DARCAS: Dynamic Association Regulator Considering Airtime Over SDN-Enabled Framework

Muhammad Salman, Jin-Ho Son, Dong-Wan Choi, Uichin Lee, *Member, IEEE*, and Youngtae Noh

***Abstract*—The massive influx of mobile devices and their increasing use in recent years have resulted in the overprovision of access points (APs) in networks. Unlike in residential environments, network administrators in enterprises and universities make every endeavor to enhance the user experience (UX) of WiFi networks where the network dynamics (e.g., traffic load and user mobility) are usually unexpected. To this end, an existing mechanism for WiFi association is client driven, i.e., users associate themselves to the AP with higher signal strength. However, they still incur dissatisfaction due to the insufficient available bandwidth. To cope with this in a centralized manner, we propose DARCAS, a software-defined network (SDN)-enabled WiFi framework for association regulation. DARCAS adopts a notion of bandwidth satisfaction ratio (BSR), which is closely related to UX. It maximizes the aggregated network throughput while satisfying the BSR of each user with sufficient airtime (i.e., channel occupancy time) provision. We use this idea in a metaheuristic genetic algorithm called DARCAS-GA, which effectively finds the suboptimal association distribution of the maximum BSR in polynomial time. We implement the DARCAS system on off-the-shelf wireless routers and an SDN controller. We report real-life experimental results in the considered scenarios and conduct extensive simulations on the NS-3 simulator to examine its performance with scalability. With fine-tuned settings, DARCAS exhibits up to 80% of the BSR gain compared to existing solutions.**

***Index Terms*—Access point (AP), airtime, association redistribution, bandwidth satisfaction ratio (BSR), software-defined network (SDN), user experience (UX).**

I. Introduction

IN RECENT years, the number of access point (AP) deployments in residential areas, enterprises, and universities has increased significantly owing to the massive influx of mobile devices and their growing variety of use cases (e.g., social media, video/audio streaming, gaming, and so forth). In residential settings, loads of APs (usually generated by a few devices per household) are relatively easy to expect and manage. However, load management in enterprise and academic settings is relatively challenging because multiple APs coexist and suffer from poor channel selection (increased co-channel interference) and unbalanced load per AP [1]–[4].

To cope with these challenges, network administrators may plan the AP deployment by considering the network coverage and frequency selection of neighboring APs. Additionally, management solutions [5], [6] can help mitigate this problem. However, operational WiFi networks have shown that user loads are often distributed unevenly among wireless APs and change over time. This significantly reduces WiFi service reliability, degrading the user experience (UX) [7]–[11]. Owing to the massive success of video (e.g., Netflix, YouTube, Amazon Prime, etc.) and audio (Spotify, Apple Music, Google Play Music, etc.) streaming applications in recent years, the proportion of bandwidth-intensive traffic has significantly increased in mobile networks [12]. These trends worsen the UX, and users switch to LTE (now being replaced with 5G) as a substitute.

The existing client-driven association based on the maximum RSSI from an AP in its vicinity does not consider the AP load and link capacity, which results in a significantly degraded WLAN performance. Moreover, the handoff overhead in such an association is costly (ranging from a few hundred milliseconds to a few seconds). Thereby, it is practically impossible to provide efficient and fair bandwidth through the association control of users in multiple-AP networks without disrupting ongoing sessions. Recent studies have found that the problem of unbalanced loads and unfair bandwidth allocation can be significantly alleviated by intelligently associating users to APs, referred to as *association control*, rather than user-driven associations to APs with the strongest received signal strength [13], [14]. However, allocating bandwidth fairly and efficiently to greedy users is challenging. Even identifying users in the bottleneck fairness group or finding their normalized bandwidth is NP-hard [15]. State-of-the-art AP association control technology mediates user (i.e., station) associations based on RSSI values and/or aggregated network throughput in a centralized manner (e.g., Aruba WLAN Controller [5] and Meraki [6]). Even so, once the association has been made, these solutions do not have a control knob for the dynamic traffic loads and location changes of the users.

With the advent of the software-defined network (SDN [16]) and its success in datacenters [17], research has been undertaken to replicate this in the wireless domain [18]–[21] and improve the UX of WiFi services. A light virtual AP

Manuscript received 27 April 2022; accepted 13 May 2022. Date of publication 18 May 2022; date of current version 7 October 2022. This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) and in part by the Ministry of Education under Grant NRF-2019R1F1A1059898 and Grant NRF-2020R1A4A1018774. *(Corresponding authors: Dong-Wan Choi; Uichin Lee; Youngtae Noh.)*

Muhammad Salman and Dong-Wan Choi are with the Department of Computer Science and Engineering, Inha University, Incheon 22212, South Korea (e-mail: salman@inha.edu; dchoi@inha.ac.kr).

Jin-Ho Son is with TmaxSoft, Seongnam-si 13595, Gyeonggi, South Korea (e-mail: jhson@nsl.inha.ac.kr).

Uichin Lee is with the School of Computing, KAIST, Daejeon 34141, South Korea (e-mail: uclee@kaist.ac.kr).

Youngtae Noh is with Energy AI, KENTECH, Naju 58330, South Korea (e-mail: ytnoh@kentech.ac.kr).

Digital Object Identifier 10.1109/JIOT.2022.3176010

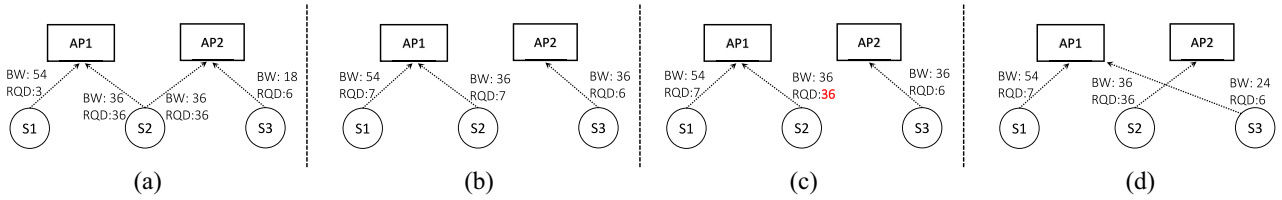

Fig. 1. DARCAS's motivation: (a) example of association control via proportional and airtime fairness with RQD; (b) example of dynamic association redistribution: initial load at time 0–15 s; (c) shows load changes at 15 s but maintains the associations until 20 s; and (d) shows association redistribution at 20 s.

(LVAP) [18], [21] or a similar abstraction technique virtualizes the states of association and separates them from the physical APs [22], [23]. These techniques introduce a seamless handoff (<80 ms) and the feasibility of the central controller-driven association. However, the central controller in existing work [24]–[30] lacks the core knowledge of the user's bandwidth satisfaction (i.e., the coarse bandwidth required by the user and how much should be effectively provided by the network resources). Therefore, the inefficient bandwidth supply causes user dissatisfaction.

To solve this problem, association control should be made aware of the multirate environment of 802.11 standards (e.g., 802.11g/n) in the global view of the network [13], [14], in addition to periodic and direct measures of residual wireless resources (i.e., channel occupancy time or *airtime*) to strike the appropriate balance between performance and fairness. Therefore, we use a standard directly related to UX, i.e., bandwidth satisfaction ratio (BSR), which is the ratio of the required bandwidth (RQD) per user to that actually provided by the direct network resource. We use this standard in a metaheuristic genetic algorithm called DARCAS-GA, which finds the best association distribution to maximize the aggregated network throughput by maximizing its residual airtime (i.e., channel occupancy time) to the user whose BSR is not fully satisfied in polynomial time. Then, we propose an SDN-enabled WiFi system called DARCAS and validate its performance through experiments involving real-life scenarios. We also provide extensive simulations using the NS-3 simulator to examine its performance with scalability. Our contributions here are threefold.

1) To enable the SDN controller for redistributing station memberships among overlapping APs and adapting the periodic semionline (i.e., *dynamic*) algorithm to the network dynamics (i.e., topology and changes in traffic pattern), we devise an airtime-measuring agent equipped with the LVAP abstraction that obtains the airtime utilization of each AP per 100 ms and reports it to an airtime association control app in the application layer. Finally, we propose an airtime-based SDN WiFi system called DARCAS, which contains an AP-level *airtime agent*, *DARCAS collector*, and *application-level airtime association control app* to enforce DARCAS-GA's decision on the entire network (Section IV-A).
2) We use a standard called the BSR, i.e., the ratio of the bandwidth required by the user to that provided, by limiting the assumption of maximum throughput usage in stations with known RQD (e.g., gaming and video/audio streaming) and developed a genetic algorithm called DARCAS-GA that maximizes the throughput of the overall wireless network by satisfying the BSR of each user. The proposed algorithm outperformed prevalent solutions and delivered near-optimal results (Section VI-B).
3) We implemented the *DARCAS agent* on off-the-shelf wireless routers, the *DARCAS collector* on an SDN controller, and the *airtime association control app* on an application server (Section IV-B). We conducted a testbed experiment in certain scenarios (Section V) and provided extensive simulations of the proposed method. It yielded an average gain in the BSR of up to 80% compared with the prevalent solutions (Section VI).

II. Motivation

Studies on association control based on proportional fairness (PF) [14], [31]–[33] and throughput-based fairness [34]–[36] have been conducted in multirate WLAN settings. Furthermore, PF has been shown to outperform throughput-based fairness [37]. The PF assumes that every station always consumes the maximum bandwidth allowed by the wireless link between it and an AP. However, this assumption does not satisfy the actual UX, i.e., providing adequate resources (airtime) to the users to fulfill their demand (RQD). We demonstrate it with the following scenarios.

A. Why Initial Association Control Matters?

Fig. 1(a) shows an example of station association control. It shows three stations (S1, S2, and S3) and two APs (AP1 and AP2) in the vicinity of their locations. BW denotes the maximum data rate allowed by the modulation coding scheme (MCS)[1] [38], and RQD represents the bandwidth required by each station. First, assume that S2 joins the AP2 network and shares the airtime of AP2 with S3. Using PF, the aggregated throughput of the network would be $54 + (36/2 + 18/2) = 81$ Mb/s. However, the aggregated throughput determined by PF only accounts for the maximum allowed bandwidth, i.e., BW for each station while ignoring the BSR of every associated station which is closely related to the UX. This inadequate assumption by PF results in misleading the aggregated throughput measurement.

