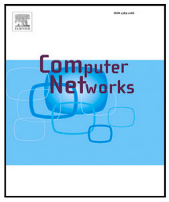
Computer Networks 212 (2022) 109043

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Learning-aided client association control for high-density WLANs

Wenjia Wu *, Yujing Liu, Jiazhi Yao, Xiaolin Fang, Feng Shan, Ming Yang, Zhen Ling, Junzhou Luo

School of Computer Science and Engineering, Southeast University, Nanjing, China

ARTICLE INFO

Keywords:
WLAN
Association control
High client density
Deep reinforcement learning

ABSTRACT

As wireless local area network (WLAN) continues to become popular, there is an increasing number of clients with huge data traffic demands. Especially, some high client-density environments are emerging, such as industrial plants, stadiums, and event centers, which poses significant challenges in terms of client association control. Under such environments, conventional client-side solutions that select access points (APs) according to simple indicators such as signal strength may result in poor network performance, and although some centralized association control mechanisms are proposed, it is still difficult that a large amount of complex global network status information needs to be effectively and efficiently utilized. To meet these challenges, we investigate the online centralized association control problem that aims to improve user quality of experience (QoE), and propose a deep reinforcement learning (DRL) aided solution, called Wi-OAC, where an image-like state pattern is designed to achieve state reformulation for deep Q-network (DQN), and the double DQN and dueling DQN strategies are combined to improve convergence speed. On the basis of offline training, Wi-OAC can determine the proper AP-client associations for the arriving clients. Both simulation experiments and real-world experiments have been conducted to validate the effectiveness of Wi-OAC. In real-world experiments, we build a Wi-OAC testbed including 3 APs and 54 clients in less than 10.5 m^2 area, and the results show that Wi-OAC can significantly improve the performance on the client throughput, AP load balancing and user QoE.

1. Introduction

With the explosive growth of user devices, 802.11 wireless local area networks (WLANs) have become one of the most popular wireless solutions to meet the ever-increasing demands for wireless traffic and user experience, which accounts for a considerable portion of global mobile traffic growth [1]. Nowadays, WLANs have been widely deployed worldwide, providing users with more convenient and higher-speed wireless services [2]. According to a recent forecast from Cisco, there will be nearly 628 million public WiFi hotspots by 2023 [3]. Consequently, the proliferation of WLANs leads to the generation of a large number of high client-density scenarios such as industrial plants, stadiums, and event centers, where many clients are expected to connect to access points (APs) within a small space.

Fig. 1 presents a typical industrial scenario where a huge number of client devices are densely deployed inside, such as environmental sensors, manufacturing facilities and user terminals. Obviously, regarding such a scenario, wireless networking is more suitable than wired networking. On the one hand, the cable-connected devices cannot support mobility well, making it inconvenient for people who need to move constantly. Besides, for wired networking, dense cables need to be deployed to achieve such a huge number of connections, which is expensive and unsafe for the industrial environment. On the other hand, WLANs can perform well in the scenario due to its high data rate, ease of deployment, and cost efficiency. Meanwhile, most of devices can easily access to the WLAN by inherent or additionally equipped WiFi modules. Therefore, the WLAN is a promising solution to provide wireless connections for heterogeneous devices in the industrial environment [4].

However, it is very challenging to guarantee the client traffic demands in such a high client-density scenario. Although the new generation WLAN technology (also known as WiFi 6 or 802.11ax) significantly improves the average throughput in densely deployed environments through advanced physical-layer and medium access control (MAC) sub-layer technologies [5], appropriate AP-client association management is still necessary to further improve the spectrum utilization and client throughput due to the scarcity of spectrum resources. Nevertheless, conventional client-side association mechanisms based on simple indicators such as received signal strength indication (RSSI) may lead to an unbalanced situation where some APs are overloaded while others

* Corresponding author.
E-mail address: wjwu@seu.edu.cn (W. Wu).

https://doi.org/10.1016/j.comnet.2022.109043
Received 30 January 2022; Received in revised form 13 April 2022; Accepted 10 May 2022
Available online 18 May 2022

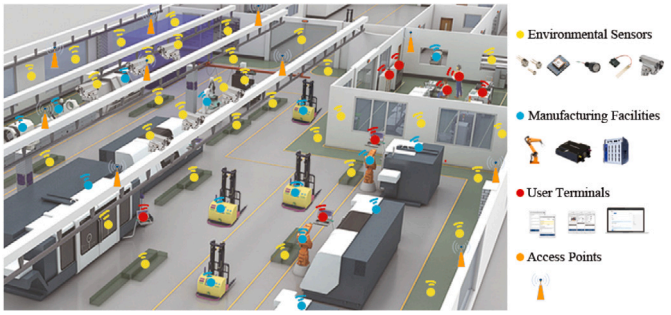


Fig. 1. High client density in industrial wireless networks. In a industrial plant, a large number of client devices are densely deployed and served by multiple APs, including different types of environmental sensors, manufacturing facilities and user terminals.

are almost idle [6,7]. Even though the metrics of AP selection are improved [8–12], the performance is still unsatisfactory. In this case, spectrum resources of the overloaded APs are shared by too many clients [13], thus resulting in a significant throughput degradation for the clients served by the overloaded APs [14].

To this end, some centralized association control mechanisms [15–17] have been proposed to solve this issue by determining the AP-client associations on the infrastructure side instead of client side. By dealing with the global network status information collected from the controlled APs such as their capabilities, operating channels and associated client lists, these infrastructure-side solutions can obtain a global view of the WLAN, and usually outperform the client-side solutions. However, given the size and complexity of the collected data in high client-density WLANs, these existing mechanisms cannot handle such a large amount of complex global network status information efficiently, which undoubtedly degrades their performance significantly. To bridge this gap, we apply the deep reinforcement learning (DRL) method into the association control mechanism, where a large amount of data is used to train the association control policies without a comprehensive analysis of the complex data. On this basis, we develop an online association control scheme for the high client-density WLANs, which assigns appropriate APs for the arriving clients dynamically with the goal of improving user quality of experience (QoE).

This paper investigates the online centralized AP-client association control problem in high client-density WLANs aimed at improving user QoE, while considering the elimination of overloaded APs. For this purpose, we first model the association control problem as the Markov decision process (MDP), the primary analysis framework of reinforcement learning (RL), and define the state, action, reward and policy accordingly. Considering the complexity of the state space in the high client-density environment, a deep neural network (DNN) is introduced to handle the dimensionality curse, which is challenging to be solved in traditional RL. Furthermore, by combining the advantages of deep learning (DL) and RL, we propose a deep reinforcement learning aided association control scheme, called Wi-OAC, which can make online association decisions for the dynamic and high-dimensional environment with the help of the offline trained model through the data of APs and associated clients collected from real-world scenarios. The main contributions of this paper can be summarized as follows:

- We consider the challenges of centralized association control for high client-density WLANs, and propose a DRL-aided solution called Wi-OAC to improve user QoE, where the global network status information collected from real-world scenarios are utilized by model training.
- Since the state space of DRL model is complicated, a state reformulation scheme is designed to transform the initial state representation into an image-like pattern, and the convolutional neural network (CNN) is utilized to effectively extract features such as AP/client positions, existing associations, and throughput of associated clients from the image-like tensors. Moreover, the double deep Q-network (DDQN) and dueling DQN strategies are combined to accelerate the convergence.
- Both simulation experiments and real-world experiments are conducted for performance evaluation. In real-world experiments, a Wi-OAC testbed with 3 APs and 54 clients is built in less than 10.5m^2 area, and the experimental results demonstrate that our solution can significantly improve the performance in terms of average throughput, AP load balancing, and user QoE.

The rest of the paper is organized as follows. Section 2 briefly provides some related work. In Section 3, we introduce the system model and formulate the association control problem in high client-density environment. Then in Section 4, the framework of Wi-OAC with system state reformulation and implementation details are presented. We validate the performance of Wi-OAC through simulation and real-world experiments in Sections 5 and 6 respectively. Finally, we draw conclusions and summarize this paper in Section 7.

2. Related work

AP-client association plays an important role in improving WLAN performance and user experience. Hence, much research effort has been devoted to designing AP-client association solutions in recent years. We categorize the related work into two main strands. The former focuses on the AP-client association mechanisms in WLANs, while the latter considers utilizing the RL method for AP-client association decision making.

2.1. AP-client association mechanisms

The existing AP-client association mechanisms can be divided into two categories: the client-side AP selection [8–12] and the infrastructure-side association control [15–24].

Client-side AP Selection: The default association mechanism of the 802.11 standard is a typical client-side AP selection mechanism, where clients always select the APs with the highest RSSI. As mentioned above, the default mechanism cannot provide satisfactory network service, and some works have been done to improve its performance. Xu et al. [8] propose an AP selection mechanism, called SmartAssoc, which makes clients select the best candidate AP according to the RSSI and AP load. In SmartAssoc, clients generate modified probe request traffic to estimate the load of AP candidates without association. Issa et al. [9] also propose an AP selection algorithm that associates the client to an AP based on AP load as well as the RSSI value of its beacon, where the AP load information can be obtained through modified beacon frames. Oni et al. [10] let the client associate with the AP with the highest Signal-to-Interference-plus-Noise Ratio (SINR), which characterizes the degree of interference of the adjacent basic service sets to the target AP and is calculated by physical layer rate, received power, and the number and size of frames received from other interfering sources. In addition, Kim et al. [11] predict the channel interference level and AP load status by continuously detecting the RSSI and receiving interval of beacon frames. As APs always send beacon frames periodically, the data used for decision making is easy to obtain and the overhead is negligible. With the goal of maximizing the global throughput subject to application QoS and AP load constraints, Dinh et al. [12] propose distributed user-to-multiple AP association methods that allow users to make more intelligent association decisions by leveraging the DQN and DDQN-based DRL frameworks.

However, the performance of these client-side association mechanisms is still limited by local information, since a single client cannot obtain the global network status information. Moreover, the improved mechanisms inevitably require modifications to the clients, which make

them impossible to be transparent to users and are challenging to be widely popularized and applied.

