DARCA: Dynamic Association Regulator Considering Airtime over SDN-enabled WiFi

***Abstract*—Recent massive influx of mobile devices and their growing use-cases lead to access point over-provisions. Unlike in residential environments, network administrators in enterprises and universities make every endeavor to enhance user experience (UX) of WiFi network where network dynamics (e.g., traffic loads and user mobility) are usually unexpectable. However, users easily fall into dissatisfaction from existing WiFi services due to client-driven association (thereby causing *sticky client problem*) and move toward a more reliable LTE (or 5G) service without hesitation. To cope with the challenge, we propose DARCA, an SDN-enabled WiFi system. DARCA adopts a notion of Bandwidth Satisfaction Ratio (BSR), which is a closely correlated to UX, maximizes BSR with an exact measure, *airtime* (*i.e.,* channel occupancy time). We put the notion into a meta-heuristic genetic algorithm called DARCA-GA that effectively finds the sub-optimal association distribution of the maximum BSRs in polynomial time. We implement DARCA system on off-the-shelf wireless routers and SDN controller. We report real-life experimental results in considered scenarios and provide extensive simulations via NS-3 simulator to show the DARCA's performance behaviors with the scalability. With the fine-tuned settings, DARCA shows up to 80% of BSR gain compared to existing solutions.**

I. Introduction

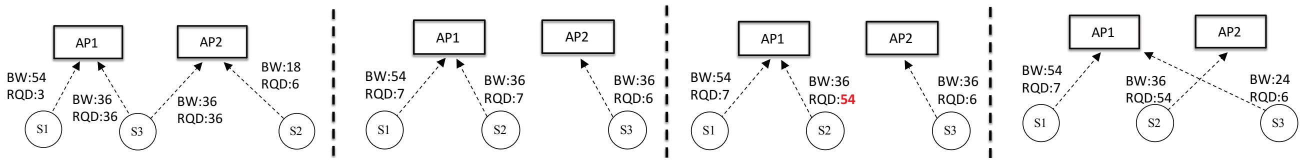
In recent years, the number of Access Point (hereafter AP) deployments in residential areas, enterprises, and universities have skyrocketed owing to massive influx of mobile devices and their growing variety of use-cases (e.g., social media, video/audio streaming, gaming, and so forth). In residential 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 selections (increased co-channel interference) and unbalanced load per AP [8]–[10], [18].

To cope with the challenges, network administrators may deploy APs in the planned manners considering network coverage and neighboring APs' frequency selections. Additionally, management solutions [2], [23] can help soothing the problems. However, operational WiFi networks have shown that user loads are often distributed unevenly among wireless access points (APs) and changed over time. These significantly degrade user experience (UX) of WiFi service reliability [3], [5], [17], [35], [42]. Due to recent huge success on video (Neflix, YouTube, Amazon Prime etc.) and audio (Spotify, Apple Music, Google Play Music etc.) streaming applications, the portion of bandwidth intensive traffic has been significantly increased in mobile network [1]. These trends make the dissatisfaction of UX even worse; finally, LTE (being replaced with 5G) is easily used as its substitute. Recent studies have found that the unbalanced loads and unfair bandwidth allocation can be greatly alleviated by intelligently associating users to APs, termed *association control*, rather than user-driven associations to APs of strongest received signal strength [20], [36].

However, allocating bandwidth fairly and efficiently to greedy users is a challenging problem; even identifying the users in the bottleneck fairness group or finding their normalized bandwidth is NP-hard [6]. State-of-the-art AP association control technology mediates user (*i.e.,* station) associations based on RSSI values or/and aggregated network throughput in a centralized manner (e.g., Aruba WLAN Controller [2] and Meraki [23]). However, after association has been made, these solutions do not have a control knob of users' dynamic traffic loads and users' location changes. Especially, client-driven association with user's mobility causes *sticky client problem* that a client first associates to an AP within its vicinity (*i.e.,* maximum data rate) but keeps staying in the same AP after moving apart. This increases the number of stations containing low Modulation and Coding Schemes (MCS) (*i.e.,* data rate) and finally degrades WiFi network performance. In addition, handoff overhead is also costly (from a few hundred milliseconds to a few seconds). Thereby, it is practically impossible to provide efficient and fair bandwidth supplies through association control of the users in multiple-AP networks without disrupting ongoing sessions; finally loses the chance of service reliability closely related to UX.

Fortunately, with advent of Software Defined Network (SDN [22]) and its huge success in datacenter domain [15], active drives have been made to gain the same "victory" in wireless domain [32]–[34], [41] to improve UX of WiFi services. Light Virtual Access Point (LVAP) [33], [34] or similar abstraction technique virtualizes association states and separates them from the physical APs [12], [43]. These techniques introduced seamless handoff ($<$ 80 ms) and feasibility of central controller-driven association control. However, the *sticky client problem* is not fully resolved yet.

To resolve this, association control should be aware of multi-rate environment of 802.11 standards (e.g., 802.11g, n, and so forth) in global view of network [20], [36] and have periodic and direct measure of residual wireless resources (*i.e.,* channel occupancy time or *airtime*) including fairness needs to be carefully considered to hit the better sweet-spot between performance and fairness. Therefore, we devise a directly UX-related standard termed as Bandwidth Satisfaction Ratio (BSR)—a ratio of required bandwidth per user to actually provided bandwidth by direct network resource (*i.e., airtime*). We then put the standard into a meta-heuristic genetic algorithm called DARCA-GA that periodically finds



(a) Association control (b) Initial traffic load (c) Load change in S2 (15 sec later) (d) Association re-distribution

Fig. 1. DARCA's motivation: (a) example of association control via proportional fairness and airtime fairness with RQD; (b) example of dynamic association redistribution: shows initial load at time 0 $\sim$ 15 seconds; (c) shows load change at 15 seconds but keep the associations same until 20 seconds; (d) shows association redistribution at 20 seconds

the best association distribution that maximize BSRs in polynomial time. We finally propose an SDN-enabled WiFi system called DARCA to validate its performance in real-life experiments considered scenarios and provide extensive simulations via NS-3 simulator to show the its performance behaviors with the scalability. Our contributions are concretely three-folds as follows:

- To make an SDN controller actively redistribute station-memberships among overlapped APs and enforce periodic-offline (*i.e., dynamic*) algorithm to be adapted to network dynamics (*i.e.,* topology and traffic pattern changes), we devise an airtime measuring agent equipped with the LVAP abstraction, which obtains airtime utilization of each AP per 100 ms and report it to airtime association control app in application layer; and finally propose an airtime-based SDN WiFi system named as DARCA, which contains AP-level *airtime agent*, *DARCA collector*, and *application-level airtime association control app* to enforce DARCA-GA's decision to the entire network (Section IV-A).
- Devise a standard of BSR, a ratio of user's required bandwidth to provided bandwidth by getting rid of the maximum throughput usage assumption in the stations with known required bandwidth (e.g., gaming, video/audio streaming, and so on), and propose a genetic algorithm called as DARCA-GA which maximizes BSR from the network, outperforms existing solutions, and performs very close to optimal (Section VI-B).
- Implement *DARCA agent* on off-the-shelf wireless routers, *DARCA collector* in an SDN controller, and *airtime association control app* in an application server (Section IV-B). We deliver testbed experiment results in considered scenarios (Section V) and provide extensive simulation results of up to 80% of averaged BSR gain compared to existing solutions. (Section VI).

