



# A User Prioritisation Algorithm for Horizontal Handover in Dense WLANs

Omar A. Aldhaibani<sup>1</sup> · Alessandro Raschellà<sup>1</sup> · Ghulam Mohi-Ud-Din<sup>1</sup> · Michael Mackay<sup>1</sup>

Received: 2 March 2021 / Revised: 20 August 2021 / Accepted: 3 December 2021  
© The Author(s) 2021

## Abstract

This paper proposes an algorithm that enhances horizontal handover (HO) in dense wireless local area networks (WLANs), which is implemented in a software-defined wireless networking (SDWN)-based architecture. The algorithm considers the concept of user prioritisation, classifying the WLAN stations (STAs) into two categories representing high and low priorities respectively, and always attempts to guarantee the best quality of experience (QoE) to the high priority users. The architecture that implements the algorithm leverages the flexibility, programmability, and centralised nature of SDWN to efficiently manage the HO process. Moreover, the paper presents a performance evaluation campaign that demonstrates significant achievements against a state-of-the-art solution in terms of the provided QoE, throughput and delay. Finally, we discuss the importance of considering user prioritisation in a HO algorithm for dense WLAN

s.

**Keywords** Wi-Fi networks · Dense WLAN · Horizontal handover · Software-defined wireless network · Quality of experience

## 1 Introduction

In a short period of time, the Internet has proved itself a powerful platform which has changed the way we do business, communicate with other people, and socialize in our day to day life. The advancement of the Internet has made the world a global village and for millions of people and it has become a source of knowledge and information at home, school, offices and other scenarios with users on the move, such as airports, shopping mall and train stations. A major means of edge connectivity with the Internet is either through a cable-based local area network (LAN) or now via Wireless LAN (WLANs). In recent years, there has been

a significant increase in the number and range of devices which use wireless technologies to connect with the Internet especially smartphones and tablets [1].

On the other hand, although the internet has many benefits it requires huge infrastructures to provide all of these services. For example, when many users want to connect to the Internet and take advantage of real-time services such as live streaming, online gaming, video on demand, or video conferencing, it poses a problem in maintaining high bandwidth and network capacity to fulfil their quality of experience (QoE) requirements.

For Wi-Fi, wireless access points (APs) are the main point of connectivity for portable devices and each AP has a finite level of radio resources which can be utilized by users. When the number of users exceeds the capacity of the AP they may start experiencing network congestion problems, mainly in dense public areas such as campuses, big exhibition halls or airports. This network congestion occurs as a result of insufficient bandwidth as the data traffic surpass network capacity. Moreover, due to packet collisions in the highly congested network, users suffer from serious performance degradation resulting in large delays, data loss and perhaps even dropping or blocking of connections.

---

✉ Omar A. Aldhaibani  
o.a.alldhaibani@ljmu.ac.uk

Alessandro Raschellà  
a.raschella@ljmu.ac.uk

Ghulam Mohi-Ud-Din  
G.MohiUddin@ljmu.ac.uk

Michael Mackay  
m.i.mackay@ljmu.ac.uk

<sup>1</sup> The School of Computer Science and Mathematics,  
Liverpool John Moores University (LJMU), Liverpool, UK

To overcome the problem of congestion in dense areas, the organization network administrators often deploy redundant APs in the network. In this way, if the number of users exceeds the capacity of a certain AP, these users can connect to the next nearest AP, which may be able to provide them with the requested services. Although the replication of APs provides a good way to bring the network out of the congested state, this replication will also increase the cost of the deployment. Moreover, the problem of interference arises when more APs are placed near each other and the coverage area of these APs starts to overlap, which again causes degradation of the bandwidth and the service received by the recipients.

Another challenge in wireless networks is the Handover (HO), which is the process of switching users from one AP to another. Traditional HOs rely on the strength of the AP signals and the users' movement but this is not always practical in a high-density area and may lead to suboptimal usage of the available spectrum. As such, HOs may also provide an approach to optimise WLANs through AP selection.

In this paper we aim to address the problem of Handovers in dense WLANs through efficient AP selection by proposing a solution which extends the Software Defined Wireless Network (SDWN) architecture presented in our previous work [2]. Moreover, our solution utilises the concept of user prioritisation to make a smart decision during the HO process by classifying users into two categories (i.e., *High* and *Low* priority user). This approach will always guarantee the best Quality of Experience (QoE) for the High priority users. Therefore, the aims of this paper are twofold. First, we propose a HO algorithm for dense Wi-Fi networks that allows users to change their connected AP even when they do not move. Secondly, we propose to manage user prioritisation by exploiting the capability of the SDWN during the HO process. Our simulation results show that addressing users' prioritization in a HO algorithm allows us to guarantee satisfactory performance results in terms of QoE, traffic delay, and throughput for High priority users in comparison with another relevant existing algorithm, which does not consider users' prioritisation.

The rest of the paper is organized as follows: Sect. 2 illustrates the works found in the state of the art addressing HOs, their limitations and our new contributions. In Sect. 3 we present the SDWN-based architecture that implements the proposed HO algorithm. Section 4 includes a detailed description of the HO algorithm based on *High* and *Low* users' priorities. In Sect. 5 we define the simulation model we implemented to assess the proposed algorithm and the analysis of the performance. Finally, we illustrate our conclusions and future work in Sect. 6.

## 2 State of the Art and New Contributions

The standard HO process in WLANs is shown in Fig. 1 where the shadowed areas represent the coverage of the APs. In the beginning, the user is connected to AP1 and receives the best received signal strength (RSS). When the user is in the overlapping area of both APs and starts moving away from AP1 towards AP2, the RSS of AP1 decreases and it affects the Quality of Service (QoS) and the QoE for the user. In this case, when the signal from AP1 drops below a threshold, the user starts receiving stronger signals from AP2 they are in its coverage area and the HO process takes place. Therefore, a standard HO depends on the RSS experienced by the Wi-Fi stations (STAs), which changes while STAs move.

All the HO processes addressed in the literature aim to achieve user satisfaction by trying to determine the most appropriate access network. Specific information gathered from the radio environment, such as RSS and signal-to-interference and noise ratio (SINR), are used to make the HO decision [3]. The decision step is critical during the HO process as it is responsible for deciding at what point to start and to which AP the connection should be made in the case of a HHO for Wi-Fi networks. Making a HO decision is a challenging task, and there are many approaches that have been proposed in the literature. Many works classify HO into the following five strategies: (1) RSS-based strategies [4]; (2) Decision function-based algorithm strategies, which include the sum of the weighted function of some parameters such as monetary cost, trust, preference, compatibility, load and capacity [5–10]; (3) QoS and User-centric algorithm strategies, which take into account different features like bandwidth, cellular cost, coverage area, SINR, and QoS requirements [3, 11–16]; (4) multiple attribute decision (MAD) strategies in which a selection is made from a limited number of candidate networks, depending on different criteria such as Multiple Attributes and Multiple Objectives [4, 17, 18]; and (5) Fuzzy logic and neural network-based algorithm strategies, which are characterised by the capability to monitor and analyse different parameters such as RSS, load and bandwidth in the

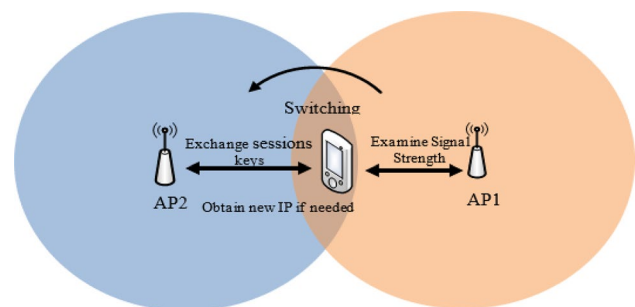


Fig. 1 HO in WLANs

case of both real-time and non-real-time applications [2, 19–22].

