Computer Networks 197 (2021) 108258

Contents lists available at ScienceDirect

Computer Networks

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

Distributed user-to-multiple access points association through deep learning for beyond 5G

Thi Ha Ly Dinh [a,*], Megumi Kaneko [a], Keisuke Wakao [b], Kenichi Kawamura [b], Takatsune Moriyama [b], Hirantha Abeysekera [b], Yasushi Takatori [b]

[a] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan
[b] Access Network Service Systems Laboratories, NTT Corporation, 1-1 Hikari-no-oka, Yokosukashi, Kanagawa, 239-0847, Japan

ARTICLE INFO

Keywords:
User association
User-to-multiple access points association
Deep reinforcement learning
Deep Q-learning

ABSTRACT

Future wireless networks will be facing unprecedented difficulties arising from mobile traffic growth, network densification, as well as diversification of applications and services. Indeed, future user devices are expected to integrate diverse radio interfaces such as 5G, WBAN or IoT, enabling each user to be served a wide range of applications at any time. This poses significant challenges in terms of wireless resource sharing and interference management, as more and more stringent Quality of Service (QoS) constraints should be jointly satisfied in dense interfering environments. Furthermore, future networks are expected to be highly autonomous and decentralized. To meet these challenges, this work proposes distributed user-to-multiple Access Points (AP) association methods, where the objective is to maximize the long-term sum-rate subject to application QoS constraints, as well as to AP load constraints. Our distributed methods enable each user to leverage their Deep Reinforcement Learning (DRL) capabilities, in particular Deep Q-Learning (DQL), to self-optimize their APs' selection solely based on their local network state knowledge, so as to best satisfy their diverse requirements. Numerical results show that, compared to baseline schemes, the proposed methods enable global throughput enhancements while reducing user QoS outage probabilities, even in large and dense networks.

1. Introduction

1.1. Background

Future Beyond 5G (B5G) networks are expected to support extreme amounts of mobile data traffic, owing to the ever increasing number of mobile subscribers worldwide, a tendency accelerated by the spread of Internet of Things (IoT) and Internet of Everything (IoE) services [1,2]. Future networks should hence enable a seamless and versatile provision of a plethora of new applications and services, characterized by a large variety of Quality of Service (QoS) constraints encompassing ultra-high data rates, ultra-low latency, extreme reliability or massive connectivity. Furthermore, next generation networks will be integrating different wireless technologies within each Access Point (AP) and user device, for instance, 5G, WLAN, WBAN or IoT radio interfaces. Thus, multiple applications with various QoS types and levels, corresponding to each wireless interface, may be requested by each user device simultaneously. Indeed, in B5G systems, each user is expected to make use of multiple applications simultaneously, each with its specific QoS requirements, e.g., Augmented Reality (AR) or Extreme Reality (XR) applications with high data rates, wireless 3D sensing through IoT devices with low latency, or health monitoring requiring high reliability. All these diverse and stringent QoS requirements can hardly be fully satisfied by restricting each user to associate with only one AP. Thus, by allowing each user to connect to different radio interfaces across different APs, much higher user satisfaction levels may be achieved. Moreover, future systems are expected to be more and more autonomous through self-learning and self-optimization, as centralized optimization of such dense and complex networks will quickly become intractable.

1.2. Related work

In such a context, new radio resource allocation and interference management techniques are needed in order to meet the technical challenges of such dense distributed networks. In particular, devising efficient user-to-AP association methods is essential for guaranteeing

* Corresponding author.
E-mail addresses: halydinh@nii.ac.jp (T.H.L. Dinh), megkaneko@nii.ac.jp (M. Kaneko), keisuke.wakao.gm@hco.ntt.co.jp (K. Wakao), kenichi.kawamura.dn@hco.ntt.co.jp (K. Kawamura), takatsune.moriyama.fe@hco.ntt.co.jp (T. Moriyama), hirantha.abeysekera.eu@hco.ntt.co.jp (H. Abeysekera), yasushi.takatori.rk@hco.ntt.co.jp (Y. Takatori).

https://doi.org/10.1016/j.comnet.2021.108258
Received 18 March 2021; Received in revised form 19 May 2021; Accepted 21 June 2021
Available online 24 June 2021

high system performance in terms of achievable throughput, spectrum-efficiency, network load balancing or user fairness [3]. However, most previous works only allow each user device to be connected to a unique AP simultaneously [4–17]. Among them, many works adopt a centralized optimization approach, such as [4] for realizing load balancing and [5] for optimizing energy efficiency. However, these methods require global and perfect channel state information knowledge of all the links at the centralized scheduler. Moreover, they incur prohibitively high costs in terms of computational complexity, power and signaling overhead, making them unsuitable for B5G networks. More recently, a centralized data-driven approach was proposed in [6], where a robust optimal user association map, pre-calculated at the BS, is used to determine the actual served users in real-time. However, this method is limited to the specific area whose optimal association data is available before hand, making it difficult to generalize to other regions.

To overcome the drawbacks of these centralized methods, distributed approaches with partial knowledge of the network environment have been considered, such as a matching theory-based method in [7], a bandit-based method in [8] or a semi-distributed method based on Alternating Direction Method of Multipliers (ADMM) [9]. However, these methods require significant time to converge as the network size grows, making them difficult to apply to large-scale networks and delay-stringent applications. This is why more recently, Deep Learning (DL)-based user association methods have been proposed. For instance, making use of a pre-trained Deep Neural Network (DNN), Ref. [10] designed a centralized user association scheme that can provide a real-time solution through Mobile Edge Computing (MEC). However, this scheme requires a long training phase, for gathering large amounts of historical data of the global network environment. A method based on an actor–critic DL for efficient joint user association and bandwidth allocation in a dense downlink mobile network was proposed in [11], while [12] designed a method using Deep Deterministic Policy Gradient in the context of online video streaming services with MEC. Furthermore, [13] provided an online Deep Reinforcement Learning (DRL) method using multiple DNNs to generate solutions for the training data set of the user association problem in heterogeneous systems. However, this method also requires the channel information of the whole network at the input of the DNNs. Refs. [14–17] also designed learning-based methods for distributed user association, where it is generally assumed that each user has knowledge of the QoS satisfaction status of all other users in the network, which is unrealistic.

1.3. Problem definition

One major observation is that, all these methods do not allow each user to be associated to multiple APs simultaneously, which is a fundamental limitation that hinders the joint satisfaction of heterogeneous types of applications and services, as will be required in future networks. To counter this drawback, we have developed in our previous work [18] a Q-Learning (QL)-based distributed user association method, where each user is able to learn its best set of APs to be connected to at any time, solely based on local knowledge of its surrounding wireless environment. It is worthwhile noting that the considered user-to-multiple APs association is fundamentally different to Coordinated Multipoint (CoMP)-like approaches, as in our case all APs are uncoordinated and take their allocation and association decisions independently. Although QL guarantees convergence towards the optimal policy as long as all states and actions are visited often enough, this method is hardly applicable to scenarios with large state/action spaces. To address this essential problem, we proposed in [19] a user-to-multiple APs association method exploiting DRL, and in particular, Deep Q-Learning (DQL) based on Deep Q-Networks [20], in the context of Sub 6 GHz/mmWave integrated networks. This method leverages the 6G vision of AI-enabled devices, whereby not only cloud/edge servers, but also user devices themselves would be equipped by DL capabilities through an embedded Deep Neural Network (DNN) [1,2]. We also showed in [19] the effectiveness of the DQN-based approach to handle this issue. However, an intrinsic drawback of DQN is the overestimation issue of Q-values which introduces bias in the optimal action selection. As explained in [21], this problem can be efficiently tackled by Double DQN (DDQN) which makes use of different DQNs for Q-value estimation and action selection.

1.4. Our contributions

Therefore, in this work, we build upon these preliminary studies by investigating the issue of user-to-multiple APs association, but under much more complex and challenging settings. Essentially, these initial schemes are extended for handling heterogeneous types of user QoS requirements in dense and large-scale interfering systems. While our previous works [18,19] focused on only one type of QoS requirement, i.e., a minimum rate requirement, we extend our proposed method to handle joint rate and delay requirements. This goes along the line of B5G challenges, where multiple applications with diverse QoS demands should be fulfilled simultaneously. Furthermore, we also leverage more advanced DRL tools, namely the Double Deep Q-Network (DDQN) technique which was proposed in [21] as an enhancement of the initial DQN method in [20], based on which new distributed association methods are designed.

Our main contributions are listed as follows.

(1) The considered user-to-multiple APs association issue is formulated as a long-term sum-rate maximization problem, but unlike [18,19], we now consider various constraints, including two different types of QoS requirements: minimum rate and maximum delay. This gives rise to an intractable combinatorial optimization problem even in a centralized setting, which becomes even more challenging in a distributed setting.

(2) To efficiently handle this problem in a distributed manner and in the context of a large-scale network, we propose a DRL-based user-to-multiple APs association method, where each user makes use of its DNN to learn the best set of APs to be simultaneously connected to, in order to fulfill its diverse QoS requirements. In addition to the DQN-based method developed in our previous work [19], we also propose a method leveraging the DDQN technique to further enhance the network performance. Moreover, two types of association algorithms are developed: the first, termed *Fully Distributed-DQN (Fully Distributed-DDQN)* method, is based on minimal decision feedback from APs towards users, and the second, termed *Partially Distributed-DQN (Partially Distributed-DDQN)* method, further improves the achievable network performance by letting each user acquire additional local information regarding its neighboring user requirements.