II. Motivation

A. How initial association control matters?

Fig. 1(a) shows a motivating example of station association control. There are three stations (namely, S1, S2, and S3) and two APs (AP1 and AP2) in the vicinity of their locations. BW stands for maximum allowed bandwidth based on channel conditions between each AP and each station such as ESNR [27] while RQD stands for required bandwidth by each station. There have been association control studies in multi-rate WLAN settings [16], [20]. It is reported that proportional fairness (PF hereafter [20]) outperforms max-min fairness (*i.e.,* throughput-based fairness [16]). Once PF is adopted, the overall throughput of this part of network would be $54 + (36 + 18)/2 = 81$ Mbps as S3 joins in AP2 network and shares airtime of AP2 with S2. However, this network actually consumes aggregated bandwidth of $(3/54) \times 54 + ((18 - 6)/18) \times 36 + (6/18) \times 18 = 33$ Mbps considering RQD and airtime utilization of all stations. More precisely, $(3/54) \times 54$ denotes S1's obtained RQD considering available BW. $((18 - 6)/18) \times 36$ denotes BW of S3 (*i.e.,* 36) multiplied by residual airtime $((18 - 6)/18)$ from S2. $(6/18) \times 18$ is S2's obtained RQD considering available BW, which is fully supported as it is not exceeding 50% among the two stations in AP2 (*i.e.,* equal share). This example clearly shows pitfalls (*i.e.,* absence of airtime utilization and maximum BW usage assumption in every station) of existing approaches. The major reason for PF's misjudgment is rooted in an assumption that every station always consumes the maximum bandwidth until the wireless link between an AP and a station allows. In reality, however, stations such as gaming (e.g, 15Kbps [19]) or video streaming (e.g., 3Mbps$\sim$15Mbps [13], [28]) on mobile devices may not require the maximum allowed bandwidth (*i.e.,* BW). Therefore, an intelligent association control considering both airtime utilization and RQD with the best knowledge (e.g., statistical [4] and machine learning [24]) and will squeeze more space up to achieve higher aggregated throughput of the network without sacrificing fairness. Therefore, the overall throughput becomes $(3/54) \times 54 + ((54 - 3)/54) \times 36 + (6/18) \times 18 = 43$ Mbps as S3 joins in AP1 instead of AP2. It is around 30% improvement of overall throughput.

B. How dynamic association control matters?

Fig. 1(b)$\sim$1(d) show another motivating example of dynamic (*i.e.,* periodic) association control considering varying network traffic loads, Fig. 1(b) shows the topology of a part of academic/enterprise network. There are three stations, which have two APs (AP1 and AP2) in the vicinity of their locations. S1 is associated with AP1 with RQD of 7 Mbps while BW is 54 Mbps. S2 is associated with AP1 with RQD of 7 Mbps and BW of 36 Mbps. S3 is associated with AP2 with RQD of 6 and BW of 36 Mbps. As shown in Fig. 1(c), required traffic load of S2 has been changed to 54 Mbps (in red) at 15 seconds. Thus, the overall throughput of the WLAN network becomes as $(7/54) \times 54 + ((54-7)/54) \times 36 + (6/24) \times 24 = 44.33$ Mbps.

However, as shown in Fig. 1(d), a periodic-offline (every 5 seconds in this setting) association control will intelligently redistribute associations of all stations in the network. Thus, S1 remains the same membership while S2 is associated with AP2 and S3 is associated with AP1 at 20 seconds. Therefore, the overall throughput becomes $7 + 36 + 6 = 49$ Mbps. Note that both S1's RQD (*i.e.,* 7) and S3's RQD (*i.e.,* 6) are fully supported as they are not exceeding 50% of airtime utilization. The state-of-the-art AP association control technology may control user (*i.e.,* station) associations initially based on RSSI values or/and aggregated network throughput in a centralized manner (e.g., Aruba WLAN Controller [2] and Meraki [23]). However, this technology does not have control on redistributing associations after the associations have been made. Thus, this will finally lose the chance of performance enhancement 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 with bandwidth satisfaction ratio (BSR) considering both required bandwidth and airtime utilization, and then devise DARCA-GA to find 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 (a.k.a. stations). Each user in U is indexed by an integer $j \in [1, n]$ and associated with only one AP. Then our optimization problem can be formulated as follows:

$$\begin{aligned} \text{Maximize:} \quad & \sum_{j \in U} \log\Big(\sum_{i \in A} r_{ij} x_{ij} p_{ij}\Big) \\ \text{subject to:} \quad & \forall j \in U : \sum_{i \in A} x_{ij} = 1, \\ & \forall i \in A, j \in U : x_{ij} \in \{0, 1\}, \text{and} \\ & p_{ij} = \text{BSR-MAX-MIN}(i, j) \end{aligned}$$

Except for newly introduced final constraint, we follow the PF optimization model introduced in [20], which aims to maximize the sum of the logarithm[1] of the bandwidth allocated to each user. The bandwidth allocation of each user 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 will be allocated with when it connects to AP i. It is worth noting that p_{ij} is newly introduced to consider airtime utilization consumed by (if known) required bandwidth between i and j. The first constraint ensures that each user is connected to only one AP, and the second constraint specifies that each indicator variable x_{ij} should be either 0 or 1.

In the final constraint, the procedure BSR-MAX-MIN(i, j) returns the fraction of time p_{ij} that will actually be assigned to user j within AP i, given the network configuration being considered. Considering some applications such as gaming, video & audio streaming, and so on, we remove the assumption of existing work that every user always consumes the maximum allowed bandwidth (*i.e.,* r_{ij}) at the AP associated with it, we now define how the fraction of time will be distributed when some users use only part of their available bandwidth (*i.e.,* known required bandwidth). BSR-MAX-MIN(i, j) works as shown in Algorithm 1.

Algorithm 1: BSR-MAX-MIN(i, j)

Input: $i \in A, j \in U$: a pair of AP and user
Output: p_{ij} : the fraction of time allocated to user j within AP i
1 $D_i = \{j \in U \mid x_{ij} = 1\}$;
2 **foreach** $j \in D_i$ **do**
3     $p_i[j] = 0$, $q_i[j] = \frac{\min(t_j, r_{ij})}{r_{ij}}$, $d_i[j] = \frac{1}{|D_i|} - q_i[j]$;
4 ALLOCATE-RECURSIVE(D_i, p_i, q_i, d_i);
5 **return** $p_i[j]$;

6 **Subroutine** ALLOCATE-RECURSIVE (D, p, q, d)**:**
7     $D^+ = \{j \in D \mid d[j] \geq 0\}$;
8   **if** D^+ *is empty* **then**
9         $\forall j \in D : p[j] = q[j] + d[j]$
10   **else**
11         $\forall j \in D^+ : p[j] = q[j]$;
12         $\forall j \in D \setminus D^+ : d[j] = d[j] + \frac{\sum_{j \in D^+} d_{ij}}{|D| - |D^+|}$;
13   **if** $D \setminus D^+$ *is not empty* **then**
14   ALLOCATE-RECURSIVE$(D \setminus D^+, p, q, d)$;