Infrastructure-side Association Control: For the infrastructure-side association control mechanism, a central controller is usually utilized to make central AP-client association decisions, which assigns a specified AP to serve each client. Murty et al. [15] introduce a central controller to make association decisions and only force the specified AP to reply with probe response frames so as to make itself visible to the client. Likewise, Zhang et al. [16] use a similar mechanism to decide the frequency bands for clients in the dual-band WLAN. Raschellà et al. [17] propose an AP association algorithm which relies on a centralized potential game developed in a software-defined wireless network framework, while considering external interference. In this algorithm, AP-client associations are decided according to the fittingness factor, a performance parameter with a value ranging between 0 and 1, which represents the suitability of the AP to meet a client's QoS demand. Given the signal quality, AP loads and minimum requirements for user traffic, Bayhan et al. [18] propose several AP-client association schemes based on a software-defined networking (SDN) controller, and leverage link-layer multicasting to handle users with same content requests so as to improve the network utilization. Huang et al. [19] formulate the online AP-client association and resource allocation problem in wireless caching networks as a stochastic network optimization problem and propose an effective scheme targeted at the minimal delivery latencies and maximum network utilities such as throughput. Wong et al. [20] jointly consider association control and random access control to achieve the maximum proportional fairness of client throughput. Gómez et al. [21] propose a SDN-based client association and channel assignment scheme that considers signal strength, channel occupancy and AP load to improve the utilization of available wireless resources and avoid the need for densification. Jian et al. [22] investigate the user association problem under the multi-association scenarios and design the mechanism for load balancing and energy efficiency by jointly considering user association, power allocation and edge node deployment.

Moreover, some researchers [23,24] consider adjusting AP-client associations through client migrations. Wong et al. [23] propose an approximation algorithm to optimize AP re-association by maximizing the minimum user throughput with a certain migration cost constraint. Bhartia et al. [24] divide APs into different cells and APs in the same cell broadcast the same basic service set identifier (BSSID) and share global information while operating on different channels. In this case, to improve the network performance, APs send unicast channel switch announcement (CSA)-enabled beacon frames and steer associated clients to other suitable APs in the same cell dynamically based on an innovative protocol called FreeSteer.

As the infrastructure-side association control mechanisms usually need very little or even no modification on the clients, they are user-transparent and can be easily deployed in actual scenarios. More importantly, by obtaining a global perspective of the WLAN through a large amount of network status information, these centralized mechanisms can make more reasonable association decisions for clients. However, these existing mechanisms may not be able to tackle new challenges in high client-density environments. For example, to make wiser association decisions, the controller needs to collect network status information from APs to obtain a global view of the WLAN. However, given the densely deployed client devices and APs in high client-density WLANs, there is a large amount of complex network status information and those existing infrastructure-side solutions cannot handle it efficiently. Besides, high client-density causes more frequent user mobility and more interference. That is, the wireless environment will become more dynamic, making it difficult for those mechanisms to understand the network status and make proper association decisions. Thus, our solution introduces the DRL method to handle the association control problem in high client-density WLANs, where a large amount of global network status information can be used to train the model. Besides, we design a state reformulation scheme to further help with feature extraction and model training.

2.2. RL based association control

Due to the capability for dealing with complex problems in an unknown environment and adapting to the dynamic system state, RL has received widespread attention over the last few years and is applied in many problems such as dynamic resource allocation problems [25] and sequential decision problems [26]. Given the dynamics of wireless environment, some RL-based association control mechanisms have been proposed. Kafi et al. [27] use a linear combination of features (LCF) to approximate the Q-value function in the association control problem, which improves the performance and efficiency of RL. There are also some solutions [25,26,28,29] that introduce DNN to solve the curse of dimensionality. Meng et al. [25] use DRL to deal with the power allocation problem in wireless cellular networks and prove the good generalization ability of their method. Wang et al. [26] study the problem of dynamic multichannel access in wireless networks and use DRL to learn a policy that could maximize the expected long-term traffic for successful transmission. The results show that their method can achieve the best performance in complex situations. Yu et al. [28] propose a DRL-based MAC protocol for heterogeneous wireless networks, and the goal is to learn the best channel access policy without understanding the operating principles of the MACs of the coexisting networks. Dinh et al. [29] propose a distributed DRL method based on DQN to solve the problem of joint AP association and beamforming in an integrated sub-6 GHz/mm Wave system, which aimed at maximizing the long-term throughput, while satisfying a large number of heterogeneous user QoS requirements.

Therefore, the existing methods demonstrate that RL can deal with the unknown dynamics and prohibitive computation of wireless environments. However, with the objective of maximizing the aggregate throughput, most of them ignore the fact that client traffic demands could be different according to the application types. Besides, the proposed models are mainly trained and evaluated through simulation data and experiments. On this basis, we propose a DRL-aided association control solution for high client-density WLANs. By defining different QoE evaluation models for different applications, our solution can make online association decisions for the arriving clients so as to maximize the average user QoE metric.

3. System model and problem formulation

In this section, we first introduce the system model and then formulate the client association control problem for high-density WLANs.

3.1. System model

We consider a multi-AP WLAN with a central controller, as depicted in Fig. 2. The number of APs is denoted by M, and the set of APs is represented by $\{1, 2, \ldots, M\}$. In this paper, we adopt a centralized association control mechanism based on active scanning [15]. When client i arrives, the controller will aggregate the client's requests and AP status information from multiple APs and then make association decisions in a centralized manner. Note that we only consider the association decisions when clients arrive, and for client departure events, given the complexity, we make no adjustment to the AP-client association relationship every time a client leaves the WLAN. However, the influence of these client departure events on network status can be well reflected once an arriving client requests for association. Since clients arrive frequently in high client-density WLANs, the proposed mechanism can provide timely feedback for the client departure events. Fig. 2 describes the association control process, including the following four steps:

1. The APs within the scanning range of client i receive probe request frames and make no response.

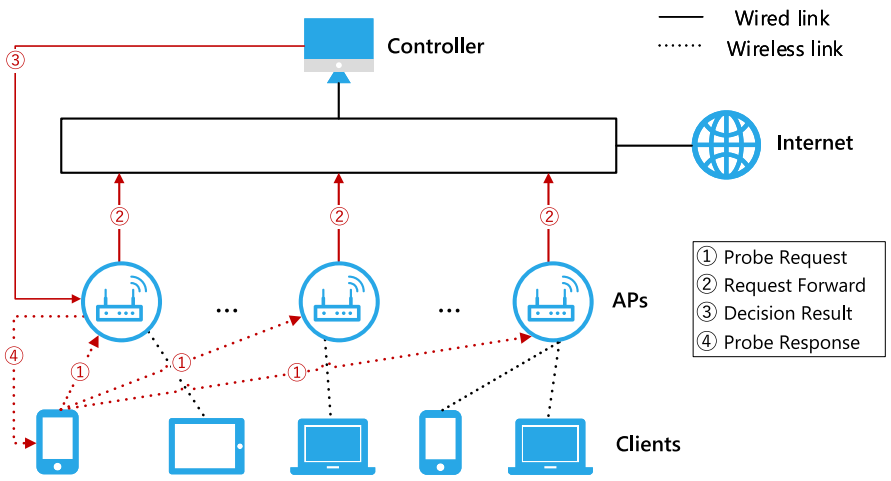


Fig. 2. A multi-AP WLAN with centralized association control mechanism. The central controller makes AP-client association decisions based on the global network status information collected by multiple APs, and assigns a specified AP to serve each arriving client.

Table 1
Notations and corresponding descriptions.

Notation	Description	Notation	Description
M	Number of APs	x_i^t	AP that serves client i at time step t
X^t	AP-client association vector of time step t	N^t	Number of connected clients at time step t
h_i^t	Throughput of client i at time step t	H^t	Client throughput vector of time step t
$L(d)$	Signal path loss of distance d	σ_1, σ_2	Parameters of path loss model
d_{BP}	Breakpoint distance	i^t	Event of client i's arrival at time step t
Ω_t	Average user QoE of time step t	q_i^t	User QoE of client i at time step t
$R_{i,j}^t$	RSSI between client i and AP j at time step t	θ	Clear channel assessment threshold
P	AP transmit power	$d_{i,j}^t$	Distance between client i and AP j at time step t
$\mathcal{S}$	State set	$\mathcal{A}$	Action set
$\mathcal{R}_t$	Reward function of time step t	Π	Policy set
s_t, a_t, r_t	System state, action and reward at time step t	γ	Discount factor in RL
$p(s_{t+1}\|s_t, a_t)$	State transition probability from s_t to s_{t+1} after a_t	α	Learning rate
R^t	RSSI vector of time step t	AU^t	Airtime utilization (AU) vector of time step t
O_A^t, O_C^t	Position matrix of APs/clients of time step t	E_π	Long-term reward expectation
$\mathcal{D}_H^t$	Set of discrete values for throughput of time step t	$\mathcal{D}_R^t$	Set of discrete values for RSSI of time step t
$\mathcal{D}_{AU}^t$	Set of discrete values for AU of time step t	$\mathcal{D}_O^t$	Set of discrete values for positions of time step t
$\psi(s_t)$	Image tensor of system state at time step t	f	Balance index
$\mathcal{T}_j$	Load of AP j	$\mathcal{N}$	Gaussian white noise in the environment
W, L	Width and length of scenarios	ϵ_s, ϵ_f	Initial/Final exploration probability

2. These APs forward the requests of client i to the controller, and then the controller will select an appropriate AP for client i based on global information.
3. The controller only notifies the specified AP of the decision result.
4. The specified AP sends probe response frames when the client i starts the next round of scanning.

The above mechanism ensures that only one AP is visible to arriving client i.

We consider the discrete time steps in our system, and assume that there is only one arriving client at each time step t. For each client, it will be associated with one AP when accessing to the network. In order to represent the AP-client association, we define $x_i^t = j$ to indicate that client i is associated with AP $j \in \{1, \ldots, M\}$, where x_i^t is an element in AP-client association vector X^t of time step t. For the current time step t, the total number of connected clients in the network is denoted by N^t, and the set of clients is represented by $\{1, 2, \ldots, N^t\}$.

We mainly focus on downlink traffic in this work as it accounts for the vast majority of the overall WLAN traffic. Thus, the downlink throughput of client i at time step t is denoted by h_i^t, and the corresponding throughput vector is denoted by $H^t = \{h_1^t, h_2^t, \ldots, h_N^t\}$. The user QoE metric of each client is represented by q_i^t, and its specific definition varies from different application scenarios [30–32].

