ABRAHAM : machine learning backed proactive handover algorithm using SDN

Reference:
Zeljkovic Ensar, Slamnik-Krijestorac Nina, Latré Steven, Marquez-Barja Johann.- ABRAHAM : machine learning backed proactive handover algorithm using SDN
IEEE transactions on netw ork and service management / IEEE [New York, N.Y.] - ISSN 1932-4537 - 16:4(2019), p. 1522-1536
Full text (Publisher's DOI): https://doi.org/10.1109/TNSM.2019.2948883
To cite this reference: https://hdl.handle.net/10067/1647640151162165141

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2948883, IEEE Transactions on Network and Service Management

ABRAHAM: Machine Learning Backed Proactive Handover Algorithm using SDN

Ensar Zeljković, Nina Slamnik-Kriještorac, Steven Latré, and Johann M. Marquez-Barja
University of Antwerp - imec, IDLab research group, Sint-Pietersvliet 7, 2000 Antwerp, Belgium.
{name.lastname}@uantwerpen.be

***Abstract*—An important aspect of managing multi access point (AP) IEEE 802.11 networks is the support for mobility management by controlling the handover process. Most handover algorithms, residing on the client station (STA), are reactive and take a long time to converge, and thus severely impact Quality of Service (QoS) and Quality of Experience (QoE). Centralized approaches to mobility and handover management are mostly proprietary, reactive and require changes to the client STA. In this paper, we first created an Software-Defined Networking (SDN) modular handover management framework called HuMOR, which can create, validate and evaluate handover algorithms that preserve QoS. Relying on the capabilities of HuMOR, we introduce ABRAHAM, a machine learning backed, proactive, handover algorithm that uses multiple metrics to predict the future state of the network and optimize the AP load to ensure the preservation of QoS. We compare ABRAHAM to a number of alternative handover algorithms in a comprehensive QoS study, and demonstrate that it outperforms them with an average throughput improvement of up to 139%, while statistical analysis shows that there is significant statistical difference between ABRAHAM and the rest of the algorithms.**

***Index Terms*—IEEE 802.11, SDN, handover algorithm**

I. Introduction

IEEE 802.11 (Wi-Fi) based networks are one of the most widespread wireless networks in today's world. To serve multiple mobile client STAs with an ever-increasing demand for resources and coverage range, today's Wi-Fi networks today consist of multiple APs. The mobility of STAs is a key feature of Wi-Fi networks, which requires proper management in order to ensure the preservation of the QoS and QoE. To do so, Wi-Fi network use handovers, a process that occurs when a STA moves its association from one AP to another to ensure continuous connectivity.

However, most handover algorithms are STA based, which means they can only create per-device optimizations and do not have a notion of global network management, in contrast to a centralized approach. The handover process is also mostly reactive, meaning it waits for a performance drop to trigger the handover and thus lacks dynamic resource management in advance. It can also take quite some time for the handover to complete, ranging from several hundred milliseconds [1] to 4 seconds [2]. Many existing handover solutions, therefore, greatly impact the QoS and QoE requirements of modern applications and services, and require changes on the STA side. Even though some researchers have moved the handover management to a centralized location, to have more global network management and avoid STA side changes, most of them are proprietary. Also, the handover AP selection algorithms mostly use only one metric, the Received Signal Strength Indicator (RSSI), and thus lack important information such as the load of the APs or STA mobility information. Very little research has been done on using more metrics, especially metrics such as location information, mobility of the STAs, as well as the load of the network. Even if those metrics were used, in most cases they were used as the single metric of decision, [3]. Finally, a significant amount of research on handover algorithms, as well as handover management has only been done in simulations. Other, real-life experiments lacked a centralized approach to handover management.

Using the SDN paradigm in Wi-Fi networks, the management of the mobility can be moved to a centralized location. Through network monitoring, this centralized location can have a global overview of the network and create global optimizations even in cases of increased network and service dynamics, without modifications to the STA. Utilizing this global information, one can create handover algorithms that can dynamically manage the resources and preserve the QoS and QoE. More important metrics can be utilized in this process, such as the RSSI, location information, mobility information, as well as the load of the network, in order to create a reliable handover algorithm.

In this paper, we present a threefold contribution. First, we introduce the Handover Management framewORk (HuMOR), which allows users to create handover algorithms that are centralized, proactive, and can use multiple metrics in their decision-making process. HuMOR is an SDN modular framework enabling the creation, validation, and evaluation of handover algorithms, running on a large scale, real-life testbed. Second, by relying on HuMOR, we introduce ABRAHAM, a mAchine learning Backed multimetRic proActive Handover AlgorithM. Using machine learning it predicts the future RSSI, which along with the predicted future STA location and predicted future AP load is used to optimize the AP load in the network and preserve the QoS and QoE. Third, in order to validate and evaluate ABRAHAM, we present a comprehensive study where we compare ABRAHAM to our previous proactive handover algorithms and the IEEE 802.11 standard handover procedure by analyzing different QoS metrics obtained through experimentation.

This paper consists of the following sections. Section II presents the related work, while Section III introduces the enabling framework HuMOR. Section IV describes the ABRAHAM handover algorithm and Section V presents the comprehensive comparison study on handover algorithms. Finally, Section VI summarizes the conclusions of this paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2948883, IEEE Transactions on Network and Service Management

TABLE I: Related Work

Handover Management	Handover Type	Trigger location	Approach	Input parameters		Works
Centralized	Vertical	N/A	N/A	context information		[4]
	Horizontal	STA-driven	Reactive	none		[5]
				AP load balancing		[6–8]
		AP-driven	Reactive	none		[9–13]
				power control and STA association information		[14,15]
				AP-STA link rate and STA throughput demand		[16]
			Proactive	current RSSI (location-aware)		[5,9,17–20]
				current and future RSSI (location and mobility-aware)	none	[21]
					throughput and load information	[22]
Decentralized	Vertical	STA-driven	N/A			[23]
	Horizontal	AP-driven	N/A			[1,24,25]

II. Related Work

Extensive research has been conducted to move the intelligence of the handover and AP selection to a centralized controller. The outcome of such research proves that the introduction of the SDN paradigm to the context of the handover process brings significant improvements in handover management, particularly in terms of QoS awareness and QoE requirements of STAs. Following the introduction of SDN in network management related to handover mechanisms, we present a brief overview of related works in Table I. These SDN solutions are intended to solve problems caused by handover management control, such as AP overloading, STA performing both the AP selection and handover, and increased delay. As it can be seen in Table I, we categorize all the works into two groups: *centralized* and *decentralized*, based on the way how the network management is realized.

Although a significantly larger effort is invested in the research of centralized solutions, there are several examples which adopt handover management in a distributed fashion. Berezin et al. [24] briefly present a decentralized handover algorithm, proposing a protocol for the direct information exchange between APs. However, this algorithm requires building a list of neighbouring APs for each AP, decreasing scalability and efficiency of the solution. In the scope of LTE networks, Ali et al. [25] anticipate QoE value for a user and take it into account while performing handover to another eNodeB. The associated eNode gathers information about the user's previous QoE from neighboring eNodeBs, and thus makes the decision about user's re-association. Another decentralized approach is presented by Yi et al. [23], adopting Media Independent Handover (MIH) to support fully distributed handovers in heterogeneous networks based on Open vSwitch and OpenFlow. Their algorithm enables mobile terminals with more than two network interfaces to make decisions upon wireless technology themselves, but being controlled by an SDN controller during the whole process.

In centralized SDN handover management solutions, authors have mainly adopted the STA-driven approach which relies on the standard 802.11 handover mechanism, or AP-driven approach but in a reactive manner. Qiang et al. [26,27] propose a scheme which reduces the number of handovers and improves QoS. Despite taking into account the mobility of STAs [26], the prediction of movement trend is based only on the QoS value. They extend their algorithm by letting the SDN controller create a request matrix using channel capacity as well as AP load. The outcome of the algorithm is a list of corresponding APs, after which the STA waits for a certain stability period to re-associate, [27]. The extensive discussion about the stability period is further provided in [1].

Based on where the handover algorithm is performed, solutions can be divided into two categories: *STA-driven* and *AP-driven*. Gilani et al. [6] and Kiran et al. [7] present STA-driven handover schemes which include AP load balancing but only among adjacent APs. Their experimental results, conducted in a testbed environment with only two APs and in Mininet-WiFi emulator, show improvements in throughput in comparison with legacy schemes. However, it is of high importance to clarify why STA-driven handover management solutions do not scale in the real-world environment. First, these solutions require changes in the mobile node stack. Second, they are reactive, which might lead to poor QoS for a long duration of time. Finally, they can only utilize metrics available at the STA in order to make AP selection, while there are other important metrics to consider on the network side [22].

Aldhaibani et al. [17] present an algorithm for a horizontal handover utilizing fuzzy logic control, and their results prove superiority to the IEEE 802.11 standard and Load-RSSI-based algorithms, but only in a simulation environment. Another fuzzy-logic based algorithm is presented by Sun et al. [21], taking into account the available bandwidth as well as the current and future RSSI value, which is predicted using grey predictive technology. Since this is one of the rare attempts to take into account the STA mobility, we will now briefly present papers which address the same problems but without any mobility consideration. For instance, Murty et al. [5,19] define a set of APIs that allow STAs and APs to send information such as radio channel conditions to a central controller. Although DenseAP [19] adds localization awareness, it only focuses on very limited movement as the location is only updated every 30 seconds, while in [5] the STA's mobility is not taken into account at all. Similarly, Bayhan et al. [16] present an approach that uses a centralized controller and aims to maximize the overall throughput using different metrics such as AP-STA link rates, throughput demands of STAs, etc. Aldhaibani et al. [17] use several metrics such as bandwidth, jitter, delay and SINR, while Broustis et al. and Ahmed et al.

consider power control and STA association information as metrics for AP selection and handover procedure. However, they all still rely on the standard IEEE 802.11 handover mechanism and do not take into account the STA mobility. A potential way to solve the problems caused by legacy handover algorithms is by using the Light Virtual Access Point (LVAP) abstraction [1], adopting the mechanism in which a physical AP uses different LVAPs for communication with each STA in order to avoid re-association and its negative effects on QoS and QoE. Suresh et al. [20] propose creating abstractions for the AP and STA which allow seamless handovers, considering only RSSI value as a metric to trigger the handover.