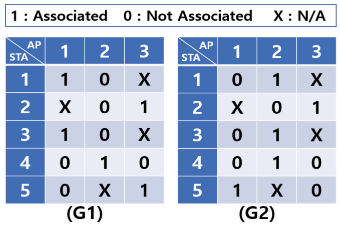
BSR-MAX-MIN initializes three arrays, namely $p_i[\ ]$, $q_i[\ ]$, and $d_i[\ ]$, where each element represents time fraction to be finally allocated to each user at AP i, time fraction actually required by each user at AP i, and remaining (or residual) fraction after assigning the time fraction as much as the equal rate (*i.e.,* $\frac{1}{|D_i|}$) 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 of user j (*i.e.,* t_j) has some slot less than r_{ij}. To represent the actual demand of time for user j, we set $q_i[j]$ to the minimum of t_j and r_{ij}.

After the initialization, the algorithm invokes a sub-procedure called ALLOCATE-RECURSIVE that recursively computes the bandwidth allocation for each user j in the max-min fairness manner. Thus, for every round, we first assign all requested resources users whose demands are less than the equal rate (*i.e.,* $\forall j \in D^+$) (Line 11), and then recalculate the remaining flows to be shared by other resources that are yet to be allocated (*i.e.,* $\forall j \in D \setminus D^+$) (Line 12). To the best of our knowledge, it is our first departure from existing work to apply the max-min fairness to *the fraction of time* to put both required bandwidth (if known) and airtime utilization into BSR in order to improve network reliability.

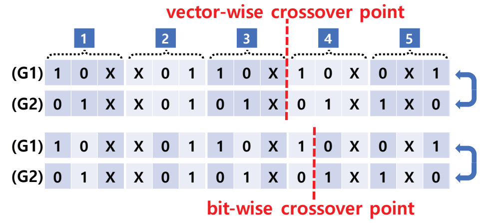
B. DARCA-GA: a genetic-based algorithm

Our optimization problem is in non-smooth as well as non-linear problem even after the relaxation of the second constraint to be $0 \leq x_{ij} \leq 1$ as each p_{ij} is determined by a non-continuous procedure, BSR-MAX-MIN. Thus, our problem belongs to a type of non-smooth optimization problem [21], where differential approaches like nonlinear programming are not applicable. Hence, we need some derivative-

[1] It is used for the fairness (*i.e.,* spreading). p_{ij} is limited and can be allocate to the stations. Thus, if the product of p_{ij} and r_{ij} is larger than base, allocating p_{ij} to the larger number of stations returns the larger.



(a) Example of association matrices of x_{ij}'s consisting of 3 APs and 5 users



(b) Crossover operation on flat lists corresponding to the association matrices of Fig. 2(a)

Fig. 2. Illustration of how to convert association matrices to their corresponding flat lists to represent chromosomes in DARCA-GA

free optimization method finding reasonably accurate approximation as well as computationally efficient because we need to do this process for every short period of time (further evaluated in Section VI). Our solution is to adapt genetic algorithms (GAs), which efficiently find near-optimal solutions particularly in a large *multi-modal* search space of x_{ij}'s [25]. As we already have a fitness function (*i.e.,* our objective function), we first define a genetic representation of the solution domain, leading to 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 done by converting the association matrix of x_{ij}'s to the corresponding *flat list* shown in Fig. 2(a). In the matrix, 1 denotes that a station is associated to the AP otherwise 0. N/A denotes that a station is out of coverage from the AP. The overall flow of DARCA-GA is as follows:

Step 1 (initialization): To improve the random population generally used in a standard GA, we additionally generate two types of baseline chromosomes, namely *nearest-best* and *round-robin*, so that the final solution is greater than or equal to these baselines. The baseline *nearest-best* selects the closest AP for each user, which is the default way of each client connecting to its best AP unless some interventions are made. The *nearest-best* can, however, result in the worst case where all the users connect to the same nearest AP. To complement this case, the *round-robin* tries to equalize the number of users associated with each AP by assigning users to APs in a round-robin fashion. This is beneficial to maintain diversity to find a globally better solution [25].

Step 2 (evaluation and selection): The initial chromosomes are evaluated by the fitness function, and then classified into one of *best-fit*, *middle-fit*, and *worst-fit*. After this classification, we eliminate the *worst-fit* and replace them with variants of the *best-fit* newly generated by crossover or mutation. This strategy, referred as *elitism*, is known as effective to guarantee the solution quality [25]. In order to keep up the quality of chromosomes in every generation, we do not discard baseline chromosomes even if they are *worst-fit*. By the elitism strategy, the *best-fit* chromosomes remain the same in the next generation. On *middle-fit* chromosomes, we perform crossover and mutation with some probability, and thereby generate new variants for the next iteration.

Step 3 (vector-wise operations): Now we have to extend crossover and mutation[2] so that they can work with our flat lists. As illustrated in Fig. 2(b), if we perform bit-wise operations like a standard GA does, each resulting flat list may end up with an incorrect association like a single user simultaneously connecting to multiple APs. In order to keep the constraint on flat lists when performing crossover or mutation, DARCA-GA suggests *vector-wise* operations, where each vector corresponds to a column of the matrix of x_{ij}'s and indicates which AP out of possible $j \in [1, n]$ to be 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 just change vectors in a manner appropriate to our problem constraints. By doing so, feasible associations are always guaranteed. We repeat *Step* 2 and *3* until no significant improvement over the fitness function.

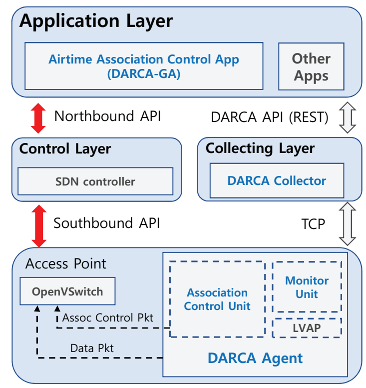


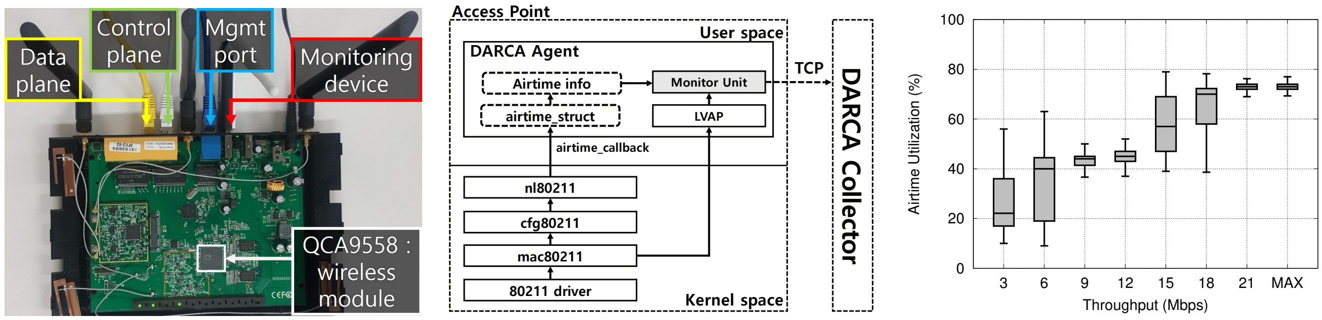
Fig. 3. DARCA system architecture

IV. DARCA Design

A. The architecture of DARCA system

As shown in Fig. 3, DARCA extends a general SDN framework with DARCA's airtime association control app in the application layer, DARCA collector in the collecting layer, and DARCA agent in each AP (*i.e.,* infrastructure layer). The application layer applies DARCA-GA algorithm by interacting with the SDN controller in the control layer and DARCA collector in the collecting layer via northbound

[2]Crossover swaps bits across chromosomes at randomly selected positions while the mutation operator changes random bits within a chromosome, and thereby both possibly leading to a better solution.



