LOCALIZED ACCESS POINT SELECTION IN INFRASTRUCTURE WIRELESS LAN

Mingming Lu and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

ABSTRACT

The poor performance of the current access point (AP) selection schemes in infrastructure WLANs motivated intensive studies to balance loads among APs. In this paper, we model the AP selection problem as the matching problem in the bipartite graph. Our objective is to maximize the total load among all APs. We propose a class of localized heuristics based on different user knowledge models. For some of these localized heuristics, we prove that there exists a constant approximation ratio in terms of expected total load through both mathematical analysis and simulations.

Keywords: *Access point selection, bipartite matching, infrastructure wireless LAN.*

I. INTRODUCTION

Over the last few years, network users have experienced enormous growth in the adoption of infrastructure wireless LANs, such as IEEE 802.11. The infrastructure wireless LAN's easy installation and low infrastructure cost make it ideal for network access in offices, campuses, airports, and hotels. The widespread deployment of IEEE 802.11 networks means that a wireless user is often in the vicinity of multiple APs with which to affiliate. For a wireless user, the selection of the proper AP from all available APs is very important because it has a great impact on his performance.

The current approach to access point selection is based on the signal strength measured by wireless users. Each individual user selects the AP with the highest signal strength from all APs within the vicinity. According to research [1], [6], [3], this approach can lead to poor performance for the users because the signal-strength metric does not consider the load of APs and the amount of contention on the wireless medium.

Numerous existing works have modeled the AP selection problem as load balance problems by assuming the existence of a central node that can collect the global information and execute the centralized LP-based algorithms, exploiting the assistance from APs, or through active (or passive) measurement on APs so that users can obtain knowledge on the available (or potential) bandwidth. In this work, we step back and look at two factors that can affect the performance of the AP selection problem: the knowledge of wireless users and the timing of the AP selection for the different users. We only consider the information that can be collected through some minor modification of the existing 802.11 protocols. Thus, we focus on the localized solutions to the AP selection problem.

We adopt the number of users connected to an AP to model the load capacity of this AP by assuming all users have the same minimum quality requirement. Thus, we can model the AP selection problem as a many-to-one $(m-1)$ matching problem in the bipartite graph, where each user can connect to only one AP, while an AP can be connected by multiple users. In our model, the optimization objective is to maximize the total load, defined as the total number of users whose minimum quality requirements are satisfied.

We first consider the localized solution to the special case, $1-1$ matching, i.e., one user matched to one AP, and extend the results to $m-1$ matching. In $1-1$ matching, we consider three types of user knowledge models separately. In each user knowledge model, we further explore the effect of the order in which users select their APs. For each case, we propose a localize heuristic. For some localized heuristics proposed in this paper, we prove that there exists an approximation ratio in terms of expected total load.

The major contributions of this paper are: 1) we propose a relatively light-weighted optimization model; 2) we study the effect of knowledge and the time of AP selection on the performance; 3) we design an approximation algorithm with constant average approximation ratio.

The remainder of this paper is organized as follows. Section II introduces models and notations, and formalizes the problem. In Section III, we study the $1-1$ matching problem from the angle of three types of user knowledge models and two different orders of the AP selection. Section IV extends our results to the $m-1$ matching problem. Section V empirically evaluates performance through our customized simulator. Section VI presents related works. Finally, Section VII concludes this work and outlines our future work.

1-4244-1513-06/07/$25.00 ©2007 IEEE

II. Preliminaries

A. The Network Model

A set of APs and a set of wireless users compose an infrastructure wireless LAN. We adopt A to denote the set of APs. Each AP has a fixed transmission range and can only serve users within its range. The coverage area of the wireless network consists of the union of the area covered by each AP in A. We use U to denote the set of wireless users that reside in the coverage area of the network. In order to access the wireless network, each user has to connect to an AP. We assume that the wireless users are free to move but they tend to stay in the same physical location for a long period of time. This assumption is backed up by recent studies of wireless user behavior [2], [8].