As it can be seen, most of the papers consider either location or mobility information, while some of the papers consider none of them. Also, most of the papers take into account one or very limited number of metrics in order to proactively trigger handover. As stated in the elaboration of the vertical handover techniques [4], excluding the context information (i.e. user preference, available resources, location and mobile capabilities) from the handover mechanism leads to its simplification but surely causes QoS disruptions. Since the importance of location and mobility information of the APs and STAs is crucial for any efficient proactive handover solution, some of the location prediction methods can be found in [21,28–30]. Besides authors' previous papers and to the best of our knowledge, Sun et al. [21] presented so far the only approach which uses dynamic handover strategy based on mobility and RSSI at the same time. Therefore, this presses the need to strive for even better solutions which imply improved throughput and reduced delay, while reducing the number of handovers within the overall network. To accomplish this, our algorithm uses an SDN approach without any STA modifications, and takes into account multiple metrics in order to trigger handovers in a proactive way.

III. Handover Management Framework

To overcome the limitations and challenges of the handover algorithms in IEEE 802.11 Wi-Fi networks, we created an SDN-based modular handover management framework HuMOR that enables creating, validating and evaluating handover algorithms. It uses the principles of SDN to move the control of the handover to a centralized location, from which it also has a global overview of the whole network. To validate and evaluate handover algorithms, we did not use simulations, but deployed the HuMOR framework on a large scale, real-life testbed. In this section, we introduce HuMOR whose architecture blocks are illustrated on Figure 1 and explained throughout this section. We present the real-life testbed HuMOR runs on and how the user can monitor the experiments thought its Graphical User Interface (GUI) to validate the handover algorithms as well as how the user can evaluate the algorithms. Finally, we briefly introduce the handover algorithms from our previous work, which will be used in the comparison study afterwards.

A. Towards seamless SDN-based handovers

HuMOR is built on top of 5G-EmPOWER [31], a multi-access edge computing operating system which supports lightweight virtualization, and also supports heterogeneous radio access technologies enabling centralized control. By adopting the SDN principle, and by using softwarization and virtualization it moves the control of the network to a centralized controller. STAs on Figure 1 connect to APs which are part of the data plane and have 5G-EmPOWER agents running on them. These agents are used to communicate to the control plane, which is the 5G-EmPOWER runtime on the 5G-EmPOWER SDN Controller. When a STA connects to the network, 5G-EmPOWER creates an abstraction for it, called the LVAP which is instantiated on the AP that the station will associate to. This abstraction takes care of all the technical details of the STAs' association, like for example the 3-way handshake, etc. It then exposes Application Programming Interfaces (APIs) to allow the programmer to create more high-level manipulations of the STA.

One such manipulation is moving the LVAP between APs which is initiated through the handover mechanism block, shown on Figure 2. By moving the LVAP block from one AP to another, one essentially moves the association of the STA from one AP to another. In essence, a handover occurs. However, because the LVAP element holds all the association information of the STA, this is a soft handover which the STA is not aware of. It has been shown by Riggio et al. [32] that this type of a handover, based on the LVAP, is seamless and transparent to the STA.

Since the handover algorithm has been moved to a centralized, network side location, it has a global overview of the whole network allowing it to make global instead of per-device optimizations. We use this as the basis of HuMOR and build on top of it.

B. Metric modules

5G-EmPOWER monitors the network by retrieving monitoring data from the APs monitoring block on Figure 1 every $500(ms)$. This monitoring data is then aggregated on the SDN controller. HuMOR utilizes this monitoring data through a block that we call $Modules$, as seen on Figure 1. Certain modules just utilize the monitored data to retrieve a metric which is then stored in the Metrics DataBase (MDB). Other modules utilize stored metrics in the MDB to calculate new metrics. Because new monitoring data is retrieved every $500(ms)$, each monitored and calculated metric also has a timestamp, which is denoted by t. When creating a handover algorithm, it is of much greater interest to view not only the metrics current values, but also their history. Historical data can not only provide valuable insight into the changes of a given metric, but can also be used for predicting future values. We, therefore, also store historical data of the metrics in the MDB. Table II gives a complete list of metrics and parameters that are used in HuMOR. Along with the description of all the metrics, a notation is defined for all of them which will be used in the rest of this paper and in the description of the handover algorithms.

1) RSSI Module: The RSSI module takes the monitored RSSI and records the current RSSI value between each AP and each STA. This is then saved to the MDB where a historical of this value is stored.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2948883, IEEE Transactions on Network and Service Management

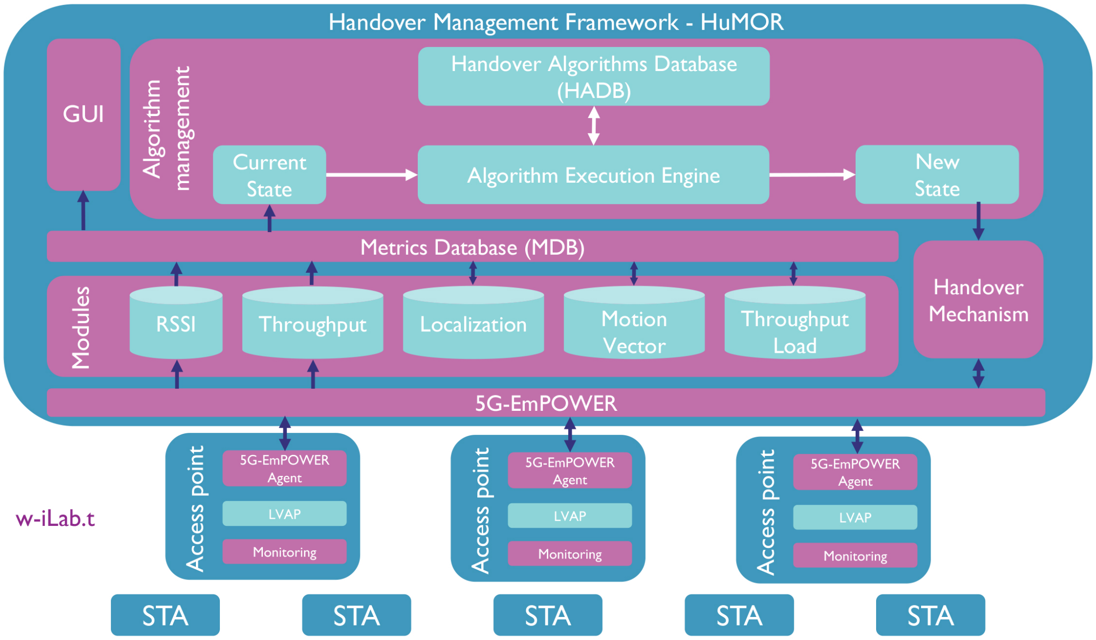


Fig. 1: Handover Management Framework - HuMOR

2) Throughput Module: The uplink and downlink throughput of all the STAs are monitored and a historical is stored in the MDB.

3) Localization Module: The localization module is used to localize each AP and each STA in the environment, as well as to calculate the estimated distances between each of the nodes. It can use any of the available metrics to calculate the locations and distances. We have implemented an RSSI based localization algorithm that has been proposed by Lim et al. in [33]. The reason we use this localization algorithm lays in its simplicity, as it only requires the RSSI information between the APs and STAs, and the fact that its localization error is within $3(m)$. The only a priori information for this algorithm to work is the position and distances between the APs, which considering the static placement of today's Wi-Fi networks should be easily obtainable. Using a technique called Singular Value Decomposition (SVD), a mapping T is created between (i) the RSSI between each AP, $rssi_{ap_i,ap_j}(t)$ and (ii) the distances between each AP $d_{ap_i,ap_j}(t)$: $T : rssi_{ap_i,ap_j}(t) \rightarrow d_{ap_i,ap_j}(t)$. This mapping also takes into account the characteristics of the environment and its changes, as it can be recalculated over time. Once this mapping is calculated, the RSSI between the STAs and APs can now be used to calculate the distances between them, $d_{ap,sta}(t) = T(rssi_{ap,sta}(t))$. Using Multi-Dimensional Scaling (MDS), the distances can be translated into locations in a 2D environment. Again, a historical of both the distances and locations are recorded and saved in the MDB.

4) Motion Vector Module: The motion vector module uses the location information as input in order to create mobility information for STAs. It takes the current and previous location of a STA, and creates a vector between these two points. This vector then shows the mobility of the STA. The vector's length represents the speed at which the STA is moving, while its angle shows the direction of the movement. Using this information, we can construct a path of the STA's movement, but also we can predict the STA's future location. By simply translating the motion vector, so that its starting point is in the current location of the STA, as depicted in Figure 3, we can

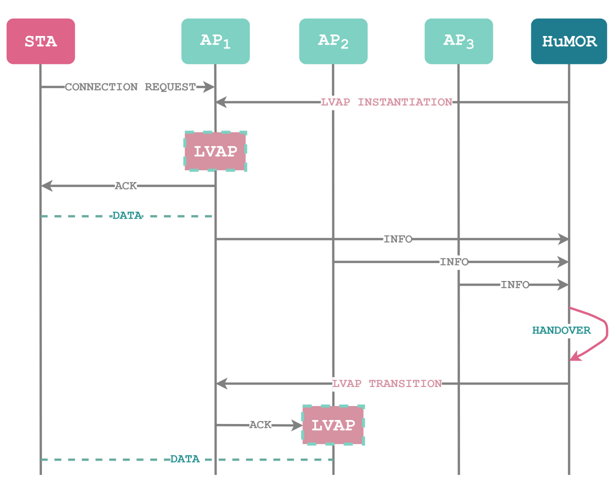


Fig. 2: MSC diagram of LVAP based handover

TABLE II: HuMOR metrics, parameters and their notation

Metric	Notation
Set of (ap, sta), where STA sta is associated to AP ap	$N(t)$
Actual RSSI value between STA sta and AP ap	$rssi_{ap,sta}(t)$
Is the STA sta associated to AP ap, 1 if it is, 0 otherwise	$a_{ap,sta}(t) = \{0, 1\}$
List of reachable APs for STA sta	$ra_{sta}(t)$
A list of all STAs that are associated to AP ap	$cs_{ap}(t)$
The throughput of STA sta	$r_{sta}(t)$
The requested throughput of STA sta	Q_{sta}
History of k STA sta throughput values	R_{sta}
Traffic load of AP ap	$b_{ap}(t)$
Capacity of AP ap	C_{ap}
Location of APs and STAs	l_{ap}, l_{sta}
Location of STA sta	$l_{sta}(t)$
History of k locations of STA sta	L_{sta}
Distance between STA sta and AP ap	$d_{ap,sta}(t)$.

say that the end point of the motion vector is the predicted future location of the STA. This takes the assumption that the STA will move in the same direction and with the same speed as in the previous point in time. Therefore, the objective of the motion vector is to determine the general direction the STAs are heading for. An average of the value is calculated and stored along with its history in the MDB.