We believe that the existing HO methods found in the state of the art do not consider a number of aspects that can improve the performance of the users' connections. Specifically, the limitations of existing works that have motivated our research are the following:

- Works presented in [3–22] do not meet the requirements of STA applications in terms of both QoS and QoE, due to the lack of intelligence of current networks, especially considering the massive increase in mobile devices and the evolution of applications. Note that [15] and [16] address only the QoS requirements in terms of data bit rates of wireless users. However, QoE has recently become a common metric to measure user satisfaction and has to be considered when designing HO strategies in order to provide satisfactory services for the users [23]. QoE represents the measurement of network system performance as perceived by the user and will be explained in detail in Sect. 3.2.
- Works presented in [3–14] and [4, 17–22] are proposed with the assumption that the wireless network is small in size or implemented in a simple scenario with a very limited number of users. However, wireless networks now and going forward are expected to become even denser due to the increasing number of mobile devices such as laptops, smartphones, tablets, Internet of Things (IoT) devices as well as network providers.
- All the solutions [2–22] have not been designed taking into account the presence of users that might expect service delivery with higher priority.

The state of the art is summarized in Table 1, which includes the reference number, the strategy and the main limitations of the analysed works.

Hence, the consideration of High priority users is a key aspect of this paper. These users are defined in order to always allow them to achieve the best available service. Specifically, these users should always receive the guaranteed service, while Low priority users are more affected during high traffic periods. However, as we will explain in Section IV, the proposed solution will always attempt to optimise the services for Low priority users also.

For instance, in exhibition halls, higher priority can be given to the users that need access to high definition (HD) video-based applications for exhibitors, while the visitors that only need to access the Internet for event content and surfing websites, can be regarded as low priority [24]. Similarly, this solution can be extended to any high-density traffic areas such as airports, where the high priority users could enjoy a premium service at an extra cost. The same scenario could be implemented in any place where the AP and user densities are high and certain users need to be prioritised [24]. Therefore, as we will explain in Section IV, the first step of our algorithm is represented by the classification of the users as High and Low priority.

Our SDWN-based controller is then able to make a smart decision to provide services to users by the evaluation of parameters monitored in the radio environment such as bandwidth, jitter, SINR and delay. As we will detail in the next section, the process involves the calculation of these parameters for each Low and High priority user which are currently connected to the APs managed by the SDWN-based controller. The controller makes the final decision by combining the statistics related to these parameters

**Table 1** Summary of the state of the art

Reference	Strategy	Limitations
[4]	RSS-based	Users' requirements in terms of QoS and QoE not considered High density not considered Presence of users that expect service delivery with high priority not considered
[5–10]	Decision function-based	Users' requirements in terms of QoS and QoE not considered High density not considered Presence of users that expect service delivery with high priority not considered
[3, 11–14]	QoS and User-centric-based	Users' requirements in terms of QoS and QoE not considered High density not considered Presence of users that expect service delivery with high priority not considered
[15, 16]	QoS and User-centric-based	Users' requirements in terms of QoE not considered Presence of users that expect service delivery with high priority not considered
[4, 17, 18]	MAD-based	Users' requirements in terms of QoS and QoE not considered High density not considered Presence of users that expect service delivery with high priority not considered
[2]	Fuzzy logic and neural network-based	Presence of users that expect service delivery with high priority not considered
[19–22]	Fuzzy logic and neural network-based	Users' requirements in terms of QoS and QoE not considered High density not considered Presence of users that expect service delivery with high priority not considered

with the priority of the users to provide them with services accordingly.

The main contributions of this paper with respect to the above-mentioned state of the art can be summarised as follows:

- We propose a novel SDWN-based architecture that extends our previous version [2] and, for the first time to the best of our knowledge, provides a framework that supports user priorities;
- We propose and assess a priority based HO algorithm that benefits High priority users. Moreover, the algorithm also supports Low priority users by using a Multi-Criteria Decision Making (MCDM) approach;
- The proposed approach optimizes the QoE of high priority users through continuous monitoring of network parameters. The details on this contribution will be provided in Sect. 4.

### 3 SDWN-Based Architecture

In this section, we present the design of the proposed SDWN architecture that supports our HO algorithm. As we will explain throughout this section, SDWN offers an extension of Software-Defined Networking (SDN) to support flexible, fast and scalable management of the WLAN network. Note that the use of SDN and SDWN for the management of wireless networks is a well-known approach that can be found in the state of the art. For instance, SDWN is employed in Wi-Fi networks for dynamic AP selection in [15, 16] and [25], and cellular networks for radio resource management in [26]. Moreover, SDN has also been considered in routing schemes for robotic systems [27]. The efficient employment of this technology analysed in the literature has motivated our use of SDWN for the management of HOs in WLANs as considered in our work. Specifically, the architecture presented in this paper will enable the programmability to manage the data plane and HOs in unlicensed frequency for a WLAN network through an SDWN-based controller. The system architecture presented in this paper follows the SDN structure defined in [28] and is composed of different layers: *Infrastructure Layer*, *Control Layer* and *Application Layer*. Figure 2 shows this architecture, including the SDWN-based controller, represented by the *Control Layer*, which implements the proposed algorithm. Note that this architecture extends our previous framework presented in [2] through the enhancement of the *Application Layer* able to address users' priorities as explained in Sect. 3.2.

The control layer is able to manage all the APs in the network, thus facilitating the execution of the HO. Moreover,

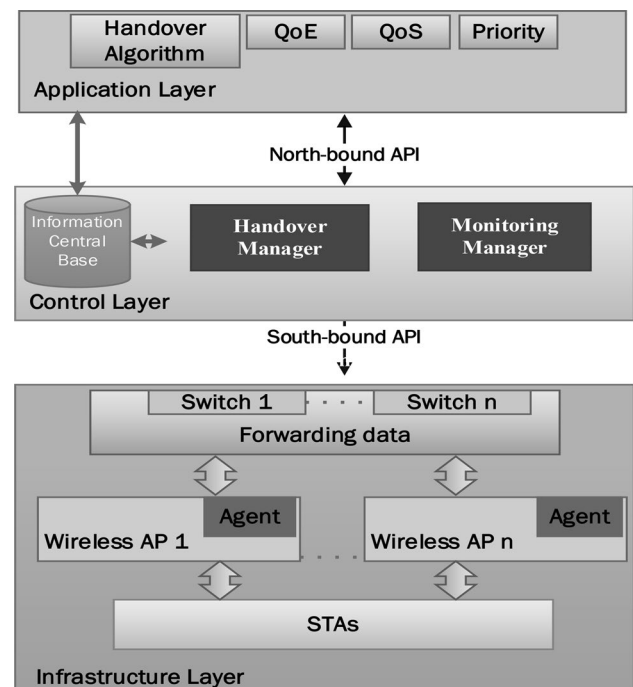


Fig. 2 SDWN-based Architecture

the centralised nature of SDWN enables the control layer to obtain a global view of the network through monitoring and measurements, which will support the HO process. This is possible through the North-bound and South-bound Application Programming Interfaces (APIs) explained in Sects. 3.4 and 3.5, respectively. The details of all the layers included in the architecture are provided in the next sub-sections.