[1]For the sake of simplicity, currently, we do not assume the channel impairments, such as the impact of slow fading and path loss, etc., to exclusively emphasize the importance of considering the UX. However, the channel impairments will reduce this data rate, which will be discussed in Section V-A. Nonetheless, the lower data rate does not affect the intended functionality of the proposed algorithms.

The actual aggregated network throughput which is consumed by the network can be measured by considering both the BSR and BW parameters as follows:

$$\text{Network}_{\text{Agg.}} = \text{BSR}_{S1} \times \text{BW}_{S1A1} + (\text{BSR}_{S2} \times RA_{S3}) \times \text{BW}_{S2A2} + \text{BSR}_{S3} \times \text{BW}_{S3A2} \quad (1)$$

where BSR_{S1} is the BSR of S1 and is equal to $RQD_{S1}/\text{BW}_{S1A1}$, i.e., the RQD demanded by the $S1$ by considering the available bandwidth between S1 and AP1 (i.e., BW_{S1A1}). Furthermore, RA_{S3} is the residual airtime from S3 that can be utilized by S2 and is equal to $(\text{BW}_{S3A2} - RQD_{S3})/\text{BW}_{S3A2}$. Therefore, using (1), the actual aggregated network throughput $\text{Network}_{\text{Agg.}}$ would be $(3/54) \times 54 + \{(36/36) \times (18 - 6)/18)\} \times 36 + (6/18) \times 18 = 33$ Mb/s. Similarly, if S2 is associated with AP1, the PF would provide a bandwidth of 27, 18, and 18 Mb/s to S1, S2, and S3, respectively. On the contrary, considering BSR and BW of each user the bandwidth allocated to the users S1, S2, and S3 would be 3, 34, and 6 Mb/s, respectively, which is almost sufficient for the requirements (i.e., RQD) of each user. It is evident from the above illustration that solely achieving a higher aggregated network throughput does not guarantee the best UX unless we consider the user level demands and adequate resource provision. The misjudgment by PF is rooted in the assumption that every station always consumes the maximum bandwidth allowed by the wireless link between it and an AP. However, in both cases (whether S2 joins the AP1 or AP2), the actual aggregated network throughput was lower than that computed by PF (i.e., 81 Mb/s). In practice, stations, such as gaming (e.g., 15 Kb/s [39]) or video streaming (e.g., 3–15 Mb/s [40]) on mobile devices may not require the maximum allowed bandwidth (i.e., BW). Therefore, an intelligent association control that considers both airtime utilization and RQD using the best available knowledge (e.g., statistical [41] and machine learning [42]) can achieve the desired aggregated throughput of the network (i.e., even if the S1 initially joins the AP2, the association control will automatically switch its association to the AP1) without sacrificing fairness.

B. Why Dynamic Association Control Matters?

Fig. 1(b)–(d) shows another motivation example of dyamic (i.e., periodic) association control by considering varying network traffic loads. Fig. 1(b) shows the topology of a part of an academic/enterprise network. It shows three stations that have two APs (AP1 and AP2) in their vicinity. S1 is associated with AP1 through an RQD of 7 Mb/s and BW of 54 Mb/s, whereas S2 is associated with AP1 with an RQD of 7 Mb/s and BW of 36 Mb/s. S3 is associated with AP2 with an RQD of 6 Mb/s and BW of 36 Mb/s. As shown in Fig. 1(b), the overall throughput of the WLAN network is $(7/54) \times 54 + ((54 - 7)/54) \times 36 + (6/36) \times 36 = 44.33$ Mb/s. The serving stations S1, S2, and S3 are allotted with bandwidths of 7, 31.33, and 6 Mb/s, respectively, which are sufficient for their RQDs. However, after 15 s, the required traffic load for S2 changes to 36 Mb/s, as shown in red in Fig. 1(c). The overall throughput of the WLAN network remains 44.33 Mb/s; nonetheless, S2 requires 36 Mb/s which is slightly higher than the provided bandwidth (i.e., 31.33 Mb/s). Furthermore, S3 requires a lower bandwidth (i.e., 6 Mb/s) than that provided, i.e., 36 Mb/s. To handle this, DARCAS performs a periodic semionline (every five seconds in this setting) association control, which intelligently redistributes the associations of all stations in the network as shown in Fig. 1(d). Thus, S1 retains its membership while S2 is associated with AP2, and S3 is associated with AP1 at the 20th second. Therefore, the overall throughput becomes $(7/54) \times 54 + (36/36) \times 36 + ((54 - 7)/54) \times 24 = 63.88$ Mb/s. This redistributed association supports the RQDs of all stations. Note that the RQDs of both S1 (i.e., seven) and S3[2] (i.e., six) are fully supported, because they do not exceed 50% of the airtime utilization. The state-of-the-art AP association control technology initially controls user (i.e., station) associations based on RSSI values and/or aggregated network throughput in a centralized manner (e.g., Aruba WLAN Controller [5] and Meraki [6]). However, this technology does not have control over redistributing associations once they have been made and, thus, cannot enhance performance, which is closely related to UX.

III. Problem Definition and Our Solution

To achieve the promising controls mentioned above, we first define our problem and objective function to maximize the aggregated network throughput by considering both the RQD and airtime utilization. We subsequently devise DARCAS-GA to obtain near-optimal solutions in polynomial time.

A. Problem Formulation

Consider a set A of APs, where each AP is represented by an integer $i \in [1, m]$, and let U denote the set of all users (also known as stations). Each user in U is indexed by an integer $j \in [1, n]$ and is associated with only one AP. The optimization problem can then be formulated as follows:

$$\text{Maximize: } \sum_{j \in U} \log\left(\sum_{i \in A} r_{ij} x_{ij} p_{ij}\right)$$

$$\text{subject to: } x_{ij} \in \{0, 1\} : \sum_{i \in A} x_{ij} = 1 \quad \forall j, \text{ and}$$

$$p_{ij} = \text{AIRTIME-MAX-MIN}(i, j).$$

The objective function in the above problem formulation aims to maximize the sum of the logarithms[3] of the bandwidth allocated to each user. This allocation is computed by the product $r_{ij}x_{ij}p_{ij}$, where r_{ij} specifies the maximum bit rate between AP i and user j, x_{ij} indicates whether AP i and user j are associated, and p_{ij} is the fraction of time that user j is allocated when it connects to AP i. Note that p_{ij} has been introduced to allocate the airtime for the (if known) RQD between i and j. The first constraint specifies that each indicator variable x_{ij} should be either zero or one, ensuring that each user is connected to only one AP. The final novel constraint

[2]As the S3 distance increases, the MCS reduces the provided bandwidth (e.g., from 36 to 24 Mb/s) depending on the channel impairments.

[3]It is used for fairness (i.e., spreading). p_{ij} is limited and can be allocated to stations. Thus, if the BSR is larger than the base, allocating p_{ij} to a larger number of stations returns a higher result.

Algorithm 1: AIRTIME-MAX–MIN(i, j)

Input: $i \in A, j \in U$: a pair of AP and user,
t_{ij}: required bandwidth of the user
Output: p_{ij}: the fraction of time allocated to user j within AP i

1 $D_i = \{j \in U \mid x_{ij} = 1\}$;
// Arrays initialization for all j∈U associated with i∈A
2 **foreach** $j \in D_i$ **do**
3 $p_i[j] = 0$, // airtime allocation
4 $q_i[j] = \frac{min(t_i[j], r_i[j])}{r_i[j]}$, // BSR
5 **if** $t_i[j] \geq r_i[j]$ **then**
6 $d_i[j] = 0$;
7 **else**
8 $d_i[j] = \frac{1}{|D_i|} - q_i[j]$; // Residual airtime
9 ALLOCATE-RECURSIVE(D_i, p_i, q_i, d_i);
10 **return** $p_i[j]$;

11 **Subroutine** ALLOCATE-RECURSIVE(D, p, q, d):
12 $D^+ = \{j \in D \mid d[j] > 0\}$, $D^- = \{j \in D \mid d[j] == 0\}$;
13 Res=$\sum_{j \in D^+} d[j]$;
// Total residual airtime
14 Ins=$|D| - |D^+|$;
// STAs with insufficient airtime
15 **if** D^+ *is empty* **then**
16 $\forall j \in D$:$p[j] = q[j] + d[j]$
17 **else**
18 $\forall j \in D^+$:$p[j] = q[j]$; // as defined in line 5
19 $\forall j \in D^-$:$d[j] = d[j] + \frac{Res}{Ins}$;
20 **if** D^- *is not empty* **then**
21 ALLOCATE-RECURSIVE(D^-, p, q, d);

AIRTIME-MAX–MIN(i, j) returns the fraction of time p_{ij} that is actually assigned to user j within AP i, given the network configuration considered. Considering applications, such as gaming, video, and audio streaming, we remove the assumption in prevalent work that each user always consumes the maximum allowed bandwidth (i.e., r_{ij}) at the associated AP. We now define how the fraction of time is distributed when some users only use a part of their available bandwidth (i.e., known RQD). AIRTIME-MAX–MIN(i, j) works as shown in Algorithm 1.

AIRTIME-MAX–MIN initializes three arrays, $p_i[\,]$, $q_i[\,]$, and $d_i[\,]$, where each element represents the fraction of time finally allocated to each user at AP i, the fraction actually required by each user at AP i (i.e., BSR), and the remaining (or residual) fraction after having assigned the time fraction identical to the equal rate (i.e., $[1/|D_i|]$) at AP i, respectively. Note that we make the best use of the remaining bandwidth (i.e., $d_i[j]$) as long as the bandwidth required by user j (i.e., $t_i[j]$) is less than $r_i[j]$. To represent the actual demand of time for user j, we set $q_i[j]$ to the minimum of t_j and r_{ij} in line 5. Finally, we accumulate the residual airtime from the users with excess airtime in $d_i[j]$ (in line 9).

After initialization, the algorithm invokes a subprocedure called ALLOCATE-RECURSIVE that recursively computes the bandwidth allocation to each user j in the manner of max–min fairness. To do this, line 13 checks the users with residual and insufficient allocated airtime. Thus, for each round, we first assign all requested resources to users whose demands are lower than the equal rate (i.e., $\forall j \in D^+$) (line 19) and then recalculate the remaining flows to be shared by other resources that have yet to be allocated (i.e., $\forall j \in D^-$) in line 20. To the best of our knowledge, DARCAS is the first departure reported from prevalent work that applies max–min fairness to *the fraction of time* to use both the RQD (if known) and airtime utilization into the BSR to improve network reliability.