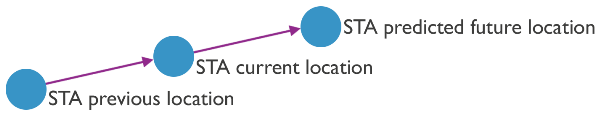


Fig. 3: Predicting STA future location using motion vector

5) AP Throughput Load Module: This module estimates the load of the APs by taking into account the throughput needs of all the STA that are associated to that particular AP.

Having all these metrics available in the MDB for the whole network, creates a better picture of the network state. Not only that, but now a handover algorithm can have multiple metrics as input, instead of only one, as most handover algorithms have in the literature. Room has been left for users to create additional modules, with other potentially interesting metrics. Also, any of the existing modules could be updated or changed. For example, if a more advanced localization algorithm is needed, the current one can be swapped. The output of the new localization module should again be the location and distances between the nodes in the network, which will be saved in the MDB.

C. Algorithm management

Figure 1 show the Algorithm Management (AM) block of HuMOR, where the handover algorithm itself is built and triggered. First, a current state of the network, $N(t)$, is created from the MDB. All the available metrics for the APs and STAs, including their historical data, are gathered to create a global overview. Next, this current state is fed to the algorithm execution engine. The algorithm executing engine has its own database that it communicates to, called the Handover Algorithms DataBase (HADB). Here, all the created handover algorithms are saved. The algorithm execution engine picks one of the handover algorithms from the HADB, feeds it the current network state and executes the handover algorithm. The output of each handover algorithm should be the new state of the network, $N(t+1)$, represented as a list of tuples. The tuples consist of a STA and the AP that the handover algorithm assigned for the STA. How the handover algorithm is executed can be changed, which directly influences the handover trigger. A handover algorithm can be triggered based on an event, to create a reactive handover, or the algorithm can constantly update the new network state, to create a proactive handover. Once the new network state is obtained, it talks to the handover mechanism block to initiate the handovers of the STAs to their assigned APs.

D. Real-life testbed

To be able to validate and evaluate handover algorithms created in HuMOR, we have decided to run it on a real-life testbed, instead of using simulations. The testbed we used is called w-iLab-t [34]. The w-iLab.t is an experimental, heterogeneous wireless testbed for development and testing of wireless applications. The reason this testbed was chosen, is because it hosts a number of different nodes, among which for the purpose of HuMOR, the most interesting are: i) the fixed wireless nodes that can serve as APs, ii) fixed server nodes which can serve to run HuMOR and iii) mobile wireless nodes that can move across the testbed and can serve as mobile STAs. w-iLab-t offers a web GUI called the robot dashboard which enables complete control of the mobile nodes. It shows a map of the whole testbed and allows the user to create a movement path for the mobile nodes. The nodes in the testbed are mounted in a $66(m)$ by $20.5(m)$ industrial environment. Once an experiment is started, the mobile nodes start moving on that defined path. This way, a number of experimental runs can be conducted using the same path to also gather statistical results. By using this testbed, we are able to run HuMOR on top of its devices and use the mobile nodes as mobile STA in order to create mobility scenarios. This way, we can validate handover algorithms created in the framework and evaluate them over a number of experimental results. Also, by evaluating multiple handover algorithms, we are able to create a comprehensive comparison study of different handover algorithms.

E. Monitored QoS parameters

To evaluate the handover algorithms, a number of QoS parameters are monitored and logged during experimentation in HuMOR. These logs can be post-processed to obtain results. To check how the handover algorithms influence QoS, HuMOR monitors and logs: throughput, latency, jitter and packet loss between the AP and STA.

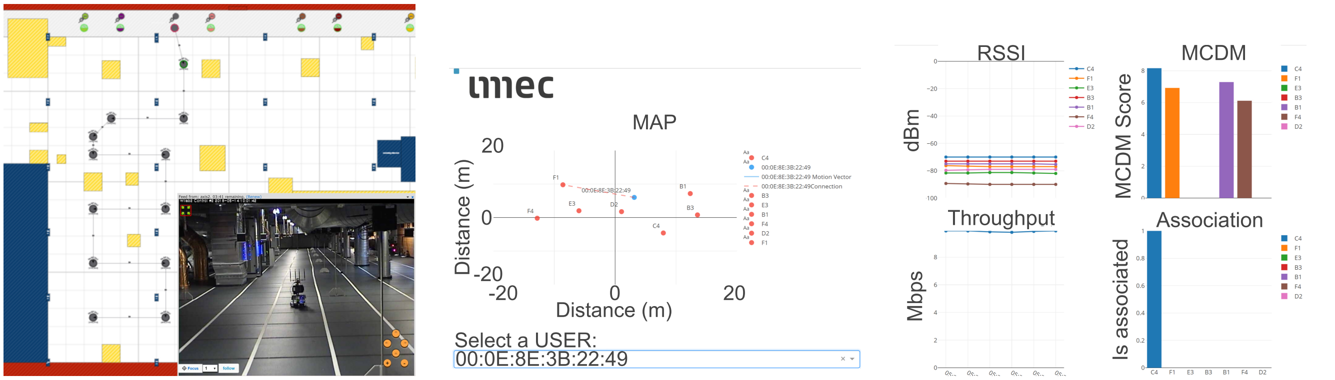
F. Graphical User Interface

Once an experiment is started, besides logging the monitored QoS parameters for post-experiment analysis, it can be of interest to also monitor the experiment in real-time. First of all, the w-iLab-t testbed already has a web GUI, the robot dashboard shown on Figure 4a, which allows a user to set up the paths of the mobile STA. However, it also allows the user to monitor the movement of the mobile STA once the experiment has started. There are two options. One is to view the 2D map of the testbed and follow the movement of the nodes there, and the other is to activate the camera system in the testbed to have a live view.

A GUI was created to accompany HuMOR, [35]. It retrieves metrics from the MDB and consists of two parts, a map shown on Figure 4b and a set of graphs shown on Figure 4c. The map is a 2D representation of the nodes in the environment. However, this map, as opposed to the map from the w-iLab-t robot dashboard, is created from the metrics available in the MDB of HuMOR alone. The map shows the location of the APs, the locations of the STAs, each STA's motion vector and its predicted future location, and to which AP each STA is associated to at the moment.

From the GUI, one can select a particular STA and view the second part of the GUI which are the graphs. There are a number of available graphs with different metric and parameters shown. One can check the RSSI value between a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2948883, IEEE Transactions on Network and Service Management



(a) Robot dashboard and camera feed (b) GUI map (c) GUI graphs

Fig. 4: Mock-up of the GUI and w-iLab.t2 robot dashboard

STA and all the APs, the throughput used by the STA, the AP that the station is currently associated to, as well as other specific parameters of the handover algorithms, which will be explained in the following sections. Using this GUI, a user can observe and monitor the experiment that is taking place in an intuitive way.

G. Existing Handover Algorithms

Based on our previous work on handover algorithms in IEEE 802.11 Wi-Fi networks [22], a number of handover algorithms are already part of HuMOR. Here, we will briefly introduce them and their main characteristics.

1) IEEE 802.11 Standard Handover Algorithm: First of all, without using much of HuMOR, the handover defined by the IEEE 802.11 standard is present. This handover algorithm is mainly here in order to provide a baseline against which other handover algorithms will be evaluated. A standard 802.11 Wi-Fi network is set up to which STAs connect. The wireless driver in the STAs are responsible for the handover, so this is a STA driven handover. This handover algorithm is triggered when the STA moves out of range of its associated AP, or when a metric such as the RSSI drops below a certain threshold. Once this happens, the AP selection part of the algorithm kicks in, and select a new AP to associate to based on a single metric, most often the RSSI. This handover is also not seamless and a complete disconnect happens during it, which means the QoS parameters are severely impacted.

2) MAX RSSI: Using HuMOR, MAX RSSI was created as a proactive handover algorithm that uses the LVAP handover mechanism. It uses one metric, the RSSI, in its decision-making process to handover STAs to whichever AP they have the highest RSSI value at any point in time, [22].

3) ADNA: ADNA was also created using HuMOR as a proactive, LVAP based handover algorithm. As opposed to the MAX RSSI, ADNA uses multiple metrics to decide which AP should each STA get handed over to. It uses RSSI, mobility information and AP load information in its decision-making process. A detailed description can be found in [22].

IV. Handover algorithm ABRAHAM

In this section we propose ABRAHAM, a mAchine learning Backed multimetRic proActive Handover AlgorithM. It uses the LVAP based handover mechanism in HuMOR, making it seamless and it is triggered in a proactive way. The goal of the algorithm is to optimize the load of the APs using predicted future metric information. ABRAHAM's AP selection algorithm is described in Algorithm 1.

A. AP Selection Algorithm

The input to the algorithm comes from the current state, $N(t)$, in HuMOR which gives it all the metrics present in HuMOR. The algorithm then assumes a clean-sheet state, where all APs are considered to be without any STA connected to them. However, for experimentation purposes, we will introduce an initial load, $INITIAL\_LOAD$, for all APs, which will be further explained in the next section for the experimental setup. ABRAHAM then goes over all the combinations of APs and STAs, and calculates a score for each (sta, ap) combination. This means, it calculates a score variable sc that shows what the preference is of associating STA sta to AP ap. The way this score is calculated is by inputting a number of metrics into the Weighted Sum Model Multi Decision Criteria Making (MCDM) algorithm, [4]. The metrics used and their weights are shown in Table III.

$b_{ap}(t+1)$ is the predicted future AP load of AP ap. This metric will be continuously updated with each iteration of the algorithm, until all the STAs have been assigned to an AP. Each time a STA is assigned to an AP, the future load of that AP will be updated by adding to it the throughput demand of the STA. As we want to optimize the load of the AP, this metric has the highest weight for the MCDM. $a_{ap,sta}(t)$ is a metric specifically added to this list to avoid the ping-pong effect. This metric is used, in situations when there is a similar score for two APs for one STA and one AP was the STA's previously connected AP, to prefer the AP that the STA was already connected to. This way an unnecessary handover is avoided. Because it's used as a tie breaker, we give this metric the least weight for the MCDM. The predicted future location $d_{ap,sta}(t+1)$ is calculated based on the motion vector in HuMOR. Finally, $rssi_{ap,sta}(t+1)$ is the predicted future RSSI between AP ap and STA sta obtained using machine learning, which will be explained in the next subsection. Because the

TABLE III: MCDM metrics and weights in ABRAHAM

Metric	Criteria	Weight
$rssi_{ap,sta}(t+1)$	MAX	0.2
$d_{ap,sta}(t+1)$	MIN	0.2
$b_{ap}(t+1)$	MIN	0.5
$a_{ap,sta}(t)$	MAX	0.1

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2948883, IEEE Transactions on Network and Service Management

RSSI and the location information are related, we give these two metrics an equal weight for the MCDM.