(a) Snapshot of DARCA's AP H/W setup (b) Components of DARCA Agent in an AP (c) Airtime utilization of 54 Mbps per MCS rate

Fig. 4. Implementation and measurement study *in the wild*

API and DARCA API respectively. We adopt floodlight [11] (a widely-adopted SDN controller) in the control layer. The controller provides flow management and (re-)association control (*i.e.,* extended OpenFlow [22]) for northbound applications and send the control plane messages to southbound DARCA agent and OpenVSwitch [31] (OVS) in each AP.

DARCA agent is located in the infrastructure layer to collect airtime utilization related data in its monitor unit:

- `CHANNEL_TIME`: time amount (in ms unit) that a radio spent on a channel
- `CHANNEL_TIME_BUSY`: time amount that primary channel is sensed as busy (activity or energy detection)

This monitoring unit reports wireless environmental information (e.g., channel time, station's received/transmitted packet, data rate, station transmission/reception power, SNR, and AP's transmission power) in every 100 ms[3] to the collecting layer via TCP connection. Association control unit does apply (re-)association rules with the help of Light Virtual Access Point (LVAP) introduced in [32], [33]. DARCA agent forwards association control packets (e.g., probe request, authentication request, association request) to the central controller by sending them to OVS. Followings will describe how DARCA's components interact with SDN framework for initial association and dynamic association controls.

Initial association control: When a new station is arrived at the network, it will send probe request messages to join in. This message will be forwarded from DARCA agent to the central SDN controller. Airtime association control app in application layer computes the DARCA-GA algorithm with the airtime and network information, and sends the association decision to the association control unit of target AP via SDN controller. The target AP sends back the probe response to the newly joined station after updating LVAP MAC address. The station will further process authentication and association processes with the related requests and responses, and finally join into the network.

Dynamic association control: Airtime association control app periodically performs DARCA-GA on reported airtime (every 100 ms) and network related information. The performance on the periodicity will further be investigated in Section VI. The application makes the association redistribution based on bandwidth gain in intended stations. In our setting, we avoid *ping-pong effect*), unnecessary dynamic exchanges of a station between two APs, by adopting a slack variable, which will further be investigated in Section VI. When re-association decision is made, airtime association control app will send the decision to association control unit of the target AP (move-in station) via SDN controller. The association control unit of the target AP will add the LVAP information of the move-in station and update network for the upcoming packets by sending gratuitous proxy ARP message. After that, airtime association control app notifies the previous AP to send a series of channel switch announcement (CSA) messages to the move-in station. The move-in station will be dissociated from the existing AP and switch to the channel (if necessary) of the target AP in order to receive one of the beacon messages. Airtime association control app will send REMOVE_LVAP message to the association control unit of previous AP and notify the target AP to send the move-in station a series of beacon messages. The move-in station will receive one of the beacons from the target AP and finally join into the network.

B. DARCA implementation

As shown in Fig. 4(a), we implement DARCA agent and extend LVAP component with airtime support on top of TP-LINK archer C7 v2 [38] equipped with Qualcomm Atheros QCA9558 CPU. We use an additional wireless LAN adapter, TL-WN722N v1 [37], to perform WiFi channel scanning of neighboring APs. We replace vendor Operating System with OpenWrt (Chaos Calmer v15.05.1) [30] on top of the TP-Link wireless router. We use the Click Modular Router to generate and manage the AP's LVAPs, and send the station information from the AP to the controller. DARCA supports OpenFlow (*de facto* an open-source instantiation of SDN) and decouples control and data planes.

In order to perform the airtime association control algorithm through the global view in the SDN controller, each AP must send station-related airtime information to application layer via the DARCA 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), DARCA agent uses the `netlink` protocol as follows:

Airtime utilization related data are stored in `cfg80211`, and `nl80211` transfers the stored data to user space

[3] We set 100 ms as longest handoff in our system took 80 ms as in [34]

in every 100 ms. DARCA agent receives the data from `nl80211` by `airtime_callback` and stores them in `airtime_struct`. The LVAP component collects data (SNR, RSSI, data rate, throughput, etc.) generated during packet exchanges with the associated stations through `mac80211`. The `airtime_struct` stored in DARCA agent and the data collected by the LVAP component are sent to DARCA collector via TCP. These will finally be transferred to airtime association control app in application layer. All the reported data in airtime association control app are finally processed by DARCA-GA. Whenever the decisions of association redistribution are made, (re-)association actions are performed in related OVSes via the SDN controller.

V. Testbed Experiments

To evaluate the effectiveness of DARCA system, we made two testbed settings, namely initial association control scenario shown in Fig 1(a) and dynamic association control scenario shown in Fig 1(b)-1(d).

A. Measurement study in the wild: airtime utilization

Before conducting testbed experiments, we performed a measurement study *in the wild*. This study shows the airtime utilization as a function of generated traffic loads. We set a close distance (e.g., lesser than 25 m) between an AP and a station so that MCS data rate had 54 Mbps (the maximum data rate of 802.11g). We generated constant bit rate (CBR) traffic from 3 Mbps to 21 Mbps and 54 Mbps. We performed the experiments 1000 times and measured `CHANNEL_TIME_BUSY` out of `CHANNEL_TIME` in the AP. Fig 4(c) shows a boxplot regarding airtime usage with the fixed link data rate (*i.e.,* 54 Mbps) as a function of generated bandwidth by the station. This figure shows a close correlation between residual data rate and airtime utilization. We also observed that the wireless link set data rate of 54 Mbps showed the maximum allowed bandwidth of 21 Mbps as reported in [39]. This measurement study clearly shows that if we intelligently associate stations to APs, we can squeeze more airtime utilization (thereby aggregated bandwidth) of the WiFi network. The following section will describe how the association control utilize the rooms to improve BSR via getting rid of a naïve assumption that every station always uses up all its allowed bandwidth.