(3) Finally, the proposed algorithms are evaluated over various network settings and compared to several performance benchmarks including a greedy method, and basic DQN/DDQN-based methods with single AP association. The numerical results demonstrate the effectiveness of our proposed methods, as they not only improve the total sum-rate of the whole system as compared to conventional methods, but also enhance the user satisfaction level by decreasing the QoS outage probabilities and enhancing the outage fairness among different applications with diverse requirements.

1.5. Organization

The remainder of the paper is organized as follows. In Section 2, we describe our system model, based on which the user-to-multiple APs association problem is formulated in Section 3. Then, in Section 4, our proposed algorithms are presented in detail. Numerical results are discussed in Section 5.3. Finally, Section 6 concludes this paper and gives directions for future work.

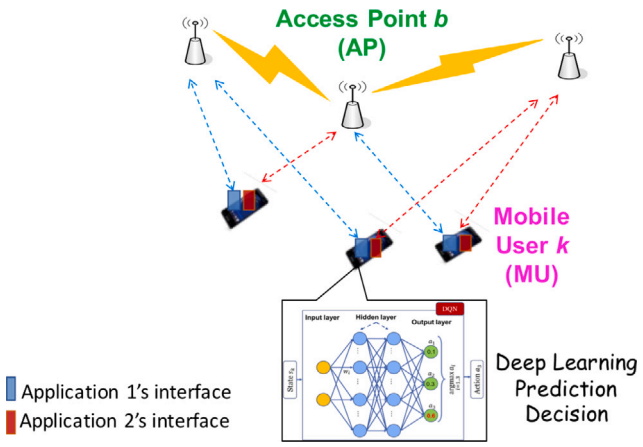


Fig. 1. Proposed User-to-multiple AP association.

2. System model

We consider a downlink network composed of a set $\mathcal{B}$ of fixed APs and a set $\mathcal{K}$ of randomly located users, as depicted in Fig. 1. All APs operate on the same bandwidth for sake of spectrum efficiency, but interfere among each other. In each scheduling frame t, each user $k \in \mathcal{K}$ requests a set of applications $\mathcal{F}_k$, in which each application f requests either a minimum rate constraint R_{kf} or a maximum delay time constraint T_{kf}. For convenience, we denote the set of applications requesting R_{kf} as $\mathcal{F}_{kr}$ and the set of applications requesting T_{kf} as $\mathcal{F}_{fd}$. Hence, $\mathcal{F}_{kr} \cup \mathcal{F}_{fd} = \mathcal{F}_k$ and $\mathcal{F}_{kr} \cap \mathcal{F}_{kd} = \emptyset$.

The achievable data rate $r_{bkf}(t)$ at user k for application f provided by AP b at frame t is computed by

$$r_{bkf}(t) = W \log(1 + \gamma_{bkf}(t)), \tag{1}$$

where W is the bandwidth. $\gamma_{bkf}(t)$ denotes the Signal to Interference-plus-Noise Ratio (SINR) at user k with application f for the transmit signal from AP b, which is given as

$$\gamma_{bkf}(t) = \frac{|h_{bk}(t)|^2 p_{bkf}(t)}{\sum_{b' \in \mathcal{B}\setminus b} |h_{b'k}(t)|^2 p_{b'}(t) + W\sigma_n^2}. \tag{2}$$

Here $h_{bk}(t) \in \mathbb{C}$ denotes the complex channel coefficient between AP b and user k, including path loss and small-scale fading effects; $p_{bkf}(t) \in \mathbb{R}^+$ is the transmit power from AP b to user k for application f, which is assumed to be known and fixed. The term σ_n^2 denotes the Additive White Gaussian Noise (AWGN) power. Since they use the same bandwidth, all other APs $b' \in \mathcal{B}\setminus\{b\}$ cause interference to user k served by AP b, with full transmit power $p_{b'}$.

Next, the delay time d_{bkf} required for serving application f of user k by AP b at frame t is given by

$$d_{bkf}(t) = \frac{s_f}{r_{bkf}(t)}, \tag{3}$$

where s_f denotes the file size of application f.

All APs are assumed to be able to serve any requested application unless their maximum load limit is violated. As in [14], the load of AP b for serving application f of user k is computed by

$$\phi_{bkf}(t) = \frac{m_{kf}}{\mu_{kf} r_{bkf}(t)}, \tag{4}$$

where m_{kf} and $\frac{1}{\mu_{kf}}$ denote the mean arrival rate in number of packets per seconds, and the mean packet size of application f in bits, respectively. Hence, assuming an orthogonal allocation of wireless resources in frequency or time for serving user applications as in [14], AP b will become overloaded if its total load exceeds the normalized value of 1, namely if

$$\Phi_b(t) = \sum_{k\in\mathcal{K}} \sum_{f\in\mathcal{F}_k} x_{bkf}(t)\phi_{bkf}(t) > 1, \tag{5}$$

where $x_{bkf}(t)$ is an association variable defined as,

$$x_{bkf}(t) = \begin{cases} 1, & \text{if AP } b \text{ serves application } f \text{ of} \\ & \text{user } k \text{ at frame } t, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

3. Problem formulation

In this work, we consider the long-term global average sum-rate maximization problem, under user QoS constraints in terms of minimum data rate and maximum delay time, and AP load constraints, which is formulated as

$$\max_{x_{bkf}(t)} \quad \lim_{T\to\infty} \left[\frac{1}{T} \sum_{t=1}^{T} \sum_{b\in\mathcal{B}} \sum_{k\in\mathcal{K}} \sum_{f\in\mathcal{F}_k} x_{bkf}(t) r_{bkf}(t) \right], \tag{7}$$

$$\text{s.t. } x_{bkf}(t) \in \{0,1\}, \forall b \in \mathcal{B}, k \in \mathcal{K}, f \in \mathcal{F}_k, \tag{7a}$$

$$\sum_{b\in\mathcal{B}} x_{bkf}(t) = 1, \quad \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_k, \tag{7b}$$

$$\sum_{b\in\mathcal{B}} x_{bkf}(t) r_{bkf}(t) \geq R_{kf}, \quad \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_{kr}, \tag{7c}$$

$$\sum_{b\in\mathcal{B}} x_{bkf}(t) d_{bkf}(t) \leq T_{kf}, \quad \forall k \in \mathcal{K}, \forall f \in \mathcal{F}_{kd}, \tag{7d}$$

$$\Phi_b(t) = \sum_{k\in\mathcal{K}} \sum_{f\in\mathcal{F}_k} x_{bkf}(t)\phi_{bkf}(t) \leq 1, \quad \forall b \in \mathcal{B}. \tag{7e}$$

The objective function (7) expresses the long-term average sum-rate over all applications, users and APs in the network. Constraint (7a) sets the binary nature of association variables $x_{bkf}(t)$ defined in (6). Eq. (7b) constrains each application requested by each user to be served by a unique AP. The minimum data rate and the maximum delay time for each application are specified by (7c) and (7d), respectively. Finally, the load constraint for each AP b is reflected in (7e).

Problem (7) is a combinatorial optimization problem which cannot be solved in polynomial time. This becomes especially intricate in a B5G setting where a large number of users with conflicting QoS constraints and creating high interference levels, should be simultaneously satisfied. Furthermore, distributed association methods based on local network and channel state information, are deemed necessary. To meet these goals, we propose to leverage self-learning and self-optimization by exploiting deep reinforcement learning at the user side, as explained in the next section.

4. Proposed algorithms

We propose two distributed association approaches based on DQL, whereby each user is an agent who takes its own decisions (action) based on its current state and reward, computed based on its local knowledge of the wireless environment. Similarly to [18], the state and action spaces of each agent are defined as follows.

- **User State:** The state $\mathbf{s}_k(t)$ of agent k in state space $\mathcal{S}_k$ is the current association between required applications f of user k and each AP b at the beginning of frame t, i.e,

$$\mathbf{s}_k(t) \in \mathcal{S}_k = \left\{ x_{bkf}(t), \forall b \in \mathcal{B}, \forall f \in \mathcal{F}_k \right\}. \tag{8}$$

Due to (7b) which constrains each application f of user k to be served by one AP, the maximum number of possible states of each agent is $(\mathbf{C}^1_{|\mathcal{B}|})^{|\mathcal{F}_k|} = B^{F_k}$.

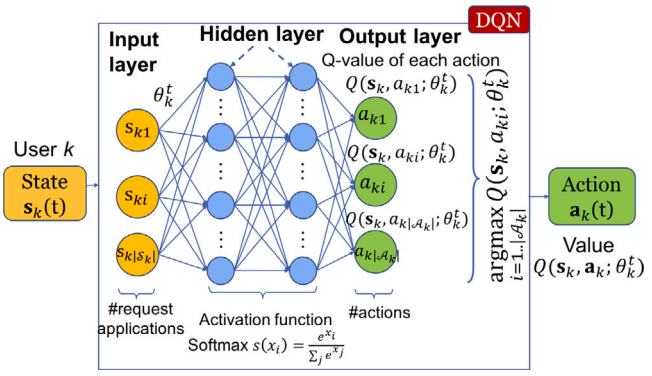


Fig. 2. DQN structure.

