Achieving Efficiency and Fairness for Association Control in Vehicular Networks

***Abstract*—Deploying city-wide 802.11 access points has made possible internet access in a vehicle, nevertheless it is challenging to maintain client performance at vehicular speed especially when multiple mobile users exist. This paper considers the association control problem for vehicular networks in *drive-thru Internet scenarios*. In particular, we aims to improve the overall throughput and fairness for all users. We design efficient algorithms to achieve the objectives through several techniques including approximation. Our simulation results demonstrate that our algorithms can achieve significantly better performance than conventional approaches.**

I. Introduction

Advanced technologies in communication have made ubiquitous network connectivity - anywhere, anytime - a reality for mobile users. Wi-Fi technology is one of them, which provides users easy access to the Internet through nearby WLAN access points. WLAN-based access points (AP) can provide cost-effective wireless Internet access with a shared gross data rate that ranges from 10 to 50 Mbit/s and can be scalable to hundreds of concurrently-active users. Not limited to corporation, office, or home use for comparably stationary user, the access points, in the *Drive-thru Internet*[1][2], are recently proposed to be deployed along the roads, thus providing continuous network connectivity to mobile users in vehicles. However, for outdoor access of mobile users at high vehicular speeds, due to the dynamically changing network structure of AP-user pairing and contentions among mobile users, it is still a big challenge to maintain a good client performance.

In order to achieve reasonable efficiency among multiple vehicular users for the above *Drive-thru Internet* scenario, several problems should be considered, e.g., rate adaptation and association control. Association control defines, while multiple users are driving along the road, how to intelligently associate these mobile users to specific APs and when to appropriately conduct handoffs of APs for users to improve the overall system performance. Compared with rate adaptation, association control considers the entire network from a macro-level perspective, which shows how to optimize system performance from a higher level viewpoint.

We notice that, albeit some recent work on association control for static networks, there is little work on how to manage AP association in this type of "Vehicular Networks". Compared with a static network, this type of network is composed of continuously moving clients that may constantly try to associate with the nearby APs. Algorithms that solve for the association control problem in a static network are not suitable because (1) a vehicular network is a much larger network composed of thousands of cars and APs compared with a rather small indoor network and (2) the existing algorithms are unable to accommodate the dynamic creation and removal of the possible communication links in the network graph. We believe that a thorough theoretical study on this problem is highly necessary for the future deployment of this type of networks. Some pitfalls can be avoided in real deployment if we have a better understanding about it first.

This paper aims to define a theoretical framework to analyze the performance of a vehicular network in the *Drive-thru Internet* scenario, in particular to investigate the association control scheme. Considering both the long-term efficiency and fairness metrics, we propose optimized schemes to associate mobile users with APs, and approximation algorithms to reduce computation complexity of calculating optimal solutions. Although there is some previous work related to this topic [1][2][3], they all focus on the measurement study on vehicular internet access. To the best of our knowledge, this is the first theoretical work that investigates the optimization problem for association control over vehicular users in Wi-Fi networks. The contributions are summarized as follows:

1) We propose a theoretical framework for association control over vehicular networks. For the efficiency metric, the problem is transformed into an optimization problem for each snapshot over the long-term service duration, and formulated as an integral linear program. For the fairness metric, we first formulate the problem as a convex program in the offline setting, and further propose a dynamic weight based online algorithm to achieve proportional fairness.

2) When the involved number of mobile users and APs along the road is rather large, to reduce the computation complexity, we propose an approximation algorithm to break the large contention group into smaller sub-groups, achieving a tradeoff between accuracy and computation complexity.

The rest of the paper is organized as follows. We present related work in Section II. We define the performance metrics and introduce our model and assumptions in Section III. We illustrate our overall optimization and snapshot solutions in Section IV, respectively for efficiency and fairness. We introduce our group-based methodology to simplify association control in Section V. We show simulation results in Section VI, and we conclude the paper in Section VII.

II. Related Work

Association control and scheduling solutions for Wireless LANs have been intensely studied [4] [5] [6], mainly targeting the efficiency and fairness metrics. Tassiulas et al. consider max-min fair allocation of bandwidth in wireless ad-hoc networks [4], and propose a fair scheduling system which assigns dynamic weights to the flows. Bejerano et al. present an efficient solution to determine the user-AP association for max-min fair bandwidth allocation [5], by leveraging the strong correlation between fairness and load balancing. To balance aggregate throughput while serving users in a fair manner, Li et al. consider proportional fairness over wireless LANs [6]. They propose two approximation algorithms for periodical offline optimization.

Internet access with roadside WiFi access points for vehicular users under vehicular speeds has been studied in recent research works [1][2][3][7][8][9][10][11][12][13]. Bychkovsky et al. study the case for vehicular clients to connect to open-access residential wireless 802.11 access points in Boston [1]. Giannoulis et al. address the problem of maintaining client performance at vehicular speeds within city-wide multi-hop 802.11 networks [3]. Ott and Kutscher report on measurements for the use of IEEE 802.11 networks in the *drive-thru Internet scenario* [2]. They measure transmission characteristics in vehicles moving at different speeds, and provide analysis on the expected performance. Mahajan et al. deploy a modest-size testbed to analyze the fundamental characteristics of WiFi-based connectivity between base stations and vehicles in urban settings [9]. Hadaller et al. give a central message that wireless conditions in the vicinity of a roadside access point are predictable and that by exploiting this information, vehicular opportunistic access can be greatly improved [10]. Kim et al. present novel association control algorithms that minimize the frequency of handoffs occurred to mobile devices [11]. In the case of a single user, the optimization of the total throughput with handoff time taken into account is studied in [12].

III. Performance Models and Metrics

A. Models and Assumptions

In the *Drive-thru Internet* scenario, vehicular users are driving through a region covered with multiple roads, and APs are deployed along the roads in a nonuniform approach. Each AP has a limited coverage range and it can only serve users that reside in its coverage area. Conventionally each user on the roads may have one or more candidate APs to associate with at any time, and each time the user can only associate with exactly one AP. Furthermore, contentions for transmission may exist among users if they associate with the same AP. If a large number of users associate with the same AP, their allocated bandwidths will be greatly reduced. We assume that different users have various velocities (including speeds and directions) which may vary over the time. Thus while users are driving along the roads, at different time instants and positions, they may be contending with different users for bandwidth from different APs. Each user associates with the first AP after first entering the Wi-Fi deployment area, then goes through a series of hand-offs among different APs while driving along the roads, and disconnects at the last associated AP before leaving the Wi-Fi deployment area.

We denote the set of APs as A indexed by $1, ..., m$ and denote the set of users as U indexed by $1, ..., n$. We consider association control over the time interval $[0, T]$. For example, 0 and T may respectively denote $0:00$ and $24:00$ time points of every day. For each user-AP pair (j, i), we assume that the effective bit rate $r_{i,j}(t)$ of the link between j and i at time t is known. We use $b_j(t)$ to denote the bandwidth allocated to user j at time t. Both bit rate and bandwidth can be measured in bits per second (bit/s). For bandwidth allocation inside each AP, we use time-based fairness for scheduling. Once an AP is associated to some users, each user is assigned an equal-sized time slot regardless its effective bit rate, and is supposed to use all the allocated bandwidth. Thus if n' users are associated with AP i at time t, then the bandwidth allocated to user j is $b_j(t) = r_{i,j}(t)/n'$.