B. DARCAS-GA: Genetic-Based Algorithm

Our optimization problem is a nonsmooth and nonlinear problem, even after relaxing of the first constraint to $0 \leq x_{ij} \leq 1$, because each p_{ij} is determined by a noncontinuous procedure, AIRTIME-MAX–MIN. Thus, our problem is a type of nonsmooth optimization problem [43], to which differential approaches, such as nonlinear programming, are not applicable. Therefore, we need a derivative-free optimization method to find a reasonably accurate and computationally efficient approximation because this process must be executed at short intervals (further evaluated in Section VI).

Our solution involves adapting the genetic algorithm (GA). Although GA is one of the oldest optimization algorithms, however, it has been widely applied in wireless network problems owing to its wide variety of benefits [44]. Our motivation for exploiting GA is threefold. First, it can efficiently find near-optimal solutions in a large *multimodal* search space of x_{ij} in quick time [45]. Second, as in our case, the decision is made in run time (i.e., semionline fashion—every 5 s). Thanks to the online-adoptive feature of GA, it can operate quickly in an unknown environment (despite the presence of dynamic channel parameters) and automatically make effective decisions. Finally, its rapid prototyping is possible on field-programmable gate arrays (FPGAs) and digital signal processing (DSP) [46].

Because we already have a fitness function (i.e., our objective function), we first define a genetic representation of the solution domain, which shows how to implement genetic operators, including *selection*, *crossover*, and *mutation*.

In a GA, each candidate solution is usually represented by an array of bits, called a *chromosome*. This can be easily achieved by converting the association matrix of x_{ij} to the corresponding *flat list*, as shown in Fig. 2(a). In the matrix, 1 denotes a station associated with the AP; this value is 0 otherwise. N/A denotes that a station is outside AP coverage. The overall flow of the DARCAS-GA is as follows.

Step 1 (Initialization): To improve the random population generally used in a standard GA, we generate two types of baseline chromosomes: 1) *nearest best* and 2) *round robin*, so that the final solution is greater than or equal to these baselines. The baseline *nearest best* selects the closest AP to each user, which is the default method for each client to connect to the most suitable AP unless interventions are made. However, the *nearest best* can result in the worst case, in which all users connect to the same nearest AP. To complement this case, the *round robin* tries to equalize the number of users associated to each AP by assigning users to APs in a round-robin fashion. This is beneficial for maintaining diversity to find a globally satisfactory solution [47].

Step 2 (Evaluation and Selection): The initial chromosomes are evaluated by a fitness function and then classified into *best*

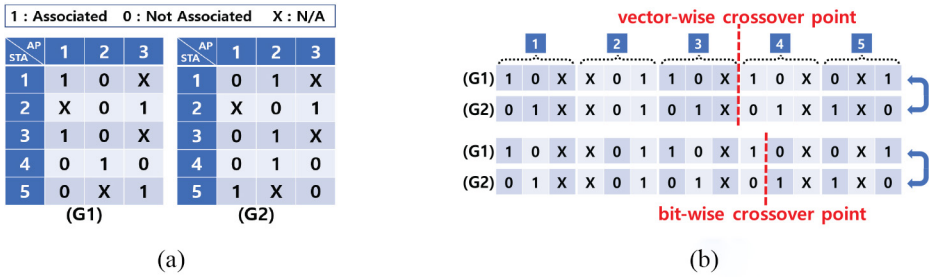

Fig. 2. Illustration of how to convert association matrices to their corresponding flat lists to represent chromosomes in DARCAS-GA. (a) Example of association matrices of x_{ij}'s consisting of three APs and five users. (b) Crossover operation on flat lists corresponding to the association matrices of Fig. 2(a).

fit, *middle fit*, and *worst fit*. After this classification, we eliminate the *worst fit* chromosomes and replace them with variants of the *best fit* chromosomes newly generated via crossover or mutation. This strategy, referred to as *elitism*, can guarantee the quality of the solution [47]. To maintain the quality of chromosomes in every generation, we do not discard the baseline chromosomes even if they are *worst fit*. Using this elitism strategy, the *best fit* chromosomes remain the same in the next generation. We perform crossover and mutation on the *middle-fit* chromosomes with some probability of generating new variants for the next iteration.

Step 3 (Vectorwise Operations): We now extend crossover and mutation[4] so that they could work with our flat lists. As illustrated in Fig. 2(b), if we perform bitwise operations similar to standard GA, each resulting flat list may end up with an incorrect association, such as a single user simultaneously connected to multiple APs. In order to maintain the constraint on flat lists when performing crossover or mutation, DARCAS-GA suggests *vectorwise* operations, where each vector corresponds to a column of the matrix of x_{ij}'s and indicates which AP out of the possible $j \in [1, n]$ is associated with user i, as shown in Fig. 2(b). Thus, we only swap column vectors (instead of bits) to satisfy the first constraint of our problem. Mutations are more straightforward as we only change vectors in a manner appropriate to the constraint of our problem. Thus, feasible associations are ensured. We repeat steps 2 and 3 until no significant improvement over the fitness function is obtained. It is noteworthy that the association control problem is NP-hard [14], [26], [48], which means that such a problem can be solved by a nondeterministic polynomial-time algorithm. To prove the association control problem solved by DARCAS-GA in a deterministic polynomial time (i.e., P), we must validate $P = \text{NP}$. Fortunately, this problem can be converted into an equivalent boolean satisfiability problem by exploiting Cooke's theorem, which makes this problem NP-complete. NP-complete can be effectively represented in the GA. To obtain the optimal solution for this problem, one way is to deliberately acquaint DARCAS-GA of 100% RQD from all the associated stations in the network.[5] This would allow us to solve the upper bound for the optimal solution in a deterministic polynomial time; thereby we can prove that the default solutions below this optimal bound are also $P = \text{NP}$.

[4]Crossover swaps bits across chromosomes at randomly selected positions while the mutation changes random bits within a chromosome. Thus, both can lead to a better solution.

[5]We will see the details in Section VI-B.

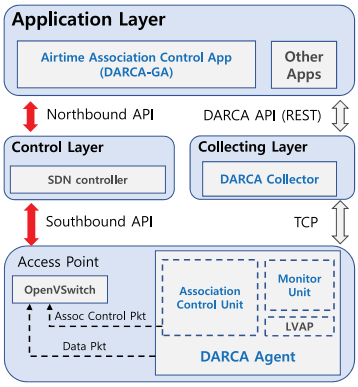

Fig. 3. DARCAS system architecture.

IV. DARCAS Design

A. Architecture of the DARCAS System

As shown in Fig. 3, DARCAS extends a general SDN framework using its airtime association control app in the application layer, the collector in the collecting layer, and the agent in each AP (i.e., the infrastructure layer). The application layer applies the DARCAS-GA algorithm by interacting with the SDN controller in the control layer and the DARCAS collector in the collecting layer via the northbound API and DARCAS API, respectively. We use floodlight [49] (a widely used SDN controller) in the control layer. The controller provides flow management and (re-)association control (i.e., extended OpenFlow [16]) for northbound applications and sends the control plane messages to the southbound DARCAS agent and Open vSwitch [50] (OVS) in each AP.

The DARCAS agent is located in the infrastructure layer to collect data related to airtime utilization in its monitor unit. The monitoring unit primarily observes the following channel parameters.

1) `CHANNEL_TIME`: Time (in ms) spent by a radio on a channel.
2) `CHANNEL_TIME_BUSY`: Time for which the primary channel is sensed as busy (activity or energy detection).

This monitoring unit reports wireless environmental information (e.g., channel time, station's transmitted/received

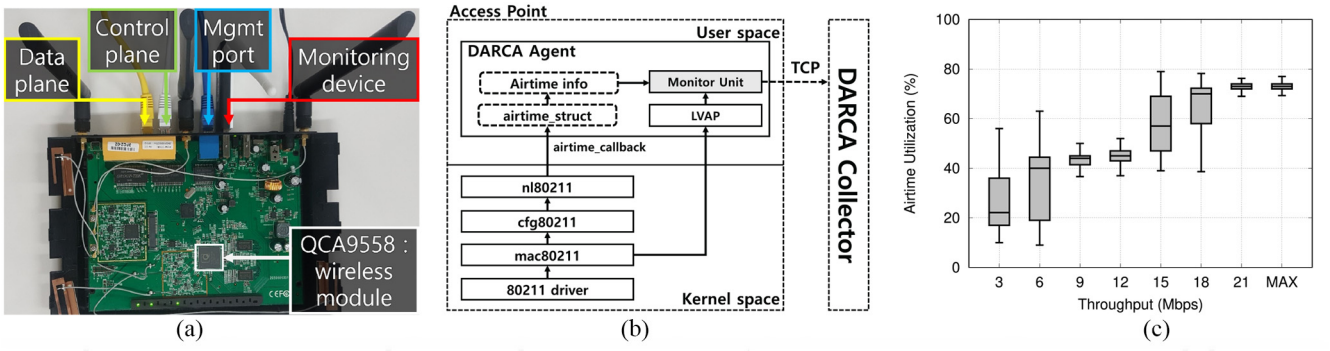

Fig. 4. Implementation and measurement study *in the wild*. (a) Snapshot of DARCAS's AP H/W setup. (b) Components of the DARCA agent in an AP. (c) Airtime utilization of 54 Mb/s per MCS rate.

packet by station, data rate, station transmission/reception power, SNR, and AP's transmission power) every 100 ms[6] to the collecting layer via TCP connection. The association control unit applies (re-)association rules using the LVAP introduced in [18] and [20]. The DARCAS agent forwards association control packets (e.g., probe request, authentication request, and association request) to the central controller by sending them to the OVS. Below, we describe how the components of DARCAS interact with the SDN framework for initial association and dynamic association control.

Initial Association Control: When a new station arrives at the network, it sends a probe request message to join. This message is forwarded from the DARCAS agent to the central SDN controller. The airtime association control app in the application layer computes the DARCAS-GA algorithm using airtime and network information and sends the association decision to the association control unit of the target AP via the SDN controller. The target AP sends back the probe response to the new station after updating the LVAP MAC address. The station then conducts the authentication and association processes using the related requests and responses and, finally, joins the network.