#### 3.1 The Control Layer

This layer is responsible for translating the application layer commands, managed through the North-bound API, to the infrastructure layer explained in Sect. 3.3, and includes the monitoring manager, the information central base (ICB) and the Handover Manager. The main role of the monitoring manager is in providing real-time monitoring to collect the data used in our algorithm from the managed APs and STAs, which are bandwidth, SINR, delay, jitter and users' priorities through the South-bound API. Further details will be provided in Sect. 3.3.

The ICB represents a central database storing the information collected by the Monitoring Manager related to the network performance and user requirements. The main role of the ICB is to keep track of active data traffic flows that are currently connected to the network. In more detail, the ICB stores all the user QoS requirements and the AP link capacity in terms of the available bit rate for the application data flows within the network [29], that will be mapped to

the QoE, and the users' priorities. The Handover Manager is the decision-making entity that uses this data to assist the wireless devices in the HO process, allowing them to always connect to the most suitable AP based on our algorithm.

### 3.2 The Application Layer

This layer consists of all the applications built on the top of the SDWN architecture. Such applications have the ability to access both lower layers (control layer and infrastructure layer) in order to manage the whole network functionality. With the help of the information gathered through the control layer, the applications abstract the network view and assist in the HO decision-making process providing information in terms of QoS, QoE and priorities.

QoE can be defined as an overall measurement of the network system performance, which depends on the perceived acceptability of service from the user's point of view. In this paper we consider the mean opinion score (MOS) as a QoE metric, which provides the human user's view of the quality of the network [15]. Specifically, the MOS is an arithmetic mean of all the individual scores achieved by the result of subjective tests and can range from 1 (worst) to 5 (best). Herein, the MOS provides a quantitative analysis of the more general form of QoE whereas the QoS is the actual bandwidth offered by the network to the user. The meaning of each score is illustrated in Table 2 in terms of quality and impairment. The MOS is obtained by observing the network parameters, i.e., delay, jitter, SINR and bandwidth available in each AP, and applying this to the type of application in use. For instance, while a good MOS score for VoIP can be obtained simply by reducing the jitter and delay experienced, for video streaming the same score might require significantly increased bandwidth. The qualities range from *Bad*, which corresponds to a *Very Annoying* impairment of the service to *Excellent* that corresponds to an *Imperceptible* impairment. Details on the achievements of QoE based on QoS requirements are given in Sect. 4.

Furthermore, users' priorities are obtained through the IP addresses of the STAs that allows us to label them as *High* or *Low* priority users. They are then used in the algorithm as explained in the next section.

**Table 2** Mean opinion score—MOS

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

### 3.3 Infrastructure Layer

This layer consists of all the data plane elements such as APs, switches and STAs, managed by the controller to respond to orders such as the forwarding of packets, HO management, and wireless parameter tuning. Moreover, it provides live monitoring represented by network status data gathered by the agents implemented in the APs and sent to the monitoring manager in the control layer through the switches using OpenFlow v1.3 protocol [30]. Therefore, the infrastructure layer enables data forwarding and processing functionalities in the network such as processing of data paths, based on live monitoring data and the algorithms implemented in the control layer.

The data gathered by the agents implemented in the APs include the bandwidth capacity, the SINR, the jitter, and the delay experienced by the STAs located in their covered area together with the priority negotiated by the users.

### 3.4 North-Bound API

The North-bound API allows different applications, such as the HO algorithm presented in this paper, to program the wireless networks as desired, based on the information obtained by the controller through the South-bound API explained below [31].

### 3.5 South-Bound API

The South-bound API provides the controller with monitoring information, statistics and events from all the network elements, which can be used as input to our algorithm. It has a direct connection with the lower component's North-bound interface. In the SDWN paradigm, the South-bound interface enables communication between a controller, switches and the routers or APs to provide effective control over the network. The South-bound interface enables routers or APs to send requests relayed from the North-bound interface and learn about the network topology. In this way, the SDWN architecture can modify network configurations according to the real-time requirements. The above-mentioned OpenFlow protocol [30] and Cisco's OpenFlex interfaces [32] are well-known examples of a South-bound API.

In our architecture, the control layer and the infrastructure layer communicate through OpenFlow v1.3. Specifically, the messages between the controller layer and the switches included in the infrastructure layer are organised according to the OpenFlow protocol and divided into three types of messages, i.e., *controller-to-switch*, *asynchronous* and *symmetric*. Controller-to-switch messages are initiated by the controller and requires a response from the switch. Examples are messages to request state information from the switches, such as AP capabilities in terms of available bandwidth, or

delay experienced by the STAs. Asynchronous messages are unsolicited and sent from the switch to the controller to denote a packet arrival. Examples are the *Packet-in* messages, which forward received packets to the control layer and the *Port-Status* message, which notifies the control layer of any addition or removal of a port on the switch. Finally, symmetric messages are unsolicited and sent from the switch to the controller or vice-versa. Examples are the *Hello* messages, which are exchanged during connection start-up.

#### 4 High and Low Priority User-Based Algorithm

The proposed Handover Algorithm is divided into two parts called Sub-Algorithm 1 based on *Priority* and Sub-Algorithm 2 based on *Multiple Criteria Decision Making (MCDM)*, respectively. Sub-Algorithm 1 is responsible for evaluating the QoE of all the connected users during high traffic periods and identifying Low priority users which are receiving QoE below an acceptable threshold, which is user-defined. The Low priority users will be sent to a specific queue where Sub-Algorithm 2 comes into play to relocate them to candidate APs based on their capacity and distance from the user. The following Sub-Algorithm 1 and Sub-Algorithm 2 describe the pseudo-code of the High and Low priority-based algorithm proposed in this paper. Furthermore, the proposed algorithm relies on the following parameters:

- $U$ : is a set including all the users connected to the network with any kind of priority, i.e., High or Low.
- $U_{\theta}$ : is a set that contains all the users connected to the network with any kind of priority and experiencing a QoE below the threshold.

- $U_{\theta L}$ : is a set that contains all the Low priority users located in APs where High priority users experience QoE below the threshold.
- $U_{TrustQ}$ : is a set that contains Low priority users.
- $\theta$ : is the threshold, which is set to a MOS value of 3.1 in order to guarantee at least the *fair* or *slightly annoying* option illustrated in Table 2.