Once the MCDM scores are calculated, we check the $b_{ap}(t+1)$ and the $rssi_{ap,sta}(t)$ to increase the robustness. If the $b_{ap}(t+1)$ is lower than the average future traffic load on all the APs, we modify the MCDM score for that ap for all STAs, by a factor of 1.5. Also, for each sta, we check the $rssi_{ap,sta}(t)$ and discard the APs that have the RSSI value lower than 74% of the rest of the RSSI values to other APs. These modifiers were obtained after an empirical comparison of several possible configurations, after which these values showed the best results.

Once the scores are calculated and updated for each combination of STAs and APs, the algorithm first finds the AP ap_m with the least future AP load. At the beginning of the algorithm, this depends on the initial load of each AP, as was described earlier. We then find the STA which has the highest score for that AP, $sc_{ap_m,sta_n}(t)$. We add this tuple, (ap_m, sta_n), to the new network state, $N(t+1)$, and mark STA sta_n as processed. To take into account that STA sta_n will be associated to AP ap_m, we update the future AP load of AP ap_m. This is done by adding the throughput demand of STA sta_m to the future AP load of AP ap_m, like illustrated in Equation 1.

$$b_{ap_m}(t+1) = b_{ap_m}(t+1) + r_{sta_n}(t) \tag{1}$$

Once this is done, the algorithm is restarted, taking into account that STA sta_n is processed, so it will be ignored in the next iteration, and that AP ap_m now has a higher future AP load. Once all the STAs have been processed and assigned to

Algorithm 1 ABRAHAM

N(t)
$\forall ap \in AP, b_{ap}(t+1) = 0 + INITIAL\_LOAD$
while STAs to be processed **do**
  for all $sta \in STA$ **do**
    for all $ap \in ra_{sta}(t)$ **do**
      $sc_{ap,sta}(t) = MCDM[rssi_{ap,sta}(t+1);$
      $d_{ap,sta}(t+1); b_{ap}(t+1); a_{ap,sta}(t)]$
      if $b_{ap}(t+1) < AVG(\bigcup b_{ap}(t+1))$ **then**
        $sc_{ap,sta}(t) = 1.5 * sc_{ap,sta}(t)$
      end if
      if $rssi_{ap,sta}(t) < 74\% \bigcup rssi_{AP,sta}(t)$ **then**
        $sc_{ap,sta}(t) = 0$
      end if
    end for
  end for
  Find ap_m with MIN $b_{ap}(t+1)$
  Find sta_n with MAX $sc_{ap_m,sta_n}(t)$
  Add (ap_m, sta_n) to $N(t+1)$
  $b_{ap}(t+1) = b_{ap}(t+1) + r_{sta}(t)$
  $STA_n \rightarrow Processed$
end while
for all $(ap, sta) \in N(t+1)$ **do**
  $Handover(ap, sta)$
end for

an AP, the LVAP handover mechanism in HuMOR is triggered and the handovers take place.

B. The Recurrent Neural Network

The predicted future RSSI, $rssi_{ap,sta}(t+1)$, between AP ap and STA sta is obtained using a Recurrent Neural Network (RNN). Because of the sequence dependence of the RSSI in the time domain, the RNN was chosen as they recognize temporal patterns. We start by creating a sequential model of three layers. To avoid the vanishing/exploding gradient problem that classical RNNs have [36], we use a variation of the RNN called Long Short-Term Memory (LSTM), [37]. LSTM is ,therefore, our first layer with the number of units set to 50. The input amount for LSTM is equal to the number of APs, because the input will be the RSSI values between a STA and all the APs in the network. The second layer we use is dropout, [38,39]. Dropout is used to avoid the overfitting problem of LSTM, and is set to drop out 50% of the output values from the LSTM layer. The output of the RNN should be the predicted RSSI values between the STA and all of the APs. Therefore, the third layer condenses the output to the number of APs in the network. We use the Adam optimizer [40], as the method for efficient stochastic optimization. As the optimization metric, we use the Mean Squared Error (MSE).

V. Comparative study

We have used HuMOR to validate and evaluate ABRAHAM. Using the monitored QoS evaluation parameters, we can compare it to the IEEE 802.11 standard handover algorithm, as well as handover algorithms which we previously created, such as MAX RSSI and ADNA. We first start by describing the experimental setup. We explain which nodes will be used and what is their configuration. We show a layout of the testbed, as well as movement patterns of the mobile nodes. We then move on to explain the experiment scenarios and their results. First, we will use one mobile STA to validate our handover algorithm ABRAHAM. We will then repeat the experiment to get statistical results. And finally, we will use more than 1 mobile STA. At the end of this section, a discussion section can be found with the analysis of the results.

A. Experimental Setup

In order to validate and evaluate handover management algorithms, a couple of tasks have to be done in order to set up the experiment. First, one needs to allocate the nodes in the testbed that will be used. For our experiments, we will allocate 7 fixed wireless nodes that will serve as APs, 1 fixed server node that will run HuMOR, and finally, 4 mobile wireless nodes which will serve as STAs. We have chosen 4 mobile STAs due to the limited amount of such nodes available in the testbed for wireless experimentation. The APs are configured to operate in the 5GHz band on channel 36. They have a capacity of $C_{ap} = 25(Mbps)$. For experimental purposes, the APs in the testbed have signal attenuators on their antennas of $20(dB)$, in order to emulate longer distances between nodes. According to the Free Space Path Loss (FSPL) model, a decrease in signal strength of $20(dB)$ results in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2948883, IEEE Transactions on Network and Service Management

an approximate 10 times distance increase. Therefore, to emulate a normal human walk, the speed of the mobile STAs is set to the default testbed speed of $16(cm/s)$, which is approximately 10 times slower than a normal human walk. The STAs establish a $Q_{sta} = 10(Mbps)$ Transmission Control Protocol (TCP) connection and their movement patterns for experimental purposes are defined using the testbeds' robot dashboard.

To train the RNN model of ABRAHAM, we have obtained RSSI data through live experiments with HuMOR in the w-iLab-t testbed. The training data was collected for a period of 8 hours, where mobile STAs moved in a random pattern, to avoid overfitting. The input for the RNN were the RSSI values from the 7 APs, while the output were the predicted RSSI values to those APs 30 seconds into the future. 75% of the gathered data was used to train the model, while 25% was used as testing data. The MSE for the testing data was $0.7(dBm)$. To evaluate the RNN model even further, we have conducted 6 experiments with 7 APs where one mobile STA moved in random patterns for each experiment. The MSE values for those experiments are shown in Table IV. These results show that the RNN model was good enough for its purpose in ABRAHAM. This model was saved and used to predict the future RSSI values for the following scenarios.

TABLE IV: MSE values for RNN model

Experiment	1	2	3	4	5	6
MSE (dBm)	2.30	2.31	2.02	2.67	2.45	1.27

1) Scenario 1 - Evolution over time: For the first scenario, we use 1 STA that moves in a Z pattern and 7 APs. Their layout, and the movement path of the STA is shown on Figure 5. The same figure also shows the initial traffic load in $(Mbps)$ on the APs, which is created to simulate more traffic and more load on particular APs.

The objective of this scenario is to validate the handover algorithm. First, the movement path has the mobile STA move from AP B1, pass by D2 and arrive near F1. The objective is to see whether the algorithm can predict the future location of the STA to be near F1 and avoid handing over the STA to AP D2. Next, the mobile STA moves from near AP F1, passes by D2 and E3, and arrives near APs B3 and C4. The objective is to confirm that the predicted future location will again skip handing over the STA to D2 and E3, but this time it will have two AP to choose from, B3 and C4. Here is where the predicted future load of the AP will come in hand, as AP B3 is already overloaded. So, the objective is to see whether the algorithm will avoid handing over the STA to B3, and choose C4 instead. Finally, the STA moves to AP F4 to end the experimental run.

2) Scenario 2 - Multiple scenario 1 runs: Because scenario 1 shows the results of a single experimental run, scenario 2 objective is to evaluate the results and conclusions of scenario 1, by repeating the scenario 1 experiment 10 times and statistically analyzing the results.

3) Scenario 3 - Multiple STAs: We use scenario 3 to involve more STAs, keeping the same number and layout of APs, as well as their initial traffic loads as described in scenario 1.

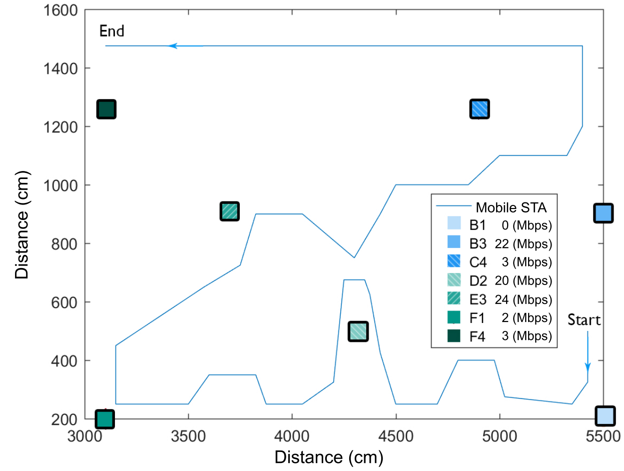


Fig. 5: Layout of APs and mobile STA movement path

In these experiments, the STAs move in random individual patterns. However, we run the same patterns for 10 times as well, with the objective of getting statistical results to evaluate the handover algorithm when multiple STAs are present.

Once the setup is complete, we can start HuMOR and using its handover management block we can validate and evaluate different handover algorithms. On top of the existing ones, such as the IEEE 802.11 standard, ADNA and MAX RSSI, we have added the newly created ABRAHAM in the handover database. From HuMOR we can run experiments with each one of these handover algorithms and obtain the monitored QoS evaluation parameters, which will be presented, analyzed and compared between handover algorithms.

B. Results

1) Scenario 1 - Evolution over time results: Here we present the results of a single mobile STA moving along the movement path defined according to Figure 5. We analyze the handovers that occur, the number of those handovers, as well as the QoS parameters along the movement path for the duration of the experiment.