Dynamic Association Control: The airtime association control app periodically executes DARCAS-GA on the reported airtime- (every 100 ms) and network-related information. The performance from the viewpoint of periodicity is investigated in Section VI. The application redistributes the associations based on the gain in bandwidth in the intended stations. Our setting avoids the *ping-pong effect* and, unnecessary dynamic exchanges in station between APs, by adopting a slack variable, which is detailed in Section VI. When a reassociation decision is made, the airtime association control app sends the decision to the association control unit of the target AP (move-in station) via the SDN controller. The association control unit of the target AP adds the LVAP information of the move-in station and updates the network for the upcoming packets by sending a gratuitous proxy ARP message. Afterward, the airtime association control app notifies the previous AP to send a series of channel switch announcement (CSA) messages to the move-in station. Then, the move-in station is dissociated from its given AP and switched to the channel (if necessary) of the target AP to receive beacon messages. The airtime association control app sends a REMOVE_LVAP message to the association control unit of the previous AP and notifies the target AP to send a series of beacon messages to the move-in station. The move-in station receives one of the beacons from the target AP and finally joins the network.

Handover Overhead Incurred by Periodic Redistribution: When a redistribution happens, the corresponding LVAP (hosted by physical AP for every station) of that particular station is detached from its existing physical AP and migrated to the target AP. Thanks to the LVAP migration between physical APs, the handover process is almost invisible from the station because layer 2, latency incurred by the LVAP handoff process, is very minute. In particular, the layer 2 average delay experienced by a station in the same channel is around 18 ms. This latency is so low that the throughput perceived by a TCP connection could not be affected even if the handover happens every 100 ms [51]. It is noteworthy that the LVAP migration is supported between APs of different channels; this feature allows the dynamic redistribution in the WLAN overlapped by APs of different frequency channels, however, this comes with a little increase in the handoff cost (i.e., $\leq$ 40 ms of layer 2 average latency [51], [52]). The slight increase in handoff cost is due to the series of events taking place during the channel switching. In summary: 1) the associated AP sends the CSA beacons to the station; 2) the station receives the CSA beacons and switches the channel accordingly; and, finally 3) the controller migrates the LVAP to the target AP [21].

B. DARCAS Implementation

As shown in Fig. 4(a), we implemented the DARCAS agent and extended the LVAP component with airtime support on top of the TP-LINK Archer C7 v2 [53] equipped with Qualcomm Atheros QCA9558 CPU. We used an additional wireless LAN adapter, TL-WN722N v1 [54], to scan the WiFi channels of neighboring APs. We replaced the vendor operating system with OpenWrt (Chaos Calmer v15.05.1) [55] on top of the TP-Link wireless router. We used the click modular router to generate and manage the AP's LVAPs and send station information from the AP to the controller. DARCAS supports

[6]We set 100 ms as the longest handoff in our system to take 80 ms as in [21].

OpenFlow (*de facto* an open-source instantiation of the SDN) and decouples control and the data planes.

To execute the airtime association control algorithm through the global view in the SDN controller, each AP must send station-related airtime information to the application layer via the DARCAS collector in the collecting layer. For this, the desired information collected from the kernel space is sent to the user space on the Linux wireless system. As shown in Fig. 4(b), the DARCAS agent uses the `netlink` protocol as follows.

In the kernel space, the data related to the stations and airtime info[7] are collected by the `mac80211` from the `80211 driver`. The configuration API `cfg80211` bridges the airtime info from the kernal space to the user space using netlink interface `nl80211`. In the user space, the DARCAS agent incorporates two important components. First, the `airtime_struct` collecting the `airtime info` through `airtime_callback`. Second, the LVAP component collecting data (e.g., throughput, RSSI, SNR, etc.)[8] generated during packet exchange with the associated stations through `mac80211`. The DARCAS agent periodically (every 100 ms) accumulates these pieces of information in its monitoring unit and then transfers them to the DARCAS collector via a TCP connection, which in turn are transferred to the airtime association control app in the application layer. All reported data in the airtime association control app are processed by DARCAS-GA.

DARCAS-GA examines the BSR of each station with the current AP association. If the BSR of a station is not satisfied with the existing association, then an alternative AP is searched to fulfill its demand. The association redistribution decision is performed by the SDN controller using the related OVS. Thanks to LVAP, the reassociation overhead is almost invisible from the station (i.e., the station does not realize disruption in its services).

V. Testbed Experiments

To evaluate the effectiveness of the DARCAS system, we created two testbed settings: 1) the initial association control scenario shown in Fig 1(a) and 2) dynamic association control scenario shown in Fig 1(b)–(d).

A. Measurement Study in the Wild: Airtime Utilization

Before conducting the testbed experiments, we performed a measurement study *in the wild*. This study showed airtime utilization as a function of generated traffic loads. We set a close distance (e.g., shorter than 25 m) between an AP and station so that the MCS data rate was 54 Mb/s (the maximum data rate of 802.11g). We generated constant bit-rate traffic ranging from 3 to 21 and 54 Mb/s. We performed the experiments 1000 times and measured `CHANNEL_TIME_BUSY` of `CHANNEL_TIME` in the AP. Fig 4(c) shows a boxplot of airtime usage at a fixed link data rate (i.e., 54 Mb/s) as a function of bandwidth generated by the station. This figure shows a close correlation between the residual data rate and airtime utilization. We also observed that the wireless link data rate of 54 Mb/s yielded a maximum allowed bandwidth of 21 Mb/s, due to the channel impairments, such as slow fading and path loss, etc., as reported in [56]. This measurement study clearly shows that if we intelligently associate stations to APs, we can increase the airtime utilization (and, thus, the aggregated bandwidth) of the WiFi network. The next section describes how association control utilizes this room to improve the BSR by eliminating a naïve assumption that each station always consumes all of its allowed bandwidth.

B. Experimental Results: Initial Association Control

For the RQD, a deterministic traffic was generated on each station using the panoramic YouTube videos, whose average bitrate was predetermined [57]. For instance, for the RQD of 3 Mb/s, we played Chicago video [58] at (1080p, 60FPS) quality, which had an average bitrate of 3.41 Mb/s. Similarly, for RQD of 10 Mb/s, we played the same video at (1440p, 60FPS) quality (equivalent to the average bitrate of 10.60 Mb/s). For some RQDs (e.g., 18 Mb/s), we used a collection of videos played simultaneously in different tabs, such as one Orleans [59] video played at (1440p, 60FPS—generating 12 Mb/s of throughput) and two Chicago videos played at (1080p, 60FPS—generating 3.41 Mb/s) qualities. In summary, we realized the RQD on each station by playing the panoramic videos streamed from YouTube at diverse frame rates and resolutions.

Testbed Setup: As shown in Fig 1(a), station #1 (STA1) was associated with access point #1 (AP1) and had an allowed bandwidth of (i.e., BW) 54 Mb/s,[9] and an RQD of 3 Mb/s. STA2 was associated with AP2 and had a BW of 18 Mb/s with an RQD of 6 Mb/s. Finally, 4 s later, STA3 arrived at the network with an RQD of 10 Mb/s and became associated with an AP based on association controls, namely, PF and DARCAS.

Experimental Results: Fig. 5(a) shows the throughput of STA1∼3 and the sum of the three stations. We conducted the experiments ten times in each testbed setup and reported the average results. As shown in Section II-A, STA3 arrived at the WiFi network after 4 s, and the central controller received its association request, allowing it to join AP2 dependent on the PF-based decision. As shown in Fig. 5(b), the airtime utilization of AP2 already reached approximately 80% of its capacity (although it fluctuated). Therefore, associating STA3 with AP2 could not improve the aggregate throughput in the WiFi network. Fig. 5(c) shows the throughput of the three stations and their sum. The aggregated throughput of the WiFi network was improved by only approximately 25%. Fig. 5(d) clearly shows that STA3 is associated with AP1 by the central controller because DARCAS detected idle airtime slots in AP1. Because of this decision, the airtime utilization of AP1

[7]Note that the channel's airtime (or channel occupancy time) depends on the offered load (i.e., RQD) from the stations. We will see the details of airtime usage as a function of bandwidth generated by stations in the following section, i.e., Section V-A.

[8]These information play a crucial role in association decision. For instance, the station throughput provides an estimate of the effective bitrate between the station and AP that can be used for computing the BSR.

[9]According to the measurement study in [56], the available bandwidth is usually around 21 Mb/s and can vary with vendor and settings.

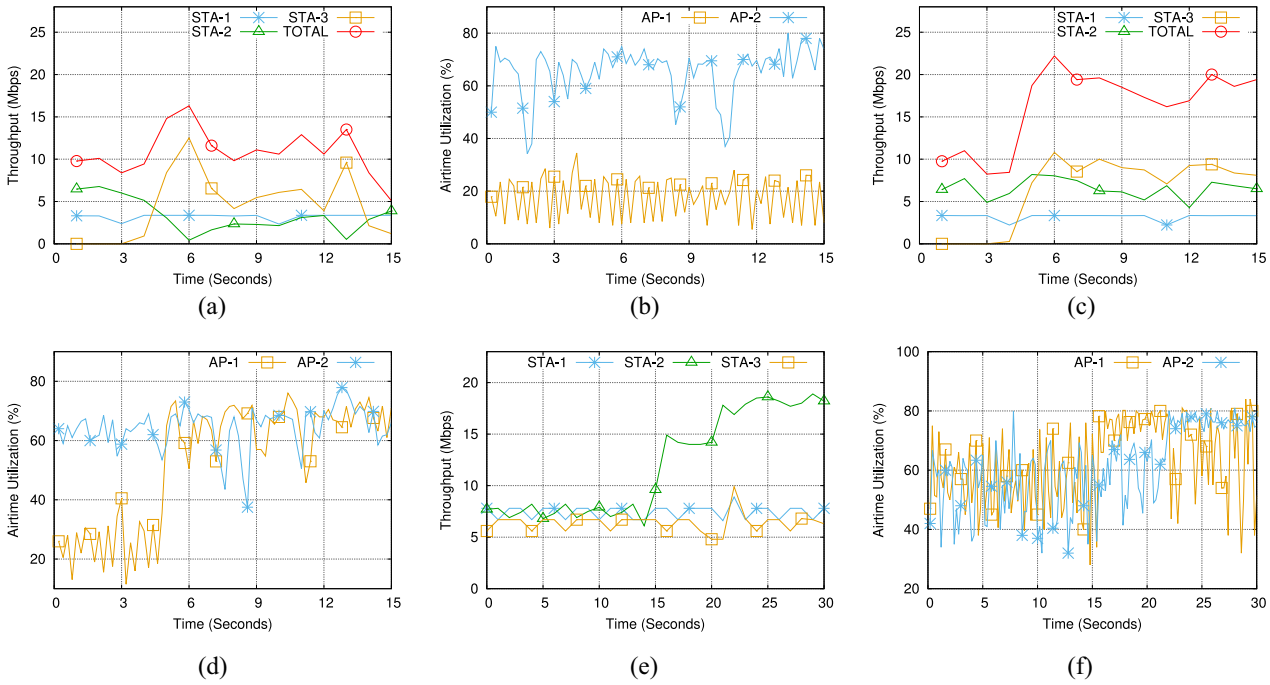

Fig. 5. Testbed experiments: (a) and (b) association control by PF; (c) and (d) association control by DARCAS; and (e) and (f) dynamic association redistribution by DARCAS.

improved from approximately 20%–70%. These experimental results clearly show that initial association control is crucial for improving network performance. This is further investigated in Section VI with scalability.