The positions of APs are fixed after the deployment and easy to acquire for the network administrators. Owing to the existence of multiple APs, each client's position can be estimated by WiFi indoor positioning [33]. Therefore, the positions of the APs and clients can be used as inputs for association decision-making.

The notations in this paper are summarized in Table 1.

3.2. Problem formulation

Consider that in a high client-density environment, APs and clients' positions are arbitrary, and the arrival of clients is uncertain. In our work, there is one arriving client at each time step, and thus the client association control problem can be regarded as a sequential decision problem, which can be solved in the following manner. A decision agent interacts with a discrete event dynamic system sequentially. In each step, the dynamic system will be in a certain state. The agent will observe the current state and select one action from a given action set according to the agent's policy. Then, the dynamic system will enter the next state and obtain a corresponding reward. In this way, the state

transition and action selection are carried out iteratively to maximize the benefits.

Therefore, we model the WLAN system as a discrete event dynamic system driven by client arrival events. Specifically, event i^t represents that client i arrives at the network at time step t. Once a new arrival event i^t occurs, the controller runs the association control algorithm and selects the appropriate AP for client i, and then the time step becomes $t+1$.

In this work, we investigate the online client association control problem for high-density WLANs. For the current time step t, the association control problem is to maximize the current average user QoE metric by determining the proper AP-client association for the client arrival event i^t, while considering the association constraints, clear channel assessment (CCA) constraints and RSSI constraints. Thus, the problem can be formulated as follows:

$$\max \quad \Omega_t = \frac{\sum_{i=1}^{N^t} q_i^t}{N^t} \tag{1}$$

$$s.t. \quad x_i^t = j \in \{1, \dots, M\}, \qquad \forall i \in \{1, \dots, N^t\} \tag{2}$$

$$R^t_{i,x_i^t} > \theta, \qquad \forall i \in \{1, \dots, N^t\} \tag{3}$$

$$R^t_{i,j} = P - L(d^t_{i,j}), \qquad \forall i \in \{1, \dots, N^t\}, \quad \forall j \in \{1, \dots, M\} \tag{4}$$

where Eq. (1) reveals the goal of maximizing the current average user QoE metric at time step t. Since client demands may vary due to different application types, we define different QoE evaluation models for three common application types and the user QoE metric at each time step is determined by the applications used by the arriving client currently. Eq. (2) presents the association constraints, that is, client i is associated with only one AP at time step t. Specifically, x_i^t represents the AP that is associated with the client i at time step t. Eq. (3) expresses the CCA constraints, namely, the RSSI between the arriving client i and AP x_i^t must be higher than CCA threshold θ. Finally, the RSSI constraints are specified by Eq. (4), that is, the RSSI value is determined by the AP's transmit power P and path loss $L(d^t_{i,j})$ based on the distance between client i and AP j at time step t.

4. Wi-OAC

In this section, we propose a deep reinforcement learning aided association control solution called Wi-OAC for high client-density WLANs. Firstly, we present the framework of Wi-OAC based on MDP, including the definitions of state, action, reward and policy. Due to the complexity of the state space, we design a state reformulation scheme that transforms the system state to an image-like pattern. Finally, the implementation details of the Wi-OAC solution are illustrated.

4.1. Wi-OAC framework

In Section 3.2, the client association control problem is regarded as a sequential decision problem, which is suitable to be described by MDP, a set of sequential decision processes with Markov attributes. At time step t, by observing state $s_t \in \mathcal{S}$, the agent takes action $a_t \in \mathcal{A}$ and interacts with the environment to gain the corresponding reward. After the reward r_t is obtained, the environment moves to the next state s_{t+1}. The goal of MDP is to find a policy that maximizes the expected cumulative reward, and the cumulative discounted reward function $\mathcal{R}_t$ is defined as

$$\mathcal{R}_t = \sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau+1} \tag{5}$$

where $\gamma \in [0, 1)$ is a discount factor that trades off the importance of immediate and future rewards, and $r_{t+\tau+1}$ denotes the reward at time step $t+\tau+1$.

In this paper, we use a model-free approach where the state transition probability $p(s_{t+1}|s_t, a_t)$ is unknown. Generally, Q-learning is used in a typical RL method to learn excellent action strategies gradually so as to maximize the expected cumulative reward in the case of knowing nothing about the environment. To perform an action according to a given state, Q-learning estimates the utility function (i.e., Q-value function) gradually according to the following rules:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_{a'} Q(s_{t+1}, a')) \tag{6}$$

where α is the learning rate, and $Q(s_t, a_t)$ indicates the estimated long-term reward after taking action a_t at state s_t. Based on the optimization goal described in Eq. (1), we define the state, action, and reward in the Q-learning as follows.

4.1.1. State

The state is the abstraction of the environment, and the basis for the agent to choose actions. According to the association control problem described in Section 3.2, the current association relationship and throughput of clients are the basic information that the agent should obtain to perceive the current environment. Moreover, some available information has also been proved to be suitable for the criteria in association control [8,9,11], including the RSSI between the arriving client and APs, and the load of APs. Besides, we consider positions of APs and the arriving client in this paper, which may be useful for wiser decision-making. After the agent obtains the above information from the environment, it will form the system state defined as follows.

$$s_t = (X^t, R^t, AU^t, H^t, O_A^t, O_C^t) \tag{7}$$

where X^t is the current association vector, R^t is the RSSI vector, AU^t is the AU vector used to measure the load of APs, H^t is the throughput vector, and O_A^t and O_C^t are the position matrices of APs and the arriving client, respectively.

4.1.2. Action

The agent needs to select an action based on the system state s_t according to its policy. For the client association control problem, the agent's action is defined as the AP that is assigned to serve the arriving client at time step t, that is, $a_t \in \mathcal{A}$ and a_t is defined as follows.

$$a_t = argmax_{a \in \mathcal{A}} Q(s_t, a) \tag{8}$$

4.1.3. Reward

In our work, the goal is to maximize the average user QoE metric at time step t. Thus, the single-step reward r_t depends on the increment of average user QoE metric caused by the action a_t, i.e.,

$$r_t = \Omega_{t+1} - \Omega_t \tag{9}$$

4.1.4. Policy

In MDP, the policy π is employed to determine the next action based on the current state. Thus, our goal is to find the optimal policy π^* with the maximum long-term reward expectation E_π, which is defined as follows:

$$\pi^* = argmax_{\pi \in \Pi} E_\pi[\sum_{t=0}^{\infty} \gamma^t r_t | s_0] \tag{10}$$

Remarkably, the traditional Q-learning algorithm forms the Q-value function gradually by constructing and updating the Q-table. However, according to the definition of the state mentioned above, there are some continuous variables in the system state, which can cause infinite rows in the Q-table. A feasible solution is to discretize the continuous variables. Nevertheless, for a high client-density WLAN, the state space after discretization may be unacceptably large. In this case, even if the Q-table is constructed, it is difficult to update and converge it.

An alternative is DNN, which can extract complex non-linear features from the input data, and is suitable for addressing RL problems

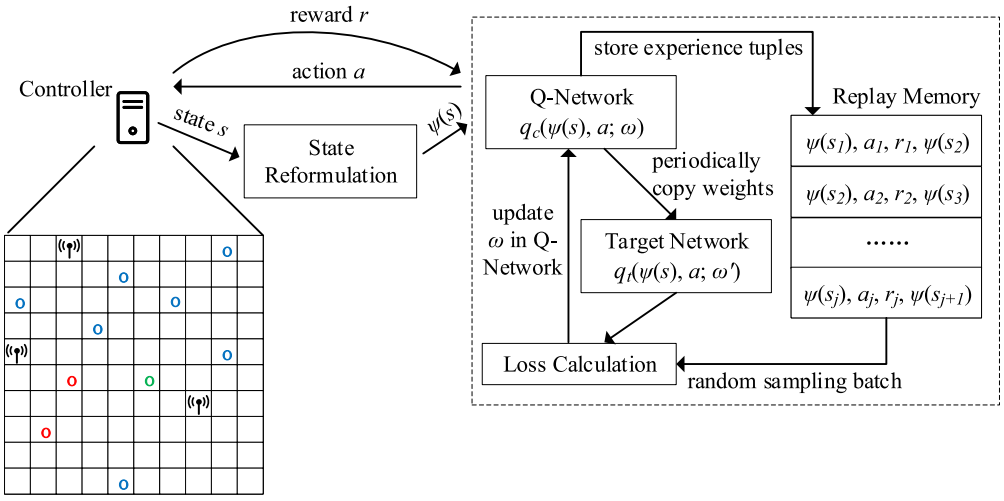


Fig. 3. The overall framework of Wi-OAC, where the DQN model receives the reformulated state as an input, outputs an action and calculates the corresponding reward. In the DQN model, Q-values for each action are calculated in the Q-network and stored in replay memory in the form of experience tuples. Besides, a batch of experience tuples is sampled randomly and periodically to calculate the loss and update the weights in the Q-network. The weights are copied to the target Q-network for every certain steps to accelerate the convergence.

with a large state space. Therefore, we use DNN to approximate the Q-value function, as shown in the following formula:

$$Q(s, a) \approx q(s, a; \omega) \tag{11}$$

where $q(s, a; \omega)$ is the approximation given by the DNN, and ω is a parameter vector containing the weights of the edges in the DNN.

Thus, by combining the advantages of the DL and RL, we propose the Wi-OAC framework to handle the client association control problem. Fig. 3 presents the overall framework of Wi-OAC that consists of two main parts, the offline training and the online association. During the offline training, we use the data collected from APs and associated clients under real-world scenarios to train the DQN model. Specifically, DQN is used as an approximator to receive the state as an input, select an action according to actions' Q-values and obtain the corresponding reward. Meanwhile, $q(s, a; \omega)$ is updated accordingly by adjusting the weights in the DQN to learn the optimal association policy gradually. After that, in the online association, by leveraging the trained model, the controller specifies an appropriate AP for each arriving client and completes the corresponding AP-client association.

4.2. State reformulation

On the basis of the Wi-OAC framework, we reformulate the system state as an image-like pattern, where different semantic information can be encoded in different channels. To be exact, the transformed state is called an image-like tensor $\psi(s_t)$. We choose to reformulate the state mainly based on the following three considerations.