For the effective bit rate setting in the *Drive-thru Internet* scenario, we adopt the model proposed in [2]. Fig. 1 depicts three different connectivity phases with respect to effective bit rate and relative distance. The entry phase and exit phase provide very weak connectivity, only the production phase provides a window of useful connectivity. As the connection is built between a user and an AP, it will maintain a constant bit rate in the production phase, which mainly depends on the AP's signal strength and the user's driving speed. Conventionally the faster the user's speed is, the lower bit rate the user can achieve. The bit rate can basically keep fixed while the user's speed does not change too much. Therefore for each specified user we can approximately model the bit rates of APs as square waves. As Fig. 2 shows, we allow nonuniform AP deployments along any user's driving trajectory which include effective ranges, neighbor distances and effective bit rates. We then divide these regions into non-overlapped *Equivalence Classes* as $Eq_1, Eq_2, ..., Eq_n$. Each Eq_i denotes a section of the roads, and within each section the candidate AP set and corresponding effective bit rates will keep fixed for the specified user.

B. Performance Metrics

We consider two important performance metrics in this study: efficiency and fairness. Efficiency is measured with the overall throughput received by all users and fairness is to regulate the association control so that all users will have a fair distribution of bandwidth as much as possible.

For efficiency, we aim to maximize the overall throughput for all vehicular users. The throughput for any user is the average message delivery rate during the user's service period and it is usually measured in bits per second (bit/s). Hence for any user j, given the service duration $[t_j, t_j + T_j]$ and the allocated bandwidth $b_j(t)$ at time $t \in [t_j, t_j + T_j]$, we can express the throughput B_j for user j as $B_j = \frac{1}{T_j} \int_{t_j}^{t_j+T_j} b_j(t) d(t)$. Consider the overall time interval $[0, T]$, during intervals $[0, t_j]$ and $[t_j + T_j, T]$, we actually have $b_j(t) = 0$, thus we have an equivalent uniform notion as $B_j = \frac{1}{T_j} \int_0^T b_j(t) d(t)$.

Association control without considering fairness may lead to the starvation of users with poor signal strength. To consider fairness, two metrics are used frequently in literature: max-min fairness [5] and proportional fairness [6][14]. We use proportional fairness in this paper because it can achieve a better trade-off between efficiency and extreme fairness. Suppose the throughput allocation for all n users can be denoted as a vector $\vec{B} = \langle B_1, B_2, ..., B_n \rangle$. By definition, an allocation $\vec{B}$ is "*proportionally fair*" if and only if, for any other feasible allocation $\vec{B}'$, $\sum_{j=1}^{|\vec{B}|} \frac{B_j^{'} - B_j}{B_j} \leq 0$. In other words, any change in the allocation must have a negative average change. It has been proved that the unique proportionally fair allocation can be obtained by maximizing $J(\vec{B}) = \sum_j \ln(B_j)$ over the set of feasible allocations [15].

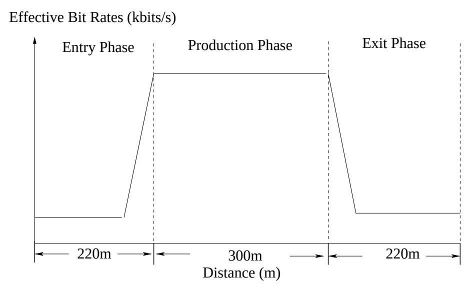


Fig. 1. Three connectivity phases

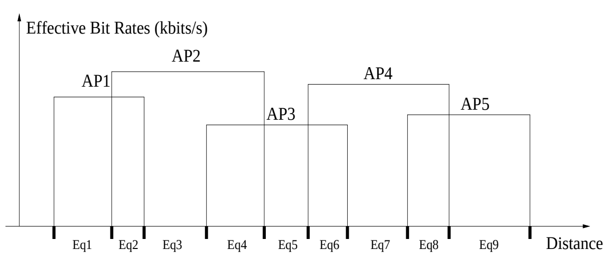


Fig. 2. Nonuniform AP deployments along user's driving trajectory

Since all APs are deployed by the same organization, a centralized control scheme is possible as proposed in [16]. Therefore based on the above models and assumptions, we aim to build a centralized association control system and our goal is to continuously construct optimized assignments of APs to users as they are driving along the roads, respectively taking the efficiency and fairness metrics into consideration. We consider both offline and online settings of the optimization problem. In the offline setting, we assume that we know the mobility patterns and trajectories of vehicular users in advance, in other words, we are given the candidate AP set $A_j(t)$ for each user j at each time $t \in [0, T]$ as part of the problem input. In the online setting, each $A_j(t)$ is revealed only at time t, at which time instant, we have to instantaneously select an AP from $A_j(t)$ to associate for each user j, without any knowledge of the future sets $A_j(t')$ for $t' \in [t, T]$.

IV. Overall Optimization and Snapshot Solution

For the efficiency metric, with a set of vehicular users U on the roads, the objective is to maximize $\sum_{j \in U} w_j B_j$, which can be further denoted as

$$\sum_{j \in U} \frac{w_j}{T_j} \int_0^T b_j(t) d(t). \tag{1}$$

Here w_j denotes priority for different users, and it is a fixed value for specified user j. Similarly if we choose proportional fairness as the metric, the optimization objective is to maximize $\sum_{j \in U} w_j \ln B_j$, which can be further denoted as

$$\sum_{j \in U} w_j \ln(\frac{1}{T_j} \int_0^T b_j(t) d(t)). \tag{2}$$

The above two objectives are optimization metrics over the duration of service period for all users. As we aim to continuously construct optimized assignments of users to APs within this duration, we use the term "snapshot" to denote the small time interval within which we have to make a decision about AP association for all users. Thus it is necessary for us to find solutions for each snapshot to achieve the overall optimal performance.

A. Snapshot Optimization for Efficiency

We first prove a theorem.

Theorem 1: For the efficiency metric, it is sufficient to maximize $\sum_{j \in U} \frac{w_j}{T_j} b_j(t)$ for each snapshot t to achieve the long-term optimization goal.

Proof: For the efficiency metric, we are to maximize $f_O = \sum_{j \in U} \frac{w_j}{T_j} \int_0^T b_j(t) d(t)$ according to (1). As w_j and T_j are constants with time for each user j, we have

$$f_O = \sum_{j \in U} \int_0^T \frac{w_j}{T_j} b_j(t) d(t) = \int_0^T \sum_{j \in U} \frac{w_j}{T_j} b_j(t) d(t).$$

As T is a constant, for each snapshot t we only have to maximize $\sum_{j \in U} \frac{w_j}{T_j} b_j(t)$ for optimization. ■

Theorem 1 essentially tells us that we can optimize for efficiency metric in each snapshot to achieve overall performance. In the offline setting, we already know T_j in the objective function. In the online setting, we have to estimate T_j based on the user's current speed $v_j(t)$. Suppose user j gives the driving trajectory to the centralized server through devises like GPS. Knowing the overall distance S_j and the distance $s_j(t)$ that user j has traveled at time t, we can continuously estimate T_j for user j at snapshot t using $T_j(t) = \frac{S_j - s_j(t)}{v_j(t)} + t$.