- **User Action:** Given its current association state $\mathbf{s}_k(t)$ and immediate reward defined in the sequel, user k selects by action $\mathbf{a}_k(t)$ in its action space $\mathcal{A}_k$, its desired future APs to be associated to for frame $t+1$, under the restriction (7b), namely

$$\mathbf{a}_k(t) \in \mathcal{A}_k = \left\{ a_{bkf}(t) \mid \sum_{b \in \mathcal{B}} a_{bkf}(t) = 1, \forall f \in \mathcal{F}_k \right\}. \tag{9}$$

Here, $a_{bkf}(t)$ are the binary variables of association requests, defined as

$$a_{bkf}(t) = \begin{cases} 1, & \text{if user } k \text{ requests AP } b \\ & \text{for application } f \in \mathcal{F}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Similarly, has a maximum of possible actions. Similarly, $\mathcal{A}_k$ has a maximum of $(\mathbf{C}^1_{|\mathcal{B}|})^{|\mathcal{F}_k|} = B^{F_k}$ possible actions.

Fig. 2 depicts the DQN structure used in our proposed algorithm. The input layer represents the current state $\mathbf{s}_k(t)$ of user k including variables $x_{bkf}(t)$, and the output layer gives the approximated Q-values for each available action in (9), calculated through the hidden layers based on the set of DNN weight parameters θ_k^t. It is worth noting that, here the number of available APs and requested applications are fixed and determine the size of the DQN's input/output layers. This is reasonable as those parameters can be assumed to vary slowly in the case of static or low mobility users, as confirmed by our simulation results in Section 5.3. In general, those parameters should undergo slow and smooth variations as compared to users' learning time, and hence users will be able to learn them online without having to re-train their DQN/DDQN from scratch

In the conventional Q-learning algorithm used in [18], only the Q-value $Q(\mathbf{s}_k(t), \mathbf{a}_k(t))$ for the current state and selected action $\mathbf{a}_k(t)$ is calculated at each iteration and memorized in a Q-table, as in [22]. By contrast, in the DQL approach taken here, the goal of the DNN at each user device is to learn an approximated Q-value function. At each iteration, this Q-function approximation is updated for all available actions at the same time based on the Deep Q-Network (DQN) [20] or Double Deep Q-Network (DDQN) [21] techniques, making Q-Learning applicable to large state/action spaces, and hence, to large-scale networks.

After performing its selected action, user k receives its immediate reward $\Gamma_k(t)$, which is used to update the set of DNN parameters θ_k^t. In the case of DQN, this update is made such that the loss function below is minimized through stochastic gradient descent, given the discount factor γ,

$$\mathcal{L}_k^{\text{DQN}} = \left[Q(\mathbf{s}_k(t), \mathbf{a}_k(t); \theta_k^t) - \left(\Gamma_k(t) + \gamma \max_{\mathbf{a}'_k} Q(\mathbf{s}_k(t+1), \mathbf{a}'_k; \theta_k^t) \right) \right]^2. \tag{11}$$

From (11), it can be observed that DQN Q is used both for action selection and evaluation, which is useful for saving the computational burden and memory storage of user devices, but may suffer from substantial Q-value overestimation issues [21]. On the contrary, by using two DQNs, one for action selection and the other for Q-value estimation, the DDQN is expected to overcome this drawback but at the cost of higher memory space consumption and increased computational complexity, which may be quite detrimental for computation and battery-limited user devices. Namely, a DDQN combines two different DQNs: in addition to the first DQN Q, a second DQN Q' is built with the same structure, however its set of parameters θ'^t_k is copied from the first only DQN periodically, i.e., every l frames. Then, DQN Q serves for action selection, while DQN Q' serves for state/action evaluation. In this case, the set of parameters θ'^t_k is updated by minimizing the following loss function,

$$\mathcal{L}_k^{\text{DDQN}} = \left[Q(\mathbf{s}_k(t), \mathbf{a}_k(t); \theta_k^t) - \left(\Gamma_k(t) + \gamma Q' \left(\mathbf{s}_k(t+1), \arg\max_{\mathbf{a}'_k} Q(\mathbf{s}_k(t+1), \mathbf{a}'_k; \theta_k^t); \theta'^t_k \right) \right) \right]^2. \tag{12}$$

Algorithm 1: Proposed DQN/DDQN-based Algorithms

```
1  Decay factor λ, greedy factor ε, weights w_1k, w_2k;
2  for each user k ∈ K do
3      Initialize DQN/DDQN Q with random weight values θ_k;
4      Random initial state s_k;
5  for t = 1, 2, ..., T do
6      for each user k ∈ K do
7          ε ← ε × λ;
8          if random number p < ε then Select action a_k randomly;
9          else Select action a_k with highest Q(s_k, a_k; θ_k^t);
10     for each AP b ∈ B do
11         Consider requests, select users/applications (if necessary)
           by greedy method ;
12         Feedback to users, data transmission;
13     for each user k ∈ K do
14         Calculate the reward of action a_k by (13) ;
15         Update θ_k^t by (11) or (12);
16         Move to the new state s_k ← s'_k;
```

Based on the above definitions, we propose our DQN and DDQN-based user association methods described in Alg. 1. Although they are based on our preliminary algorithms in [18], we recall their main steps below for sake of readability.

Step 1- At the user side, with probability 1-ε, each user selects action $\mathbf{a}_k(t)$ as in (9) from the output of its DQN/DDQN based on its current state $\mathbf{s}_k(t)$, then sends its request $\mathbf{a}_k(t)$ to the desired AP for each application f in $\mathcal{F}_k$ (Lines 6 to 9).

Step 2- After receiving all user requests, each AP decides to accept these requests or not based on its current load, i.e., if $\Phi_b(t) \leq 1$, AP b will serve all requested applications. Otherwise, AP b drops some applications by a greedy manner, namely, the user/application requests $a_{bkf}(t)$ corresponding to the highest load $\phi_{bkf}(t)$ will be eliminated until the constraint $\Phi_b(t) \leq 1$ is satisfied. After this phase, AP b sends its association decision $x_{bkf}(t)$ to its requested user through feedback. (Lines 10 to 12)

Step 3- At the user side, based on the feedback from APs, each user calculates its immediate reward $\Gamma_k(t)$ which is the weighted sum of two terms, c_{1k} and $(c^r_{2k} + c^d_{2k})$ with corresponding weights w_{1k} and w_{2k},

namely

$$\Gamma_k(t) = w_{1k}c_{1k}(t) + w_{2k}\left(c^{\mathrm{r}}_{2k}(t) + c^{\mathrm{d}}_{2k}(t)\right). \tag{13}$$

Here c_{1k} is the reward for user k as its requested application $f \in \mathcal{F}_k$ was actually served by the selected AP, and satisfied the corresponding QoS, given by

$$\begin{aligned} c_{1k}(t) = & \\ & \sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t)=1, r_{bkf}(t) \geq R_{kf}\right) a_{bkf}(t) r_{bkf}(t) \\ + & \sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t)=1, d_{bkf}(t) \leq T_{kf}\right) a_{bkf}(t) r_{bkf}(t), \end{aligned} \tag{14}$$

with the indicator function $\mathcal{I}(.)$.

On the contrary, if user k's requested applications' QoS were not satisfied or if they were dropped by APs, user k calculates its penalties, given by c^{r}_{2k} if the application has a minimum rate requirement, i.e., $f \in \mathcal{F}_{kr}$ and by c^{d}_{2k} if the application has a maximum delay requirement, i.e., $f \in \mathcal{F}_{kd}$. The immediate reward depends upon the type of feedback from its requested APs, from which users may get more or less knowledge of their local wireless environment. Thus, we considered two types of AP feedback, based on which two methods are designed, namely, the *Proposed Fully Distributed DQN (DDQN) Association* and the *Proposed Partially Distributed DQN (DDQN) Association*, described in Sections 4.1 and 4.2, respectively. Finally, user k updates the weights of its DQN or DDQN by (11) or (12) and move to its new state (Lines 13 to 16).

4.1. Proposed fully distributed DQN/DDQN association

In this algorithm, each user receives the minimal feedback $\mathbf{\Omega}_{bk}$ including only its desired APs' allocation decision, i.e,

$$\mathbf{\Omega}_{bk}(t) = \{x_{bkf}(t) | a_{bkf}(t) = 1 \quad f \in \mathcal{F}_k\}. \tag{15}$$

In other words, each user knows only about its own instantaneous channel information with its serving APs, but has no knowledge about other users' channel states nor requested QoS. Therefore, the penalties c^{r}_{2k}, c^{d}_{2k} of user k are calculated solely using its local channel state information, following (16)(a) if user k is served by AP b but at a rate lower than R_{kf}, and by (17)(a) if user k is served by AP b but with a delay time larger than T_{kf}, respectively. If user k's application f is dropped by its selected AP b, the penalties c^{r}_{2k}, c^{d}_{2k} are computed by (16)(b), (17)(b),

$$c^{r}_{2k}(t) \tag{16}$$
$$= \begin{cases} -\sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t)=1, r_{bkf}(t) < R_{kf}\right) a_{bkf}(t) \frac{R_{kf}}{r_{bkf}(t)} & \text{(a)}, \\ -\sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t)=0\right) a_{bkf}(t) \frac{R_{kf}}{r_{bkf}(t)} & \text{(b)}. \end{cases}$$

$$c^{d}_{2k}(t) \tag{17}$$
$$= \begin{cases} -\sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t)=1, d_{bkf}(t) > T_{kf}\right) a_{bkf}(t) \frac{d_{bkf}(t)}{T_{kf}} & \text{(a)}, \\ -\sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t)=0\right) a_{bkf}(t) \frac{d_{bkf}(t)}{T_{kf}} & \text{(b)}. \end{cases}$$

In (16) and (17), the ratios $\frac{R_{kf}}{r_{bkf}(t)}$ and $\frac{d_{bkf}}{T_{kf}(t)}$ enable to weight the impact of the incurred loss according to the actual rate and delay QoS, but also to the instantaneous channel quality: the higher the required rate R_{kf} (or the lower the required delay time T_{kf}) and the lower the channel quality (expressed by instantaneous rates $r_{bkf}(t)$ or by instantaneous serving time $d_{bkf}(t)$), the larger the penalty.