Figure 6 shows the connectivity and handover graphs for all 4 handover algorithms. The connectivity graphs show to which AP the STA was connected at each point of the movement path, which is also shown by the handover graphs but in the time domain, so from 0% to 100% of the time of the experiment. From these graphs, we can see that ADNA had the least number of handovers, 3, ABRAHAM had 5, the IEEE 802.11 standard had 6, and MAX RSSI had the most, 9. The handover graphs also show that MAX RSSI, ADNA and ABRAHAM, which utilize the LVAP seamless handover mechanism, have a continuous connection, without any disconnects happening when the handover occurs. This is not the case for the IEEE 802.11 handover algorithm, which experiences disconnects with every handover that occurs. This is shown on Figure 6 as gaps in the time domain, which are pointed out by arrows for clarity.

Analyzing the connectivity graphs, we can check whether the objective of this scenario is met. When the STA was moving from AP B1 towards F1, it passed by AP D2. The IEEE 802.11 standard and MAX RSSI handover algorithms at some point hand over the STA to AP D2. On the other hand, ADNA and ABRAHAM, taking into account the predicted

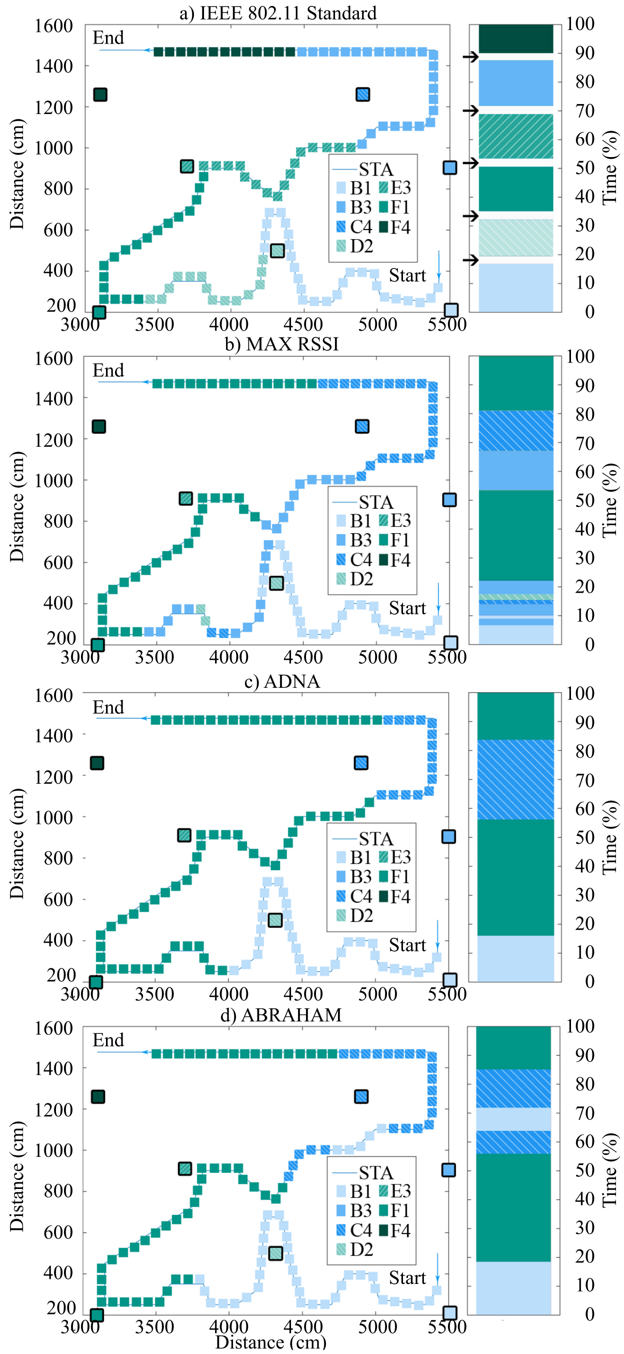


Fig. 6: Connectivity and handover graphs

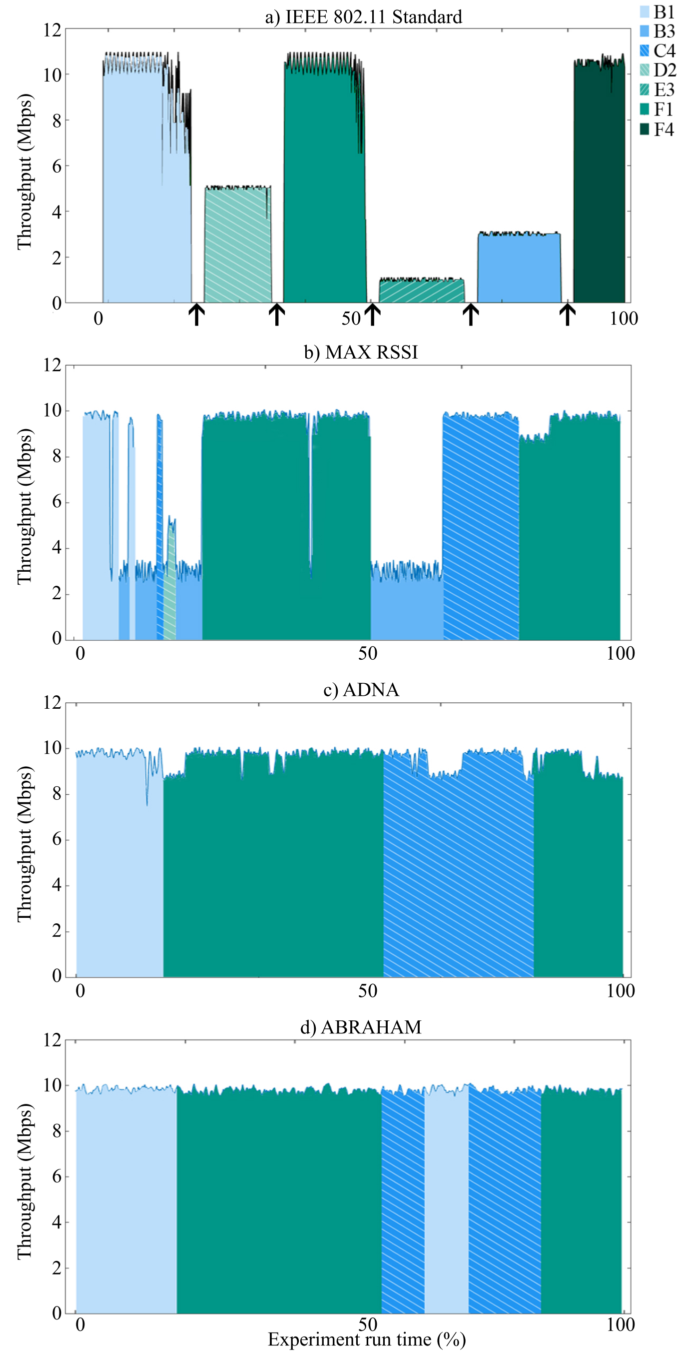


Fig. 7: Throughput graphs

future location, do not hand over the STA to AP D2, but immediately hand over the STA to AP F1. Next, the STA moved from AP F1, passed by APs E3 and D2, and arrived near APs B3 and C4. Again, ADNA and ABRAHAM avoided the handover to E3 and D2. Also, both of the algorithms were able to detect the overloaded B3 AP, and avoided handing over the STA to it. However, ADNA and ABRAHAM did behave differently at this point. ADNA just handed over the STA to C4, while ABRAHAM did a couple of handovers between C4 and B1, which was not the case for the IEEE 802.11 standard and MAX RSSI handover algorithms.

Figure 7 shows the throughput of the mobile STA over time for all 4 algorithms in $(Mbps)$. Due to the disconnects that happen with the IEEE 802.11 standard handover algorithm, we can clearly see the throughput dropping to $0(Mbps)$ during a handover. As with the handover graph, this has been pointed out by arrows on Figure 7. What can also be seen is that the throughput of the STA starts deteriorating when approaching the time to trigger a handover. On the other hand, the handover algorithms that use the LVAP seamless handover mechanism have continuous connectivity without drop in throughput to $0(Mbps)$. However, MAX RSSI can hand over the STA to an AP that is already overloaded with traffic, like APs D2, E3 or B3, which can cause a drop in throughput. This is also seen with the IEEE 802.11 standard handover algorithm. This does not happen with ADNA and ABRAHAM, so there are no high drops in throughput with these two algorithms. However, we can see that ADNA did experience slight throughput drops

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2948883, IEEE Transactions on Network and Service Management