C. Experimental Results: Dynamic Association Redistribution

Testbed Setup: As previously illustrated in Fig 1(b)–(d), STA1 was associated with AP1 and had a BW of 54 Mb/s and RQD of 7 Mb/s. STA3 was associated with AP2 and had a BW of 36 Mb/s and an RQD of 6 Mb/s. STA2 was associated with AP1 and had a BW of 36 Mb/s and an RQD of 7 Mb/s. However, after 15 s, the RQD of STA2 changed to 36 Mb/s owing to its application. The following section shows how DARCAS's dynamic association redistribution, considering traffic load changes, improves aggregated throughput.

Experimental Results: Fig. 5(e) shows the throughput behaviors of STA1∼3 whose RQDs were 7, 7, and 6 Mb/s, respectively. Between 0 and 15 s, their RQDs were fully satisfied as the bandwidths provided by the WiFi network were sufficient. However, after 15 s, the RQD of STA2 changed from 7 to 36 Mb/s. Thus, the demand exceeded the bandwidth provided by the network. As illustrated in Section II-B, DARCAS made an intelligent decision via dynamic (or periodic) association redistribution to change the associations of S2 and S3 to AP2 and AP1, respectively (i.e., AP switching). Thus, STA2 solely accessed AP2, and its throughput improved after 20 s. Fig. 5(f) shows the airtime utilization of DARCAS's dynamic association redistribution. After 20 s, the airtime utilization of both AP1 and AP2 was 80% (the practical maximum). Note that airtime utilization of AP1 fluctuated due to contention between STA1 and STA3, whereas that of AP1 was stable owing to exclusive access by STA2; this shows that dynamic association redistribution is crucial to improving the UX of the service reliability of WiFi.

VI. Simulation

In this section, we present our simulation environment, provide performance metrics, and compare four algorithms via NS-3 simulations to quantify their throughput, fairness, and BSR gains.

1) *Strongest Signal First (SSF):* Each station connects to the closest AP based on the strongest signal strength.
2) *Proportional Fairness-GA:* A GA that allocates bandwidth to clients in proportion to their data rates and maximizes the sum of the logarithms (i.e., bandwidths of all clients) in a polynomial time.
3) *DARCAS-GA:* A GA that periodically finds the best association distribution of the maximum BSRs in a polynomial time.
4) *DARCAS-PSO:* Particle swarm optimization (PSO) is a stochastic population-based searching heuristic that finds the best association distribution of the maximum BSRs in a polynomial time.
5) *DARCAS-SA:* Simulated annealing (SA) is a probabilistic search heuristic that finds the best association distribution of the maximum BSR. The run-time complexity increases with the increase in the number of stations.
6) *OPTIMAL:* All stations and APs are searched in a brute-force manner (i.e., exponential search space).
7) *Maximum Frequency Selection (MFS):* A suboptimal algorithm that assigns high throughput users to the AP

with lower airtime utilization (or higher residual airtime) in linear time.

8) *Probabilistic Frequency Selection (PFS):* A suboptimal algorithm that enforces the user to join an AP with the probability of maximum transmission.

Note that every station associated with an enterprise or academic network is connected to an AP every time its association membership can be changed in a semionline manner (i.e., periodic redistribution). We further introduce a slack parameter α (usually ranging from 1% to 16% of the throughput gain in our settings) to avoid the *ping-pong effect*, which causes unnecessary dynamic exchanges of a station between APs. We present three deployment scenarios: 1) shopping mall; 2) conference hall; and 3) offices.

A. Simulation Setup

We conducted simulations and compared the four algorithms using the discrete-event simulator NS-3 [60]. We deployed nine 802.11g APs on a 3×3 grid, where the distance between adjacent APs was set to 100 m with frequency planning (i.e., nonoverlapping channels of 2.4 GHz), as in [14], [15], [48], and [61], over an area of 300 m $\times$ 300 m. We report the results of three scenarios: 1) shopping malls (high mobility with sparse station density); 300 m $\times$ 300 m. We report the results of three scenarios: 1) shopping malls (high mobility with sparse station In the shopping mall scenario, we uniformly deployed 90 stations in the target area, where 10% of the nodes were static, whereas 90% were mobile. In the conference scenario, we deployed 90 nodes, where 50% were fixed at the center, 50 m$\times$, and 50 m grid near the center of the nine-AP grid network, and the remaining 50% of the nodes were mobile. In center, 50 m$\times$, and 50 m grid near the center of the nine-AP grid network, and the remaining 50% of the nodes were We assumed that the users moved around a corridor with a fixed boundary. We randomly set waypoints in the target areas and set predefined maximum speeds of 0.4, 0.8, and 1.6 m/s for the mobile users. Note that 1.6 m/s is equivalent to 5.76 km/h, which is 1.5 times faster than regular walking. To add a realistic factor, we periodically changed the directions of walking with an offset of $[-10, 10]$ degrees. We assumed that all users had the same priority. Unless otherwise specified, we used the log-distance path-loss model with a reference distance of 1 m, reference loss of 46.678 dBm, and loss exponent of 3. The transmission power was set to 20 dBm, and the RQD per user was randomly chosen in the range 15 Kb/s–3 Mb/s. We varied the number of mobile stations to observe the behavior of algorithms in high-, moderate-, and low-mobility scenarios. We report the average values of 50 runs with a 95% confidence interval to mitigate any dependence on specific patterns of association and to determine the general behavior of the algorithm.

B. Simulation Results

Average BSR: Fig. 6(a) shows the average BSR of the three algorithms: 1) SSF; 2) PF-GA; and 3) DARCAS-GA, in the shopping mall scenario as a function of the controller periods (i.e., periods of association redistribution). For DARCAS-GA, we plot three flavors, namely, DARCAS with 10%, 30%, and 50% of the stations with known required bandwidths (KRBs). We varied the periods of the controller to assess the performance of the algorithms in high, moderate, and low-mobility scenarios. As shown in Fig. 6(a), DARCAS with three flavors outperformed the other two algorithms in a shopping mall (high mobility) scenario. The performance of all algorithms degraded as the period increased. The second overall trend was that performance improved as the proportion of KRB increased. Fig. 6(b) and (c) shows similar trends. Comparing Fig. 6(a) and (c) demonstrates that the high-mobility (i.e., shopping mall) scenario had a smaller effect on the average BSR performance than the low-mobility (i.e., office) scenario. Interestingly, the average performance of SSF and PF-GA degraded further in the conference [dense deployment, Fig. 6(b)] scenario than in the shopping mall and office scenarios [sparse deployments, Fig. 6(a) and (c)]. The performance gain based on the proportions of the KRB increased to greater than that obtained in sparse networks, namely, the shopping mall and office scenarios. Fig. 6(d) shows the average BSR performance as a function of proportions of the KRB. DARCAS, at five different controller periods (1, 3, 5, 10, and 20 s), outperformed the PF-GA and SSF while sharing similar trends. The average BSRs of all five flavors increased when the proportions of KRB were smaller than 50% and were saturated above 50% of its proportions.

Aggregated Throughput: Fig. 6(e) shows the results of the aggregated throughput in the conference scenario. We omitted the results of the shopping mall and office scenarios for the sake of brevity as they shared similar behaviors. Again, DARCAS with five different controller periods (1, 3, 5, 10, and 20 s) outperformed the PF-GA and SSF, and the aggregated throughput increased with the proportions of the KRB. DARCAS with controller periods of 3–20 s performed worse than PF-GA because the period of the latters controller was set to 1 s providing a greater opportunity to enforce its objective function. Notably, all five flavors of DARCAS converged in terms of the aggregated throughput once the proportions of the KRB exceeded 50%.

Jain's Fairness Index (JFI): Fig. 6(f) shows JFI [62] for the three algorithms in the conference scenario. For the JFIs, we used the BSR instead of per-station throughput because each station randomly selected UDP traffic ranging from 15 Kb/s to 3 Mb/s and needed a normalized standard per station for comparison. As shown in 6(f), DARCAS with five different controller periods (1, 3, 5, 10, and 20 s) shows similar behaviors: JFIs of BSR increase as the proportion of KRB increases. DARCAS with controller period of 1 s shows the best JFI. However, PF-GA shows unfairness and remains the same because of its ignorance of the KRB.

Airtime Utilization: Fig. 6(g) shows the airtime utilization of the three algorithms. DARCAS had four flavors according to the proportions of KRB, namely, 10%, 30%, 50%, and 70%. Three algorithms (including the four flavors of DARCAS based on proportions of the KRB) show flat airtime utilization regardless of the controller period. DARCAS with both 50% and 70% of the KRB reached the same