Sub-Algorithm 1 starts by initialising the current parameters (bandwidth, jitter, delay, SINR) for each user  $u_i$  included in set  $U$  (lines 1–6 of Sub-Algorithm 1). The details on the computation of these parameters can be found in [2]. Note that in our scenario illustrated in the next section, we will consider half the users with high priority and half the users with low priority. In the next step, Sub-Algorithm 1 calculates the QoE in terms of MOS based on the above-mentioned parameters for all connected users included in set  $U$  (line 7 of Sub-Algorithm 1).

Table 3 extends the previous Table 2 to indicate how the MOS is mapped to various levels of QoE according to the QoS parameters obtained in terms of bandwidth, jitter, delay, SINR and the application type. The MOS is a commonly used subjective method to measure QoE and it is standardized by the International Telecommunication Union (ITU) [33]. The values of MOS are subdivided into 5 levels based on subjective or objective reasonings. In this paper we considered the objective reasoning. Specifically, the MOS values in Table 3 have been obtained considering the surveys conducted in [34–36] based on subjective quality evaluation tests. Customers were required to rate the QoE based on their expectation in terms of objective quality of delay, jitter, bandwidth and SINR. For instance, a H.320 video streaming session can reach an *Excellent* quality from an AP able to guarantee a delay lower than 2 ms, a jitter lower than 20 ms, a minimum SINR of 20 dBm and a minimum bandwidth of 900 kbps.

Afterwards, the controller seeks users with  $QoE < \theta$  during an interval time of 0.2 s (lines 8–12 of Sub-Algorithm 1)

**Table 3** QoS to QoE mapping

MOS	Excellent	Very good	Average	Fair	Poor
Delay	$\leq 2$ ms	$> 2$ ms $\leq 4$ ms	$> 4$ ms $\leq 8$ ms	$> 8$ ms $\leq 15$ ms	$> 15$ ms
SINR	$\geq 20$ dB	$\geq 15$ dB $< 20$ dB	$\geq 9$ dB $< 15$ dB	$\geq 3$ dB $< 9$ dB	$< 3$ dB
Jitter	$\leq 20$ ms	$> 20$ ms $\leq 80$ ms	$> 80$ ms $\leq 140$ ms	$> 140$ ms $\leq 200$ ms	$> 200$ ms
BW video H.320	$\geq 900$ kb/s	$\geq 625$ kb/s $< 900$ kb/s	$\geq 450$ kb/s $< 625$ kb/s	$\geq 220$ kb/s $< 450$ kb/s	$< 220$ kb/s
BW VoIP G.711	$\geq 64$ kb/s	$\geq 50$ kb/s $< 64$ kb/s	$\geq 37$ kb/s $< 50$ kb/s	$\geq 24$ kb/s $< 37$ kb/s	$< 24$ kb/s

which are included in the set  $U_\theta$  (line 13 of Sub-Algorithm 1). For each AP providing services to the High priority users included in  $U_\theta$ , the controller separates the Low priority users by moving them from  $U_\theta$  to  $U_{\theta_L}$  (line 15 of Sub-Algorithm 1). If the controller does not find any High priority users below the threshold, it restarts the process. The purpose of separating all Low priority users from the pool of connected users is to always maintain the minimum value of 3.1 QoE for High priority users.

For each user  $u_i$  included in  $U_{\theta_L}$ , if it can connect only to its current AP, it is moved to the set  $U_{TrustQ}$  and it can receive only limited resources, i.e., only web browsing (lines 17–19 of Sub-Algorithm 1). In the case that user  $u_i$  can connect to other APs, i.e., it is located in overlapped areas, it is moved to the set  $U_{TrustQ}$  and the controller triggers Sub-Algorithm 2 to perform the Low priority users' reallocation, which is explained below (line 20 of Sub-Algorithm 1). Note that these users can be moved again to  $U_{\theta_L}$  if resources become available, e.g., a High priority user leaves the network.

#### Sub-Algorithm 1 - Priority

```

1: Initialize:
2:  $U \leftarrow$  No. of users (low and high priority) connected to AP
3:  $B \leftarrow$  Bandwidth received by user  $u_i$ 
4:  $J \leftarrow$  Jitter experienced by user  $u_i$ 
5:  $D \leftarrow$  Delay encountered by user  $u_i$ 
6:  $SINR \leftarrow$  Signal-to-interference-to-noise ratio of user  $u_i$ 
7: Calculate QoE from B, J, D, SNIR
8: Evaluate the threshold:
9: if  $QoE \geq \theta$  then
10: wait(1)
11: goto 1
12: else
13:  $U_\theta \leftarrow$  Get users below threshold
14:  $QoE \leftarrow$  Estimate QoE based on users below threshold
15:  $U_{\theta_L} \leftarrow$  Get users with lowest priority
16:  $U_{TrustQ} \leftarrow U_{\theta_L}$ . Add lowest priority users to queue
17: if AP == 1 then
18:    $U_{TrustQ} \leftarrow$  Limit resources
19: else
20:  $U_{TrustQ} \leftarrow$  Perform user relocation to APs using Algorithm 2
21: end if
22: end

```

Sub-Algorithm 2 is based on the principle of MCDM, which is the process of selecting the best AP for Low priority users from a set of finite decision options. MCDM is used to make decisions through the evaluation of multiple conflicting criteria. In this case, the set is defined by the candidate APs for the users included in  $U_{TrustQ}$ , i.e. the APs providing coverage in the area in which the users are located. Moreover, candidate APs represent the conflicting criteria in the MCDM scenario. In detail, the APs are evaluated based on a) the distance from each user included in  $U_{TrustQ}$  which is denoted  $DS$  and b) their capacity which is denoted by  $C$ . The distance between the user and all the candidate APs is

estimated based on the RSS by the user, which decreases as the user moves further from the AP.

#### Sub-Algorithm 2 – MCDM

```

1: Initialize:
2:  $DS \leftarrow$  Distance from AP
3:  $C \leftarrow$  Capacity of the AP
4: Construct decision matrix:
5: for  $a$  in AP do
6:    $DC_{mxn} \leftarrow DS_a$ 
7:    $DC_{mxn} \leftarrow C_a$ 
8: end for
9: Standardize decision matrix (DC):
10: Calculate weights:
11: for  $m$  in row(DC) do
12:    $t \leftarrow 0$ 
13:   for  $n$  in col(DC) do
14:      $t \leftarrow t + DC_{m,n}^2$ 
15:   end for
16:    $W_{m \times 1} \leftarrow \sqrt{t}$ 
17: end for
18: Normalize decision matrix (DC):
19: for  $m$  in row(DC) do
20:   for  $n$  in col(DC) do
21:      $DC_{m,n}^* \leftarrow DC_{m,n} / W_{m \times 1}$ 
22:   end for
23: end for
24: Calculate distance between ideal and negative ideal solution:
25: Calculate the ideal solution by using equation (2)
26: Calculate the negative ideal solution by using equation (3)
27: Evaluate ideal and negative ideal solution:
28:  $DS \leftarrow$  Minimum value is ideal
29:  $C \leftarrow$  Maximum value is ideal
30: if Ideal solution found then
31:    $U_{TrustQ} \leftarrow$  Perform relocation of users to APs
32:   Return to Algorithm 1
33: else
34:   goto 1
35: end if

```

The purpose of selecting these two parameters  $DS$  and  $C$  is to best describe the capabilities of the candidate APs in terms of providing the best QoS to the Low priority users included in  $U_{TrustQ}$ . The controller starts Sub-Algorithm 2 by initialising the parameters  $DS$  and  $C$  (lines 1–3). In the next step (lines 4–8), it constructs a decision matrix with the candidate APs and their calculated parameters  $DS$  and  $C$ . Therefore, for each AP  $a$  among the candidate APs, the controller computes the matrix using  $DS_a$  and  $C_a$ .