- The effectiveness and efficiency of the DRL algorithm depend on whether the representation of system state is reasonable. To ensure that DQN can extract features from the input effectively, it is necessary to transform the initial representation of system state into a structured representation.
- CNN is suitable for extracting the spatial features of images and the logical relationship between pixels, and the information contained in them can guide the decision-making of association control. Thus, to make full use of CNN, the reformulation process is conducted to transform the input into an image-like tensor $\psi(s_t)$.
- The size of $\psi(s_t)$ can be well fixed, which means that no matter how the number of APs changes, and how the clients join or leave, the size of $\psi(s_t)$ does not need to be changed.

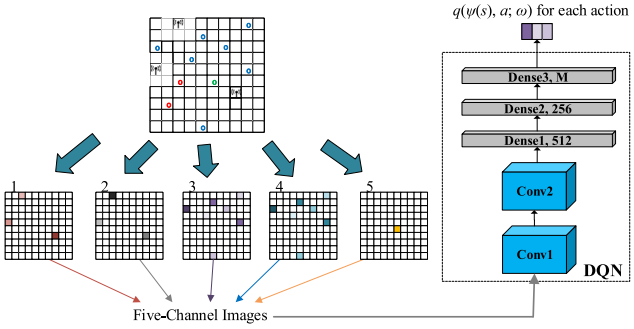


Fig. 4. State reformulation and DQN structure. The system state is reformulated into an image-like tensor with five channels where the global network status information is encoded. On the basis of the image-like tensor, Q-values for each action are estimated by DQN, which mainly contains 2 convolutional layers, 2 pooling layers, and 3 fully connected layers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We complete the reformulation through two-dimensional representation of the environment and discretization of space. The size of $\psi(s_t)$ is $W \times L \times 5$, where W and L are the width and length of the environment, respectively, and the number of image channels is 5. Therefore, there are $W \times L$ pixels in each image channel. Each pixel in the image is a grid, representing a certain size of square space in the actual environment. Different status information about APs and clients in this grid can be expressed in different channels.

Fig. 4 intuitively depicts the reformulated system state, where circles in different colors represent clients with different statuses, i.e., the red, green, and blue circles correspond to the unconnected, connecting, and connected clients, respectively. Specifically, the information encoded in each channel is defined as follows:

- **AP-position channel:** This channel indicates the positions of deployed APs. For each pixel, if there is an AP at its corresponding position, let its value be the AP's ID.
- **AU channel:** This channel indicates the airtime utilization of APs. For each pixel, if there is an AP at its corresponding position, let its value be the AP's AU.
- **Association channel:** This channel indicates the association relationship between APs and connected clients. For each pixel, if

Table 2
Specific settings of DQN.

Layer	Input shape	Kernel	Activation	Output shape
Conv1	$W \times L \times 5$	3×3, 10	ReLU	$(W-2) \times (L-2) \times 10$
Pool1	$(W-2) \times (L-2) \times 10$	2×2	None	$\lceil \frac{W-2}{2} \rceil \times \lceil \frac{L-2}{2} \rceil \times 10$
Conv2	$\lceil \frac{W-2}{2} \rceil \times \lceil \frac{L-2}{2} \rceil \times 10$	3×3, 20	ReLU	$\lceil \frac{W-6}{2} \rceil \times \lceil \frac{L-6}{2} \rceil \times 20$
Pool2	$\lceil \frac{W-6}{2} \rceil \times \lceil \frac{L-6}{2} \rceil \times 20$	2×2	None	$\lceil \frac{\lceil \frac{W-6}{2} \rceil}{2} \rceil \times \lceil \frac{\lceil \frac{L-6}{2} \rceil}{2} \rceil \times 20$
FC1	$\lceil \frac{\lceil \frac{W-6}{2} \rceil}{2} \rceil \times \lceil \frac{\lceil \frac{L-6}{2} \rceil}{2} \rceil \times 20$	N/A	ReLU	512
FC2	512	N/A	ReLU	256
FC3	256	N/A	None	M

there is a client at its corresponding position and the client has connected to an AP, the value is set to that AP's ID.

- **Throughput channel:** This channel indicates the throughput of associated clients. For each pixel, if there is a client at its corresponding position and the client has connected to an AP, the value is set to the client's current throughput.
- **Arriving-client channel:** This channel indicates the position of the arriving client which is currently being handled. For the pixel corresponding to the position of the client, its value is set to 1.

In the above five channels, the values of pixels without any correspondence are all set to 0. In addition, pixels with values other than 0 require normalization due to the different dimensions and value ranges of each channel's encoded information.

Algorithm 1 Offline Training in Wi-OAC

Input: the association vector X^t, throughput vector H^t, RSSI vector R^t, AU vector AU^t, AP position matrix O_A^t, client position matrix O_C^t, and client arrival events $\{i^1, i^2, \ldots, i^T\}$
Output: the trained model

initialize scenario parameters and replay memory $\mathcal{R}$
initialize current Q-network q_c and target Q-network q_t
for training epoch $k = 1$ to Z **do**
  for time step $t = 1$ to T **do**
    form state s_t according to arrival event i^t
    reformulate the state s_t to $\psi(s_t)$
    obtain the available action set A_i^t
    if rand $< \epsilon$ **then**
      select an action $a_t \in A_i^t$ randomly
    else
      select $a_t = \arg\max_{a \in A_i^t} q_c(\psi(s_t), a; \omega)$
    execute action a_t in the environment
    obtain r_t and $\psi(s_{t+1})$ from the environment
    store experience tuple $(\psi(s_t), a_t, r_t, \psi(s_{t+1}))$ in $\mathcal{R}$
    if replay_memory_size >= batch_size **then**
      take $(\psi(s_j), a_j, r_j, \psi(s_{j+1}))$ samples from $\mathcal{R}$
      calculate the loss and train the $q_c(\psi(s), a; \omega)$
    if time step for model updating **then**
      update q_t with the weights of q_c

4.3. Wi-OAC implementation

In this section, we introduce the implementation details of Wi-OAC, including the DQN configuration, action selection, and optimization techniques.

4.3.1. DQN configuration

DQN is the core of decision-making in the Wi-OAC scheme, which is essentially a function mapping: $\psi(s_t) \rightarrow q(\psi(s_t), a; \omega)$. CNN is used as the main structure of DQN as it is an excellent non-linear function approximator. According to a practical theory for designing deep CNN effectively [34], the DQN can be divided into two levels, that is, classifier and feature levels. For the classifier level, we use 3 fully connected layers to achieve better classification results, and the number of neurons in the fully connected layer is 512, 256, and M, respectively. For the feature level, we use 2 convolutional layers and 2 pooling layers. To balance the efficiency of feature extraction and computation complexity, the first and second convolutional layers use 10 and 20 convolution kernels of size 3×3, respectively, and use Rectified Linear Unit (ReLU) as the activation function. The structure of DQN is shown in Fig. 4, and the specific settings of DQN are presented in Table 2.

4.3.2. Action selection

We use the ϵ-greedy strategy for action selection. Specifically, the parameter ϵ is set to guide whether to explore new action or exploit existing policy. For example, we can set $\epsilon = 0.1$ to ensure that the probability of exploration is 10%, and the probability of exploitation is 90%. In this case, all possible actions in a specific state have the opportunities to be selected and executed, which helps the agent learn unknown knowledge and increases the probability of finding the optimal action.

In this paper, we initialize $\epsilon = \epsilon_s$ and gradually decrease it to ϵ_f as the training progresses. The detailed offline training algorithm is illustrated in Algorithm 1. At the beginning, the scenario parameters such as ϵ and α are set to initial values, and the replay memory $\mathcal{R}$ is initialized. Then, the current Q-network and target Q-network are initialized with random weights. There are Z training epochs in the offline training, each of which includes T time steps. At each time step, the agent deals with a new client arrival event. Firstly, the agent transforms the input into the system state according to Eq. (7), which is further reformulated into an image-like tensor. Then, the agent obtains the available action set A_i^t based on the current state and selects an action from A_i^t. Given the action selection strategy mentioned above, the action is determined by a generated random number. If the number is less than ϵ, the agent selects an action randomly, otherwise it selects the action with the maximum Q-value. After the arriving client is associated with the selected AP, the reward and new state are obtained, and the current state $\psi(s_t)$, action a_t, reward r_t and new state $\psi(s_{t+1})$ are stored in the replay buffer. The agent samples a minibatch from $\mathcal{R}$ periodically to calculate the loss and update the weights in the current Q-network. The target Q-network will also be updated with the parameters of the current Q-network at regular intervals. The online association algorithm is depicted in Algorithm 2. First of all, the offline-trained model is loaded, with the help of which the agent can make the online AP-client association decision for each arriving client. The procedure for association decision-making is similar with the offline training. Once a client arrives, the agent forms the system state based on the input, and transforms it into an image-like tensor. Then, it calculates the available action set according to the current state, and selects the action with the maximum Q-value by following the policy learned from the offline training phrase. Finally, it performs the selected action in the environment, and waits for the next client arrival event.

4.3.3. Optimization techniques

In order to further improve the performance of Wi-OAC, two optimization strategies are applied for DQN, which are DDQN [35] and dueling DQN [36].

In nature DQN, the Q-value of the current action is estimated using the maximum value of the Q-value in the next state. Although maximum estimation can quickly bring the Q-value closer to the possible optimization goal, it can also easily lead to overestimation. That is, some actions in partial states may be given overestimated rewards. To solve this problem, DDQN decouples the selection of target Q-value action and the calculation of target Q-value, and thus can usually achieve better performance and higher convergence speed. Specifically, DDQN contains two networks with the same structure and different network

Algorithm 2 Online Association in Wi-OAC

Input: the association vector X^t, throughput vector H^t, RSSI vector R^t, AU vector AU^t, AP position matrix O_A^t, client position matrix O_C^t, and client arrival event i^t

Output: the specified AP a_t

load the trained model
for time step $t = 1$ to ∞ **do**
  form state s_t according to arrival event i^t
  reformulate the state s_t to $\psi(s_t)$
  obtain the available action set A_i^t
  select $a_t = \arg\max_{a \in A_i^t} q_c(\psi(s_t), a; \omega)$
  execute action a_t in the environment
  transform to the next state $\psi(s_{t+1})$

parameters called current Q-network and target Q-network. The former is responsible for selecting actions, and the latter is used to calculate the target Q-values as its network parameters remain unchanged over a while. Notably, the parameters of the target Q-network are always copied from the current Q-network for every certain number of steps.