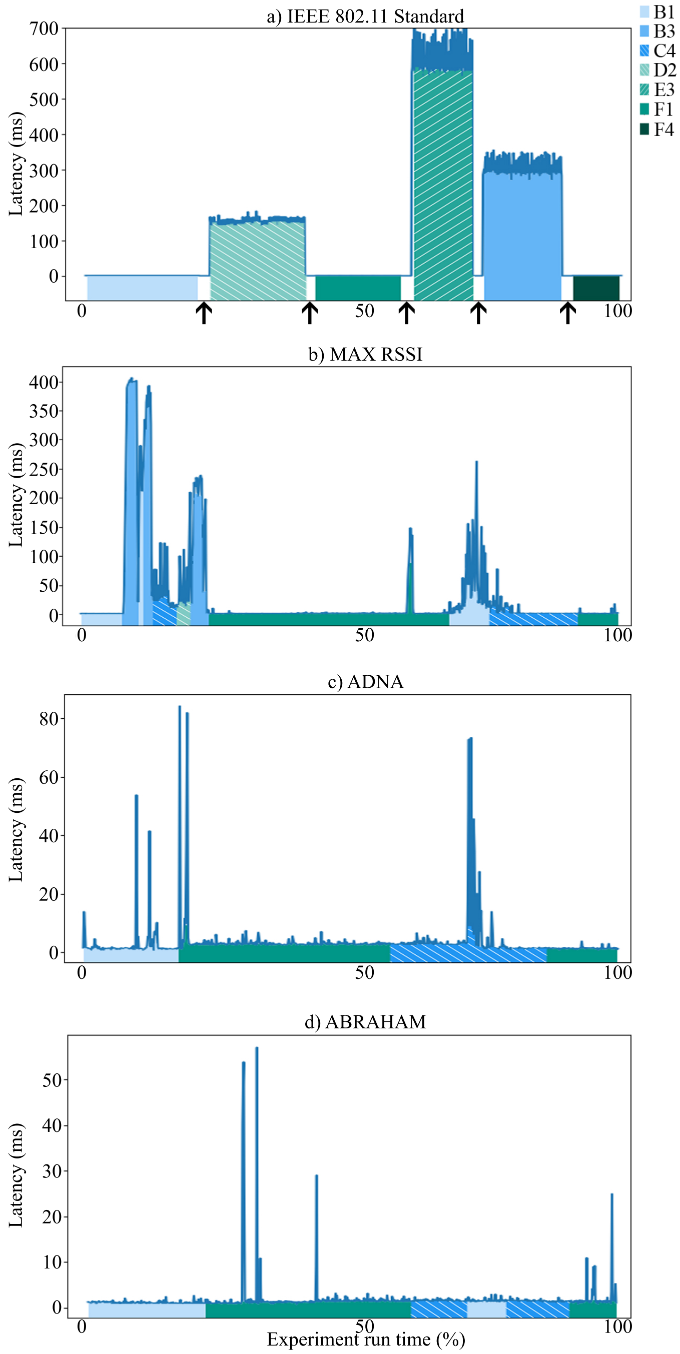


Fig. 8: Latency graphs

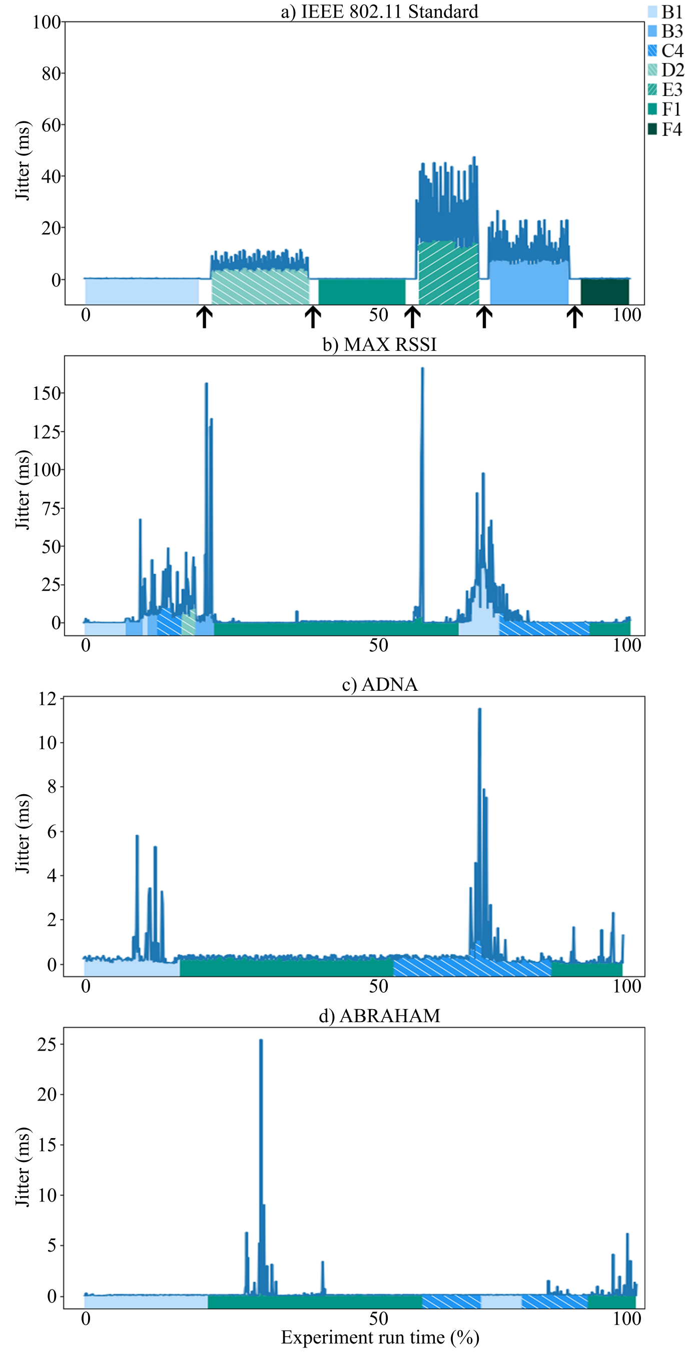


Fig. 9: Jitter graphs

when the single mobile STA was connected to AP C4.

Figure 8 shows the latency in the communication between the mobile STA and the APs that it was connected to. As can be seen for the IEEE 802.11 standard handover algorithm the latency rises as the STA gets handed over to an AP which is overloaded. The worst being around $600(ms)$ when connected to AP D3 which has only $1Mbps$ of available bandwidth. The same can be seen with the MAX RSSI algorithm. ADNA and ABRAHAM experience only slight rises to the latency. The most notable one being for ADNA around $70(ms)$ while the STA was connected to AP C4. This corresponds to the time that it experienced a slight drop in throughput as well.

Figure 9 shows jitter between the mobile STA and APs. From the graphs we can see that MAX RSSI experiences some high jitter effects, especially when the STA was handed over multiple times between AP of different loads at the beginning of the experiment. The IEEE 802.11 standard handover algorithm showed higher jitter values when the STA was handed over to higher loaded APs, and we see disconnects, pointed out by arrows on Figure 9, which interrupted the communication completely. In contrast, ADNA and ABRAHAM didn't experience severe jitter values. The only notable jitter rise was with ADNA when the STA was connected to C4.

Finally, Figure 10 shows the packet loss in the communication between the mobile STA and the APs. Here, we can see how with the IEEE 802.11 standard handover algorithm, due to it waiting until the RSSI drops below a certain threshold to trigger a handover, the RSSI becomes so low that the communication experiences packet loss. Also, we see the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2948883, IEEE Transactions on Network and Service Management

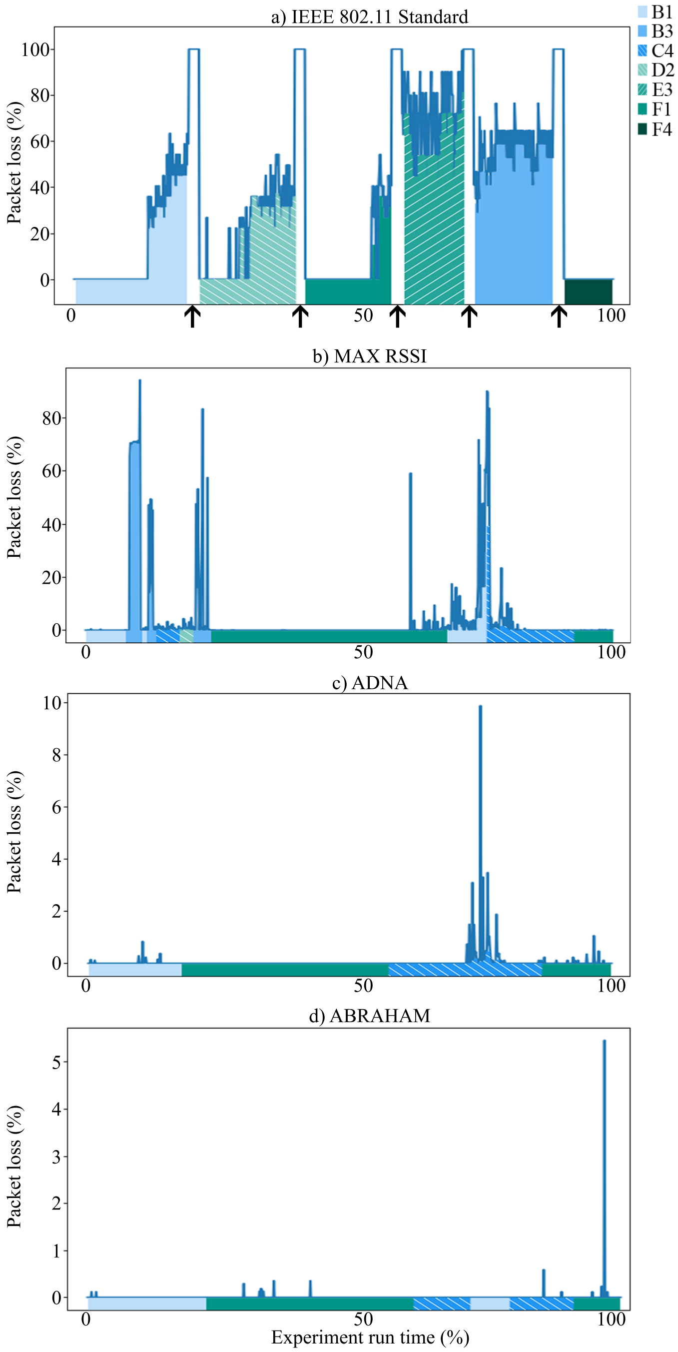


Fig. 10: Packet loss graphs

100% packet loss during the handover process, pointed out with arrows on Figure 10. MAX RSSI on the other hand does not experience 100% packet loss, but it does get high when the AP the STA is connected to is overloaded. ADNA and ABRAHAM do not experience high packet loss. ADNA does experience around 10% of traffic loss when connected to C4 due to low RSSI, while this is not the case with ABRAHAM.

2) Scenario 2 - Multiple scenario 1 runs results: In this subsection, we present the statistical results of running scenario 1 for 10 times to evaluate the results obtained in scenario 1. Because latency, jitter and packet loss are only interesting in the time domain, we focus on the throughput and the number of handovers. Figure 11 shows the average throughput of all 4 algorithms, as well as the 95% confidence intervals which are shown in numbers on the bars. ABRAHAM had the highest

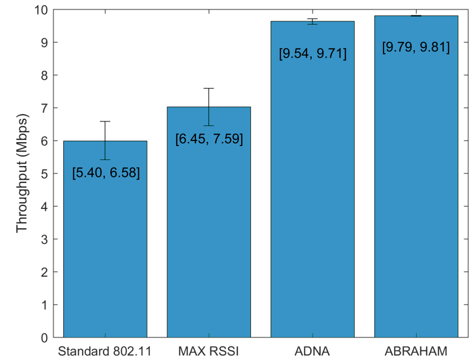


Fig. 11: Average throughput for single mobile STA

average throughput of $9.8(Mbps)$, while the IEEE 802.11 standard handover algorithm achieved only $5.99(Mbps)$.

Table V shows the average number of handovers for all 4 algorithms. ADNA has the least average number of handovers 3.33, while MAX RSSI has the most 7.83.

TABLE V: Single STAs Average number of handovers

Algorithm	IEEE 802.11	MAX RSSI	ADNA	ABRAHAM
	5.67	7.83	3.33	5.16

3) Scenario 3 - Multiple STAs results: Finally, we evaluate the 4 handover algorithms by using multiple mobile STA for the experiments. Specifically, we use 4 mobile STA which move in random patterns. We then again repeat the experiments to get statistical information on the number of handovers and the average throughput.