B. Experimental results: initial association control

Testbed setup: As shown in Fig 1(a), station #1 (STA1) was associated with access point #1 (AP1) and had allowed bandwidth (*i.e.,* BW) 54 Mbps[4] and the station had required bandwidth (*i.e.,* RQD) of 3 Mbps, which can be a video stream of 720p with the 60 fps. STA2 was associated with AP2 and had BW of 18 Mbps with RQD of 6 Mbps. Finally, STA3 arrived at the network with RQD of 10 Mbps after 4 seconds later and newly associated with an AP based on association controls, namely PF and DARCA.

Experimental results: Fig. 5(a) shows throughput of STA1$\sim$3 and sum of the three. We ran experiments 10 times in each testbed setup and report the averaged results. As illustrated in Section II-A, STA3 arrived at the WiFi network at 4 seconds and central controller received the association request of STA3 and allowed the station to join in AP2 based on PF's decision. As shown in Fig. 5(b), airtime utilization of AP2 was already reached to around 80% (although fluctuating). Therefore, associating to AP2 cannot improve aggregated throughput in the WiFi network. Fig. 5(c) shows the throughput of the three stations and sum of them again. However, the aggregated throughput of the WiFi network is around 25% improved. The Fig. 5(d) clearly shows the reason that STA3 was associated with AP1 by central controller because DARCA detected idle airtime slots in AP1 and decided to join the station to AP1 instead of AP2 right after receiving join request message from the station. Due to the decision, AP1's airtime utilization was improved from approximately 20% to 70%. These experimental results clearly show why initial association control is crucial to improve network performance. This will further be investigated in Section VI with the scalability.

C. Experimental results: dynamic association redistribution

Testbed setup: As previously illustrated in Fig 1(b)$\sim$1(d), STA1 was associated with AP1 and has BW of 54 Mbps and RQD of 7 Mbps. STA3 was associated with AP2 and had BW of 18 Mbps and RQD of 6 Mbps. STA2 associated with AP1 and had BW of 36 Mbps and RQD of 7 Mbps. However, after 15 seconds, RQD of STA2 had been changed to 54 Mbps by its application. The following section will show how DARCA's dynamic association redistribution considering traffic load changes improve aggregated throughput.

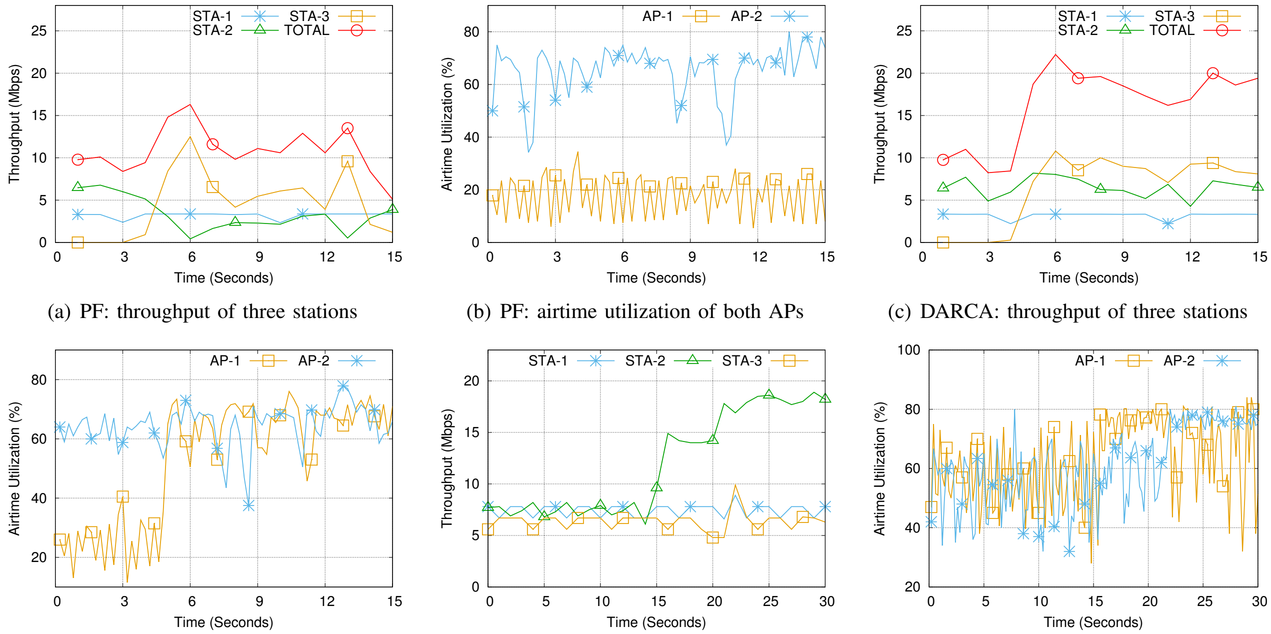
Experimental results: Fig. 5(e) shows throughput behaviors of STA1$\sim$ whose RQDs were 7 Mbps, 7 Mbps, and 6 Mbps respectively. At the time between 0$\sim$15 sec, their RQDs were fully satisfied as the provided bandwidths from the WiFi network were sufficient. However, after 15 sec, STA2's RQD had been changed from 7 Mbps to 54 Mbps. Thus, the demand exceeded the network's bandwidth provision. As illustrated in Section II-B, DARCA made an intelligent decision via dynamic (or periodic) association redistribution and thus, associations of S2 and S3 were changed to AP2 and AP1 respectively (*i.e.,* AP switching). Thus, STA2 became solely accessing the AP2 and its throughput was improved after 20 seconds. Fig. 5(f) shows airtime utilization of DARCA's dynamic association redistribution. After 20 seconds later, airtime utilization of both AP1 and AP2 became 80% (practical maximum). Note that AP1's airtime utilization was fluctuated due to contention between STA1 and STA3 while that of AP1 was stable due to exclusive access by STA2. This justifies dynamic association redistribution is crucial to improve UX of WiFi service reliability.

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 Bandwidth Satisfaction Ratio (BSR) gains:

- Strongest-Signal-First (SSF): each station connects to the closest AP based on strongest signal strength.

[4]Based on measurement study in [39], available bandwidth is usually around 21 Mbps and can vary with different vendor and settings.



(a) PF: throughput of three stations

(b) PF: airtime utilization of both APs

(c) DARCA: throughput of three stations

(d) DARCA: airtime utilization of both APs

(e) Dynamic assoc control: throughput

(f) Dynamic assoc control: airtime utilization

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

- Proportional Fairness (PF)-GA: a genetic algorithm that allocate 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.
- DARCA-GA: a genetic algorithm that periodically finds the best association distribution of the maximum BSRs in a polynomial time.
- OPTIMAL: all stations and APs are searched in a brute-force manner (*i.e.,* exponential search space).

It is noteworthy that every station associated with an enterprise or academic network is connected to an AP at each time while its association membership can be changed in semi-online manner (*i.e.,* periodic redistribution). We further introduce a slack parameter α (usually from 1% to 16% of throughput gain in our settings) to avoid *ping-pong effect* causing unnecessary dynamic exchanges of a station between two APs. We present three deployment scenarios, namely shopping mall, conference, and office scenarios.