We model the network as a bipartite graph $(A \cup U, E)$, where $E \subseteq A \times U$ is the set of links connecting users to APs. There is a link between an AP and a user if and only if the user is within the range of the AP. We call the user the AP's *neighbor user* and the AP the user's *neighbor AP*. We use u_i and a_j to denote user i and AP j, respectively. We consider two users neighbors if and only if they are within the range of the same AP.

The strategy that a user can employ to select an AP depends on the user's knowledge. The basic user knowledge is the information about his neighbor AP. We define this type of information as 0.5-*hop knowledge*. Besides that, we consider 1-*hop knowledge* and 1.5-*hop knowledge.* A user's 1-hop knowledge includes his 0.5-hop knowledge and information of his neighbor users. A user's 1.5-hop knowledge contains his 1-hop knowledge plus the neighbor APs of his neighbor users. For example, in Fig. 1, where users and APs are represented by crossed and circles, respectively, u_1's 0.5-hop knowledge includes APs a_1 and a_2, u_1's 1-hop knowledge further contains user u_i $(i = 2, 3, 4, 5, 6)$, and u_1's 1.5-hop knowledge includes all APs and users in Fig. 1. The reason that we consider 1-hop knowledge and the 1.5-hop knowledge models is that the 0.5-hop knowledge alone cannot help users self-distribute load among APs. Since the information beyond 1.5-hop knowledge requires multiple rounds of information exchange, we focus on 1-hop and 1.5-hop knowledge models.

B. Problem Formulation

Our concern focuses on the localized solutions that enable as many *satisfied users* as possible. The satisfied users denote those users whose *minimum quality requirements* are satisfied. The localized solutions means that each user self-determines which neighbor AP to connect to, and his decision is based on his local information alone.

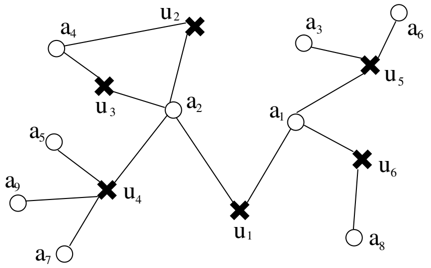


Fig. 1. The graph constructed according to u_1's 1.5-hop information.

To simplify the model, we assume that users are homogeneous, and hence, users have the same minimum requirement to the quality such as bandwidth, delay, and etc. We also assume that there is a limitation on APs' capacities. If the number of users connecting to an AP simultaneously exceeds a certain threshold, the quality of network access will fall under users' minimum quality requirement.

Based on the above discussion, the AP selection problem can be modeled as the $m - 1$ *matching problem* in the bipartite graph. In the $m - 1$ matching problem, each user can connect to only one AP, whereas an AP can be connected by multiple users simultaneously. The objective of this problem is to find a matching scheme so that the number of satisfied users is maximized.

There are two major differences between the traditional max-flow based algorithms [4] and our localized solutions for the bipartite matching problem: 1) our localized solutions do not require central node to collect global information and compute the optimal solution accordingly; 2) our localized solutions are non-preemptive, i.e., once a user matches an AP, this matching is determined and other users cannot force him to switch to other APs.

III. Solutions to the $1 - 1$ Matching problem

We first consider a special case of the $m - 1$ bipartite matching problem: the $1 - 1$ bipartite matching problem, where at most one user is allowed to connect to an AP, i.e., the threshold of any AP is 1. Under each user knowledge model, we consider two different AP-selection orders: simultaneous connection and sequential connection.

A. The 0.5-*hop Knowledge Model*

The existing 802.11 based network protocols provide the 0.5-hop knowledge through signal detection of APs on the user side. We first consider the simultaneous connection, where users connect to their intended APs simultaneously.

1) Simultaneous Connection: The simultaneous-connection based schemes can be regarded as a variations of the greedy method in the current 802.11 protocols, where each user connects to the neighbor AP with the strongest signal. Because users have no knowledge about their neighbor users, user strategies can regarded as being made at the same time, even though they may not be made simultaneously.

In Algorithm 1, we formally present a generic user strategy for simultaneous connection in the 0.5-hop knowledge model. We assume user u_i has k neighbor APs denoted as $\{a_1, a_2, \cdots, a_k\}$. u_i connects to a_j with probability $p(a_j)$.