Figure 12 shows the overall average throughput of the STAs across 10 experiments, as well as the 95% confidence intervals which are shown in numbers on the bars. As can be seen, ABRAHAM has the highest overall average throughput of $8.35(Mbps)$, while the IEEE 802.11 standard handover algorithm had $3.5(Mbps)$

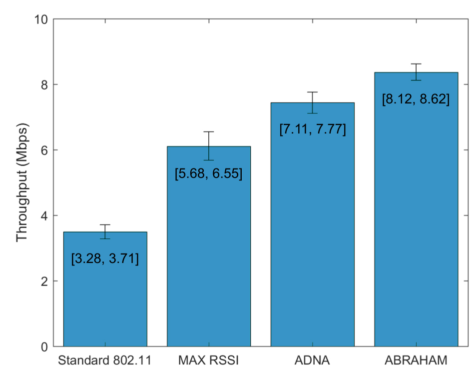


Fig. 12: Average throughput of multiple mobile STAs

Table VI shows the average number of handovers that occur in the experiment per algorithm. As can be see, ADNA has the lowest average number of 4.33. ABRAHAM is just slightly above at 4.67, while MAX RSSI has the highest of 7.67.

TABLE VI: Multiple STAs Average number of handovers

Algorithm	IEEE 802.11	MAX RSSI	ADNA	ABRAHAM
	6.63	7.67	4.33	4.67

VI. Discussion

A. Results discussion

Scenario 1 gives insight into the time domain, by analyzing the QoS parameters (throughput, latency, jitter and packet loss) and the number of handovers that occur. From the results of the IEEE 802.11 standard handover algorithm, we can clearly see no data transfer and 100% packet loss during the handover process because of the disconnects that happen. The reactive handover trigger waits for the RSSI to drop below a certain threshold, at which point the other QoS parameters already start to deteriorate. Another weakness is the AP selection algorithm which relies only on the RSSI, so it can hand over a STA to an AP that is already overloaded. We can see this occurring when the STA gets handed over to APs D2, E3 or B3. This results in the drop of throughput, rise in latency and packet loss, and a more variable jitter. The results for the IEEE 802.11 standard handover algorithm show that it does not have any notion of preserving the QoS parameters.

On the other side, the handover algorithms created with HuMOR are seamless with no disconnects in communication thanks to the LVAP handover mechanism. During a handover, there is no significant drop in throughput or increase in latency, jitter or packet loss. MAX RSSI, ADNA and ABRAHAM are all proactive handover algorithms, meaning they don't wait for the QoS parameters to deteriorate before triggering a handover. They all continuously monitor the network and decide when to handover a STA. MAX RSSI only monitors the RSSI which results in a large number of handovers occurring, due to the noisy nature of the RSSI. Even though it is seamless, it only relies on the RSSI, which means it also hands over the STA to AP which are overloaded. This results in the QoS parameters deteriorating, similar the IEEE 802.11 standard handover.

ADNA and ABRAHAM, on the other hand, preserve the QoS parameters well during experimentation. Both of them have a global overview of the network thanks to HuMOR, which means they can create global optimizations of the network, instead of doing per-device optimizations. They rely on multiple metrics when deciding to which AP a STA should get handed over to, including the RSSI, location and mobility information, as well as the AP load. Because of the AP load information, both algorithms avoid handing over the STAs to AP which are overloaded. Therefore, the throughput does not experience high drops. However, we can see that ADNA did experience slight throughput drops when the single mobile STA was connected to AP C4. This also resulted in an increase of the latency and packet loss. ABRAHAM did not experience this as it did not keep the single STA on C4 as long as ADNA. At the point in time that ADNA experienced a drop in throughput on C4, ABRAHAM handed over the STA to B1. This resulted in the throughput to be preserved, and no packet loss to be experienced. This was due to ABRAHAM using the LSTM predicted future RSSI, for which the MSE during the experimentation was $2.26(dBm)$. The drop in throughput with ADNA was experienced due to a low RSSI, however, the two other metrics, AP load and future location, indicated that C4 was the best candidate. ABRAHAM was able to predict the drop in RSSI, and temporarily handed over the STA to B1.

The average latency of the STA with ABRAHAM was $1.71(ms)$, average jitter was $0.2(ms)$, average packet loss was $0.01(\%)$. Even though the average values do not say much by themselves, it is interesting to see what % of time will these values be higher with the other algorithms, meaning what % of time is ABRAHAM better than the rest of the algorithms. These results can be seen in Table VII. For 43.75% of the time of the experiment, the latency of the IEEE 802.11 standard handover algorithm was higher than ABRAHAMs average latency. This means that on average, ABRAHAMs' latency was lower than the latency of the IEEE 802.11 standard handover algorithm for 43.75% of the time. Similar results can be concluded by comparing ABRAHAM to the rest of the algorithms, as well as the rest of the QoS parameters.

TABLE VII: Percentage of time values were higher than ABRAHAMs' average

	IEEE 802.11	MAX RSSI	ADNA	ABRAHAM
Latency	43.75(%)	37.13(%)	16(%)	11.88(%)
Jitter	45.97(%)	59.12(%)	16.79(%)	8.23(%)
Packet loss	57.12(%)	28.77(%)	8.24(%)	1.84(%)

In scenario 2, we evaluate the conclusions of scenario 1 over 10 experimental runs. ABRAHAM had a 64% higher average throughput than the IEEE 802.11 standard, even higher than ADNA. ABRAHAMs' 95% confidence interval does not overlap with the one from ADNA. Also, to compare ABRAHAM and ADNA, and see if the results show statistical improvement of ABRAHAM over ADNA, we conducted the one-way Analysis of variance (ANOVA). Setting the alpha value to 0.05, the result of the ANOVA test is $F(1,18) = 15.205, p = 0.0011$, with F critical of 4.414. The ANOVA, therefore, concludes that with 95% probability there are statistically significant differences between the means of ABRAHAM and ADNA.

Scenario 3, with multiple STAs involved, ABRAHAM had a 139% higher overall average throughput than the IEEE 802.11 standard handover. ABRAHAM even outperformed ADNA in the overall average throughput by 11%, while their 95% confidence intervals don't overlap, and the result of the ANOVA with alpha set to 0.05 was $F(1,18) = 19.618, p = 0.0003$, with F critical of 4.414. This proves that with 95% probability there are significant statistical differences between ABRAHAM and ADNA.

ADNA had the least amount of handover in both scenario 2 and 3, however with multiple STAs ADNA only had 8% lower average number of handovers than ABRAHAM. Looking at the QoS parameters, because the LVAP handover process does not impact the QoS, having more handovers in return for a higher throughput is justified.

B. Deployment discussion

HuMOR runs on general-purpose hardware. The APs run on top of OpenWRT, [41], which can be deployed on a wide range of commercially available APs. So, deploying HuMOR in other testbeds should not present a challenge. The prerequisite to use ABRAHAM is the a priori knowledge of the location of the APs. One such environment could be enterprise networks, where the locations of APs are known. The HuMOR framework concept can also be deployed in a more production-oriented environment, such as Long Term Evolution (LTE) mobile networks. LTE has a similar control management interface where statistics from the User Equipments (UEs) and network side are gathered. The handover is executed from the network side and the locations of base stations are a priori known. The difference is in the handover mechanism, for which HuMOR uses the LVAP one. The localization module of HuMOR can also be updated so that it does not require any a priori AP location knowledge.

Handover algorithms created with HuMOR benefit from the use of the LVAP seamless handover, meaning the QoS do not degrade during a handover. When creating a new handover algorithm, one can use a number of exposed metrics and focus on the AP selection process and the trigger. This handover algorithm can be stored in the HADB of HuMOR, and then validated and evaluated in the testbed.

ABRAHAM can be implemented without HuMOR, however, its performance would depend on the framework it is deployed in. Even if it had access to the same metrics as in HuMOR, it would lack the support of the LVAP handover mechanism. This would result in the algorithm to be able to spread the load across the APs and take into account the QoS parameters, but the QoS would degrade during a handover. ABRAHAM also requires a training period for its RNN model. In a real deployment scenario, such as the ones discussed above, one could use ADNA as the handover algorithm. During its use, enough RSSI data can be gathered to train the RNN specific to the deployment environment. Once the RNN model is trained and tested, ABRAHAM can be activated to replace ADNA and provide the added benefits demonstrated in this paper.

VII. Conclusion

Controlling the handover is crucial for having proper mobility management in IEEE 802.11 Wi-Fi networks. To overcome the challenges of handover algorithms, we first proposed an enabling framework called HuMOR. HuMOR is an SDN modular handover management framework capable of creating, validating and evaluation handover algorithms that are centralized and proactive. It utilizes the LVAP handover mechanism to make them seamless and transparent to the STAs, and is capable of gathering metrics such as RSSI, location, mobility and AP load. Relying on these capabilities of HuMOR, the second contribution of this paper is a proactive handover algorithm called, ABRAHAM, a mAchine learning Backed multimetRic Handover AlgorithM. Using the metrics in HuMOR it predicts the future location of STAs, the future predicted AP load and using LSTM predicts the future RSSI. Its goal is to optimize the load on the AP by handing over STAs to APs in a way that will preserve the QoS. In a comparative study, we validate and evaluate ABRAHAM by comparing it to other handover algorithms in literature. We show that with multiple mobile STAs, ABRAHAM had a 139% higher overall throughput compared to the IEEE 802.11 standard handover algorithm. Compared to ADNA, this was higher for 11% and the ANOVA showed significant statistical difference between ABRAHAM and ADNA.

Acknowledgment

The research leading to these results received funding from the VLAIO SBO SAMURAI project, as well as FlexNet, a Celtic-Next project.

References

[1] L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, and J. Almodovar, "Building an SDN enterprise WLAN based on virtual APs," *IEEE Communications Letters*, vol. 21, no. 2, pp. 374–377, 2017, doi: http://dx.doi.org/10.1109/LCOMM.2016.2623602.

[2] A. Zubow, S. Zehl, and A. Wolisz, "BIGAP — Seamless handover in high performance enterprise IEEE 802.11 networks," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, pp. 445–453, doi: http://dx.doi.org/10.1109/NOMS.2016.7502842.

[3] J. Márquez-Barja, C. T. Calafate, J.-C. Cano, and P. Manzoni, "An overview of vertical handover techniques: Algorithms, protocols and tools," *Computer Communications*, vol. 34, no. 8, pp. 985–997, 2011, doi: http://dx.doi.org/10.1016/j.comcom.2010.11.010.

