D. Balbi, D. Passos, J. Vieira, R. C. Carrano, L. C. Magalhães, and C. Albuquerque, "Towards a fast and stable filter for RSSI-based handoff algorithms in dense indoor WLANs," *Comput. Commun.*, vol. 183, pp. 19–32, Feb. 2022.

[24] *IEEE Standard for Information technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2016 (Revision of IEEE Standard 802.11-2012), 2016, doi: 10.1109/IEEESTD.2016.7786995.



Juan Vieira received the B.Sc. and M.Sc. degrees in computer science from Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil, in 2017 and 2019, respectively, and the D.Sc. degree in computer science from Universidade Federal Fluminense, with a period abroad in University of Pittsburgh, in 2024. He has been involved in projects regarding Smart Grids with strict latency requirements and resource allocation in Edge Computing environments. His research interests include wireless networks, Internet of Things, and software-defined networks.

Daniel Mosse received the B.S. degree in mathematics from the University of Brasilia, Brazil, in 1985, and the M.S. and Ph.D. degrees in computer science from the University of Maryland at College Park, College Park, in 1990 and 1993, respectively. He has been a Professor with the University of Pittsburgh since 1992, including six years as a department chair, and has co-founded HiberSense, a startup company in the area of Smart Homes. His main research interest is in the allocation of resources in the realm of sustainable computing, computing for sustainability, and real-time, with the main concerns being power management, security, and fault tolerance. For the last 20 years, most of his systems research has focused on power and energy management, and for the last decade on how to increase diversity and promote reproducible research in computing.

Diego Passos (Member, IEEE) received the B.Sc., M.Sc., and D.Sc. degrees in computer science from Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil, in 2007, 2009, and 2013, respectively. From 2013 to 2014, he worked as a Postdoctoral Fellow Researcher with the Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil, where he was a Professor with the Computer Science Department from 2014 to 2021. He is currently a Professor with the Department of Electronical Engineering, Telecommunications and Computers of the Instituto Superior de Engenharia de Lisboa. His research interests include multihop wireless networks, network coding, and wireless routing.