Another optimization strategy concerns about the fact that although both of the state and action contribute to the Q-value, the degree of influence on it is different. Hence, we hope that the Q-value can reflect the differences in the following two aspects. On the one hand, in some state, the Q-values for choosing different APs should fully reflect the differences. On the other hand, in different states, the rewards for choosing the same AP may differ significantly, so the Q-values for different states should also be well distinguished. Therefore, dueling DQN is used to tackle this issue, which considers dividing the Q-network into two parts, i.e., the value function and the advantage function. Specifically, the value function is only related to the current state and has nothing to do with the adopted action, while the advantage function is related to both the state and action. In this work, two sub-network structures are added before the output layer, corresponding to the value function network and advantage function network respectively. Notably, the final output of the Q-network is a linear combination of the value function's output and the advantage function's output.

5. Simulations

In this section, we validate the effectiveness of our Wi-OAC solution under different scenario scales through simulation experiments.

5.1. Setup

Firstly, the settings of simulation experiments are presented. We consider network scenario as a two-dimensional rectangular area, which is determined by its length L and width W. The positions of APs and clients are randomly generated according to the normal distribution and are different from each other. According to Cisco's report [37], an environment with a large number of concentrated clients (≥1 client every 1.5 m ^{2}) can be defined as high client-density environment. In addition, a typical ratio of the number of clients to the number of AP is 15 : 1. Therefore, we refer to these criteria to set up network scenarios with different scales that are presented in Table 3. In the following, we use the number of clients to represent the corresponding scenario scale, i.e., 45, 75, 105, 135, 165, 195, 225, and 255, respectively.

In the simulation experiments, we adopt the average client throughput as average user QoE metric regardless of specific user application. We set the parameters of path loss model according to the model B of the standard WiFi channel models [38], which represents a typical large open space and office environment. Besides, Gaussian white noise $\mathcal{N}$ is applied to the environment, and $\mathcal{N}$ is set be −82 dBm. We assign

Table 3
Parameter settings of network scales.

Number of clients	Number of APs	Length L & width W(m)
45	3	9, 9
75	5	11, 11
105	7	13, 13
135	9	15, 15
165	11	16, 16
195	13	18, 18
225	15	19, 19
255	17	20, 20

Table 4
Simulation parameters.

Parameters	Values
Path loss model parameters	
Path loss of reference distance $L_0(dB)$	106.73
Breakpoint distance d_{BP}(m)	5
Slope before breakpoint σ_1	2
Slope after breakpoint σ_2	3.5
Scenario parameters	
Channel bandwidth (MHz)	40
AP transmit power P (mW)	100
CCA θ (dBm)	−80
Gaussian white noise $\mathcal{N}$ (dBm)	−82
Algorithm parameters	
Discount factor γ	0.9
Replay memory capacity	1×10^6
Minibatch size	32
Target network update period	200
Activation function	ReLU
Initial exploration probability ϵ_s	1.0
Final exploration probability ϵ_f	0.001

40MHz orthogonal channels to each AP that supports 802.11ac. We set that clients are under the saturated traffic conditions and share spectrum resources in a temporary fairness way. Hence, the client throughput can be obtained through a mapping table of SNR to modulation and coding scheme (MCS) values [39]. The specific simulation parameters are summarized in Table 4.

In addition to the average client throughput, we also evaluate the performance on AP load balancing. To this end, we adopt a balance index based on Jain's Fairness Index [40], which is defined as follows:

$$f = \frac{(\sum_{j=1}^{M} \mathcal{T}_j)^2}{|M| \sum_{j=1}^{M} (\mathcal{T}_j)^2} \tag{12}$$

where $\mathcal{T}_j$ represents the load of AP j. The AP load is usually measured by AU, which refers to the percentage of time when the wireless channel is busy. Since it is difficult to obtain the real-time AU in the simulation environment, the AP load is characterized by the aggregate throughput of AP.

For each set of simulation experiments, the DRL model for online association is trained offline under the corresponding network scenario, and then loaded into the controller for the performance evaluation. Besides, we compare our Wi-OAC solution with two other association control mechanisms, that is, strongest-RSSI-first (SRF), and RL method based on LCF [27].

5.2. Simulation results

According to the above settings, we generate 8 scenarios with different scales, where the number of arriving clients is 45, 75, 105, 135, 165, 195, 225, and 255, respectively. Experiments on each scenario are

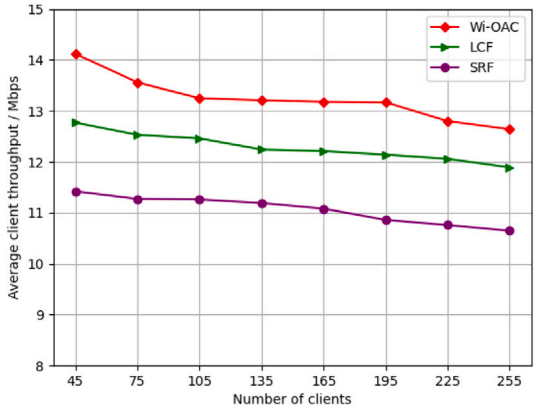


Fig. 5. Performance on average client throughput varying scenario scales.

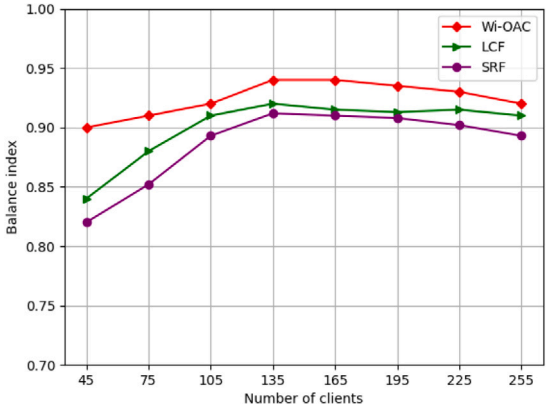


Fig. 6. Performance on AP load balancing varying scenario scales.

carried out three times to eliminate the contingency. The average client throughput and balance index are measured after all clients connect to the WLAN.

Fig. 5 illustrates the average client throughput obtained by the three association control mechanisms when scenario scales are varying. It can be observed that our Wi-OAC scheme achieves significantly higher average client throughput compared to the LCF and SRF schemes on all scenario scales. Moreover, we can also see from the figure that the average client throughput has a slightly decreasing trend with the expansion of network scenarios. It can be explained that more APs are deployed in large-scale scenarios to serve more clients and ensure that user experience will not be reduced significantly. However, the slightly decrease of average client throughput is inevitable, because some clients may connected to an AP that is further away from itself when the scenario expends.

Fig. 6 shows the performance on AP load balancing. It can be seen from the figure that our Wi-OAC scheme achieves the best performance on AP load balancing regardless of scenario scales. That is because the association decision making of Wi-OAC is on the basis of the global network information, and can effectively prevent clients from associating with overloaded APs.

6. Testbed and real-world experiments

In this section, we implement a prototype of Wi-OAC, deploy a realistic high client-density testbed and conduct real-world experiments for further performance validation in terms of average client throughput, AP load balancing and user QoE.

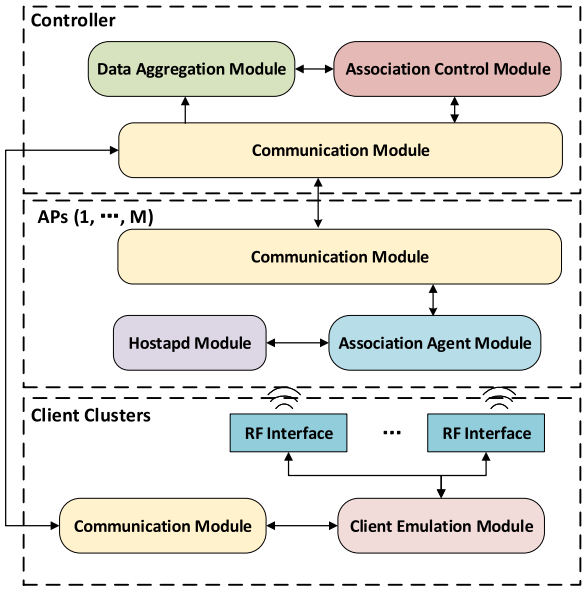


Fig. 7. Architecture of Wi-OAC prototype. The prototype consists of a central controller, multiple APs and several client clusters.

6.1. Wi-OAC prototype and testbed

To verify the effectiveness of our Wi-OAC solution in real-world scenarios, we implement a prototype which consists of a central controller, multiple APs and several client clusters. The architecture is shown in Fig. 7, and the implementation details are explained as follows:

6.1.1. Controller-side implementation

The central controller aggregates the global network status information and makes DRL-aided AP-client association decisions. Specifically, the data aggregation module deals with the status information about APs and clients, which is received by the communication module. The data is stored, and can be utilized as the input of offline training in the association control module. Meanwhile, with the help of the offline-trained model, the association control module makes online association decision for each arriving client according to the current system state tensor. Moreover, a Web-based management system is built on the controller for network manager to configure and manage the WLAN.

6.1.2. AP-side implementation

We adopt the embedded board Compex WPQ864,[1] with Qualcomm QCA9984 chipset as AP devices. The APs are flashed with Linux-based open-source OpenWrt.[2] In the Wi-OAC, the APs not only provide WiFi connections for clients, but also acts as the data collector and command executor for the central controller. Specifically, the association agent model collects the status information about itself and its connected clients, and sends it to the controller through the communication module. In addition, the association agent module forwards probe request frames sent by clients and receives the commands of AP-client association decisions from the controller. Then, the commands are conducted by Hostapd[3] an open-source userspace program integrated in OpenWrt for AP management.

[1] https://compex.com.sg/shop/embedded-board/multi-slots/wpq864/ (last accessed: January 29, 2022).
[2] https://openwrt.org/ (last accessed: January 29, 2022).
[3] https://w1.fi/hostapd/ (last accessed: January 29, 2022).

6.1.3. Client-side implementation

We build a PC equipped with multiple wireless NICs (WNICs), called a client cluster, which can be utilized to emulate multiple clients. Several client clusters can be deployed in the same area, which forms a high client-density environment. In the experiments, the Wi-OAC assigns each WNIC installed on the client clusters to associate with a specified AP. Once connecting to the WLAN, the corresponding independent threads are created to bind with the WNICs, respectively, and perform different kinds of user applications. Meanwhile, the QoE score can be measured for each thread (client).