To describe the constraints in this problem formulation for each snapshot t, we formulate the association problem into a linear program (LP) as proposed in [6]. We use a fractional variable $p_{i,j}(t)$ to denote the fraction of time that AP i devotes to user j. For each AP i and user j, if j is associated with i, then $p_{i,j}(t)$ is a fraction between 0 and 1; if user j is not associated with i, then the fraction is 0. Since each user j is assigned to only one AP for the integral solution, there is exactly one non-zero $p_{i,j}(t)$ for each $i \in A$. We can first relax this constraint and assume that one user can associate with multiple APs for the fractional solution. Then the bandwidth $b_j(t)$ allocated to each user j can be depicted as $b_j(t) =$

$\sum_{i\in A} r_{i,j}(t)p_{i,j}(t)$. Thus we can obtain a fractional solution from the following linear program formulation:

$$maximize \sum_{j\in U} \frac{w_j}{T_j} b_j(t) \tag{3}$$

subject to

$$\forall j \in U \quad b_j(t) = \sum_{i\in A} r_{i,j}(t) \cdot p_{i,j}(t) \tag{4}$$

$$\forall i \in A \quad \sum_{j\in U} p_{i,j}(t) \le 1 \tag{5}$$

$$\forall j \in U \quad \sum_{i\in A} p_{i,j}(t) \le 1 \tag{6}$$

$$\forall i \in A, j \in U \quad 0 \le p_{i,j}(t) \le 1 \tag{7}$$

$$\forall j \in U \quad b_j(t) \ge C \tag{8}$$

The first constraint defines $b_j(t)$, the bandwidth allocated to user j at time point t. The second constraint means that the overall allocated time fraction of each AP i to all users cannot be more than 1. The third constraint states that the overall allocated time fraction of each user j that communicates with all APs cannot be more than 1. The fourth constraint shows that the time fraction is between 0 and 1. To ensure that every user is able to maintain connectivity to the internet within the service duration, the fifth constraint guarantees that every user has a minimum bandwidth of C at any time t, where C is a constant value for the lower bound. For the *pure efficiency* goal we set $C = 0$ by default.

For completeness, we describe briefly in the following how to find the integral solution based on the fractional solution. After we obtain $p_{i,j}(t)$ for each user-AP pair, we can further calculate the fractional assignment $x_{i,j}(t) = \frac{r_{i,j}(t)\cdot p_{i,j}(t)}{b_j(t)}$, which reflects the fraction of user j's total bandwidth that it expects to get from AP i. Apparently $0 \le x_{i,j}(t) \le 1$. We can view the assignment as a bipartite graph. Then the final integral solution is a set of binary variables $\hat{x}_{i,j}(t)$ for all user-AP pairs, where $\hat{x}_{i,j}(t)$ is equal to 1 if user j is associated to AP i and 0 otherwise. We use the rounding algorithm proposed by Shmoy and Tardos [17] to calculate the integral solution $\hat{x}_{i,j}(t)$. Readers can refer to [6] for detailed description.

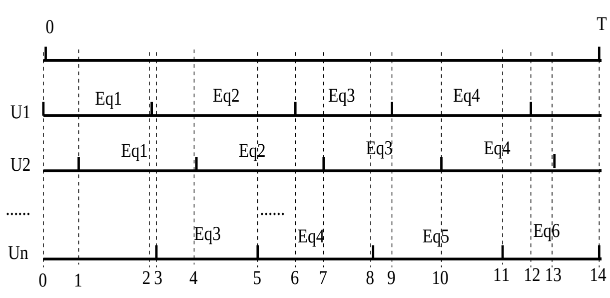


Fig. 3. Time line for users to drive through the *Equivalence Classes*

Since we have obtained the optimized strategy for association control over each snapshot, we need to consider the handoff strategy for efficiency. Continuously computing the optimized association solution for each snapshot is definitely not an appropriate solution, as it incurs too much computing and communication cost. Without loss of generality, we assume that the boundaries of a AP's effective range will not coincide with the others. Fig. 3 shows an example of the vehicular scenario, where a set of users $U_1, U_2, ..., U_n$ are trying to drive through *Equivalence Classes* over the time span $[0, T]$. Thus we can divide the overall time span $[0, T]$ into smaller time intervals according to the boundaries of *Equivalence Classes* over the time span. We denote these time intervals as $[t_0^{'}, t_1^{'}], [t_1^{'}, t_2^{'}], ..., [t_{L-1}^{'}, t_L^{'}]$. We rely on the following theorem to devise an efficient handoff strategy for efficiency metric.

Theorem 2: For optimal association control to maximize the efficiency metric, handoffs to new association solutions for users only happen when at least one user is crossing the boundaries of *Equivalence Classes*. At each boundary the user will meet with one of the following cases: (1) new candidate AP is detected; (2) original optimal AP is lost and (3) original candidate AP is lost. For cases (1) and (2), new association control is necessary. For case (3), new association control is not needed, so the original optimized solution holds.

Proof: According to objective function $\sum_{j\in U} \frac{w_j}{T_j} b_j(t)$, the weight w_j and service duration T_j for each user j are fixed all the time. When all users are within the boundaries of *Equivalence Classes*, the effective bit rates $r_{i,j}(t)$ are never changed, which indicates that all parameters for the optimization problem are not changed, thus the optimal solution holds. For case (1), as a new candidate AP is detected, one effective bit rate $r_{i,j}(t)$ will change from 0 to a positive value, so the computation of a new optimal association solution is necessary. For case (2), since the original optimal AP is lost, a new optimal association solution is definitely necessary. For case (3) we prove by contradiction that new association control is not necessary. We denote the former equivalence class as Eq_i and the new equivalence class as Eq_j. Assume in Eq_j some user will switch to a different AP for a new optimal solution. Thus we can use new bandwidth allocation $b_j^{'}(t)$ for users to improve the objective function $\sum_{j\in U} \frac{w_j}{T_j} b_j^{'}(t) > \sum_{j\in U} \frac{w_j}{T_j} b_j(t)$. As only one original candidate AP is lost for some user in Eq_j, each user's candidate AP set for Eq_j is a subset or equivalent set of the one for Eq_i. So we can apply this new solution to Eq_i, so as to further improve the objective function, but that contradicts with the fact that we already have optimal solution for Eq_i. Thus the assumption does not hold, the theorem gets proved. ■

According to *Theorem* 2, for the efficiency metric we only have to compute the optimized association solution each time when one or more users cross the boundary of *Equivalence Classes*. We can further prevent unnecessary computations by checking the special patterns of adjacent *Equivalence Classes*.

B. Offline Optimization for Proportional Fairness

In the above subsection we have demonstrated that for the efficiency metric we can transform the long-term overall optimization into the snapshot optimization. However, for proportional fairness, as each snapshot decision for the optimal solution may depend on its former and future situations, we cannot simply conduct this transformation.

As illustrated in Fig. 3, within each refined time interval $[t'_{l-1}, t'_l]$ for $l = 1, ..., L$, the candidate AP sets and corresponding bit rates for all users are fixed. We can devise a fixed pattern of association solution in an optimized approach. Following the offline setting described in Section III, and using the detailed information over the time span $[0, T]$ given in advance, we devise an offline algorithm to achieve optimization for proportional fairness. Here we use $b_{j,l}$ to denote the average bandwidth allocated in the time interval $[t'_{l-1}, t'_l]$, and we define $t_l = t'_l - t'_{l-1}$ for $l = 1, ..., L$. So we have $B_j = \frac{1}{T_j}\sum_{l=1}^{L}(b_{j,l} \cdot t_l)$. Since the objective is to maximize $\sum_{j\in U} w_j \ln B_j$, we have

$$\sum_{j\in U} w_j \ln B_j = \sum_{j\in U} w_j \ln(\sum_{l=1}^{L} b_{j,l} \cdot t_l) - \sum_{j\in U} w_j \ln T_j.$$