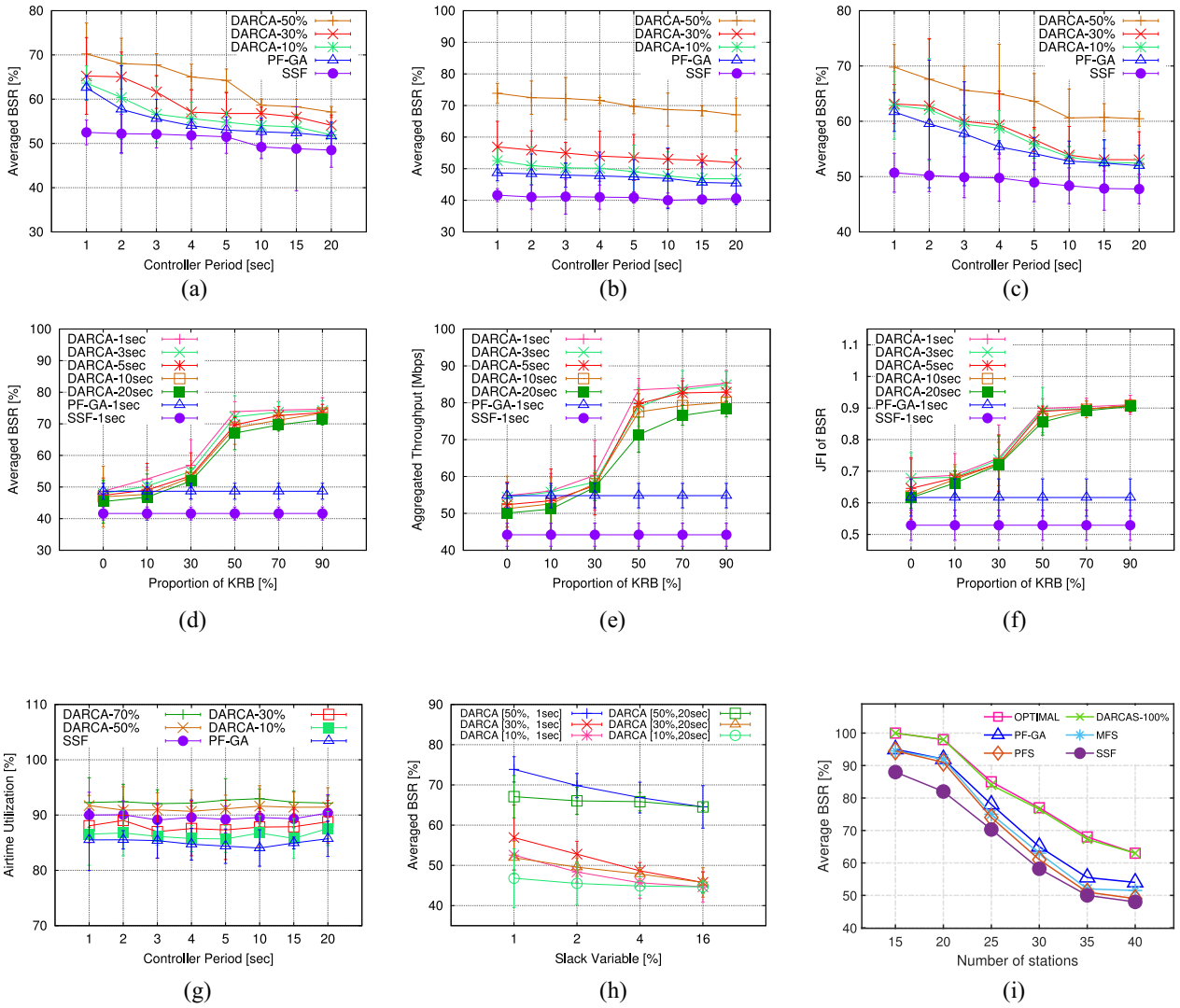

Fig. 6. Simulation results: (a)–(c) average BSR for shopping mall, conference, and office scenarios; (d)–(f) average BSR, aggregated throughput, and JFI of BSR in conference scenario; (g) airtime utilization with periods of the controller ; (h) averaged BSR with slack variable α; and (i) comparison of diverse algorithms with different computational complexities.

maximum airtime utilization. The SSF and PF-GA utilized more than 80% of the airtime of the network, but the difference in airtime utilization between them and the four flavors of DARCASs was approximately 10%. However, the average BSR, aggregated throughput, and JFI were different. This proves that the DARCAS algorithm is cost effective (airtime versus performance).

Slack Variable α: Fig. 6(h) shows the average BSR performance of DARCAS as a function of the slack variable α. We introduce α as a slack variable to avoid the *ping-pong effect*, which causes unnecessary dynamic exchanges of a station between APs by setting a bar of reassociation (i.e., handoff threshold). A value of α of 1% meant that the station changed its association to another AP if the gain in BSR after switching to a new AP was higher than 1%. We varied the proportions of the KRB and controller periods to observe the behavior of DARCAS. All the flavors of DARCAS yielded a worse performance as α increased. This means that the 1% setting is adequate for the threshold to avoid the *ping-pong effect*. We also confirmed that DARCAS with higher proportions of KRB and shorter controller periods yielded a better average BSR performance.

Comparison With Different Computational Complexity Schemes: Fig. 6(i) shows the DARCAS-GA comparison with other algorithms of diverse computational costs. The OPTIMAL algorithm is a cost-prohibitive algorithm and requires exponential computational time, i.e., $2^{\text{poly}(n)}$ to solve the problem. On the contrary, the PF-GA and DARCAS-GA are metaheuristic algorithms that require polynomial time, i.e., poly(n) to find a solution. Finally, the suboptimal schemes, namely, MFS and PFS [46], [63] require linear time, i.e., $O(n)$ for finding a solution. We have slightly modified the default mechanism[10] of MFS and PFS in order to make them compatible for our scenario. The y-axis represents the average BSR,

[10]That is, the default MFS and PFS are intended for searching for the best frequency for the maximum transmission of a cognitive node, whereas in our case, they are searching for the best AP that can better service the maximum transmission of stations.

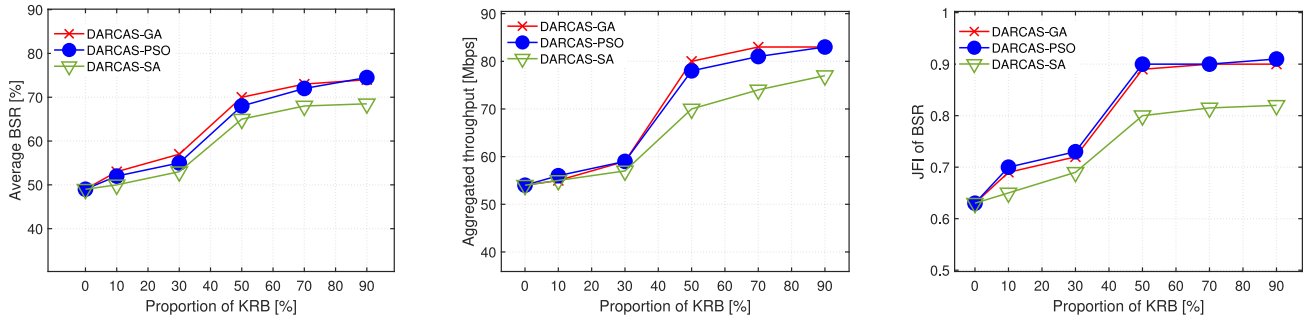

Fig. 7. Simulation results: (a)–(c) comparison of average BSR, aggregated throughput, and JFI of DARCAS variants.

and the x-axis represents the number of deployed stations (i.e., user density). We show the results of a small-scale setup (i.e., three APs and 40 stations at most) owing to the exponential search space of the OPTIMAL scheme. Interestingly, the metaheuristics of DARCAS-GA with 100% of KRB delivered a performance very close to the OPTIMAL while saving a large amount of computational time. As evident, the PF-GA's average BSR degrades exponentially when the number of stations is increased (i.e., beyond 20 stations). The MFS and PFS, although, show a similar trend of declining with an increased number of stations. However, with a meticulous look, the PFS has a slightly lower average BSR than MFS. This impact is because of using a penalty metric in PFS for reducing the reassociation delays. Finally, as expected, the SSF has the lowest performance in terms of average BSR. The reader might notice that 100% of KRB is not realistic, considering that streaming (e.g., video and audio) and gaming traffic comprise less than 70% of Internet traffic, the maximum RQD of which can be easily estimated [12]. We set an unrealistic value (100%) to show how the DARCAS-GA heuristic algorithm approaches the optimal value.

DARCAS Variants: Herein, we present variants of DARCAS by exploiting three different metaheuristic algorithms for solving our objective problem. We consider the conference setting and evaluated the average BSR, aggregated throughput, and JFI of BSR in Fig. 7. In general, the DARCAS-SA shows the worst performance for all the three cases in Fig. 7(a)–(c). The reason behind this is SA algorithm has a considerably higher complexity than the other two variants and does not converge sufficiently with the same number of iterations. In contrast, since both the GA and PSO are population-based search methods, therefore they almost deliver the same trend for all the cases. More specifically, in Fig. 7(a) and (b), the DARCAS-GA has a slightly improved performance than DARCAS-PSO. However, in terms of fairness of BSR in Fig. 7(c), the DARCAS-PSO is slightly better.

VII. Related Work

AP Association Control in WLANs: Studies on operational wireless LANs (WLANs or WiFi) have shown that user load is often unevenly distributed among wireless APs, and this significantly degrades the UX of the service reliability of WiFi [7], [9], [10], [24], [27]. The conventional AP selection mechanism (i.e., association) on the users side relies on the RSSI as an association criterion [36], [68]–[70]. Thus, users associated with APs in their vicinity with uneven load distributions receive a poor UX and a lower aggregated network throughput. Recent studies on operational WLANs have shown that traffic imbalance can be alleviated by intelligently associating the user to APs, called association control, reported as NP-hard [14], [26], [48]. Benjerano *et al.* [48] proposed approximate algorithms that achieve a max–min fair bandwidth allocation. However, max–min throughput fairness can significantly reduce the aggregated throughput in multirate WLANs because the max–min fairness problem is designed for single-rate WLANs. To overcome this limitation (i.e., throughput-based fair bandwidth allocation), Li *et al.* [14] defined the PF (or the time-based fair scheduling) problem, which is also NP-hard, and demonstrated that the algorithm improves the aggregated throughput 2.3 fold, compared to that of max–min fair allocation. However, PF allocates the maximum bandwidth allowed by the wireless link between the station and AP while ignoring the service reliability to the user. Furthermore, maintaining a fair and efficient bandwidth allocation in PF is not feasible for a long time, as the dynamic reassociation would cause disruption of ongoing sessions, which would yield excessive handoff latency.

SDN-Enabled WLAN Frameworks: In conventional WLANs, clients solely decide on which AP to associate with. Murty *et al.* proposed DenseAP [71] which uses the idea that better AP association decisions can be made with a global view of the WLAN network instead of a local view. In DenseAP, only modifications on the AP side are made to aggregate workload information and provide enhanced association controls using the global view. However, the system has certain performance limitations because it cannot work around the delay involved in the handoff (i.e., reassociation delay). To cope with this delay, Schulz-Zander *et al.* [18], [20] proposed Odin. Odin builds on an LVAP abstraction to address the complexity of the IEEE 802.11 protocol stack. LVAPs virtualize the states of association and separate them from the physical APs. They also allow the associated clients to handoff without triggering the entire reassociation mechanism.