4.2. Proposed partially distributed DQN/DDQN association

In this algorithm, the users will receive additional information from the feedback of its desired APs. Namely, in addition to their own instantaneous channel information and requested APs' association decisions, each user will also gain knowledge about its own load relative to that of neighboring competitors who had requested the same APs. Hence, the feedback $\mathbf{\Omega}_{bk}$ sent to user k, is now given as

$$\begin{aligned} \mathbf{\Omega}_{bk}(t) = \{ & x_{bkf}(t), \phi_{bkf}(t) | a_{bkf}(t) = 1 \ \& \ \forall f \in \mathcal{F}_k; \\ & N^{\mathrm{r}}_b(t), N^{\mathrm{d}}_b(t)\}, \end{aligned} \tag{18}$$

where $N^{\mathrm{r}}_b(t), N^{\mathrm{d}}_b(t)$ are normalization factors for dropped applications with minimum rate and maximum delay time requirements respectively, given as

$$N^{\mathrm{r}}_b(t) = \sum_{\substack{k'\in\mathcal{K} \\ f'\in\mathcal{F}_{kr}}} \mathcal{I}\left(a_{bk'f'}(t)=1, x_{bk'f'}(t)=0\right) \frac{R_{k'f'}}{r_{bk'f'}(t)}. \tag{19}$$

$$N^{\mathrm{d}}_b(t) = \sum_{\substack{k'\in\mathcal{K} \\ f'\in\mathcal{F}_{kd}}} \mathcal{I}\left(a_{bk'f'}(t)=1, x_{bk'f'}(t)=0\right) \frac{d_{bk'f'}}{T_{k'f'}(t)}. \tag{20}$$

Taking advantage of this improved local environment knowledge, we now update the computation of penalty terms c^{r}_{2k}, c^{d}_{2k} as

$$c^{\mathrm{r}}_{2k}(t) \tag{21}$$
$$= \begin{cases} -\sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t)=1, r_{bkf}(t) < R_{kf}\right) a_{bkf}(t) \frac{R_{kf}}{r_{bkf}(t)}, & \text{(a)} \\ -\sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kr}} \mathcal{I}\left(x_{bkf}(t)=0\right) a_{bkf}(t) \phi_{bkf}(t) \frac{\frac{R_{kf}}{r_{bkf}(t)}}{N^{\mathrm{r}}_b(t)}. & \text{(b)} \end{cases}$$

$$c^{\mathrm{d}}_{2k}(t) \tag{22}$$
$$= \begin{cases} -\sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t)=1, r_{bkf}(t) < R_{kf}\right) a_{bkf}(t) \frac{d_{bkf}}{T_{kf}(t)}, & \text{(a)} \\ -\sum_{b\in\mathcal{B}}\sum_{f\in\mathcal{F}_{kd}} \mathcal{I}\left(x_{bkf}(t)=0\right) a_{bkf}(t) \phi_{bkf}(t) \frac{\frac{d_{bkf}}{T_{kf}(t)}}{N^{\mathrm{d}}_b(t)}. & \text{(b)} \end{cases}$$

In (21)(b) and (22)(b), the penalty terms in the case of overloaded APs, are improved by weighting each dropped application by its load contribution $\phi_{bkf}(t)$ upon its requested AP b, and also by normalizing the previous weight $\frac{R_{kf}}{r_{bkf}(t)}$, $\frac{d_{bkf}(t)}{T_{kf}}$ by the term $N^{\mathrm{r}}_b(t)$, $N^{\mathrm{d}}_b(t)$ respectively, which incorporates the rate as well as delay time requirements and instantaneous channel qualities of all other dropped users. These new definitions enable to set adequate penalties to each user, relatively to each other's loads, required rates, required serving time, and instantaneous channels, yet with only local information.

5. Numerical results

5.1. Simulation settings

The proposed algorithms are evaluated in two scenarios: firstly, scenario 1 composed of 9 APs and 10 users (Fig. 3), similarly as in [18]; secondly, scenario 2 composed of 25 APs and 50 users (Fig. 4). Users are uniformly distributed over the network areas and are assumed fixed during each episode of 10 000 frames. It is reasonable to assume that user positions will remain fixed during the users' learning time under low user mobility scenarios, as will be confirmed by the simulations results. However, the results are averaged over 100 random user positions for scenario 1, and over 20 random user positions for scenario 2. Moreover, all user to AP channels undergo random block Rayleigh fading where each channel coefficient remains fixed during a frame, but changes randomly across frames, in addition to slow fading.

In scenario 1, we consider two cases: first, each user requires two applications with minimum rate requirements $R_{k1} = 6$ Mbps, $R_{2k} = 3$

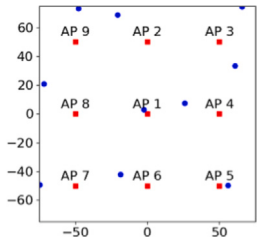


Fig. 3. Scenario 1.

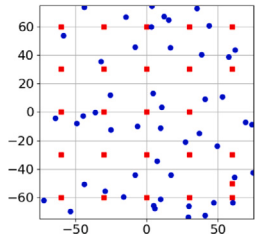


Fig. 4. Scenario 2.

Mbps, and second, each user requests three applications among which two applications have the same minimum rate requirements as in the first case, while the third application has a maximum delay time requirement of $T_{3k} = 1$ ms. Finally, in scenario 2, each user requires two applications with minimum rate requirements $R_{k1} = 6$ Mbps and maximum delay time $T_{3k} = 1$ ms. Remaining parameter values are given in Table 1. For fair comparison, reward weights (w_1, w_2) have been manually tuned to yield the "best" performance for each algorithm. That is, preliminary evaluations over varying values of (w_1, w_2) have shown that the best setting for both proposed and baseline algorithms was (0.8, 0.2), though the algorithms do not exhibit large performance variations for $(w_1, w_2) \in [0.1, 0.9]^2$. Therefore, in the sequel, (w_1, w_2) will be fixed to (0.8, 0.2), though marginal performance gains may be achieved by an exhaustive search over (w_1, w_2), at the cost of higher computational complexity.

The DQN/DDQN is built with two hidden fully connected layers using Softmax activation function. The number of neural nodes per hidden layer is 16, the memory size is set to 100. The period l for updating the weights of the DDQN target network is set to 5. Finally, it is worth mentioning that in our proposed algorithms, the training phase is performed online, which allows to assess their performance when users need to learn the mobile environment from scratch. This is possible thanks to the simple DQN structure with reduced number of nodes and layers that guarantees good convergence behaviors, as shown in the sequel.[1]

5.2. Baseline algorithms

We compare the proposed algorithms with the following two benchmarks:

- **Reference greedy** (*Ref. Greedy*): At each scheduling frame, the APs providing best instantaneous Signal-to-Noise Ratios (SNR) will be requested by each user. Namely, the user will send its association request to the AP with higher SNRs, according to the order of QoS priorities. Then the number of requested APs is equal to the number of required applications of each user.
- **Reference Basic DQN (DDQN)** (*Ref. Basic DQN (DDQN)*): This method is similar to the reference Q-learning scheme in [18] but uses DQN (DDQN). Namely, user state and action are the current association and its desired association between its applications and APs. Users receive the minimum feedback and their penalties are given by the gap between the served rate/time and their corresponding QoS requirements.
- **Reference Basic DQN (DDQN)-Single AP** (*Ref. Basic DQN (DDQN)-Single AP*): This method is similar to *Ref. Basic DQN (DDQN)*, but constrains each user to request only one AP for all its applications at each scheduling frame.

Table 1
Simulation parameters.

Parameter	Description
Transmit power p_{bkf}	5 dBm
Noise power σ_n^2	−169 dBm/Hz
Bandwidth	10 MHz
Path loss model	140.7 + 37.6$\log_{10}(d)$ (d: AP-user distance [km])
Mean packet arrival rate ($m_{kf}\frac{1}{\mu_{kf}}$)	0.1 Mbps
Epsilon ε	0.5
Decay factor λ	0.995
Discount factor γ	0.9
Weight (w_1, w_2)	(0.8, 0.2)

For convenience, we denote the *Proposed Fully Distributed DQN (DDQN) Association* and *Proposed Partially Distributed DQN (DDQN) Association* algorithms as *Prop. Fully Distributed DQN (DDQN)* and *Prop. Partially Distributed DQN (DDQN)*, respectively.