As $-\sum_{j\in U} w_j \ln T_j$ is constant, we can set the objective function as $\sum_{j\in U} w_j \ln(\sum_{l=0}^{L} b_{j,l} \cdot t_l)$. Then we obtain the following convex program formulation:

$$maximize \sum_{j\in U} w_j \ln(\sum_{l=0}^{L} b_{j,l} \cdot t_l) \tag{9}$$

subject to

$$\forall j \in U, l \in \{1, ..., L\} \quad b_{j,l} = \sum_{i\in A} r_{i,j}(l) \cdot p_{i,j}(l) \tag{10}$$

$$\forall i \in A, l \in \{1, ..., L\} \quad \sum_{j\in U} p_{i,j}(l) \le 1 \tag{11}$$

$$\forall j \in U, l \in \{1, ..., L\} \quad \sum_{i\in A} p_{i,j}(l) \le 1 \tag{12}$$

$$\forall i \in A, j \in U, l \in \{1, ..., L\} \quad 0 \le p_{i,j}(l) \le 1 \tag{13}$$

The constraints (10)-(13) are coherent with those constraints depicted in (4)-(7), here we respectively use $r_{i,j}(l)$ and $p_{i,j}(l)$ to denote the bit rate and fraction of time used between AP-user pair (i, j) in the time interval $[t'_{l-1}, t'_l]$. Since the objective function is a convex function and the other constraints are all linear, we can solve this convex program in polynomial time. The fractional solution of $p_{i,j}(l)$ is the exact solution for association control in the lth interval for $l = 1, .., L$, as we allow multiple handoffs within each interval to achieve the fractional solution. Now we consider how to devise optimized handoff strategies to achieve the optimized fractional solution.

Assume within any time interval $[t'_{l-1}, t'_l]$, there are K phases of association control as $AC_1, AC_2, ..., AC_K$. In each phase AC_k, for $k = 1, ..., K$, any user can associate with one unique AP in the integral solution. We use $x_{i,j}(k)$ to denote the integral association solution between AP i and user j, and $\alpha(k)$ to denote the time proportion used for phase k. Here we aim to minimize the overall number of handoffs as too many handoffs will incur heavy delays and communication costs. We use k_j to denote the number of handoffs for user j. Thus we have the following non-linear program formulation:

$$minimize \sum_{j\in U} k_j \tag{14}$$

subject to

$$\forall j \in U \quad k_j = \sum_{k=1}^{K-1} \sum_{i\in A} \max\{0, x_{i,j}(k+1) - x_{i,j}(k)\} \tag{15}$$

$$\forall i \in A, j \in U, k \in \{1, ..., K\} \quad x_{i,j}(k) \in \{0, 1\} \tag{16}$$

$$\forall j \in U, k \in \{1, ..., K\} \quad \sum_{i\in A} x_{i,j}(k) \le 1 \tag{17}$$

$$\forall i \in A, k \in \{1, ..., K\} \quad n_i(k) = \sum_{j\in U} x_{i,j}(k) \tag{18}$$

$$\forall i \in A, j \in U \quad \sum_{k=1}^{K} \frac{\alpha(k)}{n_i(k)} \cdot x_{i,j}(k) = p_{i,j}(l) \tag{19}$$

$$\forall k \in \{1, ..., K\} \quad 0 \le \alpha(k) \le 1 \tag{20}$$

$$\sum_{k=1}^{K} \alpha(k) = 1 \tag{21}$$

Here the first constraint calculates k_j by comparing each adjacent integral solutions for user j during the K phases. If for user j there exists some $i \in A$ for which $x_{i,j}(k)$ changes from 0 to 1 in the next solution $x_{i,j}(k+1)$, it means that a handoff happens for user j. The second constraint requires that $x_{i,j}(k)$ is an integral solution. The third constraint requires that any user can only associate with one unique AP for each phase. The fourth constraint states that $n_i(k)$ is the number of users associated to AP i for each phase. The fifth constraint states that the overall time proportion for AP i to associate to user j should equal to $p_{i,j}(l)$. The sixth and seventh constraints define the time proportion $\alpha(k)$ for each phase k.

To solve the optimization problem, we set K as a large number and calculate the series $x_{i,j}(1), x_{i,j}(2), ..., x_{i,j}(K)$ and $\alpha(1), \alpha(2), ..., \alpha(K)$, and reduce the series by merging those adjacent phases with equal $x_{i,j}$ matrices. The drawback of this approach is that if we do not set a large enough value for K, we may just get a “feasible” solution instead of an “optimal” solution.

C. Online Algorithm for Proportional Fairness

From the above subsection we know that the exact optimal solution can only be achieved with information obtained over the whole time span $(0, T)$ in advance. However, in practice we cannot precisely know users’ future mobility trajectory, thus no information about which users will be contending for specified APs in the future can be obtained beforehand. In this subsection, according to the online setting described in Section III, we design an online algorithm. Our solution relies on the following theorem.

Theorem 3: Maximizing the long-term objective function $\sum_{j\in U} w_j \ln(\epsilon + \int_0^T b_j(t)dt)$ is consistent with maximizing the long-term objective function $\int_0^T \sum_{j\in U} \frac{w_j}{\epsilon+\int_0^t b_j(t)dt} b_j(t)dt$. Here, $\int_0^t b_j(t)dt$ denotes the accumulated bandwidth in time span $[0,t]$, w_j denotes the original fixed weight as priority for each user j, and $\epsilon > 0$ is a small constant number.

Proof: Using the fact that T is constant and setting $x_j = \epsilon + \int_0^t b_j(t)dt$, we have

$$\int_0^T \sum_{j\in U} \frac{w_j b_j(t)}{\epsilon + \int_0^t b_j(t)dt} dt = \sum_{j\in U} \int_0^T \frac{w_j b_j(t)}{\epsilon + \int_0^t b_j(t)dt} dt$$
$$= \sum_{j\in U} \int_0^T \frac{w_j}{\epsilon + \int_0^t b_j(t)dt} d(\epsilon + \int_0^t b_j(t)dt)$$
$$= \sum_{j\in U} \int_\epsilon^{\epsilon+\int_0^T b_j(t)dt} \frac{w_j}{x_j} d(x_j)$$
$$= \sum_{j\in U} w_j \ln x_j \Big|_\epsilon^{\epsilon+\int_0^T b_j(t)dt}$$
$$= \sum_{j\in U} w_j \ln(\epsilon + \int_0^T b_j(t)dt) - \sum_{j\in U} w_j \ln \epsilon. \quad (22)$$

As ϵ is constant, maximizing (22) is equivalent to maximizing

$$\sum_{j\in U} w_j \ln(\epsilon + \int_0^T b_j(t)dt). \quad (23)$$

The theorem gets proved. ■

Recall that the original long-term goal is to maximize

$$\sum_{j\in U} w_j \ln(\int_0^T b_j(t)dt). \quad (24)$$

The only difference between the above two objective functions (23) and (24) is ϵ, which may have an impact on the corresponding optimal solution. However, as long as we set ϵ small enough ($\epsilon \to 0$) in (23), the long-term goal in (23) becomes very near to $\sum_{j\in U} w_j \ln(\int_0^T b_j(t)dt)$.

In order to maximize $f_O = \int_0^T \sum_{j\in U} \frac{w_j}{\epsilon+\int_0^t b_j(t)dt} b_j(t)dt$, we use the following heuristic snapshot objective $f'_O = \sum_{j\in U} \frac{w_j}{\epsilon+\int_0^t b_j(t)dt} b_j(t)$ along with the constraint depicted in (4)-(8) to approximate the long-term optimization solution. The intuition is that maximizing f'_O at each t contributes to the maximization of f_O. We thus propose an online algorithm DWOA based on dynamic weight $W_j(t) = \frac{w_j}{\epsilon+\int_0^t b_j(t)dt}$. Since it is possible that $\int_0^t b_j(t)dt = 0$, we let $\epsilon > 0$ to prevent $W_j(t)$ from equal to $+\infty$. This online algorithm is illustrated in Algorithm 1. Here Step 1 takes care of the fairness metric by setting $W_j(t)$ inversely proportional to the accumulated bandwidth. Step 2 considers the efficiency metric by attempting to maximize the sum of the weighted bandwidths. We update the association solution for every Δt time interval. Conventionally the less Δt we use, the better solution we can obtain, but the drawback is that it may cause too many handoffs.