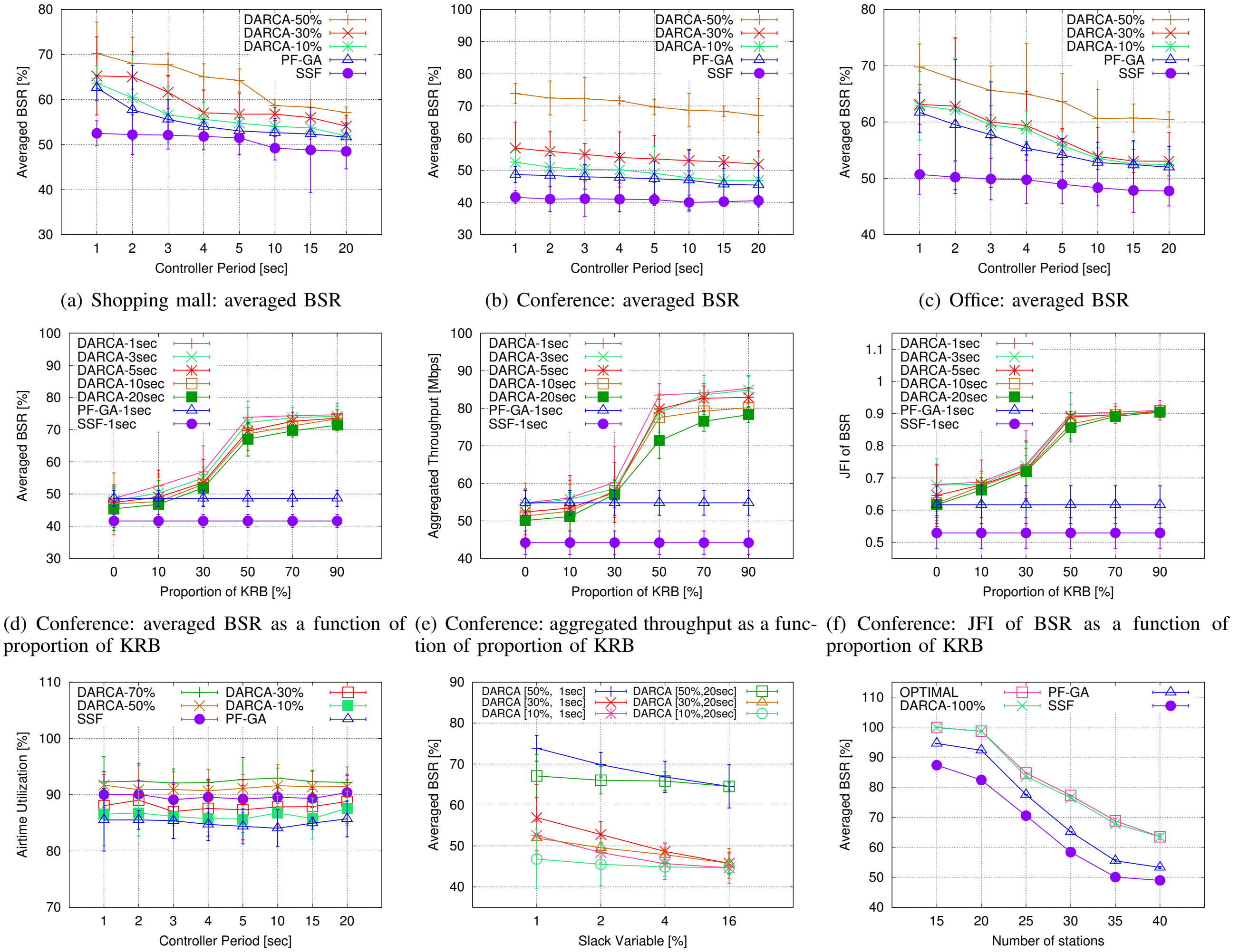
A. Simulation Setup

We conduct extensive simulations and compare the above four algorithms on a discrete-event simulator NS-3 [29]. In our simulation settings, we deployed 802.11g 9 APs on a 3×3 grid, where the distance between adjacent APs is set to 100 meters with frequency planning (*i.e.,* non-overlapping channels of 2.4 GHz) as in [6], [7], [20], [40] in an area of 300m$\times$300m. We report the simulation results of three scenarios, namely shopping mall (high mobility with sparse station density), conference (low mobility with dense station density), and office (moderate mobility with sparse station density). In a shopping mall deployment, we uniformly deployed 90 stations in the target area where 10% of nodes are static while 90% of nodes are mobile. In a conference deployment, we deployed 90 nodes where 50% of the nodes are fixed nodes in the center 50 m$\times$ 50 m grid near the center of the 9 AP grid network and remaining 50% of the nodes are mobile. In an office deployment, we placed 90 stations uniformly where stationary nodes are 30% and mobile nodes are 70%.

For the user mobility, we assumed users are moving around in a corridor that has fixed boundary. We randomly set waypoints in the target areas and set pre-defined maximum speed of 0.4, 0.8, and 1.6 m/s to mobile users. Note that 1.6 m/s is equivalent to 5.76 km/h which is about 1.5 times faster than regular walk. To add more realistic factor, we periodically change the directions of walking with an offset of [-10, 10] degrees. We assume all users have the same priority. Unless otherwise specified, we used log-distance path loss model with reference distance 1 meter, reference loss 46.678 dBm and loss exponent 3. Transmit power was set to be 20 dBm. Required bandwidth per user was randomly selected between 15Kbps$\sim$3 Mbps. We varied the number of mobile stations to see the algorithms' behaviors in a high, moderate, and low mobility scenarios. We reported an average value of 50 runs with 95% confidence interval to mitigate any dependence on specific association pattern and to understand general algorithms' behaviors.

B. Simulation Results

Averaged BSR: Fig. 6(a) shows averaged BSR of three algorithms, namely SSF, PF-GA, and DARCA-GA in a shopping mall scenario as a function of controller periods (*i.e.,* periods of association re-distribution). For the DARCA-GA, we plot three flavors, namely DARCA with 10%, 30%, and 50% of stations with known required bandwidth (hereafter



(a) Shopping mall: averaged BSR

(b) Conference: averaged BSR

(c) Office: averaged BSR

(d) Conference: averaged BSR as a function of proportion of KRB

(e) Conference: aggregated throughput as a function of proportion of KRB

(f) Conference: JFI of BSR as a function of proportion of KRB

(g) Conference: airtime utilization

(h) Conference: slack variable α

(i) Optimality of DARCA-GA, PF-GA, and SSF

Fig. 6. Simulation results: (a)-(c) Averaged BSR in shopping mall, conference, and office scenarios; (d)-(f) Averaged BSR, aggregated throughput, JFI of BSR in conference scenario; (g) Airtime utilization with controller periods; (h) Averaged BSR with slack variable α; (i) optimality of three algorithms

KRB). We vary periods of controller in order to see the algorithms' performance in high, moderate, and low mobility scenarios.