Algorithm 1 Simu-Connection (u_i) in 0.5-hop Model

```
Input: $\{a_1, a_2, \cdots, a_k\}$
calculate $p(a_j), j \in \{1, \cdots, k\}$
connect to $a_j$ with probability $p(a_j)$
```

Different implementations of the probability calculation in the above strategy can derive different methods. For example, by assigning probability 1 to the neighbor AP with the strongest signal and probability 0 to all the other neighbor APs, the generic scheme can be reduced to the greedy method. By assigning probability $\frac{1}{k}$ to each neighbor AP, the generic scheme is reduced to the method in that each user connects to his neighbor APs with equal probability.

In this generic scheme, the conflict is incurred by multiple users connecting to the same AP. In that cases those users can reduce their connection probabilities to reduce conflict. For example, if the number of users that select an AP is m, each of those users can set his connecting probability to $\frac{1}{m}$. A trade-off exists between the conflict probability and the connection probability.

2) Sequential Connection: To reduce the conflict that caused by multiple users connecting to the same AP at the same time, we consider a back-off based sequential connection scheme. We assume that the time consists of multiple continuous time slots. Initially, each user sets a back-off counter to each neighbor AP. The value of the counter (in terms of the number of back-off slots) associated with each AP is reverse proportional to the received signal strength indicator (RSSI) of the AP, i.e., the signal strength of the AP sensed by the user. The counter of each neighbor AP will decrease by 1 for every l slots. At the beginning of every l slots, if any counter becomes 0, a user randomly selects a slot within the next l slots to connect to the corresponding AP. Once an AP is connected by a user, the AP will send an acknowledgement message to the user. By overhearing the message, all the neighbor users of this AP will know this AP is occupied. Those unconnected neighbor users will remove the AP from their neighbor AP sets.

Note that the conflict probability is proportional to the value of l. The larger the value of l, the lower the conflict probability. In Algorithm 2, we formally present our back-off based strategy for each user. We assume that each user initially has at least one neighbor AP. The following notations and operations are used in the back-off based strategy: 1) t: the timer that reduced 1 at the beginning of every l time slots; 2) $N(u_i)$: the set of neighbor APs of user u_i, i.e., the set of APs that u_i can connect to; 3) $B_{u_i}(a_j)$: the back-off counter for neighbor AP a_j; 4) B_{min}: the back-off counter with the smallest value; 5) a_1: the neighbor AP with the smallest back-off counter; 6) $RANDOM(k)$: the random number generator that generates random integer number range from 0 to $l-1$.

Algorithm 2 Sequential Connection (u_i) in 0.5-hop Model

```
Input: $N(u_i) \leftarrow \{a_1, a_2, \cdots, a_k\}$
status ← unconnected; /*initialize status*/
initialize $B_{u_i}(a_j)$ for $j \in \{1, \cdots, k\}$;
while status = unconnected do
    if $a$ is connected by $u$ then
        remove AP $a$ from $N(u_i)$;
    if $N(u_i) = \emptyset$ then
        quit;
    $B_{min} \leftarrow \min_{a_j} B_{u_i}(a_j)$;
    $a_1 \leftarrow \arg\min_{a_j} B_{u_i}(a_j)$;
    if $B_{min} = 0$/*back-off timeout*/ then
        $r \leftarrow RANDOM(k)$;
        schedule connection to $a_1$ at the $r$-th slot from now;
    if $t \mod k = 0$ and $B_{u_i}(a_j) > 0$ then
        $B_{u_i}(a_j) \leftarrow B_{u_i}(a_j) - 1$ for $j \in \{1, \cdots, k\}$ ;
```

B. The 1-hop Knowledge Model

Although the back-off based sequential connection scheme can reduce the conflict probability, it cannot avoid conflict. In this subsection, we consider a conflict-free heuristic based on 1-hop knowledge. The basic idea of this heuristic can be simply described in the following steps: 1) each user selects the neighbor AP with the strongest signal strength; 2) each user informs his selected AP about RSSI through a message; 3) upon receiving messages from users, each AP sends feedback to inform those users that sends RSSIs to the AP about which user sensed the highest RSSI; 4) only the user that sensed the highest RSSI will connect to his selected AP. Unlike the greedy method in the 0.5-hop knowledge model, where each user connects to his selected AP, in this localized heuristic, a user connects to his selected AP only if he is the best user among all the users that also select that AP. The best user denotes the user that senses the highest RSSI from that AP.