D. Performance Analysis of DWOA

Due to the mathematical complexity of our objective function $\sum_{j\in U} w_j \ln(B_j)$, we take the following steps to define a metric R to evaluate our online algorithm DWOA.

Algorithm 1 DWOA: Dynamic Weight based Online Algorithm

1: $t = 0$
2: **while** $t < T$ **do**
3: **Step1.** For each user j, calculate $W_j(t) = \frac{w_j}{\epsilon+\int_0^t b_j(t)dt}$ for snapshot at t.
4: **Step2.** For snapshot at t, set the object function as to maximize $\sum_{j\in U} W_j(t)b_j(t)$, calculate and apply the solution for association.
5: **Step3.** $t = t + \Delta t$, go to Step 1.

First, realizing the equivalance of its maximization to that of $\sum_{j\in U} w_j \ln(B_j)$, we use $\prod_{j\in U}(B_j)^{w_j}$ as the objective function in the definition of R. Then similar to the approach used in [6], we define

$$R = \ln \frac{\prod_{j\in U}(B_j^*)^{w_j}}{\prod_{j\in U}(B_j)^{w_j}} = \sum_{j\in U} w_j \ln B_j^* - \sum_{j\in U} w_j \ln B_j.$$

Here B_j^* and B_j respectively denotes the overall optimal solution and the online solution. According to the definition of B_j, we can further obtain

$$R = \sum_{j\in U} w_j \ln \int_0^T b_j^*(t)dt - \sum_{j\in U} w_j \ln \int_0^T b_j(t)dt.$$

Theorem 4: The upper bound of $R = \sum_{j\in U} w_j \ln B_j^* - \sum_{j\in U} w_j \ln B_j$ for DWOA is

$$\sum_{j\in U} w_j \cdot \ln(\frac{r_{max}T_{max}}{r_{min}T_{min}} \cdot \frac{n}{\rho}),$$

where $\rho = \frac{n\cdot\min_{j\in U} w_j}{\sum_{j\in U} w_j}$, $T_{min} = \min_{j\in U}(T_j)$, $T_{max} = \max_{j\in U}(T_j)$, r_{min} is the minimum value of bit rate, and r_{max} is the maximum value of bit rate.

Proof: Since DWOA attempts to maximize $\sum_{j\in U} w_j \ln(\epsilon + \int_0^T b_j(t)dt)$, we first define $R(\epsilon) = \sum_{j\in U} w_j \ln(\epsilon + \int_0^T b_j^*(t)dt) - \sum_{j\in U} w_j \ln(\epsilon + \int_0^T b_j(t)dt)$. We then have

$$R(\epsilon) = \sum_{j\in U} w_j \ln(\frac{\epsilon + \int_0^T b_j^*(t)dt}{\epsilon + \int_0^T b_j(t)dt}).$$

As $\int_0^T b_j^*(t)dt \leq T_{max} \cdot r_{max}$ and $\int_0^T b_j(t)dt \geq \min_{j\in U}(\int_0^T b_j(t)dt)$, we have

$$R(\epsilon) \leq \sum_{j\in U} w_j \cdot \ln(\frac{\epsilon + T_{max}\cdot r_{max}}{\epsilon + \min_{j\in U}(\int_0^T b_j(t)dt)}).$$

Then according to the definitions of R and $R(\epsilon)$, we have

$$R - R(\epsilon) = \sum_{j\in U} w_j \ln(\frac{(\int_0^T b_j^*(t)dt)\cdot(\epsilon + \int_0^T b_j(t)dt)}{(\epsilon + \int_0^T b_j^*(t)dt)\cdot(\int_0^T b_j(t)dt)})$$
$$\leq \sum_{j\in U} w_j \ln(\frac{\epsilon}{\int_0^T b_j(t)dt} + 1).$$

Therefore we obtain

$$\begin{aligned}
R &\le \sum_{j\in U} w_j \ln(\frac{\epsilon}{\min_{j\in U}(\int_0^T b_j(t)dt)}+1)+R(\epsilon)\\
&\le \sum_{j\in U} w_j\cdot\ln(\frac{\epsilon+\min_{j\in U}(\int_0^T b_j(t)dt)}{\min_{j\in U}(\int_0^T b_j(t)dt)})\\
&+\sum_{j\in U} w_j\cdot\ln(\frac{\epsilon+T_{max}\cdot r_{max}}{\epsilon+\min_{j\in U}(\int_0^T b_j(t)dt)})\\
&=\sum_{j\in U} w_j\cdot\ln\frac{\epsilon+T_{max}\cdot r_{max}}{\min_{j\in U}(\int_0^T b_j(t)dt)}. \qquad (25)
\end{aligned}$$

Now we need to estimate $\min_{j\in U}(\int_0^T b_j(t)dt)$ in Eq. (25). We consider the worst case that all users are driving in a very close neighborhood, and so they are always contending for the same APs. We assume for the n users $w_1 > w_2 > ... > w_n$. According to the algorithm, for the optimal snapshot solution, each AP will only allocate bandwidth to the user with the largest weight $W_j(t)$ within its range. As $W_j(t)$ is changing all the time, we adopt the following procedure: user 1 first get all the bandwidth from the AP until $W_1(t) = W_2(t)$, then user 1 and 2 will further get bandwidth until $W_1(t) = W_2(t) = W_3(t)$, ..., finally we can achieve $W_1(t) = W_2(t) = ... = W_n(t)$. From then on, all users will share the bandwidth distribution and keep $W_1(t) = W_2(t) = ... = W_n(t)$, in this way each user j further gets the bandwidth from the AP with proportion $\frac{w_j}{\sum_{j\in U} w_j}$. In order to obtain $\min_{j\in U}(\int_0^T b_j(t)dt)$, we further assume that user n has the minimum service duration $T_n = \min_{j\in U}(T_j)$, thus among all users $j \in U$ user n obtains the least bandwith.

We denote the time interval $[0, t^*]$ during which user n gets no bandwidth and $[t^*, T_n]$ during which user n shares the AP's bandwidth distribution. Thus during the interval $[0, t^*]$, the total downloaded bits for user $1, ..., n-1$ is $\sum_{j=1}^{n-1}\int_0^{t^*} b_j(t)dt$. As the minimum bit rate is r_{min}, we obtain $t^* \le \frac{1}{r_{min}}\sum_{j=1}^{n-1}\int_0^{t^*} b_j(t)dt$. At the time point t^*, as $W_1(t^*) = W_2(t^*) = ... = W_n(t^*)$, according to the definition of $W_j(t)$, we have

$$\forall j = 1, 2..., n-1, \quad \frac{w_j}{\epsilon+\int_0^{t^*} b_j(t)dt} = \frac{w_n}{\epsilon}.$$

Thus we get $\int_0^{t^*} b_j(t)dt = (\frac{w_j}{w_n}-1)\epsilon$, and further more

$$t^* \le \frac{\epsilon}{r_{min}}(\frac{1}{w_n}\sum_{j\in U} w_j - n). \qquad (26)$$