Then, the decision matrix, which is called  $DC$ , has the form of  $mxn$  and is initialised using the  $DS$  and  $C$  for each AP  $a$  belonging to the set of candidates (lines 6 and 7). Here,  $m$  represents the number of users connected to the AP and  $n$  indicates the number of parameters  $DS$  and  $C$ . Next, the  $DC$  is standardised and later normalised in order to convert different dimension parameters into dimensionless ones. After the normalisation all the parameters will have equal effect in the algorithm, which simplifies making a comparison among multiple conflicting criteria.

The normalization is performed by calculating  $W$ , which is the root of the sum of the square of the values in the  $DC$ , using two nested loops (lines 11–17 of Sub-Algorithm 2). The first loop iterates through the rows of the  $DC$ , while the second loop iterates through the columns. Firstly, the sum of the square of the values is computed across each row of the decision matrix and assigned to variable  $t$  on each iteration of the second loop (as shown in line 14 of Algorithm 2). Then, after all the iterations of the second loop, the square root is computed for variable  $t$  and assigned to  $W_{m \times 1}$  (see line 16 of Sub-Algorithm 2).  $W_{m \times 1}$  is a one-dimensional array, where,  $m$  represents the number of rows in the decision matrix.

Subsequently, each value in the decision matrix is divided by  $W_{m \times 1}$  to normalise the values. The equation below computes the normalised  $DC$ :

$$DC'_{m \times n} = \frac{DC_{m \times n}}{W_{m \times 1}} \quad (1)$$

The normalisation will convert different dimension parameters into same scale parameters, which will have an equal effect in the algorithm. After the normalisation process, the ideal and negative ideal solutions are calculated using the  $DC$ .

In this sub-algorithm, the ideal solution is represented by the candidate AP that has the least distance from the user and has high capacity. The opposite of the ideal solution is assumed as the negative ideal solution. As Sub-Algorithm 2 proceeds (lines 24–26), the distance of each candidate AP is measured from these ideal solutions using a Euclidean distance measure as shown below.

$$DS_i = \sqrt{(DS_{i1} - DS_{i2})^2 + (C_{i1} - C_{i2})^2} \quad (2)$$

where  $i$  is a subset of  $DS$  when it is minimum and  $C$  is maximum. While the negative ideal solutions is:

$$DS_n = \sqrt{(DS_{n1} - DS_{n2})^2 + (C_{n1} - C_{n2})^2} \quad (3)$$

Similarly,  $n$  is a subset of  $DS$  when it is maximum and  $C$  is minimum. Finally, the candidate APs are ranked based on their relative nearest distance from the ideal solution.

As a result, when Sub-Algorithm 2 finds an ideal solution, it performs the relocation and assigns  $U_{TrustQ}$  users to the selected AP. At this stage, Sub-Algorithm 2 finalises its operation and returns to Sub-Algorithm 1 for further execution (lines 30–32). Hence, our contribution lies in proposing a method to construct a decision matrix based on the ideal and negative ideal solutions, which assists in relocating Low priority users efficiently.

## 5 Performance Evaluation

### 5.1 Simulated Scenario

The SDWN-based architecture illustrated in Fig. 2, which implements the proposed HO algorithm, has been designed and assessed using OPNET. Moreover, in order to benchmark the performance of our HO algorithm, we analyse its performance against our previous work based on fuzzy logic control theory (FLCT) [2]. Note that this work outperforms the IEEE 802.11 standards and another relevant approach for HO found in the state of the art [37]. Our choice of this algorithm is justified by the fact that it uses a similar SDWN-based architecture, has the same aim and, in turn, improved the performance of previous relevant works. The evaluation of our algorithm against this strategy focuses on the following performance metrics, which have been averaged and updated every time a new STA connected to the network: *MOS*, *Delay*, and *Throughput*.

The simulated scenario consists of a dense WLAN deployed in an area of  $500 \times 500$  m including 25 APs and 250 STAs uniformly distributed. All the simulation configuration parameters are illustrated in Table 4 [15, 31].

The application types that we have considered are VoIP G. 711 transmitting in uplink and downlink with a bit rate of 64 kbps and download video streaming H.320 with a bit rate of 438 kbps and a resolution of  $525 \times 384$  pixels. We created 50% of the users with a High priority and 50% with a Low priority to represent a challenging scenario of mixed network usage. Specifically, every 5–10 s we created one Low priority user and one High priority user, respectively. The algorithm is triggered only when at least one of the users receives a QoE that drops below the threshold of 3.1 and the HO is performed simultaneously for all users that experience the QoE decrease.

**Table 4** Simulations parameters

Parameters	Value
AP coverage area	Circular with one cell
Radius of the cell	25 m
Overall AP coverage area	$500 \times 500$ m
Number of APs	25
Number of STAs	250
mac type	802.11 g
Transmit power	0.005 Watts
Reception power threshold	– 95 dBm
AP beacon interval	0.02 s

## 5.2 Performance Results

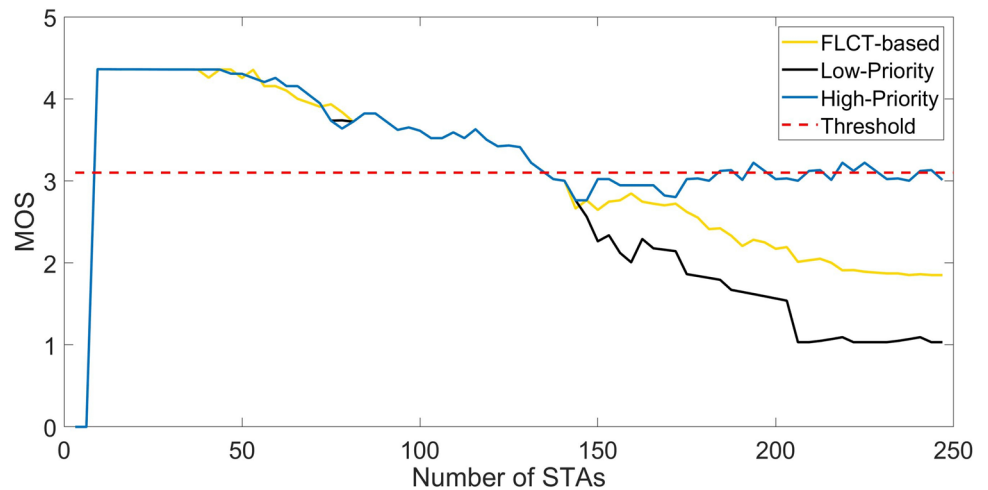
Figure 3 illustrates the result in terms of MOS together with the threshold selected in this simulation campaign. Note that we have calculated the MOS for *High* and *Low* Priority STAs separately. Therefore, the figure shows the value of MOS averaged for all the STAs obtained for all the algorithms every time a new STA connects to the network. The blue line illustrates how the High priority STAs are mainly above the threshold. However, this result also shows a significant drop in the service provided for Low Priority STAs which are represented by a black line. As we have mentioned, this result has been compared with our previous FLCT-based algorithm, which is represented by a yellow line. From this figure we can observe that the priority-based algorithm achieves on average a MOS of 2.6 when all 250 users are connected to the network.