6.2. QoE metrics

To evaluate the performance of our Wi-OAC solution in real-world scenarios, we consider three types of typical user applications, i.e., file downloading, video playing, and web browsing. In the experiments, the QoE metric is used to represent the users' actual experience and obtained by normalizing the mean opinion scores (MOS). Specifically, we specify a MOS evaluation model for each of the above applications.

6.2.1. File downloading

We use the Iperf3[4] tool to generate TCP downlink traffic, which can simulate the file downloading process, and use the MOS-throughput logarithmic relationship [30] to calculate the MOS. The relationship stems from the assumption that the utility function of elastic traffic (e.g., FTP services) is increasing, strictly concave, and continuously differentiable with respect to throughput. The MOS function for the file downloading application is defined as follows.

$$\mu_f(\eta) = a \times \lg(b\eta) \tag{13}$$

where $\mu_f(\eta)$ is the MOS function for file downloading, parameters a and b are determined by the worst and best quality perceived by the user, and η represents the client throughput.

In this paper, we set the upper limit of client throughput to be 20Mbps, and the corresponding MOS reach the maximum value of 5. While the lower limit of client throughput is set to be 5Mbps, and the value of the corresponding MOS is 1. According to the these settings, we can obtain the values of parameters a and b.

6.2.2. Video playing

We use dashc [31] as the video player, which is a software tool for dynamic adaptive streaming over HTTP (DASH) video. It also includes a lightweight testing function to evaluate the performance of real DASH video stream traffic. During video playback, the MOS following the ITU-T Rec. P.1203 standard can be obtained by the tool dashc, which is calculated according to the video buffering and playback related logs.

6.2.3. Web browsing

We select 37 sites from the top 50 portal sites in China and divide them into two categories according to web pages' actual loading speed, namely short-duration sites and long-duration sites. Then, we use headless Chrome browser with Node.js[5] and Puppeteer[6] to perform automated web browsing. Although several influencing factors have been proposed to account for the QoE of web browsing, the user waiting time, i.e., page load time (PLT), is still the main factor. Therefore, we use the PLT as the primary application service quality indicator. The ITU-T G 1030 single-page web-QoE model [32] defines a logarithmic relationship between PLT and MOS. According to this relationship, the MOS function of web browsing can be defined as follows.

$$\mu_w(\lambda) = \begin{cases} 4.38 - 1.30 \times \ln \lambda & \text{for short-duration sites} \\ 4.79 - 1.03 \times \ln \lambda & \text{for long-duration sites} \end{cases} \tag{14}$$

where $\mu_w(\lambda)$ denotes the MOS function for web browsing and λ represents the PLT.

6.3. Deployment

On the basis of the prototype, we build a Wi-OAC testbed. Fig. 8 depicts the deployment of Wi-OAC testbed. As shown in this figure, the network scenario is in a rectangular room, where a controller, 3 APs and 3 client clusters are deployed in less than 10.5 m^2 area. The details of these devices are presented as follows:

- **Controller:** PC server HP EliteDesk 880 G3TWR, Ubuntu 16.04 LTS 64 bit.
- **AP:** Embedded board Compex WPQ864, wireless module QCA99-84, 4 antennas, Openwrt 19.07.3.
- **Client cluster:** PC with 18 WNICs (Intel 9260/8265/3168), 18 antennas, Ubuntu 18.04 LTS 64 bit.

According to the area of the real-world scenario, the side length of each pixel in the image-like tensor is set to 5 cm. In the experiments, all APs support 802.11ac, and each AP is assigned an 80MHz orthogonal channel. To ensure the normal operation of the association control mechanism described in Section 3.1, we set the same SSID and modified beacon frames with an empty SSID field for all APs, and block APs' direct replies to the probe request frames.

We conduct four groups of experiments, the first three groups are for three kinds of user applications, respectively, i.e., file downloading, video playing, and web browsing, and in the last group, a mixture of the three types of applications are considered.

For each group of experiments, the DRL model is firstly trained offline through the data collected from the corresponding realistic scenario. After that, 20 rounds are performed for each group of experiments by following the same procedure used in the model training, and the mean value is taken as the final result. The detailed procedure is as follows. In each round, the controller makes 54 clients access to the WLAN one by one in a random order. Before each client i is associated, the controller inputs the current system state into the model and outputs the corresponding AP-client association decisions. After client i is associated with the specified AP, it immediately runs the application program, and the running period of the application program is set to be 10 s. After the application program is completed, the client obtains the QoE metric and feeds it back to the controller. Simultaneously, the controller reassigns the application for associated clients to run and selects the next client $i + 1$ to continue associating. The above process is repeated until all 54 clients are connected to the WLAN, and this round of experiments ends. The controller collects the relevant data generated from this round of experiments as the basis for judging the performance of the schemes.

6.4. Real-world experimental results

In the following, the results of four groups of experiments are presented.

6.4.1. File downloading

The traffic of such applications is characterized by user demands for high-throughput and continuous data transmission. In general, as the number of clients connecting to WLAN continues to increase, the overall network throughput increases at a relatively rapid rate. However, when one or several APs reach the load limit, associating clients to such APs will not improve the overall network throughput, and such APs are unable to meet the traffic demand of newly connected clients for file downloading. With uneven load distribution, APs with a high number of associated clients will quickly approach the load limit. Besides, the actual available bandwidth of other clients gradually decreases due to the channel contention among clients, which in turn affects user QoE.

[4] https://iperf.fr/ (last accessed: January 29, 2022).
[5] https://nodejs.org/en/ (last accessed: January 29, 2022).
[6] https://github.com/puppeteer/puppeteer (last accessed: January 29, 2022).

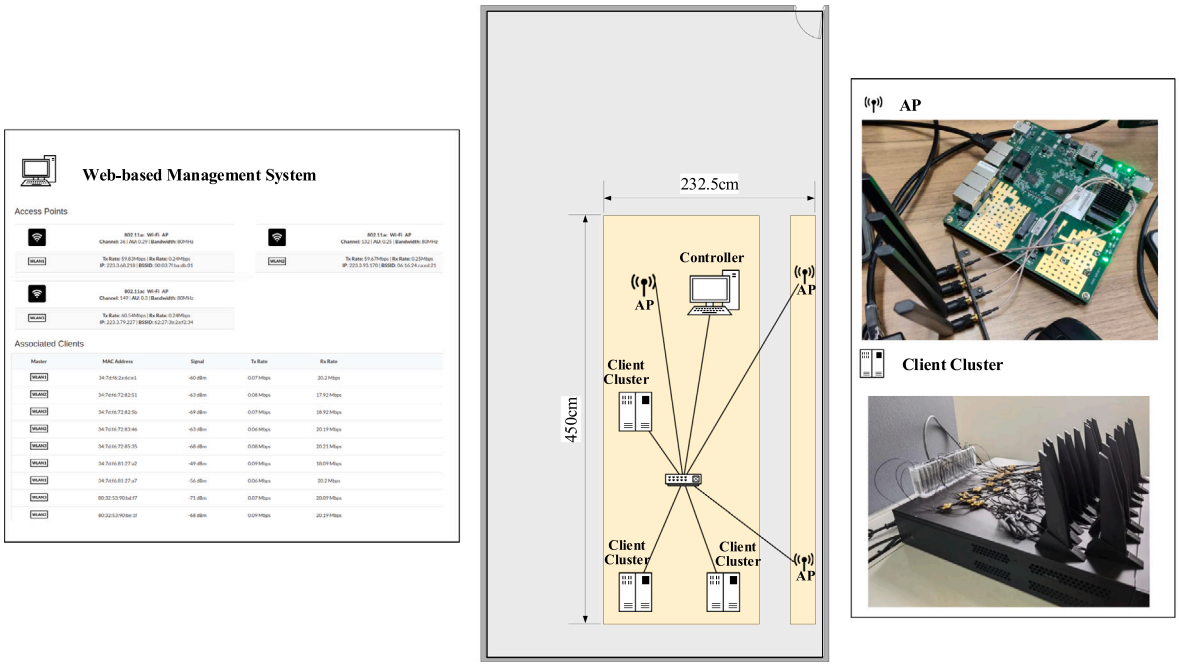
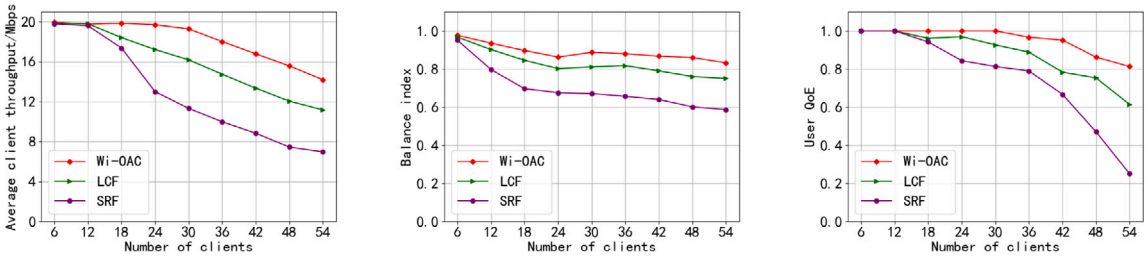


Fig. 8. Deployment of Wi-OAC testbed. The Wi-OAC testbed includes a central controller, 3 APs and 3 client clusters (54 clients) and is deployed in less than 10.5 m^2 area.



(a) Average client throughput (b) AP load balancing (c) User QoE

Fig. 9. Performance on average client throughput, AP load balancing and user QoE for file downloading applications in real-world experiments.

Fig. 9(b) shows that the load balancing of the SRF method has a large gap with the other two methods starting from the arrival of the 12th client, so there is bound to be a severe load imbalance in the network. With the association of subsequent clients, the load imbalance leads to a significant decline in the average throughput. Therefore, as shown in Fig. 9(c), after all clients are connected, the average user QoE of the SRF method is only about 0.2, and its performance is unacceptable.

Combined with Fig. 9(a) and (b), it is obvious that the two types of RL-based association control schemes, i.e., LCF and Wi-OAC, significantly outperform the default SRF method in both average throughput and load balancing, which demonstrates that RL methods that value long-term reward expectations have higher effectiveness and applicability for AP-client association decision-making in high-density WLANs.