As shown in Fig. 6(a), DARCA with three flavors outperform other two algorithms in a shopping mall (high mobility) scenario. An overall trend is that the performance of all algorithms decrease as the controller periods increase. The second overall trend is that the performance increases as the proportions of KRB increases. Fig. 6(b) and Fig. 6(c) shows the similar trends. Comparing Fig. 6(a) and Fig. 6(c), it is shown that high mobility (*i.e.,* shopping mall) scenario has less effect on the average BSR performance than low mobility (*i.e.,* office) scenario. Interestingly, averaged performance of SSF and PF-GA is degraded more in conference (dense deployment, Fig. 6(b)) scenario than in shopping mall and office scenarios (sparse deployments, Fig. 6(a) and Fig. 6(c)). The performance gain from proportions of KRB becomes larger than sparse networks, namely shopping mall and office scenarios. Fig. 6(d) shows the averaged BSR performance as a function of proportions of KRB. DARCA with five different controller periods (1, 3, 5, 10, and 20 seconds) outperform PF-GA and SSF and share the similar trends. It is interesting that averaged BSRs of all five flavors increase when proportions of KRB is lesser than 50% while they are saturated after 50% of proportions of KRB.

Aggregated Throughput: Fig. 6(e) shows the results of aggregated throughput in conference scenario. We omitted the results of shopping mall and office scenarios for the sake of brevity as they share the similar behaviors. Again, DARCA with five different controller periods (1, 3, 5, 10, and 20 seconds) outperform PF-GA and SSF and the aggregated throughput is increased as the proportions of KRB are increased. DARCA with controller periods of 3∼20 seconds perform poor than PF-GA. It is because PF-GA's controller period is set as 1 second so that it is has more chances to enforce its objective function. It is noteworthy that all five flavors of DARCA converge their aggregated throughput after the proportions of KRB are more than 50%.

Jain's Fairness Index: Fig. 6(f) shows Jain's Fairness Index (JFI) [14] of the three algorithms in conference scenario. For JFIs, we adopted BSR instead of per-station throughput because each station randomly selects UDP traffic from 15Kbps to 3 Mbps and needed a normalized standard per station to compare with. As shown in 6(f), DARCA with five different controller periods (1, 3, 5, 10, and 20 seconds) show the similar behaviors that JFIs of BSR increase as the proportions of KRB increase. DARCA with controller period of 1 second shows the best JFI. However, PF-GA shows unfair and remains the same due to its ignorance of KRB.

Airtime utilization: Fig. 6(g) shows the airtime utilization of three algorithms. DARCA has four flavors according to proportions of KRB, namely 10%, 30%, 50%, and 70%. Three algorithms (including four flavors of DARCA based on proportions of KRB) show the flat airtime utilization regardless of controller periods. DARCA with 50% of KRB already reach to the maximum airtime utilization as DARCA with 70% of KRB. It is interesting that SSF and PF-GA utilize more than 80% of airtime in the network but the difference of airtime utilization between these algorithms and four flavors of DARCAs are about 10% but the averaged BSR, aggregated throughput, and JFI are distinct. This justify that DARCA algorithm is cost-effective (airtime versus performance).

Slack variable α: Fig. 6(h) shows the DARCA's averaged BSR performance as the function of slack variable α. We introduced α as a slack variable to avoid *ping-pong effect* causing unnecessary dynamic exchanges of a station between two APs by setting a bar of re-association (*i.e.,* handoff threshold). The α 1% means the station will change of its association to another AP if the BSR gain after the switch to the new AP is more than 1%. We vary the proportions of KRB and controller periods to observe DARCA behaviors with the settings. All DARCA flavors shows the degraded performance as the α value increases. This means that 1% of setting is enough for the threshold to avoid *ping-pong effect*. We can also confirm that DARCA with higher proportions of KRB and shorter controller periods show better averaged BSR performance.

Optimality: Fig. 6(i) shows the optimality of the three algorithms. PF-GA and DARCA-GA are meta-heuristic algorithms which require polynomial time to find a solution while OPTIMAL spends exponential computation time (prohibitive computation cost). The y-axis represents averaged BSR and x-axis represent the number of deployed stations (*i.e.,* user density). We report the results of small-scale setup (*i.e.,* 3 APs and 40 stations at most) due to the exponential search space of OPTIMAL. SSF and PF-GA show the similar patterns as in Fig. 6(a) and 6(d). Surprisingly, meta-heuristics of DARCA-GA with 100% of KRB performs very close to optimal while achieving large saving in computation time. A careful reader might notice that 100% of KRB is not realistic considering streaming (e.g., video and audio) and gaming traffic composes less than 70% of Internet traffic whose maximum required bandwidth is easily estimated [1]. We set the unrealistic value (100%) to show how DARCA-GA heuristic algorithm gets close to the optimal.

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 access points (APs); thereby this significantly degrades UX of WiFi service reliability [3], [5], [17]. The conventional AP selection mechanism (*i.e.,* association) from user side relies on RSSI as the association criteria; thereby users with uneven load distributions associate with APs in their vicinity and more importantly user mobility will also cause the *sticky client problem*—a client stick to an AP in its vicinity (*i.e.,* maximum data rate) but keep staying in the AP after moving apart. Recent studies on operational WLANs have shown that the traffic imbalance can be alleviated by intelligently associating user to APs—termed association control, reported as NP-hard [6], [7], [20]. Benjerano *et al.* proposed [6], [7] approximate algorithms that achieves max-min fair bandwidth allocation. However, max-min throughput fairness can significantly reduce aggregated throughput in multi-rate 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.* defined a proportional fairness (or time-based fair scheduling) problem, which is also NP-hard [20] and demonstrated the algorithms improve the aggregated throughput 2.3 times more than aggregate throughput of max-min fair allocation.

In existing proposals, however, it is practically impossible to provide fair and efficient bandwidth allocations through association controls in WiFi networks without disrupting ongoing sessions (*i.e.,* re-association); finally loses the chance of service reliability which is closely correlated to UX. However, this is not the case in DARCA as DARCA can seamlessly handoff clients without triggering with the help of LVAP technique.

SDN-enabled WLAN frameworks: In conventional WLANs, clients solely decide on which AP to associate with. Murty *et al.* proposed DenseAP [26] which uses the idea that better AP association decisions can be made by a global view of WLAN network instead of local view. In DenseAP, modifications only on AP side were made to aggregate workload information and provide enhanced association controls by global view. However, the system has certain performance limitations because it cannot work around the delay involved in handoff (*i.e.,* re-association delay). To cope with the delay, Schulz-Zander *et al.* [32], [33] proposed Odin. Odin builds on a light virtual access point (LVAP) abstraction for addressing the IEEE 802.11 protocol stack complexity. LVAPs virtualize association state and separate them from the physical APs as well as they allow associated clients to handoff without triggering the whole re-association mechanism. Unfortunately, existing proposals are limited to introduction to seamless handoff demonstration via LVAP and additional configuration extensibility to WLAN features (*i.e.,* wireless channel selection, interference mitigation, and mobility management), which have not fully solved the *sticky client problem* yet. However, DARCA set UX-related standard termed as BSR and put the standard into a algorithm called DARCA-GA that periodically finds the sub-optimal association distribution in polynomial time.