Now according to Eq. (26) we can estimate the minimum accumulated bandwidth among the users

$$\begin{aligned}
&\min_{j\in U}(\int_0^T b_j(t)dt) \ge (T_n - t^*)\cdot\frac{w_n r_{min}}{\sum_{j\in U} w_j}\\
&\ge \frac{w_n r_{min}}{\sum_{j\in U} w_j}\min_{j\in U}(T_j) + (\frac{w_n\cdot n}{\sum_{j\in U} w_j}-1)\epsilon.
\end{aligned}$$

Then according to Eq. (25), if we set $\rho = \frac{n\cdot\min_{j\in U} w_j}{\sum_{j\in U} w_j}$, we have

$$R \le \sum_{j\in U} w_j\cdot\ln[\frac{\epsilon + r_{max}T_{max}}{(\rho-1)\epsilon+\rho r_{min}T_{min}/n}].$$

Here since ϵ is an adjustable parameter, it is possible to determine an optimal value for ϵ to minimize the bound. As $\rho \le 1$, when $\epsilon \to 0$, we can achieve the minimum value for R's upper bound as $\sum_{j\in U} w_j\cdot\ln(\frac{r_{max}T_{max}}{r_{min}T_{min}}\cdot\frac{n}{\rho})$. ■

For the optimal solution B_j^* and the online solution B_j, we respectively denote the results of objective function as $f_O(B_j^*)$ and $f_O(B_j)$. We define $D = \frac{r_{max}T_{max}}{r_{min}T_{min}}\cdot\frac{n}{\rho}$. According to *Theorem* 4, we have

$$\sum_{j\in U} w_j\cdot\ln B_j^* - \sum_{j\in U} w_j\cdot\ln D \le \sum_{j\in U} w_j\cdot\ln B_j.$$

So we have $f_O(B_j^*/D) \le f_O(B_j)$, implying that the total utility of the bandwidth allocation vector $\vec{B}$ given by *DWOA* is greater than that of the optimal bandwidth allocation vector $\vec{B^*}$ scaled down by a factor of D. Since $D = \frac{r_{max}T_{max}}{r_{min}T_{min}}\cdot\frac{n}{\rho}$, by achieving the smaller value gaps for (r_{min}, r_{max}) and (T_{min}, T_{max}), as well as the smaller value of n, better performance can be obtained by DWOA.

V. GROUP-BASED APPROACH

The above proposed snapshot solution as well as the online algorithm solution both require continuously solving a linear program for each snapshot. The computation complexity may be huge in our considered scenario, where we have to deal with an extremely large Wi-Fi deployment. For example, according to polynomial-time algorithms for linear program like the ellipsoid method[18], the computation complexity is $O(N^4\cdot L)$, where N is the number of parameters and L is the length of encoding bits for parameters. For our optimization problems, we have $N = m\cdot n$ for all parameters $p_{i,j}$, (m is the number of APs and n is the number of users). Thus, if n and m are large enough, N will be a huge number and the computation complexity will be tremendous. Hereby we propose a group-based approach that partitions the network into groups. In each group, we apply the aforementioned LP method so that the computation complexity will be reduced. In the rest of this section, as we only focus on the snapshot solution, for the ease of presentation, we omit the "(t)" for snapshot parameters in the formulas. Next, we give a formal definition of "*group*".

Definition 1. We say that a set of users belong to the same *group* if and only if for any two users j' and j from the set, there is a series of users $j_1, j_2, ..., j_s$ from the set such that $A_{j'}\bigcap A_{j_1} \ne \phi, A_{j_1}\bigcap A_{j_2} \ne \phi, ..., A_{j_s}\bigcap A_j \ne \phi$, where A_j is user j's candidate AP set.

According to this definition, different groups are *mutually exclusive* (each user can only belong to one unique group) and *independent* (users belonging to different groups will have no shared candidate APs for contention). For any snapshot, the users over the roads can be divided into one or more groups.

Therefore our association control over all users is reduced to association control within each group inferred by *Theorem* 5.

Theorem 5: For snapshot solution of the efficiency metric and online solution of the fairness metric, the optimization achieved within each group is consistent with the overall optimization.

Due to lack of space, we omit the proof of *Theorem* 5. As we have learned from Section IV, both snapshot solution for efficiency and online solution for fairness can be unified in the formulation as to maximize $\sum_{j\in U} W_j \cdot b_j$ for each snapshot. Here for efficiency, we have $W_j = w_j/T_j$; and for proportional fairness, we have $W_j = \frac{w_j}{\epsilon+\int_0^t b_j(t)dt}$.

A. Breaking into Smaller Sub-Groups

In a dense traffic scenario, the distance between adjacent users may be close to share some candidate APs, thus we cannot divide them into separate groups. Hence we may still have extremely large-sized *groups* over the roads. In order to reduce the computation complexity, we need to break the large groups into smaller sub-groups. Therefore before we conduct optimization with the linear program based method, we first perform pre-processing. For each user j, we delete those weak links (i', j) with bit rate $r_{i',j}$ low enough to satisfy $r_{i',j} < \beta_j \cdot \max_{i\in A_j} r_{i,j}$. Here, $\max_{i\in A_j} r_{i,j}$ is the maximum bit rate that user j can achieve within its candidate AP set A_j, and

$$\beta_j = \begin{cases} \gamma/c_{i^*} & \text{If } c_{i^*} \geq \gamma \\ 1 & \text{If } c_{i^*} < \gamma \end{cases} \tag{27}$$

where i^* is the specific AP with the maximum bit rate, c_{i^*} is the number of users within AP i^*'s effective range, and $\gamma \geq 1$ is a constant parameter. When $\gamma = 1$, the intuition is that from user j's perspective, even choosing i' as its own AP with no contention is worse than contending for i^* with c_{i^*} users, so it is too weak to worth association. We use the constant parameter γ to control the effect for weak link eliminations. Actually the larger γ is, the greater number of links will be deleted. Thus we are able to break a large group into more sub-groups. After deleting these weak links, we check the bipartite graph $G(V, E)$ of AP-user links corresponding to the original large group. Here V denotes the set of users and APs, and E denotes links after weak link eliminations. If the graph is disconnected somewhere due to the effect of weak link elimination, we will obtain new sub-groups. Algorithm 2 illustrates the detailed procedure.

Algorithm 2 Breaking a Large Group into Smaller sub-groups

```
1: for each user j within group G do
2:     Find the maximum r_{i,j} as R_j for user j from its candidate AP set A_j
3:     for each AP i within j's candidate AP set A_j do
4:         if r_{i',j} ≤ β_j · R_j then
5:             Delete edge (i', j).
6: Check the bipartite graph G(V, E) of group G.
7: if G(V, E) is disconnected. then
8:     Break G into subgroup G_1, G_2, ..., G_k according to graph G(V, E).
9:     Output: G_1, G_2, ..., G_k.
```

B. Approximation Ratio Analysis

Assume that for any user j in the original optimized solution without edge removal, we have bandwidth b_j for the fractional solution and $\hat{b_j}$ for the integral solution; and similarly in the approximate solution with edge removal, we have bandwidth b'_j for the fractional solution and $\hat{b'_j}$ for the integral solution. For both the efficiency and fairness metrics, we define the approximation ratio as $\frac{\sum_{j\in G} W_j \cdot \hat{b_j}}{\sum_{j\in G} W_j \cdot \hat{b'_j}}$. Apparently the ratio is at least 1 as $\sum_{j\in G} W_j \cdot \hat{b_j} \geq \sum_{j\in G} W_j \cdot \hat{b'_j}$, since the candidate link set for the original solution is a superset of the one for approximate solution. In the following we give an upper bound for this approximation ratio.