Therefore, the proposed algorithm outperforms the FLCT-based approach by around 30%, which on average provides a MOS of 1.8 to all 250 users. This improvement

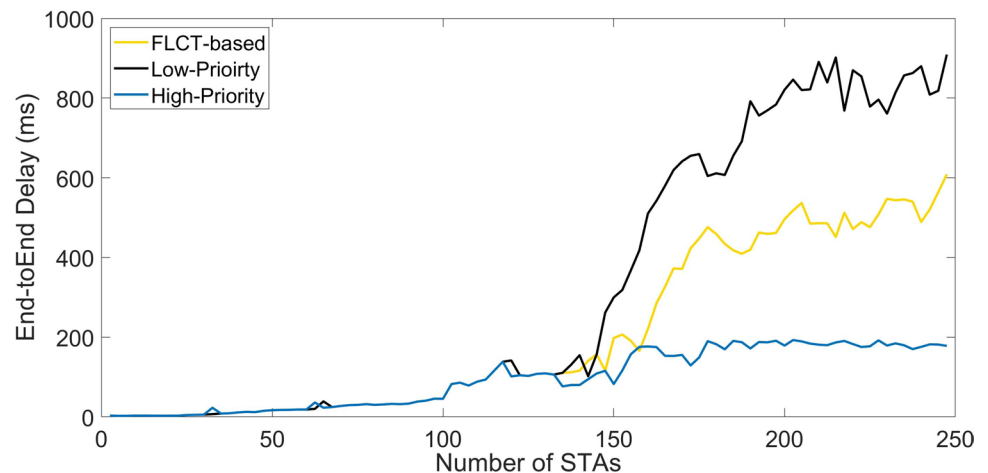
is due to the *MCDM* strategy described by Sub-Algorithm 2, which allows us to select the best option for the Low priority users, while the *Priority* approach, described by Sub-Algorithm 1, guarantees the minimum required QoE to the High priority users. The FLCT-based algorithm does not distinguish high priority users from low priority one, decreasing the average MOS below the *fair* or *slightly annoying* threshold gradually while the number of connected STAs increases. This undoubtedly jeopardizes the QoE that should be guaranteed to the High priority users.

Figure 4 illustrates the end-to-end delay of all the packets received by the STAs. The figure shows how the proposed solution keeps the delay for the High priority users below 200 ms when all the users are connected, while for the FLCT-based algorithm, it is about 600 ms. However, the Low priority STAs experience a delay reaching up to 900 ms. Hence, the proposed algorithm, considering the connection of all the users, i.e., High and Low priority, are characterized on average by a delay of 550 ms and, therefore,

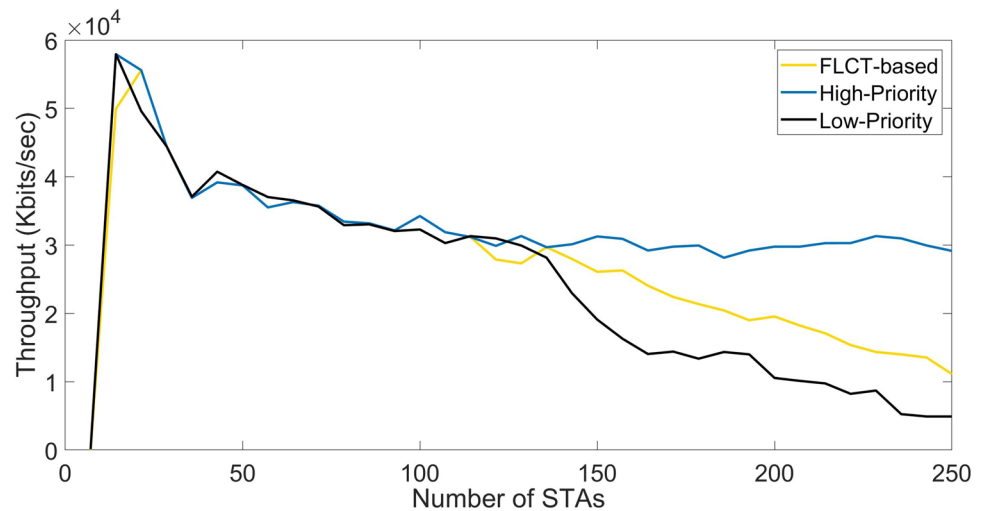
**Fig. 3** MOS as a function of the number of STAs



**Fig. 4** Delay as a function of the number of STAs



**Fig. 5** Throughput as a function of the number of STAs



outperforms the FLCT-based approach by around 8% in terms of this metric.

Next, Fig. 5 shows the average throughput achieved by the STAs. The yellow line again represents the average throughput obtained by the STAs in the FLCT-based algorithm, the blue line denotes average throughput obtained by only the High priority users in the simulated scenario, while the black line represents the average throughput obtained by the Low priority users. From this figure, we can observe that users with higher priority get the highest throughput compared with the rest of the solutions. Specifically, the value of throughput tends to decrease up to approximately 120 STAs, and then it is maintained even when all the STAs are connected to the network at the expense of the Low priority users. In fact, we can observe from the figure that the Low priority users receive the lowest throughput even when compared to the FLCT-based algorithm.

Finally, from this figure we can observe that the priority-based algorithm obtains on average a throughput of  $1.8 \times 10^4$  kbps when all 250 users are connected to the network. Therefore, the proposed algorithm outperforms the FLCT-based strategy by around 28%, which on average provides a throughput of  $1.3 \times 10^4$  kbps to all 250 users.

### 5.3 Discussion

Wi-Fi networks have not been designed to guarantee users' satisfaction in dense deployments, such as airport, campus or shopping mall networks. Therefore, the 802.11 standards lack solutions that guarantee high performance services to specific users in current WLANs. On the other hand, the latest generation of Wi-Fi is expected to provide very high capacity for users demanding bandwidth-hungry applications in densely deployed scenarios [24]. Moreover, as we have mentioned in Sect. 2, to the best of our knowledge, user

prioritization in HO mechanisms has not been addressed in the literature to date.