Algorithm 3 Simultaneous Connection (u_i) in 1-hop Model

```
Input: ${a_1, a_2, \cdots, a_k}$
sense signal strength $S_{u_i}(a_j), j \in \{1, \cdots, k\}$;
$a \leftarrow \arg\max_{a_j} S_{u_i}(a_j)$; /*find the best AP*/
send message to $a_j$ about $S_{u_i}(a_j), j \in \{1, \cdots, k\}$;
if receive feedback from $a$ then
    if $u_i = \arg\max_u S_u(a)$ then
        connect to $a$;
```

The formal description of the heuristic is presented in Algorithm 3, where $S_u(a)$ represents the RSSI of a sensed by user u. Although this heuristic is relatively simple, it can achieve a constant approximation ratio under the reasonable assumptions that the RSSI of an AP sensed by a user is proportional to their distance, and both APs and users are uniformly deployed in the coverage area of the wireless network. Theorem 1 presents the average approximation ratio of our heuristic, which is the average number satisfied users by our heuristic to the maximum number of satisfied users by the optimal solution.

Theorem 1: If both APs and users are uniformly deployed in the coverage area, the average approximation ratio of Algorithm 3 is at least $1 - \frac{1}{e}$.

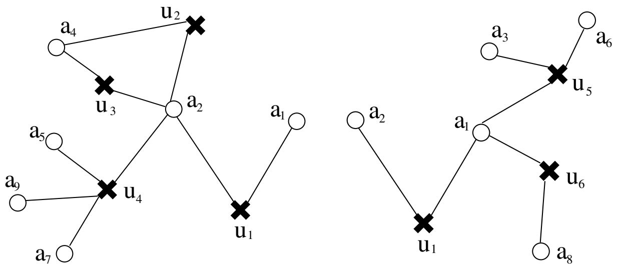
Due to space limitation, we omit the proof of this theorem. The above heuristic is conflict-free. Therefore, the performance of this heuristic can be further increased through iterative executions.

C. The 1.5-hop Knowledge Model

In this subsection, we consider the model where each user has 1.5-hop knowledge, i.e, users know not only their neighbor users but also those neighbor users' neighbor APs. For example, consider the example shown in Fig. 1. If u_1 has 1.5-hop knowledge, he knows not only his neighbor user u_6 but also u_6's neighbor AP a_8.

To obtain this 1.5-hop knowledge, we assume that each user needs to register at each neighbor AP beforehand. To register at a neighbor AP, a user not only provides his own information such as ID but also the RSSIs of his neighbor APs. After the registrations of all neighbor users, an AP notifies its neighbor users through a feedback message, containing the IDs of each neighbor user, the neighbor AP set of each neighbor user, and the associated RSSIs.

Based on the 1.5-hop knowledge, each user can construct a *local graph* for each neighbor AP. The local graph for an AP includes the AP itself, all of its neighbor users, and all of the neighbor APs of those users, i.e., the 1.5-hop knowledge of this AP. For example, Fig. 2 contains two local graphs for u_1's two neighbor APs, a_1 and a_2, as shown in Fig. 2 (a) and 2(b), respectively. An important property of the local graph for an AP is that the local graph is the common knowledge of the neighbor users of the AP.



(a) The local graph for a_2. (b) The local graph for a_1.

Fig. 2. The graphs constructed according to the knowledge that user u_1 has, as shown in Fig. 1. The local graph for a_2 is the common knowledge for users u_1, u_2, u_3, u_4, and u_5, while the local graph for a_1 is the common knowledge of users u_1, u_2, u_5, and u_6.

In this model, we assume that each user selects its AP in a greedy way, i.e., a user considers his first choice, the neighbor AP with the best signal strength. If its first choice is not available (i.e., connected by other users), a user will consider its second choice, and so forth.