For any user j, we denote A_j as its original candidate AP set, and S_j as a subset of A_j for the remaining candidate AP set after the weak link elimination. Thus $A_j - S_j$ is the set of those eliminated candidate APs. We respectively utilize matrix $p_{i,j}$ and $p'_{i,j}$ to denote the optimized parameters for the original solution and the approximate solution. Then according to the definition we have

$$b_j = \sum_{i\in A_j} r_{i,j} \cdot p_{i,j} = \sum_{i\in S_j} r_{i,j} \cdot p_{i,j} + \sum_{i\in A_j - S_j} r_{i,j} \cdot p_{i,j} \tag{28}$$

$$b'_j = \sum_{i\in A_j} r_{i,j} \cdot p'_{i,j} = \sum_{i\in S_j} r_{i,j} \cdot p'_{i,j}. \tag{29}$$

We have the property in *Lemma* 1.

Lemma 1: For optimal parameters $p_{i,j}$ in set S_j for the original solution and optimal parameters $p'_{i,j}$ in set S_j for the approximate solution, $\sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} p_{i,j} \leq \sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} p'_{i,j}$.

Proof: After removing some weak links, those $p_{i',j}$ with $i' \notin S_j$ corresponding to the weak links are set to 0. Consequently some constraints for the remaining $p_{i,j}$ with $i \in S_j$ get relaxed. We can then add a slack value to adjust $p_{i,j}$ to $p'_{i,j}$ so as to further increase the objective function $\sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} p_{i,j}$, which at least can remain the same by setting $p'_{i,j} = p_{i,j}$. Therefore the objective function of the optimal solution should be equal to or larger than the previous one, i.e. $\sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} p_{i,j} \leq \sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} p'_{i,j}$. The lemma gets proved. ■

Theorem 6: For the snapshot solution of the weight based objective function, the upper bound of the approximation ratio for group breaking is $\theta + \gamma$, where θ is the approximation ratio for the rounding algorithm in Shmoy and Tardos[17].

Proof: According to Eq. (28) and Eq. (29),

$$\sum_{j\in G} W_j b_j = \sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} \cdot p_{i,j} + \sum_{j\in G}\sum_{i\in A_j - S_j} W_j r_{i,j} \cdot p_{i,j}.$$

$$\sum_{j\in G} W_j b'_j = \sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} \cdot p'_{i,j}.$$

According to Lemma 1,

$$\sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} p_{i,j} \le \sum_{j\in G}\sum_{i\in S_j} W_j r_{i,j} p'_{i,j} = \sum_{j\in G} W_j b'_j.$$

Then

$$\sum_{j\in G} W_j b_j \le \sum_{j\in G} W_j b'_j + \sum_{j\in G}\sum_{i\in A_j - S_j} W_j r_{i,j}\cdot p_{i,j}.$$

Thus by the definition of weak link (i,j) with $i \in A_j - S_j$, we have $r_{i,j} < \beta_j \cdot R_j$, where $R_j = \max_{i\in A_j} r_{i,j}$. Then

$$\sum_{j\in G} W_j b_j \le \sum_{j\in G} W_j b'_j + \sum_{j\in G}(\beta_j R_j W_j \cdot \sum_{i\in A_j - S_j} p_{i,j}).$$

Since $\sum_{i\in A_j - S_j} p_{i,j} \le 1$, then

$$\sum_{j\in G} W_j b_j \le \sum_{j\in G} W_j b'_j + \sum_{j\in G} W_j \beta_j R_j.$$

As we have $\sum_{j\in G} W_j \cdot \widehat{b_j} \le \sum_{j\in G} W_j \cdot b_j$ because the fractional optimal solution can always achieve a better result than the integral optimal solution, then

$$\frac{\sum_{j\in G} W_j\cdot\widehat{b_j}}{\sum_{j\in G} W_j\cdot\widehat{b'_j}} \le \frac{\sum_{j\in G} W_j\cdot b_j}{\sum_{j\in G} W_j\cdot\widehat{b'_j}} \le \frac{\sum_{j\in G} W_j\cdot b'_j + \sum_{j\in G} W_j\cdot\beta_j\cdot R_j}{\sum_{j\in G} W_j\cdot\widehat{b'_j}}.$$

We set $\theta = \frac{\sum_{j\in G} W_j\cdot b'_j}{\sum_{j\in G} W_j\cdot\widehat{b'_j}}$ to be the approximation ratio for the rounding algorithms proposed by Shmoy and Tardos[17], and $1 \le \theta \le 2$. So,

$$\frac{\sum_{j\in G} W_j\cdot\widehat{b_j}}{\sum_{j\in G} W_j\cdot\widehat{b'_j}} \le \theta + \frac{\sum_{j\in G} W_j\cdot\frac{\gamma}{c_{i^*}}\cdot R_j}{\sum_{j\in G} W_j\cdot\widehat{b'_j}}.$$

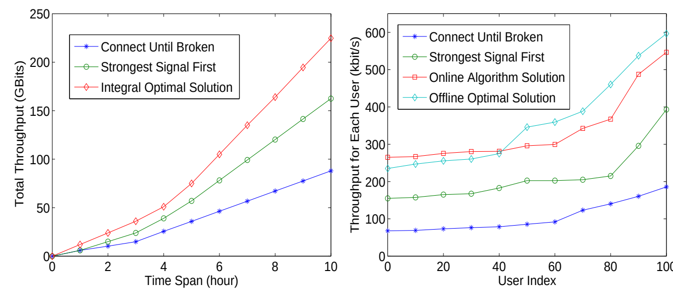
After the weak link elimination, the optimal integral solution may come with a worst case, where a bunch of users share only one candidate AP which has the largest bit rate for any of them, with the other candidate APs all eliminated. Then the optimal method for each user j is to select the unique AP with R_j, which is the largest $r_{i,j}$ for j. Thus we have $\sum_{j\in G} W_j \cdot \frac{R_j}{c_{i^*}}$ as the lower bound for the optimal result $\sum_{j\in G} W_j\cdot\widehat{b'_j}$. So,

$$\frac{\sum_{j\in G} W_j\cdot\widehat{b_j}}{\sum_{j\in G} W_j\cdot\widehat{b'_j}} \le \theta + \frac{\sum_{j\in G} W_j\cdot\frac{\gamma}{c_{i^*}}\cdot R_j}{\sum_{j\in G} W_j\cdot\frac{R_j}{c_{i^*}}} \le \theta+\gamma$$

Therefore we obtain the approximation ratio as $\theta+\gamma$, where γ is a constant. This indicates we can well control the approximation ratio by adjusting γ. ■

VI. Performance Evaluations

We have implemented a simulator to simulate the Drive-thru Internet scenario over a Wi-Fi deployed square region (20km $\times$ 20km) covered with 10 roads. Among them 5 roads are along the east-west direction and 5 roads are along the north-west direction, thus forming 5×5 intersections within the square region. We randomly place a total of 2000 APs over these roads and adopt the experiment results from [2] to simulate effective bit rates of APs. We make sure that at any location of the roads the user is within effective range of at least one AP, and that their peak bit rates range from 1000kpbs to 3500kbps for vehicular users. In our simulation, we simulate a total of 100 users driving over these roads with speed ranged from 40km/h to 100km/h, and utilize a *Poisson Process* with parameter λ to simulate the series of vehicular users within time span $[0h, 10h]$. On average, every $10/\lambda$ second a new user will drive into this region. Among these multiple roads, each user randomly selects the trajectory to drive through the region. We solve the linear program and convex program using MATLAB.