TABLE I
THOROUGH COMPARISON OF DARCAS WITH THE RELATED WORKS

DARCAS vs. Related Work	Association (User-AP)	Approach	Handover Cost	Considering UX	Dynamic Redistribution	Experimental Evaluation	Simulation Evaluation
DARCAS	AP-Driven	Semi-online	Low	*o*	*o*	*o*	*o*
Bayhan *et al.* [64]	AP-Driven	Semi-online	High	*o*	*o*	×	*o*
Manzoor *et al.* [65]	AP-Driven	Semi-online	High	×	*o*	*o*	×
Chen *et al.* [66]	AP-Driven	Semi-online	High	×	*o*	*o*	*o*
Suresh *et al.* [51]	AP-Driven	Semi-online	Low	×	×	*o*	×
Nofal *et al.* [67]	AP-Driven	Offline	High	×	×	*o*	*o*
Li *et al.* [14]	AP-Driven	Offline	High	×	*o*	×	*o*
Bejerano *et al.* [15]	User-Driven	Offline	High	*o*	*o*	×	*o*
Balbi *et al.* [68]	User-Driven	Online	High	*o*	×	*o*	*o*
Xu *et al.* [36]	User-Driven	Online	High	*o*	×	*o*	*o*
Bangolae *et al.* [69]	User-Driven	Online	High	*o*	×	*o*	×
Nah *et al.* [70]	User-Driven	Online	High	*o*	×	×	*o*

Current proposals do not consider the bandwidth required by the user and that allocated to the user by the network resource (i.e., airtime utilization) in order to achieve optimal user satisfaction. However, DARCAS sets a UX-related standard called the BSR and implements the standard in an algorithm called DARCAS-GA that periodically finds the suboptimal distribution of associations in a polynomial time. In summary, DARCAS improves the fairness and the aggregated throughput of the network while maintaining the user's service reliability. More specifically, DARCAS controls the reassociation distribution dynamically in a semionline fashion such that overall network throughput increases while the UX of the user is satisfied (i.e., by satisfying its BSR). Additionally, DARCAS keeps the L2 and L3 levels handover invisible from the user by leveraging LVAP, thereby preventing the users from experiencing reassociation latencies. A detailed comparison of DARCAS versus the related works is shown in Table I.

VIII. DISCUSSION AND FUTURE WORK

Although we provided a number of important insights from the testbed experiments and simulation studies. However, there are some limitations that remain as our future work.

Fine-Tuning Periodicity: There is a tradeoff in choosing an association control algorithm. The online-association algorithms abruptly make an association decision (despite insufficient information) as the user arrives that further deteriorates fairness and aggregated throughput over time [14], [72]. In contrast, offline association algorithms encounter delays in the association decision that worsens the aggregated throughput [15]. To strike a balance in collecting sufficient monitored data while making the association decision in the due period (if necessary), DARCAS adopts a semionline algorithm by incorporating a parameter called controller period. In particular, this parameter controls the periodicity of the semionline algorithm. In our current setting, the threshold for the controller period is set to 5 s owing to its decent performance in terms of average BSR, aggregated throughput, and fairness as illustrated in Fig. 6(d)–(f). In the future, we would like to further explore a self-adaptive value for this parameter to ensure intelligent association decisions in optimal time.

Impact of Handover Delays on Burst Flows: Thanks to the LVAP, the layers 2 and 3 handover costs are negligible for the associated stations when they are moving from one physical AP to another (either in the same or different channels). However, a short or a burst flow whose duration is less than the handoff latency could be lost if its transmission coincidentally happens during the handover process. In the current implementation, we have introduced slack parameter α (discussed in Section VI) for preventing the unnecessary and frequent exchange of a station between APs, which ultimately reduces the packet drop of such bursty flows. In the future, we would like to explore more effective ways to tackle this issue.

Required Bandwidth Estimation: In this article, we used the traffic generated from YouTube due to the fact that a large volume of the WLAN traffic (around 82% [73]) is composed of Video on Demand (VoD) services, such as YouTube, Netflix, etc. [74]. Additionally, the VoD flows have strict quality requirements, and by leveraging QoS flags, their throughput can be estimated [75], [76]. We envision that the throughput estimated from VoD traffic could be one way to determine the RQD of a reasonable proportion of the associated stations (as discussed in Section VI-A). However, the stations with flows generated by different applications, such as FTP, HTTP, etc., in that WLAN would still be unknown to the DARCAS. In the future, we would like to explore more effective ways to estimate the maximum possible RQD of the stations.

IX. CONCLUSION

Herein, we propose and verify an SDN-enabled WiFi system called DARCAS with a GA called DARCAS-GA to fully resolve the poor UX problem of WiFi service reliability through periodic association redistribution in a central SDN controller. Our real-life experiments and extensive simulations showed that DARCAS can achieve a better tradeoff between performance and fairness than prevalent methods and improve UX by improving the BSR. Although we provided several important insights from the testbed experiments and simulation studies, unexplored parameters persist and will be addressed in future work. Because the prototype of DARCAS ran on 802.11g, we used the same settings for our simulation study of DARCAS-GA. We envision that the 802.11n standard or higher (e.g., 802.11ac) requires a more dynamic association redistribution of the network. In future work, we plan to investigate the performance gains and tuning parameters of DARCAS-GA in such settings.

REFERENCES

[1] Y. Daldoul, D. Meddour, and A. Ksentini, "IEEE 802.11ac: Effect of channel bonding on spectrum utilization in dense environments," in *Proc. IEEE Int. Conf. Commun.*, 2017, pp. 1–6.

[2] L. Kriara, E. C. Molero, and T. R. Gross, "Evaluating 802.11ac features in indoor WLAN: An empirical study of performance and fairness," in *Proc. 10th ACM Int. Workshop Wireless Netw. Testbeds Exp. Eval. Characterization*, 2016, pp. 17–24.

[3] A. Bhartia *et al.*, "Measurement-based, practical techniques to improve 802.11ac performance," in *Proc. Internet Meas. Conf.*, 2017, pp. 205–219.

[4] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-E, A. Bhartia, and D. Aguayo, "Large-scale measurements of wireless network behavior," in *Proc. ACM Conf. Special Interest Group Data Commun.*, 2015, pp. 153–165.

[5] "Aruba Gateways and Controllers." [Online]. Available: https://tinyurl.com/2z76nct2 (Accessed: Jul. 7, 2021).

[6] "Meraki." [Online]. Available: https://meraki.cisco.com/products/wi-fi/ (Accessed: Nov. 3, 2021).

[7] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan, "Characterizing user behavior and network performance in a public wireless LAN," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2002, pp. 195–205.

[8] K. Sui *et al.*, "Characterizing and improving WiFi latency in large-scale operational networks," in *Proc. 14th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2016, pp. 347–360.

[9] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," *Wireless Netw.*, vol. 11, no. 1, pp. 115–133, Jan. 2005.

[10] M. Balazinska and P. Castro, "Characterizing mobility and network usage in a corporate wireless local-area network," in *Proc. 1st Int. Conf. Mobile Syst. Appl. Services*, 2003, pp. 303–316.

[11] Y. Zeng, I. Pefkianakis, K.-H. Kim, and P. Mohapatra, "MU-MIMO-aware AP selection for 802.11ac networks," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2017, pp. 1–10.

[12] "2018 Global Internet Phenomena Report." [Online]. Available: https://tinyurl.com/kwdtjzx2 (Accessed: Dec. 3, 2021).

[13] G. Tan and J. Guttag, "Time-based fairness improves performance in multi-rate WLANs," in *Proc. Annu. Conf. USENIX Annu. Tech. Conf.*, 2004, p. 23.

[14] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *Proc. 27th Conf. Comput. Commun.*, 2008, pp. 1004–1012.

[15] Y. Bejerano, S.-J. Han, and L. Li, "Fairness and load balancing in wireless LANs using association control," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 560–573, Jun. 2007.

[16] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[17] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined wan," in *Proc. ACM SIGCOMM Conf.*, 2013, pp. 3–14.

[18] J. Schulz-Zander, L. Suresh, N. Sarrar, A. Feldmann, T. Hühn, and R. Merz, "Programmatic orchestration of WiFi networks," in *Proc. USENIX Conf. USENIX Annu. Tech. Conf.*, 2014, pp. 347–358.

[19] Y. Yiakoumis, M. Bansal, A. Covington, J. van Reijendam, S. Katti, and N. McKeown, "BeHop: A testbed for dense WiFi networks," in *Proc. 9th ACM Int. Workshop Wireless Netw. Testbeds Exp. Eval. Characterization*, 2014, pp. 1–8.

[20] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann, "OpenSDWN: Programmatic control over home and enterprise WiFi," in *Proc. 1st ACM SIGCOMM Symp. Softw. Defined Netw. Res.*, 2015, p. 16.

[21] L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, and J. Almodovar, "Building an SDN enterprise WLAN based on virtual APs," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 374–377, Feb. 2017.

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

[23] P. Gawlowicz, A. Zubow, M. Chwalisz, and A. Wolisz, "UniFlex: A framework for simplifying wireless network control," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–7.

[24] A. Raschellà, F. Bouhafs, M. Mackay, K. Zachariou, V. Pilavakis, and M. Georgiades, "Smart access point selection for dense WLANs: A use-case," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2018, pp. 1–6.

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

[26] X. Qin, X. Yuan, Z. Zhang, F. Tian, Y. T. Hou, and W. Lou, "Joint user-AP association and resource allocation in multi-AP 60-GHz WLAN," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5696–5710, Jun. 2019.

[27] M. A. Kafi, A. Mouradian, and V. Vèque, "On-line client association scheme based on reinforcement learning for WLAN networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–7.

[28] S. Bayhan, E. Coronado, R. Riggio, and A. Zubow, "User-AP association management in software-defined WLANs," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 3, pp. 1838–1852, Sep. 2020.

[29] Z. Mao, "Throughput optimization based joint access point association and transmission time allocation in WLANs," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 899–914, 2021.

[30] J. Saldana *et al.*, "Attention to Wi-Fi diversity: Resource management in WLANs with heterogeneous APs," *IEEE Access*, vol. 9, pp. 6961–6980, 2021.

[31] W. Li *et al.*, "AP association for proportional fairness in multi-rate WLANs," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 191–202, Feb. 2014.

[32] S. Lin, N. Che, F. Yu, and S. Jiang, "Fairness and load balancing in SDWN using handoff-delay-based association control and load monitoring," *IEEE Access*, vol. 7, pp. 136934–136950, 2019.

[33] L. Xie, Q. Li, W. Mao, J. Wu, and D. Chen, "Achieving efficiency and fairness for association control in vehicular networks," in *Proc. 17th IEEE Int. Conf. Netw. Protocols*, 2009, pp. 324–333.