5.3. Simulation results

Scenario 1, $R_{k1} = 6$ Mbps, $R_{k2} = 3$ Mbps

We show the evolution of data rates over scheduling time frames, as well as of outage given by the DQN/DDQN-based algorithms in Figs. 5 and 6, respectively, then the average sum-rate and outage of all algorithms including *Ref. Greedy* are given in Table 2. Firstly, Fig. 5 shows the achievable data rate per application averaged over all users and positions by DQN/DDQN-based algorithms. We can observe that all algorithms converge well. Namely, *Ref. Basic DQN/DDQN-Single AP* converge after about 500 frames, whereas *Ref. Basic DQN/DDQN* and the proposed algorithms need 1000 and 2000 frames to converge, respectively.[2] This is because in the reference basic DQN/DDQN-Single AP algorithms, each user requests the same APs for all of its applications, thereby reducing the action space and taking a shorter time to converge. In general, the DQN-based methods achieve similar average rates as their corresponding DDQN-based algorithms. It can be observed that all user-to-multiple AP association algorithms achieve higher sum-rate than *Ref. basic DQN/DDQN-Single AP*. This proves the aforementioned necessity of allowing users to associate with several APs in the context of B5G systems. Compared to *Ref. Basic DQN/DDQN*, all proposed algorithms significantly improve the rates of each application. Namely *Prop. fully distributed DQN/DDQN* achieve a 69% rate increase for application 1 and 106% for application 2, whereas *Prop. partially distributed DQN/DDQN* achieve 68% and 85% rate increase for applications 1 and 2, respectively. In addition, Fig. 5 also shows that all algorithms can allocate a higher rate to the application with higher requirement R_{kf}, especially for *Ref. Basic-DQN/DDQN* and *Prop. partially distributed DQN/DDQN*.

[1] The training complexity of our proposed algorithms may be further improved by making the training phase offline and only re-training periodically, especially in very dynamic network environments with high mobility users. These aspects require more in-depth investigations, and will be the object of our future work.

[2] If we consider each frame to be of 1 ms, the proposed algorithms can converge after 2s for the small network Scenario 1.

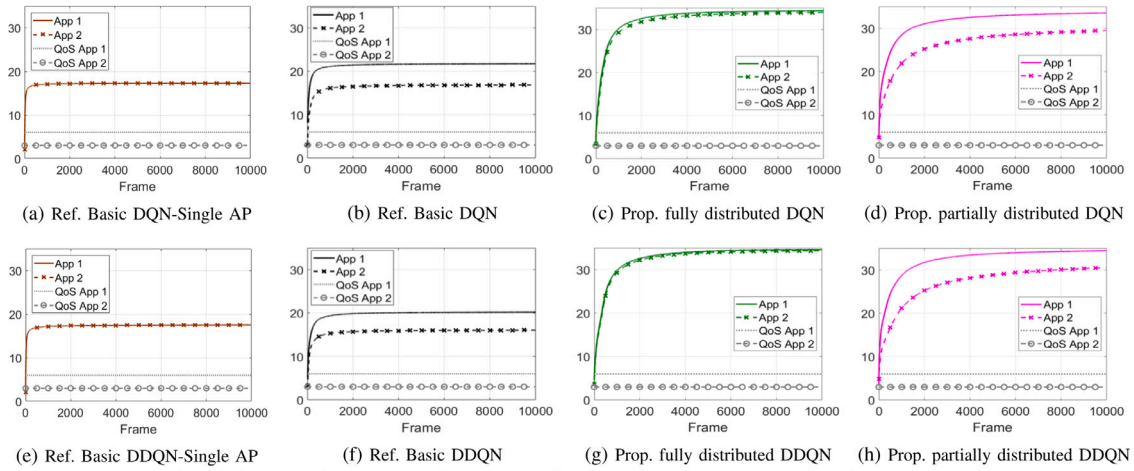


Fig. 5. Average data rates [Mbps] per application, scenario 1, two applications per user.

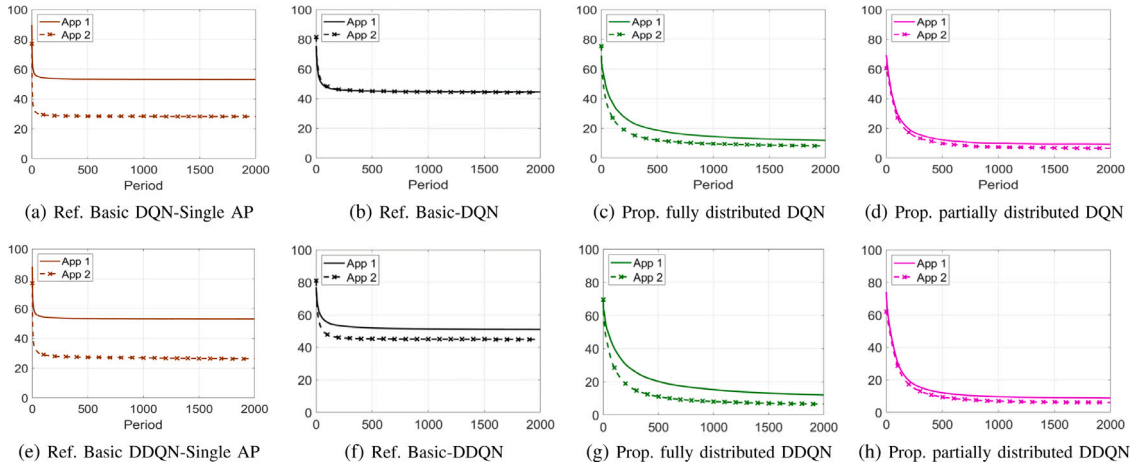


Fig. 6. Average user outage [%] per application, scenario 1, two applications per user.

Next, we consider the evolution of user outage, averaged over all users and positions. Similarly to [18], this outage performance is evaluated for short-term achieved rate $\bar{r}_{bkf} = \frac{\sum_{\tau=1}^{q} r_{bkf}(\tau)}{q}$, with $q = 5$. Then, if $\bar{r}_{bkf} < R_{kf}$, user k with application f is in outage. Interestingly, although reference user-to-single AP association methods provide the same data rate for all applications, their outage are much different as shown in Figs. 6(a), (e) since they have different QoS requirements. This again indicates the advantage of proposed user-to-multiple APs association methods. As shown in Fig. 6, both DQN and DDQN-based reference algorithms are outperformed by proposed algorithms. Namely, *Prop. fully distributed DQN/DDQN* provide 76% and 85% lower outage probabilities for applications 1, 2 respectively, whereas 83% and 86% lower outage levels are observed for applications 1 and 2 by *Prop. partially distributed DQN/DDQN*. Compared to their DQN-based counterparts, the DDQN-based proposed methods provide slightly better outage performance, especially for application 2. In addition, compared to *Prop. fully distributed DQN/DDQN*, *Prop. partially distributed DQN/DDQN* achieve better outage fairness between applications, as seen by the smaller gaps between each outage curve as shown in Fig. 6.

From Table 2, we can observe that *Ref. Greedy* achieves the lowest outage level for application 1 because users always request the best AP for this application. However, application 2 suffers much higher outage, up to 73.5% compared to 0.39% of application 1. Therefore, this algorithm cannot guarantee the fairness among different QoS applications.

Table 2
Average sum-rate [Mbps] and outage [%], after convergence scenario 1, two applications per user.

Algorithms	Sum-rate	Outage	
		App 1	App 2
Ref. Greedy	37.9	0.39	73.5
Ref. Basic DQN-Single AP	34.6	53.1	28.3
Ref. Basic DDQN-Single AP	35.1	53.0	26.2
Ref. Basic DQN	38.6	44.6	44.3
Ref. Basic DDQN	36.3	51.0	44.8
Prop. Fully Distributed DQN	68.3	11.8	8.2
Prop. Fully Distributed DDQN	69.0	12.0	6.54
Prop. Partially Distributed DQN	64.7	9.35	6.49
Prop. Partially Distributed DDQN	65.0	8.80	6.06

Moreover, this algorithm is also outperformed by proposed algorithms in terms of sum-rate.

Scenario 1, $R_{k1} = 6$ Mbps, $R_{k2} = 3$ Mbps, $T_{k3} = 1$ ms

Fig. 7 presents the average user rates for application 1 and 2 for DQN/DDQN-based algorithms. Similarly to the previous case, *Ref. Basic DQN/DDQN-Single AP* provides the same data rate for all applications, while user-to-multiple AP association methods provide higher data rate for application 1 and lower data rate for application 2. Again, among algorithms enabling user-to-multiple APs association, it can be observed that proposed algorithms outperform reference ones. Meanwhile, for

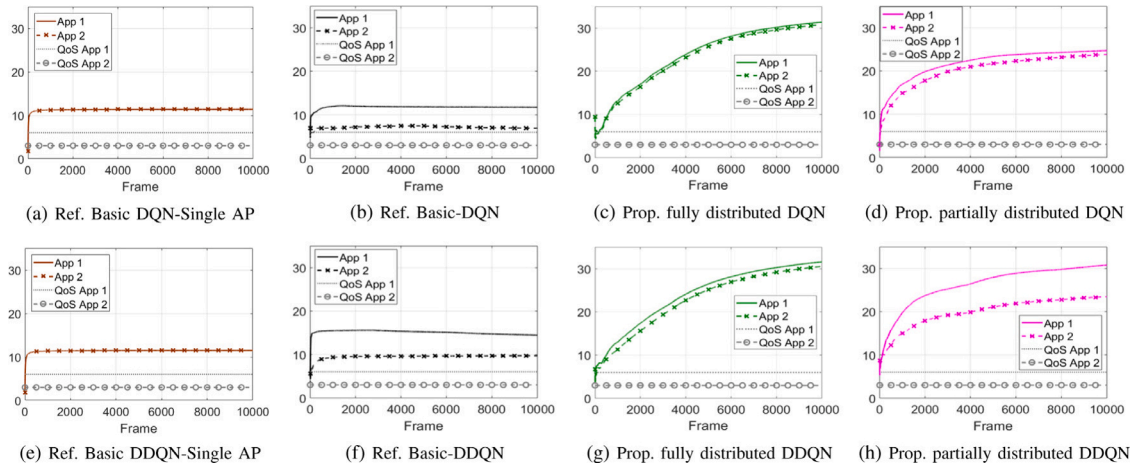


Fig. 7. Average data rates [Mbps] per application, scenario 1, three applications per user.