VIII. CONCLUSION

In this paper, we presented an SDN-enabled WiFi system called DARCA with a genetic algorithm called as DARCA-GA to fully resolve *sticky client problem* via periodic association re-distribution in a central SDN controller. Our real-life experiments and extensive simulation results validate that DARCA can achieves the better trade-off between performance and fairness, and improves user experience by improving BSR. Although we provided a number of important insights from the testbed experiments and simulation studies, there are unexplored parameters that will be addressed in future work. Due to the prototype of DARCA running on 802.11g, we stick our simulation study of DARCA-GA to the same setting. We envision that 802.11n or higher (e.g., 802.11ac) will require more dynamic association re-distribution of the network. We will further investigate the performance gains and tune-up parameters of DARCA-GA in those settings, which remain as our future work.

REFERENCES

[1] 2018 Global Internet Phenomena Report. https://www.sandvine.com/hubfs/downloads/phenomena/2018-phenomena-report.pdf.

[2] Aruba WLAN Controller. https://www.arubanetworks.com/pdf/technology/whitepapers/wp_ARM_EnterpriseWLAN.pdf.

[3] A. Balachandran, G. M. Voelker, P. Bahl, and P. V. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In *Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 195–205, 2002.

[4] H. Balakrishnan, M. Stemm, S. Seshan, and R. H. Katz. Analyzing Stability in Wide-area Network Performance. In *Proceedings of the 1997 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 2–12, 1997.

[5] M. Balazinska and P. Castro. Characterizing Mobility and Network Usage in a Corporate Wireless Local-area Network. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pages 303–316, 2003.

[6] Y. Bejerano, S. Han, and L. Li. Fairness and Load Balancing in Wireless LANs Using Association Control. *IEEE/ACM Transactions on Networking*, 15(3):560–573, Jun 2007.

[7] Y. Bejerano, S.-J. Han, and L. E. Li. Fairness and Load Balancing in Wireless LANs Using Association Control. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 315–329, 2004.

[8] A. Bhartia, B. Chen, F. Wang, D. Pallas, R. Musaloiu-E, T. T.-T. Lai, and H. Ma. Measurement-based, practical techniques to improve 802.11ac performance. In *Proceedings of the 2017 Internet Measurement Conference*, pages 205–219, 2017.

[9] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-E, A. Bhartia, and D. Aguayo. Large-scale Measurements of Wireless Network Behavior. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 153–165, 2015.

[10] Y. Daldoul, D. Meddour, and A. Ksentini. IEEE 802.11ac: Effect of channel bonding on spectrum utilization in dense environments. In *2017 IEEE International Conference on Communications*, pages 1–6, 2017.

[11] Floodlight. http://www.projectfloodlight.org/.

[12] P. Gawlowicz, A. Zubow, M. Chwalisz, and A. Wolisz. UniFlex: A framework for simplifying wireless network control. In *2017 IEEE International Conference on Communications*, pages 1–7, 2017.

[13] How much bandwidth is required to stream Live Video? https://www.quora.com/How-much-bandwidth-is-required-to-stream-Live-Video.

[14] R. Jain, D. Chiu, and W. Hawe. A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. *CoRR*, cs.NI/9809099, Jan 1998.

[15] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat. B4: Experience with a Globally-deployed Software Defined Wan. In *Proceedings of the ACM SIGCOMM 2013 Conference*, pages 3–14, 2013.

[16] F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, Mar 1998.

[17] D. Kotz and K. Essien. Analysis of a Campus-Wide Wireless Network. *Wireless Networks*, 11(1):115–133, Jan 2005.

[18] L. Kriara, E. C. Molero, and T. R. Gross. Evaluating 802.11ac Features in Indoor WLAN: An Empirical Study of Performance and Fairness. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*, pages 17–24, 2016.

[19] J. Lakkakorpi, A. Heiner, and J. Ruutu. Measurement and characterization of Internet gaming traffic. *Research Seminar on Networking, Helsinki University of Technology, Networking Laboratory*, pages 1–12, Jan 2002.

[20] L. Li, M. Pal, and Y. R. Yang. Proportional Fairness in Multi-Rate Wireless LANs. In *The 27th Conference on Computer Communications*, pages 1004–1012, 2008.

[21] P. N. Marko M Mkel. *Nonsmooth optimization: analysis and algorithms with applications to optimal control*. World Scientific, 1992.

[22] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, Mar 2008.

[23] Meraki. http://www.meraki.com/products/wireless/enterprise-cloud-management.

[24] M. Mirza, J. Sommers, P. Barford, and X. Zhu. A Machine Learning Approach to TCP Throughput Prediction. In *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 97–108, 2007.

[25] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1996.

[26] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill. Designing High Performance Enterprise Wi-Fi Networks. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 73–88, 2008.

[27] R. U. Nabar, H. Bolcskei, and F. W. Kneubuhler. Fading relay channels: performance limits and space-time signal design. *IEEE Journal on Selected Areas in Communications*, 22(6):1099–1109, Aug 2004.

[28] Netflix Connection Speed Recommendations. https://help.netflix.com/en/node/306.

[29] NS3 Simulator. https://www.nsnam.org.

[30] OpenWrt. https://openwrt.org/releases/15.05/start.

[31] B. Pfaff, J. Pettit, T. Koponen, E. J. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar, K. Amidon, and M. Casado. The Design and Implementation of Open vSwitch. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, pages 117–130, 2015.

[32] J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann. OpenSDWN: Programmatic Control over Home and Enterprise WiFi. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, pages 16:1–16:12, 2015.

[33] J. Schulz-Zander, L. Suresh, N. Sarrar, A. Feldmann, T. Hühn, and R. Merz. Programmatic Orchestration of WiFi Networks. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, pages 347–358, 2014.

[34] L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, and J. Almodovar. Building an SDN Enterprise WLAN Based on Virtual APs. *IEEE Communications Letters*, 21(2):374–377, Feb 2017.

[35] K. Sui, M. Zhou, D. Liu, M. Ma, D. Pei, Y. Zhao, Z. Li, and T. Moscibroda. Characterizing and Improving WiFi Latency in Large-Scale Operational Networks. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 347–360, 2016.

[36] G. Tan and J. Guttag. Time-based Fairness Improves Performance in Multi-rate WLANs. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, pages 23–23, 2004.

[37] TPLINK TL-WN722N V1. https://www.tp-link.com/us/download/TL-WN722N_V1.html.

[38] TPLINK ARCHER C7 V2. https://www.tp-link.com/us/download/Archer-C7_V2.html.

[39] 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 *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network*, pages 220–225, 2005.

[40] W. Wong, A. Thakur, and S. H. G. Chan. An approximation algorithm for AP association under user migration cost constraint. In *The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, 2016.

[41] Y. Yiakoumis, M. Bansal, A. Covington, J. van Reijendam, S. Katti, and N. McKeown. BeHop: A Testbed for Dense WiFi Networks. In *Proceedings of the 9th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, pages 1–8, 2014.

[42] Y. Zeng, I. Pefkianakis, K.-H. Kim, and P. Mohapatra. MU-MIMO-Aware AP Selection for 802.11ac Networks. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 19:1–19:10, 2017.

[43] A. Zubow, S. Zehl, and A. Wolisz. BIGAP Seamless handover in high performance enterprise IEEE 802.11 networks. In *2016 IEEE/IFIP Network Operations and Management Symposium*, pages 445–453, 2016.