Through further comparative analysis, we find that as the number of clients that have connected to the networks increases, the performance of Wi-OAC is always better than that of LCF, and the gap is becoming larger and larger. This trend reveals that the RL approach based on a linear combination of features has its limitations. As shown in Fig. 9(c), the average user QoE of Wi-OAC is still above 0.8, even when all 54 clients have joined the WLAN. Hence, it is demonstrated that the Wi-OAC can provide good user experience for file-downloading applications.

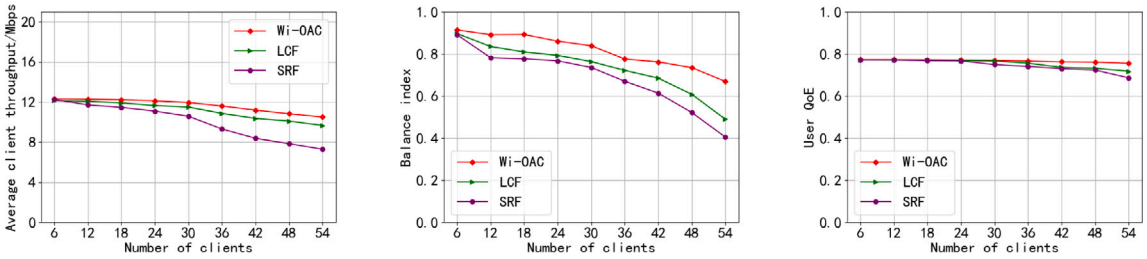
6.4.2. *Video playing*

We use a DASH video source from MMSys18 datasets,[7] and its average throughput requirements are lower than those of the file downloading application. For such video playing applications, our Wi-OAC solution also performs better than other two schemes in terms of average client throughput, AP load balancing, and user QoE, as shown in Fig. 10.

The traffic characteristic of DASH video is that clients no longer generate traffic temporarily after the video buffering is completed, and the user QoE of the application is relatively insensitive to the decrease of the actual throughput. Since DASH always requests the highest quality video clips that the network can support and tries to avoid the events such as buffer exhaustion and frame loss that may seriously impair user experience. Hence, it significantly reduces the impact of network deterioration on user QoE, as shown in Fig. 10(c).

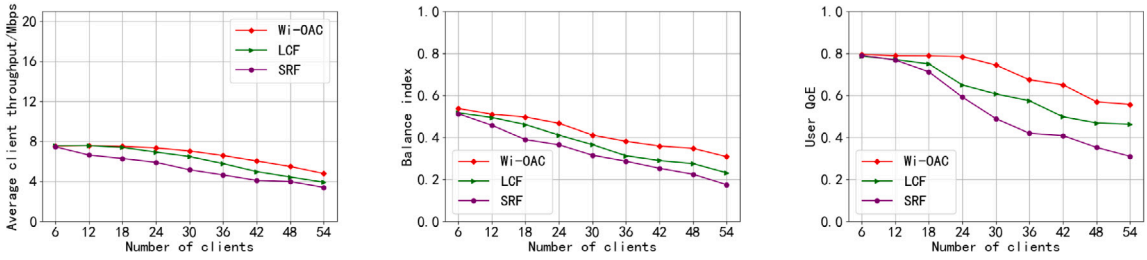
As shown in Fig. 10(a) and (b), the performance on the average client throughput and AP load balancing are similar to the trend of file downloading applications. However, the value of average client throughput is significantly smaller, which is consistent with its throughput requirements. In addition, the balancing index is lower in this group of experiments, mainly due to its different traffic generation pattern from that of the file downloading applications. Clients may

[7] http://ftp.itec.aau.at/datasets/mmsys18/ (last accessed: January 29, 2022).



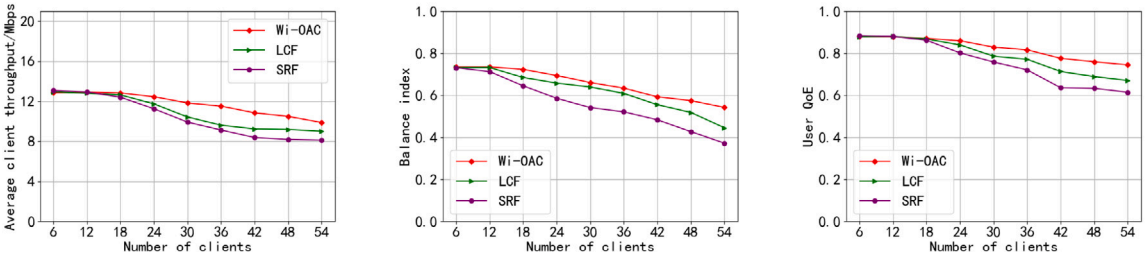
(a) Average client throughput (b) AP load balancing (c) User QoE

Fig. 10. Performance on average client throughput, AP load balancing and user QoE for video playing applications in real-world experiments.



(a) Average client throughput (b) AP load balancing (c) User QoE

Fig. 11. Performance comparison on average client throughput, AP load balancing and user QoE for web browsing applications in real-world experiments.



(a) Average client throughput (b) AP load balancing (c) User QoE

Fig. 12. Performance on average client throughput, AP load balancing and user QoE for mixed applications in real-world experiments.

have zero traffic demand at some time, which makes its overall load balancing slightly lower than that of applications where traffic demands are always present.

6.4.3. *Web browsing*

The traffic characteristic of web browsing applications is that the average throughput requirements are the lowest among the three types of applications involved in this paper, and the traffic requirements are time-variant during the web page loading process depending on the specific web page. For such web browsing applications, our Wi-OAC solution still performs best among these three client association schemes on average client throughput, AP load balancing, and user QoE, as shown in Fig. 11.

It can be seen from Fig. 11(a) that the average client throughput of web browsing applications align with the above traffic characteristic. In addition, the traffic of web browsing applications is considered during the time from when the browser is assigned a URL to when the multimedia content is fully loaded. In fact, clients' traffic demands in each step vary greatly, and the number and proportion of static resources and dynamic resources in different web pages are also quite different. Due to these differences, different browser processes may execute different stages of page loading at the same time. Even if they execute the same stage, their throughput may still be different. This leads to the large difference on real-time throughput of clients, which ultimately makes the AP load balancing at a low level, as presented in Fig. 11(b).

6.4.4. *Mixed applications*

In the group of experiments, every arriving client selects one of the above three types of applications, and we assure that the number of clients for each kind of applications is roughly the same at every moment. Therefore, the traffic characteristics of mixed applications are the average and synthesis of the above three types of applications' traffic characteristics when they are executed separately. For example, the trend and value of average client throughput in Fig. 12(a) are similar to the average of results in the above three groups of experiments.

By running the mixed applications, we obtain the results consistent with those of the three groups of experiments described above. Besides, the scenario of densely deployed clients running a mix of three typical applications with high, medium, and low throughput requirements is a better representation of the real user experience in a high client-density environment. As is shown in Fig. 12, compared with the LCF and SRF methods, the curve representing Wi-OAC has a slower decay rate and higher values on different aspects, i.e., average client throughput, AP load balancing, and user QoE, while clients are continuously arriving, which indicates that Wi-OAC can effectively improve user QoE in a high client-density environment.

7. Conclusion

In this paper, we investigated the online centralized association control problem in high-client density WLANs with the objective of

improving average user QoE metric. To solve this issue, a deep reinforcement learning aided solution, called Wi-OAC, was proposed, where the client association control problem was regarded as a sequential decision problem driven by client arrival events. Firstly, we presented the framework of Wi-OAC based on MDP, including the offline training and online association. With the help of the offline-trained model, Wi-OAC could assign an appropriate AP to serve each arriving client in the online association phase. Given the complexity of state space, the state reformulation scheme was designed to transform the initial representation of system state into an image-like pattern. Besides, the DDQN and dueling DQN strategies were combined to accelerate the convergence. After that, both simulation experiments and real-world experiments were conducted to evaluate the performance of Wi-OAC. In the real-world experiments, a Wi-OAC testbed was built with 3 APs and 54 clients in less than 10.5 m^2 area. The experimental results demonstrated that Wi-OAC could significantly improve the performance in terms of average client throughput, AP load balancing and user QoE.

In the future, we will study dynamic client migration mechanism so as to adapt to time-variant network environment, and further optimize network performance and user QoE.

CRediT authorship contribution statement

Wenjia Wu: Conceptualization, Methodology. **Yujing Liu:** Investigation, Writing – original draft. **Jiazhi Yao:** Software, Validation. **Xiaolin Fang:** Resources, Data curation. **Feng Shan:** Formal analysis. **Ming Yang:** Visualization. **Zhen Ling:** Writing – review & editing. **Junzhou Luo:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (Nos. 62072102, 62132009, 62072103, 61972083, 62072101, 62022024, 61972088, and 62061146001); Jiangsu Provincial Key Laboratory of Network and Information Security, China (No. BM2003201); the Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China (No. 93K-9); the Fundamental Research Funds for the Central Universities (No. 2242022k30029).

References

[1] Cailian Deng, Xuming Fang, Xiao Han, Xianbin Wang, Li Yan, Rong He, Yan Long, Yuchen Guo, IEEE 802.11be Wi-Fi 7: New challenges and opportunities, IEEE Commun. Surv. Tutor. 22 (4) (2020) 2136–2166.

[2] Imad Jamil, Laurent Cariou, Jean-François Hélard, Efficient MAC protocols optimization for future high density WLANS, in: 2015 IEEE Wireless Communications and Networking Conference, WCNC, 2015, pp. 1054–1059.

[3] Cisco, Cisco annual internet report (2018–2023) white paper, 2020, [EB/OL], https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf.

[4] Gianluca Cena, Stefano Scanzio, Adriano Valenzano, Improving effectiveness of seamless redundancy in real industrial Wi-Fi networks, IEEE Trans. Ind. Inf. 14 (5) (2017) 2095–2107.

[5] Daan Weller, Raoul Dijksman Mensenkamp, Arjan van der Vegt, Jan-Willem van Bloem, Cees de Laat, Wi-Fi 6 performance measurements of 1024-QAM and DL OFDMA, in: ICC 2020-2020 IEEE International Conference on Communications, ICC, IEEE, 2020, pp. 1–7.

[6] Wei Li, Shengling Wang, Yong Cui, Xiuzhen Cheng, Ran Xin, Mznah A. Al-Rodhaan, Abdullah Al-Dhelaan, AP association for proportional fairness in multirate WLANs, IEEE/ACM Trans. Netw. 22 (1) (2014) 191–202.