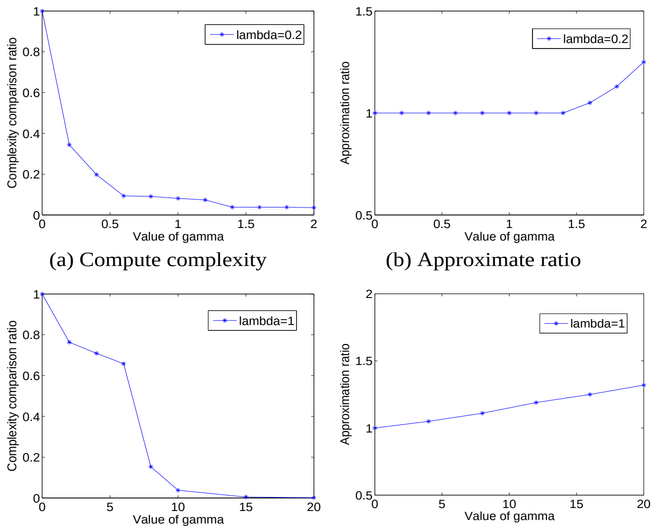


(a) (b)

Fig. 4. Performance comparison for (a) efficiency and (b) proportional fairness

A. Efficiency and Fairness

In Fig. 4(a) and Fig. 4(b) we respectively evaluate performance of our optimal strategies for efficiency and fairness. We compare performance with two heuristic strategies. The first strategy is *Connect Until Broken (CUB)*, which maintains a connection with a user and an AP until the user considers the link to be broken. Upon disconnection, the user will be associated with a new AP which yields the largest signal strength. The second strategy is *Strongest Signal First (SSF)*, which always associates a user with the AP yielding the strongest received signal strength at all times. Fig. 4(a) shows the result for the efficiency metric, where the X axis is the time span as users are driving over the roads and the Y axis is the users' total received data measured in GBits. For the ease of comparison, we set $w_j = 1$ and $T_j = 1h$ for each user, so comparing the total received bits is equal to comparing the overall throughput. We observe that the integral optimal solution outperforms both the *SSF* solution and the *CUB* solution. The *SSF* solution achieves about 70% of the total throughput of the integral optimal solution, while the *CUB* solution always performs worst for it only reaches 38% of the total throughput in comparison with the integral optimal solution. Fig. 4(b) shows the result for the proportional fairness metric, where the X axis is the user index and the Y axis is users' throughput in Kbps. The users are sorted by their throughput in increasing order. For the ease of comparison, we set $w_j = 1$ for each user. For the online algorithm we set $\epsilon = 0.01$ and every 5 second we recalculate the association solution. We observe that although the two heuristics have similar growth trend as our optimal solutions, and that both the offline optimal solution and online solution outperform the two heuristic solutions for overall throughput. For instance, the median user's bandwidth value of the online algorithm solution is 69% higher than the *SSF* solution and 300% higher than the *CUB* solution. The offline optimal solution has better performance than the online solution, since the median user's bandwidth value of the former is 12.9% higher than the latter.



(a) Compute complexity (b) Approximate ratio

(c) Compute complexity (d) Approximate ratio

Fig. 5. Performance evaluation for group breaking approximate algorithm

B. Group Breaking Mechanism

In Fig. 5 we evaluate the performance for the group breaking approximate algorithm in terms of computation complexity and approximate ratio. Here we normalize the computation complexity as a ratio between the complexity after group breaking and the original complexity. We respectively consider two conventional cases with different λ for the *Poisson Process*. For the sparse case with $\lambda = 0.2$, the average distance between adjacent users is large, thus we can break the large group of users into many sub-groups. As Fig. 5 (a) shows, when γ varies from 0 to 0.6, the computation complexity reduces dramatically from 1 to 0.1, and when γ further increases from 0.6 to 2, as many of the weak links has already been eliminated, the computation complexity reduces gently. As Fig. 5 (b) shows, when γ varies from 0 to 1.4, the average approximate ratio keeps close to 1 as those weak link deleted among the sparse group of users will not impact on the final optimal solution, and when γ further increases from 1.4 to 2, the approximate ratio grows slightly from 1 to 1.25. For the dense case with $\lambda = 1$, the average distance between adjacent users is rather small so that eliminating some weak links will not be enough to break the large group of users into many sub-groups. As Fig. 5 (c) shows, the computation complexity first reduces gently when γ increases from 0 to 6, then reduces dramatically when γ increases from 6 to 10 due to weak links, and finally the computation complexity reduces slightly. As Fig. 5 (d) shows, when γ varies from 0 to 20, the averaged approximate ratio grows near linearly from 1 to 1.32. We observe that it is much smaller than the worst case ratio $\theta + \gamma$, which demonstrates for conventional cases we achieve tight bound for the expected approximation ratio.

VII. Conclusion

In this paper, we conduct a theoretical research on association control over the *Drive-thru Internet* scenario. We consider both efficiency and fairness metrics, and present corresponding offline and online solutions to achieve overall optimization objectives. We further propose an approximation algorithm for group breaking in order to reduce computation complexity. Simulation results confirm the performance of our optimized algorithms for association control.

References

[1] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ Wi-Fi networks," in *ACM Proc. of MOBICOM*, 2006.

[2] J. Ott and D. Kutscher, "Drive-thru internet: IEEE 802.11b for 'automobile' users," in *IEEE Proc. of INFOCOM*, 2004.

[3] A. Giannoulis, M. Fiore, and E. W. Knightly, "Supporting vehicular mobility in urban multi-hop wireless networks," in *ACM Proc. of MobiSys*, 2008.

[4] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 23, no. 1, pp. 163–173, 2005.

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

[6] L. Li, M. Pal, and Y. R. Yang, "Proportional fairness in multi-rate wireless LANs," in *IEEE Proc. of INFOCOM*, 2008.

[7] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using Wi-Fi," in *ACM Proc. of MOBICOM*, 2008.

[8] J. Ott and D. Kutscher, "A disconnection-tolerant transport for drive-thru internet environments," in *IEEE Proc. of INFOCOM*, 2005.

[9] R. Mahajan, J. Zahorjan, and B. Zill, "Understanding WiFi-based connectivity from moving vehicles," in *Internet Measurement Conference*, 2007.

[10] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular opportunistic communication under the microscope," in *ACM Proc. of MobiSys*, 2007.

[11] M. Kim, Z. Liu, S. Parthasarathy, D. Pendarakis, and H. Yang, "Association control in mobile wireless networks," in *IEEE Proc. of INFOCOM*, 2008.

[12] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. R. Das, "Mobisteer: using steerable beam directional antenna for vehicular network access," in *IEEE Proc. of MobiSys*, 2007.

[13] X. Zhang, J. Kurose, B. N. Levine, D. F. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing," in *ACM Proc. of MOBICOM*, 2007.

[14] L. B. Jiang and S. C. Liew, "Proportional fairness in wireless LANs and ad hoc networks," in *IEEE Proc. of WCNC*, 2005.

[15] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, "Achieving mac layer fairness in wireless packet networks," in *ACM Proc. of MobiCom*, 2000, pp. 87–98.

[16] R. Murty, J. Padhye, A. Wolman, and B. Zill, "Designing high performance enterprise Wi-Fi networks," in *ACM Proc. of NSDI*, 2008.

[17] D. B. Shmoys and E. Tardos, "An approximation algorithm for the generalized assignment problem," *Math. Program*, vol. 62, no. 3, pp. 461–474, 1993.

[18] W. H. Freeman, "Linear programming," 1983.