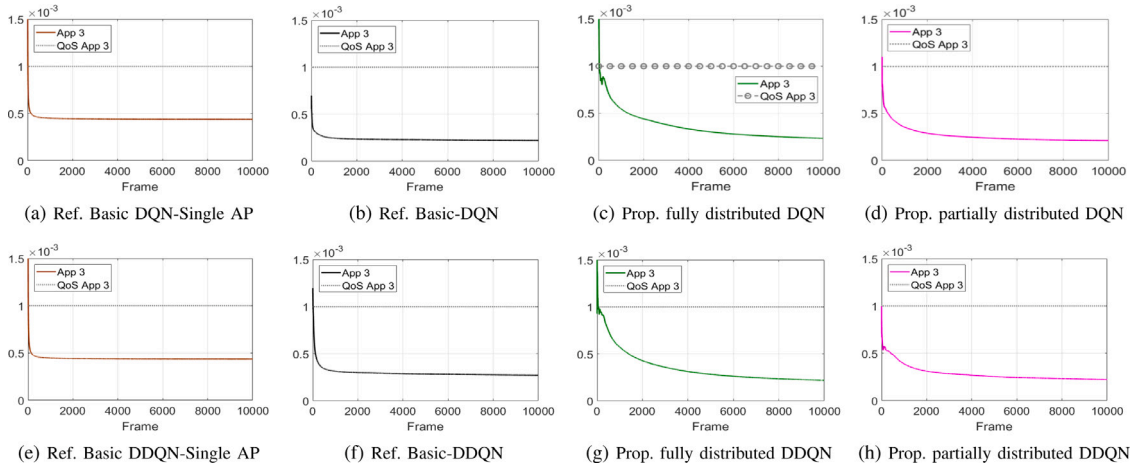


Fig. 8. Average delays [s] per application, scenario 1, three applications per user.

the delay-stringent application 3, Fig. 8 shows that all DQN/DDQN-based algorithms provide the same performance. However, as shown in Fig. 9, the proposed algorithms provide lower outage probabilities not only for applications 1 and 2, but also for application 3 as compared to *Ref. Basic DQN/DDQN-Single AP* and *Ref. Basic-DQN/DDQN*. This proves that our proposed algorithms significantly improve the association decisions as well as rate allocation compared to the benchmarks.

It can be also observed in Fig. 7 that the algorithms based on DDQN achieve slightly higher average rates compared to their DQN-based counterparts, in particular for application 1 regarding *Prop. partially distributed DDQN*, and for application 2 regarding *Ref. Basic-DDQN*. The rate gap between the two application curves is also more pronounced for DDQN-based algorithms than that for DQN-based ones.

In addition, it is interesting to observe that *Prop. partially distributed DQN/DDQN* provide the best fairness among applications as the three curves in Fig. 9 can be hardly distinguished, whereas *Ref. Basic-DQN* shows the largest gap between the two rate-constrained applications and the delay-constrained application (Fig. 9(b)). In conclusion, the reference algorithms cannot handle well the QoS diversity of the multiple applications required by each user, for instance, rate versus delay in this case. This can be explained as follows: *Ref. Basic-DQN/DDQN* calculate the penalties $c^{\mathrm{r}}_{2k}, c^{\mathrm{d}}_{2k}$ by the difference between the served rate/delay and the corresponding QoS requirement, however these raw differences are hardly comparable among themselves. By contrast, by using the ratio between each QoS requirement and the served rate or delay, the proposed algorithms can alleviate this drawback and hence guarantee better fairness among applications, as observed in Fig. 9.

Table 3
Average sum-rate [Mbps] and outage [%], after convergence scenario 1, three applications per user.

Algorithms	Sum-rate	Outage		
		App 1	App 2	App 3
Ref. Greedy	37.0	0.40	99.9	97.4
Ref. Basic DQN-Single AP	34.4	53.3	30.0	46.2
Ref. Basic DDQN-Single AP	34.5	53.1	28.5	45.8
Ref. Basic DQN	41.3	73.0	72.1	37.0
Ref. Basic DDQN	42.6	48.3	64.5	68.3
Prop. Fully Distributed DQN	80.6	33.3	32.3	23.9
Prop. Fully Distributed DDQN	85.1	32.4	29.6	18.5
Prop. Partially Distributed DQN	72.2	19.1	16.3	16.8
Prop. Partially Distributed DDQN	76.7	17.6	16.3	15.0

The average sum-rate and outage after convergence per application of all algorithms for Scenario 1 with three required applications are given in Table 3. Again, we can see that the greedy method cannot work well in the context of various QoS constraints, as only application 1 with highest QoS requirement is served well while the other applications undergo high outage. In addition, *Prop. Fully Distributed DQN/DDQN* obtains highest sum-rate, but at the cost of higher outage probability for all applications as compared to *Prop. Partially Distributed DQN/DDQN*.

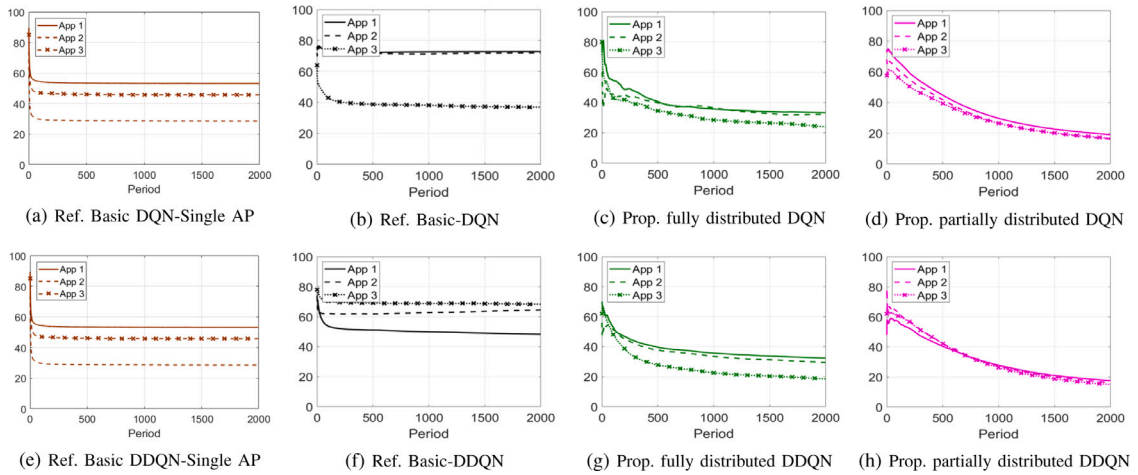


Fig. 9. Average user outage [%] per application, scenario 1, three applications per user.

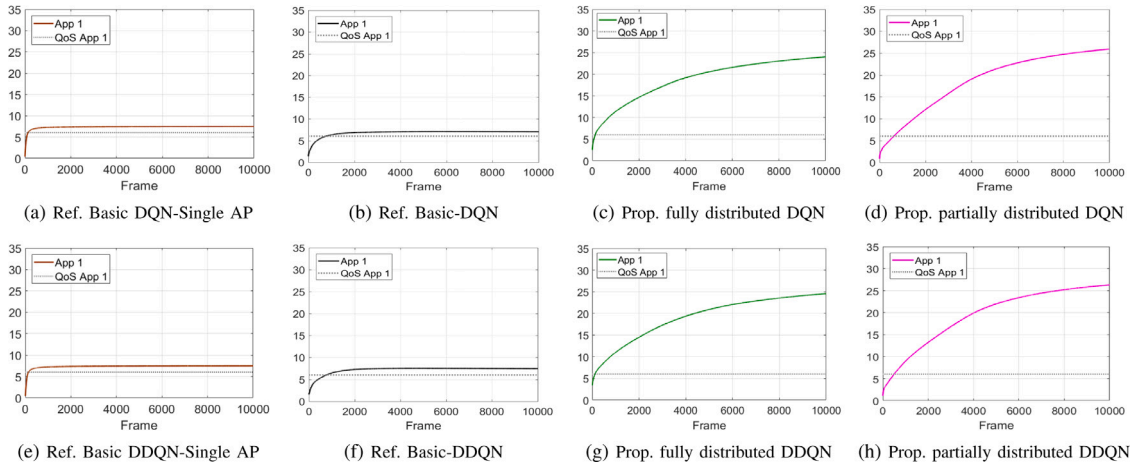


Fig. 10. The average data rate [Mbps] per application, scenario 2, two applications per user.