[34] S. Keranidis, T. Korakis, I. Koutsopoulos, and L. Tassiulas, "Contention and traffic load-aware association in IEEE 802.11 WLANs: Algorithms and implementation," in *Proc. Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw.*, 2011, pp. 334–341.

[35] F. Xu, C. C. Tan, Q. Li, G. Yan, and J. Wu, "Designing a practical access point association protocol," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[36] F. Xu, X. Zhu, C. C. Tan, Q. Li, G. Yan, and J. Wu, "SmartAssoc: Decentralized access point selection algorithm to improve throughput," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 12, pp. 2482–2491, Dec. 2013.

[37] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: Shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, Mar. 1998.

[38] E. Hossain and K. K. Leung, *Wireless Mesh Networks: Architectures and Protocols*. New York, NY, USA: Springer, 2007.

[39] J. Lakkakorpi, A. Heiner, and J. Ruutu, "Measurement and characterization of Internet gaming traffic," in *Proc. Res. Seminar Netw. Helsinki Univ. Technol. Netw. Lab.*, Jan. 2002, pp. 1–12.

[40] "Netflix Connection Speed Recommendations." [Online]. Available: https://help.netflix.com/en/node/306 (Accessed: Jun. 5, 2020).

[41] H. Balakrishnan, M. Stemm, S. Seshan, and R. H. Katz, "Analyzing stability in wide-area network performance," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 1997, pp. 2–12.

[42] M. Mirza, J. Sommers, P. Barford, and X. Zhu, "A machine learning approach to TCP throughput prediction," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2007, pp. 97–108.

[43] P. N. Marko and M. Mäkelä, *Nonsmooth Optimization: Analysis and Algorithms With Applications to Optimal Control*. Singapore: World Sci., 1992.

[44] U. Mehboob, J. Qadir, S. Ali, and A. Vasilakos, "Genetic algorithms in wireless networking: Techniques, applications, and issues," *Soft Comput.*, vol. 20, no. 6, pp. 2467–2501, 2016.

[45] G. D. Edward, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley, 2002.

[46] D. Gözüpek and F. Alagöz, "Genetic algorithm-based scheduling in cognitive radio networks under interference temperature constraints," *Int. J. Commun. Syst.*, vol. 24, no. 2, pp. 239–257, 2011.

[47] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1996.

[48] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in wireless LANs using association control," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 315–329.

[49] "Floodlight." [Online]. Available: http://www.projectfloodlight.org/ (Accessed: Jan. 6, 2020).

[50] B. Pfaff *et al.*, "The design and implementation of open vSwitch," in *Proc. 12th USENIX Conf. Netw. Syst. Design Implement.*, 2015, pp. 117–130.

[51] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise WLANS with Odin," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 115–120. [Online]. Available: https://doi.org/10.1145/2342441.2342465

[52] M. E. Berezin, F. Rousseau, and A. Duda, "Multichannel virtual access points for seamless handoffs in IEEE 802.11 wireless networks," in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, 2011, pp. 1–5.

[53] "TPLINK ARCHER C7 V2." [Online]. Available: https://www.tp-link.com/us/download/Archer-C7_V2.html (Accessed: Jan. 5, 2020).

[54] "TPLINK TL-WN722N V1." [Online]. Available: https://www.tp-link.com/us/download/TL-WN722N_V1.html (Accessed: Jan. 10, 2020).

[55] "OpenWrt." [Online]. Available: https://openwrt.org/releases/15.05/start (Accessed: Feb. 11, 2020).

[56] A. L. Wijesinha, Y. Tae Song, M. Krishnan, V. Mathur, J. Ahn, and V. Shyamasundar, "Throughput measurement for UDP traffic in an IEEE 802.11g WLAN," in *Proc. 6th Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distrib. Comput. First ACIS Int. Workshop Self-Assembling Wireless Netw.*, 2005, pp. 220–225.

[57] J. Meng, Q. Xu, and Y. C. Hu, "Proactive {energy-aware} adaptive video streaming on mobile devices," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2021, pp. 303–316.

[58] *Chicago City Tour—Downtown [Amazing Travel Experience]*. (2020). [Online Video]. Available: https://youtu.be/UWdfaNWThnA.

[59] *New Orleans 4K—Sunset Drive—Driving Downtown*. (2019). [Online]. Available: https://tinyurl.com/4fsc65nv

[60] "NS3 Simulator." [Online]. Available: https://www.nsnam.org

[61] W. Wong, A. Thakur, and S. H. G. Chan, "An approximation algorithm for AP association under user migration cost constraint," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[62] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Jan. 1998, *arXiv:cs/9809099*.

[63] D. Gözüpek and F. Alagöz, "Throughput and delay optimal scheduling in cognitive radio networks under interference temperature constraints," *J. Commun. Netw.*, vol. 11, no. 2, pp. 148–156, 2009.

[64] S. Bayhan and A. Zubow, "Optimal mapping of stations to access points in Enterprise wireless local area networks," in *Proc. 20th ACM Int. Conf. Model. Anal. Simulat. Wireless Mobile Syst.*, 2017, pp. 9–18. [Online]. Available: https://doi.org/10.1145/3127540.3127556

[65] S. Manzoor, Z. Chen, Y. Gao, X. Hei, and W. Cheng, "Towards QoS-aware load balancing for high density software defined Wi-Fi networks," *IEEE Access*, vol. 8, pp. 117623–117638, 2020.

[66] C. Chen, C. Wang, H. Liu, M. Hu, and Z. Ren, "A novel AP selection scheme in software defined networking enabled WLAN," *Comput. Electr. Eng.*, vol. 66, pp. 288–304, Feb. 2018. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2017.10.013

[67] R. A. Nofal, N. Tran, B. Dezfouli, and Y. Liu, "A framework for managing device association and offloading the transport layer's security overhead of WiFi device to access points," *Sensors*, vol. 21, no. 19, p. 6433, 2021.

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

[69] S. Bangolae, C. Bell, and E. Qi, "Performance study of fast BSS transition using IEEE 802.11r," in *Proc. IWCMC*, 2006, pp. 737–742. [Online]. Available: https://doi.org/10.1145/1143549.1143696

[70] J.-W. Nah, S.-M. Chun, S. Wang, and J.-T. Park, "Adaptive handover method with application-awareness for multimedia streaming service in wireless LAN," in *Proc. Int. Conf. Inf. Netw.*, 2009, pp. 1–7.

[71] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, "Designing high performance Enterprise Wi-Fi networks," in *Proc. 5th USENIX Symp. Networked Syst. Design Implement.*, 2008, pp. 73–88.

[72] J. Hwang, J. Choi, J. Yoo, and C.-K. Kim, "A theoretical approach to optimal association control in vehicular Wi-Fi networks," *EURASIP J. Wireless Commun. Netw.*, vol. 1, no. 1, pp. 1–12, 2014.

[73] "Cisco: Online Video to Account for 82% of all Internet Traffic by 2021!" (2017). [Online]. Available: https://tinyurl.com/yckaac4x

[74] P. Cerwall *et al.*, *Ericsson Mobility Report June 2020*, Ericsson Stockholm, Sweden, 2020.

[75] A. Toppan, P. Toppan, C. De Castro, and O. Andrisano, "A testbed about priority-based dynamic connection profiles in QoS wireless multimedia networks," in *Telecommunications Networks*, J. H. Ortiz, Ed. Rijeka, Croatia: IntechOpen, 2012, ch. 4, doi: 10.5772/38243.

[76] M. Morshedi and J. Noll, "Estimating PQoS of video streaming on Wi-Fi networks using machine learning," *Sensors*, vol. 21, no. 2, p. 621, 2021.

Muhammad Salman received the B.S. degree in electronic engineering from Balochistan University of Information Technology, Engineering and Management Sciences, Quetta, Pakistan, in 2010, and the M.S. degree in electronic engineering from Politecnico di Torino, Turin, Italy, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Inha University, Incheon, South Korea.

From October 2014 to June 2019, he worked as a Lecturer with the Electrical and Computer Engineering Department, Effat University, Jeddah, Saudi Arabia. His research interests include spy camera detection, bufferbloat mitigation, wireless networks, and software-defined networks.

Jin-Ho Son received the B.S. degree in information and communication engineering and the M.S. degree in computer engineering from Inha University, Incheon, South Korea, in 2018 and 2020, respectively.

He is currently working with TmaxSoft, Chicago, IL, USA, as a Research Engineer. His research interests include application and cloud monitoring.

Dong-Wan Choi received the B.S. degree from Inha University, Incheon, South Korea, in 2008, and the M.S. and Ph.D. degrees from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2010 and 2014, respectively.

From 2016 to 2017, he worked as a Research Associate with the Department of Computing, Imperial College London, London, U.K., and, from 2015 to 2016, he was a Postdoctoral Fellow with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. He is currently an Associate Professor with the Department of Computer Engineering, Inha University. He is currently working on designing more efficient and lightweight deep learning models and algorithms from a database perspective. His research interests span the areas of data mining, databases, big data algorithms, and scalable machine learning.

Uichin Lee (Member, IEEE) received the B.S. degree in computer engineering from Chonbuk National University, Jeonju, South Korea, in 2001, the M.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2003, and the Ph.D. degree in computer science from UCLA, Los Angeles, CA, USA, in 2008.

He continued his studies with UCLA as a Postdoctoral Research Scientist from 2008 to 2009 and then worked with Alcatel-Lucent Bell Laboratories, Holmdel, NJ, USA, as a member of Technical Staff till 2010. He is currently an Associate Professor with the School of Computing, KAIST. His research interests include humancomputer interaction, social computing, and ubiquitous computing.

Youngtae Noh received the B.S. degree in computer science from Chosun University, Gwangju, South Korea, in 2005, the M.S. degree in information and communication from Gwangju Institute of Science Technology, Gwangju, in 2007, and the Ph.D. degree in computer science from the University of California at Los Angeles, Los Angeles, CA, USA, in 2012.

He is an Associate Professor with Energy AI, KENTECH, Naju, South Korea. Before joining KENTECH, he worked as an Associate Professor with Inha University, Incheon, South Korea, from 2015 to 2022. Moreover, he has also worked with Cisco Systems, San Jose, CA, USA, as a Staff Member from 2012 to 2014. His research areas include mobile/pervasive computing, mobile systems, mobile data science, mobile-HCI, cloud computing, data center networking, wireless networking, and future Internet.