The figures above illustrate the importance of considering prioritisation in a HO algorithm, which guarantees a satisfactory QoE higher or equal to the *fair* or *slightly annoying* threshold chosen in this paper to High priority users. This algorithm implemented through SDWN can be leveraged in any scenario where maintaining satisfactory performance in terms of QoE, throughput and delay for a subset of High priority users is crucial. In this context, examples of this use cases are: airport and train stations in which approximately the 50% of the users expect higher bandwidth for HD video applications and 10% expect low delay for online gaming; and exhibition halls where 60% of the users expect to access high bandwidth-based content prepared by the exhibitors, such as HD videos [24].

On the other hand, the Low priority users are adversely affected during this process because they experience a significant reduction of their performance. However, note that this algorithm has been designed for scenarios where Low priority users do not expect to utilise either bandwidth-hungry or low delay services. In fact, in both above-mentioned use cases, the 40% of the users are expected to connect to the Internet for low bandwidth-based applications only, such as email, Twitter and web surfing [24].

Moreover, the internet of things (IoT) is another scenario where devices can be considered as Low priority users that do not expect high performance. In fact, due to their specific data traffic characteristics, IoT devices have stricter requirements in terms of high energy efficiency rather than high bandwidth or low delay [38].

Finally, on average, when considering all the users connected to the network, our proposed algorithm provides better performance compared to another work from the state of the art in terms of QoE, throughput and delay. In summary, based on the comparison illustrated in Sub-section V.C and

the discussion provided in this Sub-section, we can conclude that our solution for Handover in dense WLANs outperforms the state of the art [2] by a gain of 30%, 8% and 28% in terms of MOS, Delay and Throughput, respectively, efficiently satisfying the High priority users, while minimising the impact on the Low priority ones. Moreover, the obtained results show how our solution based on SDWN can benefit realistic scenarios with both users that expect to experience high performance applications, and users that connect to the Internet only for low priority applications such as surfing websites.

## 6 Conclusions and Future Works

In this paper we presented a priority-based handover (HO) algorithm implemented in a SDWN-based architecture. The algorithm is made up of two parts named *Priority* and *MCDM*, respectively, and aims to optimise the QoE for users according to their priorities. Specifically, the concept of priority is introduced in order to always provide the best QoE to the High priority users at the expense of Low priority ones. However, the proposed algorithm also attempts to maintain an acceptable QoE for the Low priority users by relocating them to the best candidate APs through the MCDM algorithm. The use of SDWN provides key input from the radio environment through the South-bound API.

The results of the proposed algorithm are compared with another approach from the state of the art, named FLCT-based algorithm. The results indicate that our solution based on users' prioritization outperforms the state of the art providing on average better QoE, throughput and delay to all the connected users. However, this study has been performed through simulations and lacks an analysis of performance in a more realistic scenario. We believe that simulations are crucial as a first step to carry out appropriate studies and evaluate the performance of innovative solutions before progressing to a prototype or full-scale deployment on realistic platforms such as a real-time testbed. Motivated by the encouraging results we have achieved through simulations, as part of our future work, the algorithm proposed in this paper will be implemented and assessed in the SDWN-based testbed designed in the context of the Wi-5 project [39] in order to evaluate our proposal also in a more realistic scenario. Note that the OpenFlow protocol implemented in the South-bound API has been extended in the Wi-5 SDWN architecture in order to handle connection requests from users and their AP allocations and, therefore, to manage HO applications.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long

as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- [Online]. Available: "Ofcom. n.d. Which is the most important device you use to connect to the internet, at home or elsewhere? Statista", last access August 2021, available from: <https://www.statista.com/statistics/387447/consumer-electronic-devices-by-internet-access-in-the>.
- O. Aldhaibani, F. Bouhaf, M. Mackay, and A. Raschella, "An SDN-Based Architecture for Smart Handover to Improve QoE in IEEE 802.11 WLANs," *2018 32nd Int. Conf. Adv. Inf. Netw. Appl. Work.*, pp. 287–292, 2018.
- Y. Taenaka, S. Kashihara, K. Tsukamoto, S. Yamaguchi, and Y. Oie, "An implementation design of a cross-layer handover method with multi-path transmission for VoIP communication," *Ad Hoc Networks*, vol. 13, no. PART B, pp. 462–475, 2014.
- B. J. Chang and J. F. Chen, Cross-layer-based adaptive vertical handoff with predictive RSS in heterogeneous wireless networks, *IEEE Trans. Veh. Technol.*, Vol. 57, No. 6, pp. 3679–3692, 2008.
- H. S. Oh, D. G. Jeong, W. S. Jeon, "Joint Radio Resource Management of Channel-Assignment and User-Association for Load Balancing in Dense WLAN Environment," *IEEE Access*, vol. 8, pages: 69615 – 69628, Apr. 2020.
- A. Ahmed, L. M. Boulahia and D. Gaiti, Enabling vertical handover decisions in heterogeneous wireless networks: a state-of-the-art and a classification, *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 2, pp. 776–811, Jan. 2014.
- M. H. Dwijaksara, W. S. Jeon, D. G. Jeong, User association for load balancing and energy saving in enterprise WLANs, *IEEE Systems Journal*, Vol. 13, No. 3, pp. 2700–2711, 2019.
- J. Gozalvez, M. C. Lucas-Estañ and J. Sanchez-Soriano, Joint radio resource management for heterogeneous wireless systems, *Wirel. Networks*, Vol. 18, No. 4, pp. 443–455, 2012.
- R. P. Ray and L. Tang, Hysteresis margin and load balancing for handover in heterogeneous network, *International Journal of Future Computer and Communication*, Vol. 4, No. 4, pp. 231–235, 2015.
- A.K. Rangiseti, B. H. Bhopabhai, B. Pradeep Kumar, and B. R. Tamma, Load-aware hand-offs in software defined wireless LANs, *Int. Conf. Wirel. Mob. Comput. Netw. Commun.*, pp. 685–690, 2014.
- Z. Chen, Z. Luo, X. Duan, L. Zhang, Terminal handover in software-defined WLANs, *Springer EURASIP Journal on Wireless Communications and Networking*, vol. 68, Mar. 2020.
- K. Košťál, R. Bencel, M. Ries, P. Trúchly, and I. Kotuliak, High performance SDN WLAN architecture, *MDPI Sensor*, Vol. 19, No. 8, p. 1880, Apr. 2019.
- A. Lee and L. Chen, A framework of handoffs in wireless overlay networks based on mobile IPv6, *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 11, pp. 2118–2128, 2005.
- K. Sood, S. Liu, S. Yu, and Y. Xiang, Dynamic Access Point Association Using Software Defined Networking, *International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, Australia, 18–20 Nov. 2015.