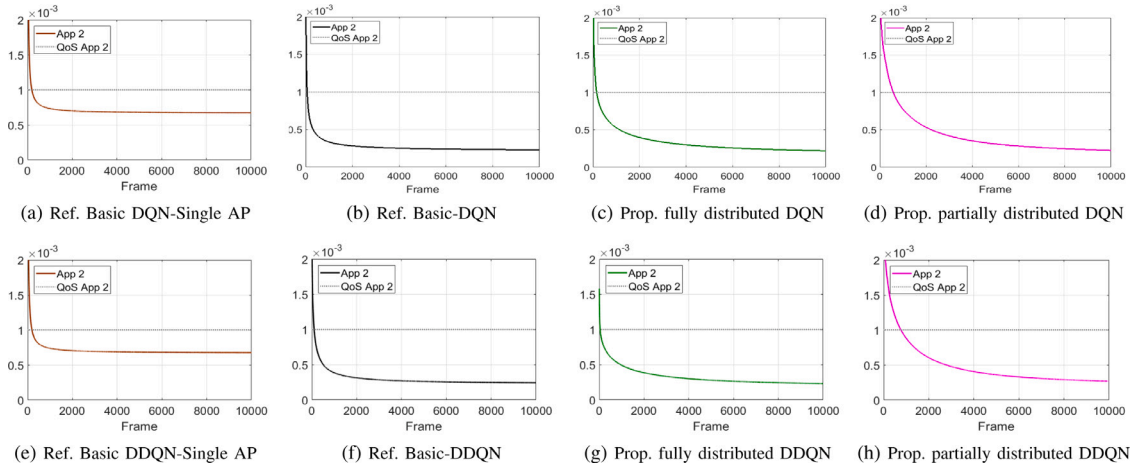
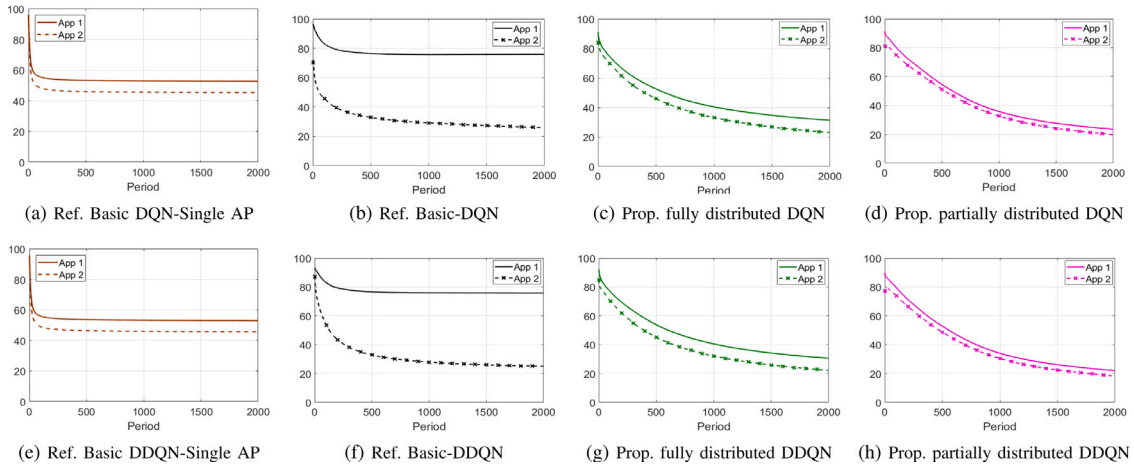


Fig. 11. Average delay [s] per application, scenario 2, two applications per user.

Scenario 2, $R_{k1} = 6$ *Mbps,* $T_{k2} = 1$ *ms*

Considering a dense and large-scale interfering network, Figs. 10 and 11 show the average user rate of application 1 and the average delay of application 2 for all DQN/DDQN-based algorithms. It can be observed that even in the large-scale network, *Ref. Basic DQN/DDQN-Single AP* still converge quickly after 500 frames, while *Ref. Basic DQN/DDQN* and the proposed algorithms take more time, though limited to 2000 and 4000 frames, respectively.[3] In spite of a slightly slower

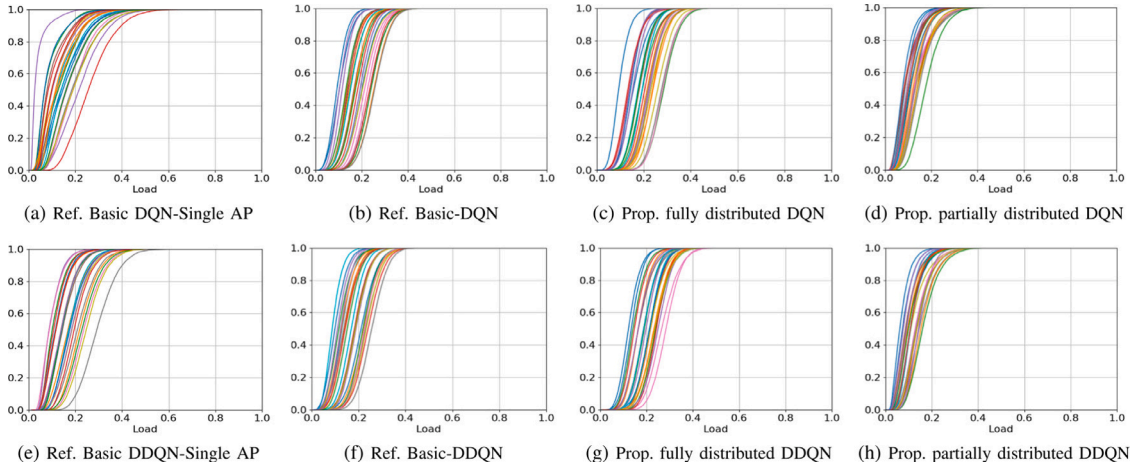
[3] The convergence time may be further reduced by conducting offline training, which will be investigated in our future work.



(a) Ref. Basic DQN-Single AP (b) Ref. Basic-DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN

(e) Ref. Basic DDQN-Single AP (f) Ref. Basic-DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Fig. 12. Average user outage [%] per application, scenario 2, two applications per user.



(a) Ref. Basic DQN-Single AP (b) Ref. Basic-DQN (c) Prop. fully distributed DQN (d) Prop. partially distributed DQN

(e) Ref. Basic DDQN-Single AP (f) Ref. Basic-DDQN (g) Prop. fully distributed DDQN (h) Prop. partially distributed DDQN

Fig. 13. CDF load per AP, scenario 2.

convergence compared to the reference schemes, the proposed methods provide higher rate for application 1 which requires the minimum rate QoS, and the same delay for application 2 as compared to *Ref. Basic-DQN/DDQN*, which is lower than that of *Ref. Basic DQN/DDQN-Single AP*. Like previous scenarios, *Ref. Basic-DQN/DDQN* also result into the largest gaps between both outage probability curves as shown in Fig. 12(b), (f). In particular, the average outage probability of application 1 amounts to 76%, and to 25% for application 2. *Prop. fully distributed DQN/DDQN* significantly reduce this gap with 34% outage for application 1 and 25% outage for application 2. As expected, *Prop. partially distributed DQN/DDQN* still provide the best fairness and lowest outage for both applications with 23% and 19% outage probabilities for applications 1 and 2, respectively.

Table 4 summarizes the average sum-rate and outage after convergence per application given by all algorithms for Scenario 2. Similarly to previous scenarios, *Ref. Basic DQN/DDQN-Single AP* achieves the lowest average sum-rate, whereas the greedy method gets the most unfair outage between two applications. In spite of allowing users to associate with multiple APs, *Ref. Basic DQN/DDQN* still obtain low average sum-rate, only about 28 Mbps while getting very high outage for application 1.

Finally, the cumulative distribution function (CDF) of the load per AP is presented in Fig. 13. We observe that all algorithms satisfy the load constraint of each AP. In particular, *Prop. partially distributed DQN/DDQN* achieve the best fairness in terms of load among APs, as shown by the more compact distribution of the CDF curves of all APs, as compared to *Ref. Basic-DQN/DDQN-Single AP*, *Ref. Basic-DQN/DDQN* and to *Prop. fully distributed DQN/DDQN*. Furthermore, *Prop. partially distributed DQN/DDQN* achieves the lowest burden per AP as well.

Table 4
Average sum-rate [Mbps] and outage [%], after convergence scenario 2, two applications per user.

Algorithms	Sum-rate	Outage	
		App 1	App 2
Ref. Greedy	34.8	0.12	97.7
Ref. Basic DQN-Single AP	14.9	52.7	45.4
Ref. Basic DDQN-Single AP	15.0	53.0	45.7
Ref. Basic DQN	28.0	75.8	25.4
Ref. Basic DDQN	29.2	75.7	25.0
Prop. Fully Distributed DQN	45.8	33.6	25.4
Prop. Fully Distributed DDQN	46.2	33.3	25.0
Prop. Partially Distributed DQN	45.0	23.5	19.8
Prop. Partially Distributed DDQN	48.2	22.2	18.3

Interestingly, we can observe from our simulation results that, although the DDQN-based proposed methods perform generally better compared to their DQN-based counterparts, the performance gains are rather limited. Therefore, the DQN-based approach may be more

appropriate for computation and battery-limited user devices, whereas DDQN-based methods can be useful if there are no such constraints. Hence, the most appropriate proposed method may be chosen according to the specific needs in terms of performance levels, and depending on the user devices' processing, memory and battery capabilities.