[7] Ouldooz Baghban Karimi, Jiangchuan Liu, Jennifer Rexford, Optimal collaborative access point association in wireless networks, in: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, IEEE, 2014, pp. 1141–1149.

[8] Fengyuan Xu, Xiaojun Zhu, Chiu C. Tan, Qun Li, Guanhua Yan, Jie Wu, Smartassoc: Decentralized access point selection algorithm to improve throughput, IEEE Trans. Parallel Distrib. Syst. 24 (12) (2013) 2482–2491.

[9] Omneya Issa, Ying Ge, Aizaz U. Chaudhry, Bernard Doray, User association algorithm for throughput improvement in high-density wireless networks, in: 2017 26th International Conference on Computer Communication and Networks, ICCCN, IEEE, 2017, pp. 1–8.

[10] Phillip B. Oni, Steven D. Blostein, Decentralized AP selection in large-scale wireless LANs considering multi-AP interference, in: 2017 International Conference on Computing, Networking and Communications, ICNC, IEEE, 2017, pp. 13–18.

[11] Hyunsoon Kim, Woonghee Lee, Mungyu Bae, Hwangnam Kim, Wi-Fi seeker: A link and load aware AP selection algorithm, IEEE Trans. Mob. Comput. 16 (8) (2016) 2366–2378.

[12] Thi Ha Ly Dinh, Megumi Kaneko, Keisuke Wakao, Kenichi Kawamura, Takatsune Moriyama, Hirantha Abeysekera, Yasushi Takatori, Distributed user-to-multiple access points association through deep learning for beyond 5G, Comput. Netw. (2021) 108258.

[13] Yigal Bejerano, Seung-Jae Han, Li Li, Fairness and load balancing in wireless LANs using association control, IEEE/ACM Trans. Netw. 15 (3) (2007) 560–573.

[14] Wooi King Soo, Teck-Chaw Ling, Aung Htein Maw, Su Thawda Win, Survey on load-balancing methods in 802.11 infrastructure mode wireless networks for improving quality of service, ACM Comput. Surv. 51 (2) (2018) 1–21.

[15] Rohan Murty, Jitendra Padhye, Ranveer Chandra, Alec Wolman, Brian Zill, Designing high performance enterprise Wi-Fi networks, in: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, Vol. 8, NSDI, 2008, pp. 73–88.

[16] Jun Zhang, Guangxing Zhang, Qinghua Wu, Binbin Liao, Gaogang Xie, A data-driven approach to client-transparent access selection of dual-band WiFi, IEEE Trans. Netw. Serv. Manag. 16 (1) (2018) 321–333.

[17] Alessandro Raschellà, Faycal Bouhafs, Michael Mackay, Qi Shi, Jorge Ortín, José Ramón Gállego, Maria Canales, A dynamic access point allocation algorithm for dense wireless lans using potential game, Comput. Netw. 167 (2020) 106991.

[18] Suzan Bayhan, Estefanía Coronado, Roberto Riggio, Anatolij Zubow, User-AP association management in software-defined WLANs, IEEE Trans. Netw. Serv. Manag. 17 (3) (2020) 1838–1852.

[19] Xi Huang, Shuang Zhao, Xin Gao, Ziyu Shao, Hua Qian, Yang Yang, Online user-AP association with predictive scheduling in wireless caching networks, IEEE Trans. Mob. Comput. (2020) 1.

[20] Wangkit Wong, Kam-Wa Chau, S.-H. Gary Chan, Joint client association and random access control for MU-MIMO WLANs, IEEE Trans. Mob. Comput. 19 (12) (2020) 2818–2832.

[21] Blas Gómez, Estefanía Coronado, José M. Villalón, Roberto Riggio, Antonio Garrido, WiMCA: Multi-indicator client association in software-defined Wi-Fi networks, Wirel. Netw. (2021) 1–17.

[22] Xin Jian, Langyun Wu, Keping Yu, Moayad Aloqaily, Jalel Ben-Othman, Energy-efficient user association with load-balancing for cooperative IIoT network within B5G era, J. Netw. Comput. Appl. (2021) 103110.

[23] Wangkit Wong, Avishek Thakur, S.-H. Gary 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, 2016, pp. 1–9.

[24] Apurv Bhartia, Bo Chen, Derrick Pallas, Waldin Stone, Clientmarshal: Regaining control from wireless clients for better experience, in: The 25th Annual International Conference on Mobile Computing and Networking, 2019, pp. 1–16.

[25] Fan Meng, Peng Chen, Lenan Wu, Power allocation in multi-user cellular networks with deep Q learning approach, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6.

[26] Shangxing Wang, Hanpeng Liu, Pedro Henrique Gomes, Bhaskar Krishnamachari, Deep reinforcement learning for dynamic multichannel access in wireless networks, IEEE Trans. Cognit. Commun. Network. 4 (2) (2018) 257–265.

[27] Mohamed Amine Kafi, Alexandre Mouradian, Véronique Vèque, On-line client association scheme based on reinforcement learning for WLAN networks, in: 2019 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2019, pp. 1–7.

[28] Yiding Yu, Taotao Wang, Soung Chang Liew, Deep-reinforcement learning multiple access for heterogeneous wireless networks, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1277–1290.

[29] Thi Ha Ly Dinh, Megumi Kaneko, Keisuke Wakao, Kenichi Kawamura, Takatsune Moriyama, Hirantha Abeysekera, Yasushi Takatori, Deep reinforcement learning-based user association in sub6GHz/mmWave integrated networks, in: 2021 IEEE 18th Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2021, pp. 1–7.

[30] Andre B. Reis, Jacob Chakareski, Andreas Kassler, Susana Sargento, Distortion optimized multi-service scheduling for next-generation wireless mesh networks, in: 2010 INFOCOM IEEE Conference on Computer Communications Workshops, IEEE, 2010, pp. 1–6.

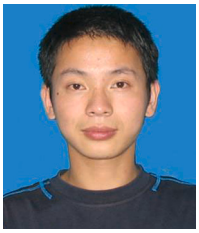
[31] Aleksandr Reviakin, Ahmed H. Zahran, Cormac J. Sreenan, Dashc: A highly scalable client emulator for DASH video, in: Proceedings of the 9th ACM Multimedia Systems Conference, 2018, pp. 409–414.

[32] ITUT Rec, G. 1030-Estimating End-to-End Performance in IP Networks for Data Applications, Vol. 42, International Telecommunication Union, Geneva, Switzerland, 2005.
[33] Han Zou, Ming Jin, Hao Jiang, Lihua Xie, Costas J. Spanos, WinIPS: WiFi-based non-intrusive indoor positioning system with online radio map construction and adaptation, IEEE Trans. Wireless Commun. 16 (12) (2017) 8118–8130.
[34] Xudong Cao, A Practical Theory for Designing Very Deep Convolutional Neural Networks, 2015, Unpublished Technical Report.
[35] Hado Van Hasselt, Arthur Guez, David Silver, Deep reinforcement learning with double Q-learning, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 2094–2100.
[36] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, Nando Freitas, Dueling network architectures for deep reinforcement learning, in: International Conference on Machine Learning, 2016, pp. 1995–2003.
[37] Cisco, Wireless high client density design guide, 2018, [EB/OL], https://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-7/b_wireless_high_client_density_design_guide.html.
[38] Vinko Erceg, Laurent Schumacher, Persefoni Kyritsi, et al., IEEE 802.11-03/940R4 TGn channel models, vol. 11, 2004, IEEE P802.
[39] Andrew von Nagy, Wi-Fi SNR to MCS data rate mapping reference, 2014, [EB/OL], http://revolutionwifi.blogspot.com/2014/09/wi-fi-snr-to-mcs-data-rate-mapping.html.
[40] Rajendra K. Jain, Dah-Ming W. Chiu, William R. Hawe, et al., A Quantitative Measure of Fairness and Discrimination, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, 1984.

Wenjia Wu received the B.S. and Ph.D. degrees in computer science in 2006 and 2013, respectively, from Southeast University. He is an associate professor at the School of Computer Science and Engineering in Southeast University. His research interests include wireless and mobile networks.

Yujing Liu received the B.S. degree from Southeast University, Nanjing, China, in 2020. She is currently a M.S. student in the School of Computer Science and Engineering in Southeast University, Nanjing, China. Her research interests include wireless and mobile networks.

Jiazhi Yao received the B.E. degree from Fuzhou University, Fuzhou, China, in 2014, and the M.E. degree from Southeast University, Nanjing, China, in 2021. His research interests include wireless and mobile networks.



Xiaolin Fang received the B.S. degree from Harbin Engineering University, China in 2007. And he received the M.S. and Ph.D. degree from Harbin Institute of Technology, Harbin, China, in 2009 and 2014, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. His research interests include sensor networks, data processing, image processing, and scheduling.

Feng Shan received the Ph.D. degree in computer science from Southeast University, Nanjing, China, in 2015. He was a visiting student with the School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA, from 2010 to 2012. He is currently an Associate Professor with the School of Computer Science and Engineering, Southeast University. His research interests include the areas of Internet of Things, wireless networks, swarm intelligence, and algorithm design and analysis.

Ming Yang received the M.S. and Ph.D. degrees in computer science and engineering at Southeast University, China, in 2002 and 2007, respectively. He is currently a professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. His main research interests include network security, privacy and Internet of Things.

Zhen Ling received the B.S. degree (2005) and Ph.D. degree (2014) in Computer Science from Nanjing Institute of Technology, China and Southeast University, China, respectively. He is a full professor in the School of Computer Science and Engineering, Southeast University, Nanjing, China. He won ACM China Doctoral Dissertation Award and China Computer Federation (CCF) Doctoral Dissertation Award, in 2014 and 2015, respectively. His research interests include artificial intelligence of things, mobile system security, network security and privacy, and trusted computing.

Junzhou Luo received the B.S. degree in applied mathematics and the M.S. and Ph.D. degrees in computer network, all from Southeast University, China, in 1982, 1992, and 2000, respectively. He is a full professor in the School of Computer Science and Engineering, Southeast University. He is a member of the IEEE Computer Society and co-chair of IEEE SMC Technical Committee on Computer Supported Cooperative Work in Design, and he is a member of the ACM and chair of ACM SIGCOMM China. His research interests are next generation network architecture, network security, cloud computing, and wireless LAN.