15. A. Raschellà, F. Bouhafs, M. Seyedebrabimi, M. Mackay and Q. Shi, Quality of service oriented access point selection framework for large Wi-Fi networks, *IEEE Transactions on Network and Service Management*, Vol. 14, No. 2, pp. 441–455, 2017.
16. A. Raschellà, F. Bouhafs, M. Mackay, Q. Shi, J. Ortín, J.R. Gállego, and M. Canales, A dynamic access point allocation algorithm for dense wireless LANs using potential game, *Elsevier Computer Networks*, Vol. 167, p. 106991, Feb. 2020.
17. A. Stamou, N. Dimitriou, K. Kontovasilis, and S. Papavassiliou, Autonomic handover management for heterogeneous networks in a future internet context: a survey, *IEEE Communications Surveys & Tutorials*, Vol. 21, No. 4, Fourth quarter 2019.
18. H. Zhao, S. Zhang, J. Wei, and Y. Li, Channel width adaptation and access in high-density WiFi networks, *2014 XXXIth URSI Gen. Assem. Sci. Symp. (URSI GASS)*, pp. 14–17, 2014.
19. J. Li, X. Li and R. Zhang, Cross-layer collaboration handoff mechanism based on multi-attribute decision in mobile computation offloading, *Soft Comput.*, Vol. 23, No. 1, pp. 323–341, 2019.
20. S. Goutam, and S. Unnikrishnan, Algorithm for handover decision in IEEE 802.11 WLAN Environment based on Fuzzy Logic, *IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, Bangalore, India, 15–16 Nov., 2019.
21. M. Zekri, B. Jouaber and D. Zeghlache, A review on mobility management and vertical handover solutions over heterogeneous wireless networks, *Computer Communications*, Vol. 35, No. 17, pp. 2055–2068, Oct. 2012.
22. M. M. R. Mou and M. Z. Chowdhury, Service aware fuzzy logic based handover decision in heterogeneous wireless networks, *ECCE 2017 - Int. Conf. Electr. Comput. Commun. Eng.*, vol. 10, no. 1, pp. 686–691, 2017.
23. S. Ghahfarokhi, Distributed QoE-aware channel assignment algorithms for IEEE 802.11 WLANs, *Wirel. Networks*, Vol. 21, No. 1, pp. 21–34, Jan. 2015.
24. [Online]. Available: Usage models for IEEE 802.11 High Efficiency WLAN study group (HEW SG) – Liaison with WFA, IEEE 802.11-11-13/0657r6, Last Access May 2020, available from: <https://mentor.ieee.org/802.11/dcn/13/11-13-0657-06-0hew-hew-sg-usage-models-and-requirements-liaison-with-wfa.ppt>.
25. B. Gomez et al. WiMCA: multi-indicator client association in software-defined Wi-Fi networks, *Springer Wireless Networks*, vol. 27, pp 3109–3125, Mar. 2021.
26. E. Coronado, et al., “5G-EmPOWER: A Software-Defined Networking Platform for 5G Radio Access Networks, *IEEE Transactions on Network and Service Management*, Vol. 6, No. 2, pp. 715–728, Jun. 2019.
27. H. Rutvij, Real-time QoS-aware routing scheme in SDN-based robotic cyber-physical systems, *IEEE International Conference on Mechatronics System and Robots (ICMSR)*, Singapore, 3–5 May, 2019.
28. E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, Software-defined networking (SDN): layers and architecture terminology, RFC 7426, 2015, Last Access May 2020.
29. Q.-T. Nguyen-Vuong, N. Agoulmine, E. H. Cherkaoui and L. Toni, Multicriteria optimization of access selection to improve the quality of experience in heterogeneous wireless access networks, *IEEE Transactions on Vehicular Technology*, Vol. 62, No. 4, pp. 1785–1800, 2013.
30. Software-defined networking: The new norm for networks, Palo Alto, CA, USA, White Paper, Apr. 2012.
31. F. Bouhafs, et al., Wi-5: A Programming Architecture for Unlicensed Frequency Bands, *IEEE Communications Magazine*, Vol. 56, No. 12, pp. 178–185, Dec. 2018.
32. [Online]. Available: “Cisco ACI and OpFlex Connectivity for Orchestrators”, Last Access April 2020, available from: [https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/kb/b\\_Cisco\\_ACI\\_and\\_OpFlex\\_Connectivity\\_for\\_Orchestrators.pdf](https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/kb/b_Cisco_ACI_and_OpFlex_Connectivity_for_Orchestrators.pdf).
33. ITU-T Recommendation, (2003) “ITU-T Rec. P.800.1: Mean Opinion Score (MOS) Terminology”.
34. S. Danilo, B. Boban, P. Boban and P. Vladimir, The impact of quality of service parameters to the subjective and objective video quality assessment, *Serbian Journal of Electrical Engineering*, Vol. 15, No. 1, pp. 97–114, 2018.
35. Pal, T. Triyason, A survey of standardized approaches towards the quality of experience evaluation for video services: an ITU perspective, *International Journal of Digital Multimedia Broadcasting*, vol. 2018, Article ID 1391724, 25 pages, 2018.
36. S. Leandro, G. Edjair, M. Edjair, and Y. Iano, Mean opinion score measures based on E-model during a VoIP call, *The Eleventh Advanced International Conference on Telecommunications (AICT 2015)*.
37. A. K. Rangiseti, B. H. Bhopabhai, B. Pradeep Kumar, and B. R. Tamma, Load-aware hand-offs in software defined wireless LANs, *Int. Conf. Wirel. Mob. Comput. Netw. Commun.*, pp. 685–690, 2014.
38. G. C. Deepak, F. Bouhafs, A. Raschellà, M. Mackay, and Q. Shi (2019) Radio Resource Management Framework for Energy-Efficient Communications in the Internet-of-Things. *Transactions on Emerging Telecommunications Technologies*, 30:12. Doi:<https://doi.org/10.1002/ett.3766>
39. H2020 Wi-5 Project (What to do With the Wi-Fi Wild West). <https://github.com/Wi5>.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Omar A. Aldhaibani** received his B. S. degree in control and systems engineering from the University of Technology, Iraq, in 2009, and the M. Sc. degree in Computer Systems and Networks Engineering from Kharkov University, Ukraine in 2012 and, the Ph.D. degree in wireless communications from Liverpool John Moores university. Currently, working as a research associated in sensor technology and microwave at Liverpool John Moores University.



**Alessandro Raschellà** received the M.Sc. degree in telecommunications engineering from the University Mediterranea of Reggio Calabria (UNIRC), Italy, in 2007, and the Ph.D. degree in wireless communications from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 2015. From 2007 to 2009, he was a Research Assistant with UNIRC. He is currently a Post-Doctoral Research Associate with Liverpool John Moores University, U.K. His research interests include dynamic spectrum access and management,

wireless networks optimization, cognitive radio networks, and HetNets.



**Ghulam Mohi-Ud-Din** is working in the Built Environment and Sustainable Technologies (BEST) Research Group at Liverpool John Moores University (UK). He received his M.Sc. degree in Computer Science from Bradford University, (UK) and Ph.D. degree in Computer Science from Liverpool John Moores University, (UK). His research interests include artificial intelligence, deep learning approaches, big data, cyber security, networking, telecommunication, and application-specific

architectures for IoT-based electronics systems.



**Michael Mackay** received the Ph.D. degree in IPv6 transition management from Lancaster University, in 2005. He has been a Senior Lecturer with the Department of Computer Science, Liverpool John Moores University, since 2010. His main research areas currently include cloud networking, IoT and smart applications, and wireless spectrum management.