6. Conclusion

We have investigated the issue of user-to-multiple AP association, where a user requiring various applications with different QoS may be served by multiple APs simultaneously. This issue was formulated as a long-term network sum-rate maximization problem subject to the QoS requirements for each user and application, and AP load constraints. To solve this problem in a large-scale and distributed setting, we have proposed two user-to-multiple AP association methods with different amounts of local feedback from requested APs, and exploiting the DQN and DDQN-based deep reinforcement learning frameworks at the user device side. Numerical results show the effectiveness of the proposed method against reference methods. The proposed algorithms not only improve multiple objectives such as sum-rate and QoS satisfaction levels, but also enhance outage fairness among applications, as well as AP load balancing. Unlike reference schemes, the proposed methods are particularly well suited for handling heterogeneous types of QoS requirements.

In the future work, the proposed deep learning-based association methods will be further extended for jointly optimizing other resource allocation parameters such as transmit power and beamforming, and to cope with both Uplink and Downlink data transmissions. Moreover, the proposed methods will be extended to cope with high user mobilities as well as rapid variations of the network environment.

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.

Acknowledgement

This work was supported by the NTT-NII collaboration research project "Machine-learning based distributed wireless environment prediction" funded by NTT Corporation, Japan.

References

[1] I.F. Akyildiz, A. Kak, S. Nie, 6G and beyond: The future of wireless communications systems, IEEE Access 8 (2020) 133995–134030.
[2] W. Saad, M. Bennis, M. Chen, A vision of 6G wireless systems: Applications, trends, technologies, and open research problems, IEEE Netw. 38 (2020) 134–142.
[3] D. Liu, L. Wang, Y. Chen, et al., User association in 5G networks: A survey and an outlook, IEEE Commun. Surv. Tutor. 18 (2) (2016) 1018–1044.
[4] T. Zhou, Y. Huang, W. Huang, et al., QoS-aware user association for load balancing in heterogeneous cellular networks, in: IEEE VTC Fall, 2014, pp. 1–5.
[5] Y. Xu, et al., QoE-aware mobile association and resource allocation over wireless heterogeneous networks, in: IEEE GLOBECOM, IEEE, 2014, pp. 4695–4701.
[6] N. Liakopoulos, G. Paschos, T. Spyropoulos, Robust user association for ultra dense networks, in: IEEE INFOCOM, 2018, pp. 2690–2698.
[7] S. Bayat, et al., Distributed user association and femtocell allocation in heterogeneous wireless networks, IEEE Trans. Commun. 62 (8) (2014) 3027–3043.
[8] S. Maghsudi, E. Hossain, Distributed user association in energy harvesting small cell networks: A probabilistic bandit model, IEEE Trans. Wireless Commun. 16 (3) (2017) 1549–1563.
[9] X. Ge, X. Li, H. Jin, J. Cheng, V.C. Leung, Joint user association and user scheduling for load balancing in heterogeneous networks, IEEE Trans. Wireless Commun. 17 (5) (2018) 3211–3225.
[10] R. Dong, C. She, W. Hardjawana, Y. Li, B. Vucetic, Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin, IEEE Trans. Wireless Commun. 18 (10) (2019) 4692–4707.
[11] Q.V. Do, I. Koo, Actor-critic deep learning for efficient user association and bandwidth allocation in dense mobile networks with green base stations, Wirel. Netw. 25 (8) (2019) 5057–5068.
[12] P.-Y. Chou, W.-Y. Chen, C.-Y. Wang, R.-H. Hwang, W.-T. Chen, Deep reinforcement learning for MEC streaming with joint user association and resource management, in: IEEE ICC, 2020, pp. 1–7.
[13] Z. Li, M. Chen, K. Wang, C. Pan, N. Huang, Y. Hu, Parallel deep reinforcement learning based online user association optimization in heterogeneous networks, in: IEEE ICC, 2020, pp. 1–6.
[14] A.H. Arani, M.J. Omidi, A. Mehbodniya, F. Adachi, A distributed learning-based user association for heterogeneous networks, Trans. Emerg. Telecommun. Technol. 28 (5) (2017) 1–13.
[15] N. Zhao, X. He, M. Wu, P. Fan, M. Fan, C. Tian, Deep Q-network for user association in heterogeneous cellular networks, in: Conf. on Complex, Intelligent, and Software Intensive Sys., 2018, pp. 398–407.
[16] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, Y. Jiang, Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Networks, in: IEEE GLOBECOM, 2018, pp. 1–6.
[17] H. Ding, F. Zhao, J. Tian, D. Li, H. Zhang, A deep reinforcement learning for user association and power control in heterogeneous networks, Ad Hoc Netw. 102 (2020) 102069.
[18] T.H.L. Dinh, M. Kaneko, K. Wakao, H. Abeysekera, Y. Takatori, Reinforcement learning-aided distributed user-to-access points association in interfering networks, in: IEEE GLOBECOM, 2019, pp. 1–6.
[19] T.H.L. Dinh, M. Kaneko, K. Wakao, K. Kawamura, T. Moriyama, H. Abeysekera, Y. Takatori, Deep reinforcement learning-based user association in sub6ghz/mmwave integrated networks, in: IEEE CCNC, 2021, p. 7.
[20] V. Mnih, K. Kavukcuoglu, D. Silver, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529—533.
[21] H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, in: Thirtieth AAAI Conference on Artificial Intelligence, 30, (1) 2016.
[22] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT press, 2018.

Thi Ha Ly Dinh received her BSc. and MSc. degrees in computer science in 2015 and 2017 from Hanoi University of Science and Technology, Vietnam. She is currently working towards the Ph.D degree at the Graduate University of Advanced Studies (Sokendai), National Institute of Informatics, Tokyo, Japan. Her research interests include optimization problems, evolutionary algorithms, wireless communications, radio resource and interference management.

Megumi Kaneko (S'06, M'08, SM'17) received her Diplôme d'Ingénieur from Télécom SudParis (French Grande Ecole), France, in 2004, jointly with a MSc. degree from Aalborg University, Denmark, where she received her Ph.D. degree in 2007. In May 2017, she obtained her HDR degree (French Doctoral Habilitation for Directing Researches at Professor position) from Paris–Saclay University, France. She was a JSPS post-doctoral fellow at Kyoto University from April 2008 to August 2010. From September 2010 to March 2016, she was an Assistant Professor in the Department of Systems Science, Graduate School of Informatics, Kyoto University. She is currently an Associate Professor at the National Institute of Informatics as well as the Graduate University for Advanced Studies (Sokendai), Tokyo, Japan. Her research interests include wireless communications, 5G and beyond, IoT wireless systems, and PHY/MAC design and optimization. She serves as an Editor of IEEE Transactions on Wireless Communications, IEEE Communication Letters and IEICE Transactions on Communications. Since September 2020, she is as a member of the Advisory Board for Promoting Science and Technology Diplomacy at the Ministry of Foreign Affairs of Japan. She received the 2009 Ericsson Young Scientist Award, the IEEE Globecom 2009 Best Paper Award, the 2011 Funai Young Researcher's Award, the WPMC 2011 Best Paper Award, the 2012 Telecom System Technology Award, the 2016 Inamori Foundation Research Grant and the 2019 Young Scientists' Prize from the Minister of Education, Culture, Sports, Science and Technology of Japan. She is a Senior Member of IEEE.

Keisuke Wakao received B.E. and M.E. degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 2013 and 2015, respectively. Since joining NTT Access Network Service Systems Laboratories in 2015, he has been engaged in the research and development of wireless communication systems. He received the Young Researcher's Award from the IEICE of Japan in 2018. He is a member of the IEICE.

Kenichi Kawamura received B.E. and M.I. degrees from Kyoto University, Japan, in 1999 and 2001, respectively. Since joining Nippon Telegraph and Telephone Corporation (NTT) Corporation in 2001, he has been engaged in the research and development of wireless LAN systems, mobile routers, and network architecture for wireless access systems. Currently, he is a senior research engineer in the NTT Access Network Service Systems Laboratories. He is a member of the IEICE.

Takatsune Moriyama received his B.E. and M.E. degrees from Muroran Institute of Technology, Japan, in 1991 and 1993. He joined NTT in 1993. Since 1999, he worked at NTT Communications, where he was in charge of network service development and operation for corporate customers. He has been in his current position since July 2019.

Hirantha Abeysekera received the B.Eng., M.Eng., and Ph.D. degrees in communications engineering from Osaka University, Japan, in 2005, 2007, and 2010, respectively. In 2010, he joined NTT Network Innovation Laboratories, Yokosuka, Japan. He has been engaged in the research and development of next-generation wireless LAN systems. He received the IEEE VTS Japan Student Paper award in 2009. He is a member of IEEE.

Yasushi Takatori received a B.E. degree in electrical and communication engineering and an M.E. degree in system information engineering from Tohoku University, Miyagi, Japan, in 1993 and 1995, respectively. He received a Ph.D. degree in wireless communication engineering from Aalborg University, Aalborg, Denmark, in 2005. He joined NTT in 1995. He is currently working on research and development of a high-efficiency wireless access platform. He served as a co-chair of COEX Adhoc in IEEE 802.11ac from 2009 to 2010. He was a visiting researcher at the Center for TeleInFrastrutur (CTIF), Aalborg University, from 2004 to 2005. He received Best Paper Awards from the IEICE in 2011 and 2016. He was honored with the IEICE KIYASU Award in 2016. He also received the IEEE Standards Association's Outstanding Contribution Appreciation Award for the development of IEEE 802.11ac-2013 in 2014. He is a senior member of the IEICE and a member of the IEEE.