[4] J. Marquez-Barja, C. T. Calafate, J.-C. Cano, and P. Manzoni, "Performance trade-offs of a IEEE 802.21-based vertical handover decision algorithm under different network conditions," in *2011 IEEE 10th International Symposium on Network Computing and Applications*. IEEE, 2011, doi: http://dx.doi.org/10.1109/nca.2011.50.

[5] R. Murty, J. Padhye, A. Wolman, and M. Welsh, "Dyson: An Architecture for Extensible Wireless LANs," *Proc. of the 2010 USENIX Annual Technical Conference, (ATC)*, p. 14, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855840.1855855

[6] 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 Journal on Wireless Communications and Networking*, vol. 2017, no. 1, 2017, doi: http://dx.doi.org/10.1186/s13638-017-0856-9.

[7] N. Kiran, Y. Changchuan, and Z. Akram, "AP load balance based handover in software defined WiFi systems," in *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*. IEEE, 2016, doi: http://dx.doi.org/10.1109/icnidc.2016.7974526.

[8] P. Dely, A. Kassler, L. Chow, N. Bambos, N. Bayer, H. Einsiedler, C. Peylo, D. Mellado, and M. Sanchez, "A software-defined networking approach for handover management with real-time video in WLANs," *Journal of Modern Transportation*, vol. 21, no. 1, pp. 58–65, 2013, doi: http://dx.doi.org/10.1007/s40534-013-0007-x.

[9] A. Sen and K. M. Sivalingam, "An SDN framework for seamless mobility in enterprise WLANs," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2015, doi: http://dx.doi.org/10.1109/pimrc.2015.7343624.

[10] J. Chen, B. Liu, H. Zhou, Q. Yu, L. Gui, and X. S. Shen, "QoS-driven efficient client association in high-density software-defined WLAN," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7372–7383, 2017, doi: http://dx.doi.org/10.1109/tvt.2017.2668066.

[11] D. Sajjadi, M. Tanha, and J. Pan, "Meta-heuristic solution for dynamic association control in virtualized multi-rate WLANs," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, 2016, doi: http://dx.doi.org/10.1109/lcn.2016.53.

[12] W. Wong, A. Thakur, and S.-H. G. Chan, "An approximation algorithm for AP association under user migration cost constraint," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, doi: http://dx.doi.org/10.1109/infocom.2016.7524397.

[13] M. Kawada, M. Tamai, and K. Yasumoto, "A trigger-based dynamic load balancing method for WLANs using virtualized network interfaces," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2013, doi: http://dx.doi.org/10.1109/wcnc.2013.6554716.

[14] I. Broustis, K. Papagiannaki, S. V. Krishnamurthy, M. Faloutsos, and V. P. Mhatre, "Measurement-driven guidelines for 802.11 WLAN design," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 722–735, 2010, doi: http://dx.doi.org/10.1109/tnet.2009.2031971.

[15] N. Ahmed and S. Keshav, "SMARTA," in *Proceedings of the 2006 ACM CoNEXT conference on - CoNEXT '06*. ACM Press, 2006, doi: http://dx.doi.org/10.1145/1368436.1368449.

[16] S. Bayhan and A. Zubow, "Optimal mapping of stations to access points in enterprise wireless local area networks," in *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM '17*. ACM Press, 2017, doi: http://dx.doi.org/10.1145/3127540.3127556.

[17] O. Aldhaibani, F. Bouhafs, M. Makay, and A. Raschella, "An SDN-based architecture for smart handover to improve QoE in IEEE 802.11 WLANs," in *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*. IEEE, 2018, doi: http://dx.doi.org/10.1109/waina.2018.00100.

[18] B. Ng, A. Deng, Y. Qu, and W. K. G. Seah, "Changeover prediction model for improving handover support in campus area WLAN," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2016, doi: http://dx.doi.org/10.1109/noms.2016.7502821.

[19] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, "Designing high performance enterprise wi-fi networks," in *NSDI'08 Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1387589.1387595

[20] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANS with odin," in *Proceedings of the first workshop on Hot topics in software defined networks - HotSDN '12*. ACM Press, 2012, doi: http://dx.doi.org/10.1145/2342441.2342465.

[21] M. Sun, , and H. Qian, "Handover management scheme in SDN-based wireless LAN," *Journal of Communications*, 2016, doi: http://dx.doi.org/10.12720/jcm.11.3.282-289.

[22] E. Zeljkovic, J. M. Marquez-Barja, A. Kassler, R. Riggio, and S. Latre, "Proactive access point driven handovers in ieee 802.11 networks," in *2018 14th International Conference on Network and Service Management (CNSM)*. IEEE, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8584977

[23] G. Yi and S. Lee, "Fully distributed handover based on SDN in heterogeneous wireless networks," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication - ICUIMC '14*. ACM Press, 2014, doi: http://dx.doi.org/10.1145/2557977.2558047.

[24] M. E. Berezin, F. Rousseau, and A. Duda, "Multichannel virtual access points for seamless handoffs in IEEE 802.11 wireless networks," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. IEEE, 2011, doi: http://dx.doi.org/10.1109/vetecs.2011.5956552.

[25] Z. Ali, N. Baldo, J. Mangues-Bafalluy, and L. Giupponi, "Machine learning based handover management for improved qoe in lte," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 794–798.

[26] L. Qiang, J. Li, and C. Huang, "A software-defined network based vertical handoff scheme for heterogeneous wireless networks," in *2014 IEEE Global Communications Conference*. IEEE, 2014, doi: http://dx.doi.org/10.1109/glocom.2014.7037545.

[27] L. Qiang, J. Li, Y. Ji, and C. Huang, "A novel software-defined networking approach for vertical handoff in heterogeneous wireless networks," *Wireless Communications and Mobile Computing*, vol. 16, no. 15, pp. 2374–2389, 2016, doi: http://dx.doi.org/10.1002/wcm.2690.

[28] S. Mumtaz, V. Monteiro, J. Rodriguez, and C. Politis, "Positioning for performance enhancement in shared LTE and HSPA networks," in *2012 International Conference on Telecommunications and Multimedia (TEMU)*. IEEE, 2012, doi: http://dx.doi.org/10.1109/temu.2012.6294711.

[29] T. Ahmed, K. Kyamakya, and M. Ludwig, "A context-aware vertical handover decision algorithm for multimode mobile terminals and its performance," 2006. [Online]. Available: https://bit.ly/2G92xEp

[30] T. Kim, S. wook Han, and Y. Han, "A QoS-aware vertical handoff algorithm based on service history information," *IEEE Communications Letters*, vol. 14, no. 6, pp. 527–529, 2010, doi: http://dx.doi.org/10.1109/lcomm.2010.06.100176.

[31] R. Riggio, "The EmPOWER mobile network operating system," in *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization - WiNTECH '16*. ACM Press, 2016, doi: http://dx.doi.org/10.1145/2980159.2980173.

[32] R. Riggio, K. M. Gomez, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and M. K. Marina, "Programming software-defined wireless networks," in *10th International Conference on Network and Service Management (CNSM) and Workshop*. IEEE, 2014, doi: http://dx.doi.org/10.1109/cnsm.2014.7014148.

[33] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, "Zero-configuration, robust indoor localization: Theory and experimentation," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*. IEEE, 2006, doi: http://dx.doi.org/10.1109/infocom.2006.223.

[34] imec - IDLab w-iLab.t Testbed. [Online]. Available: http://doc.ilabt.imec.be/ilabt-documentation/

[35] E. Zeljkovic, J. M. Marquez-Barja, and S. Latré, "A large-scale demonstration of SDN-based handover management in IEEE 802.11 networks," in *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos - SIGCOMM '18*. ACM Press, 2018, doi: http://dx.doi.org/10.1145/3234200.3234223.

[36] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994, doi: http://dx.doi.org/10.1109/72.279181.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[38] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014. [Online]. Available: http://arxiv.org/abs/1409.2329

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[41] OpenWRT Project. [Online]. Available: https://openwrt.org/

Ensar Zeljković is currently pursuing a PhD in the Computer sciences. He is a PhD student at the University of Antwerp and the imec research center in Belgium. He holds a Master's degree in Electrical engineering, department of telecommunications from the Faculty of Electrical engineering, University of Sarajevo, Bosnia & Herzegovina. He is currently doing research on virtual slicing based management and SDN in heterogeneous wireless networks, as well as research in the domain of resource management in 5G mobile networks. His research resulted in 9 articles published in international peer-reviewed journals and conference proceedings, as well as in 2 submitted patent applications.

Nina Slamnik-Kriještorac has recently started her PhD adventure in the field of Applied Engineering Sciences, as a PhD Student at the University of Antwerp and the imec research center in Belgium. She obtained Master degree in telecommunications engineering at Faculty of Electrical Engineering, University of Sarajevo, Bosnia & Herzegovina, in July 2016. Her current research is mostly based on sharing models for next generation end-to-end network resources, based on enabling programmable networks for flexible smart cities.

Steven Latré is an associate professor at the University of Antwerp and director at the research centre imec, Belgium. He is leading the IDLab Antwerp research group (85+ members), which is performing applied and fundamental research in the area of communication networks and distributed intelligence. His personal research interests are in the domain of machine learning and its application to wireless network optimization. He received a Master of Science degree in computer science from Ghent University, Belgium and a Ph.D. in Computer Science Engineering from the same university with the title "Autonomic Quality of Experience Management of Multimedia Services". He is author or co-author of more than 100 papers published in international journals or in the proceedings of international conferences. He is the recipient of the IEEE COMSOC award for best PhD in network and service management 2012, the IEEE NOMS Young Professional award 2014 and is a member of the Young Academy Belgium.

Johann M. Marquez-Barja currently is an Associate Professor at University of Antwerpen, as well as adjunct researcher in IMEC. He is the Head of the Wireless Cluster at imec IDLab UAntwerp. He is involved in several European research projects, being Principal and Co-Principal Investigator for many projects. He is a Senior member of the IEEE Communications Society as well as the IEEE Education Society, where he participates in the Standards Committee and contributes to the IEEE P1876 - Networked Smart Learning Objects for Online Laboratories standards working group. His main research interests are: 5G advanced architectures; elastic and flexible future wireless networks and its integration and impact on optical networks; IoT clustering; softwarization and virtualization; provisioning and dynamic resource allocation towards dynamic networks. He is also interested in vehicular networks, mobility and handovering within smart cities. He received a personal grant from the EU under the prestigious Alban Programme to purse a PhD degree. Prof. Marquez-Barja has been given several keynotes and invited talks in different major events. He is serving as Editor and Guest editor for four International Journals, as well as participating in several Technical Programme and Organizing Committees for different worldwide conferences/congresses.