1) Simultaneous Connection: We first consider the simultaneous connection. The formal description of the heuristic is in Algorithm 4, where the following additional notations are used: 1) Q_{u_i}: the priority queue of user u_i, which stores the neighbor APs of u_i according to their priority; 2) Q_a: the priority queue for AP a, which records the neighbor users of a and is maintained at the user side; 3) DEQUE($\cdot$): fetch the highest-priority element in the priority queue and remove this element from the queue thereafter.

Algorithm 4 Simu-Connection (u_i) in 1.5-hop Model

```
1:  $status_{u_i} \leftarrow$ unconnected;
2:  $a \leftarrow DEQUE(Q_{u_i})$;
3:  build local graph for $a$ if $a$ is not empty; otherwise,
    quit;
4:  construct $Q_a$ for users in the local graph;
5:  $u \leftarrow DEQUE(Q_a)$;
6:  while $u_i \neq u$ do
7:      $status_u \leftarrow$ connected;
8:      same operations as lines 8-11;
9:      while $status_u$ =connected do
10:         $u \leftarrow DEQUE(Q_a)$;
11: if $u_i = u$ ($u_i$ has the highest priority) then
12:     status $\leftarrow$ connected;
13:     establish Connection($u_i, a$);
```

In the above strategy, user u_i repeatedly constructs the local graph for his first choice, second choice, and so forth, until a connection can be established between a neighbor AP and u_i, or no neighbor AP is available. Lines 2-5 represent the set of operations that fetch the AP with the highest priority in user u_i's priority queue, construct the local graph for the AP if the AP exists, and find the users with the highest priority in the local graph. To determine whether a user can establish a connection to a neighbor AP, the user has to maintain two priority queues: one for the neighbor APs, and the other for the neighbor users in the corresponding local graph. The former is fixed in the strategy, while the latter is dynamic because the user dynamically constructs the local graph for his neighbor APs. In the local graphs for u_i's second choice, third choice, etc, if u_i finds that the users with higher priority has already connected to some other APs, u_i will ignore those users.

We should note that Algorithm 3 can be regarded as a special case of Algorithm 4 because users in Algorithm 3 consider the AP with the highest signal strength alone, while users in Algorithm 4 take into account all neighbor APs one by one in the order of the sensed signal strength. Therefore, the performance of Algorithm 4 should not be worse than that of Algorithm 3, and hence, the approximation ratio of Algorithm 3 can also be regarded as a low bound for Algorithm 4.

Although we provide additional chances for those users who lose in the competitions for their first choices, not every user can obtain the additional chances that belong to him because every user has only 1.5-hop knowledge. For example, in Fig. 2, the first choices of u_2, u_3 and u_4 are all a_2. The second choices of u_2 and u_3 are both a_4. The second choice of u_4 is a_7, which is the first choice of another user (not drawn in Fig. 2) and therefore be selected by another user. The third choices of u_3 and u_4 are both a_5. There is no collision if u_4 selects a_5 because u_3 has already taken a_4, but u_4 does not know it, and hence resigns from the competition for a_5.

2) Sequential Connection: The above problem of the simultaneous AP connection can be mitigated by iteratively running Algorithm 4. We assume that, at the end of each iteration, the matched user-AP pairs are the common knowledge for their neighbor users.

IV. Extension to the m-1 matching problem

With some modifications, the localized heuristics for the $1-1$ matching problem can also be extended to the $m-1$ matching problem. For the simultaneous AP connection in the 0.5-hop knowledge model, the users' strategies for the $1-1$ matching and the $m-1$ matching are the same. The only difference is the classification of the satisfied users and the unsatisfied users. In $1-1$ matching, if more than one users connect to the same AP, all users connecting to this AP are dissatisfied, while in $m-1$ matching, a user is unsatisfied if and only if the number of users connecting to the AP selected by the user is larger than the AP's threshold.

For the sequential AP connection in the 0.5-hop knowledge model, we need to change the lines $4-5$ in Algorithm 2 to the following pseudo codes, where we introduce two notations: 1) C_a: the counter to record the number of users connected to AP a; 2) T_a: the threshold of AP a, i.e., the maximum number of users that can connect to a and still meet the minimum quality requirement.

Algorithm 5 Modification (u_i)

```
if $C_a < T_a$ then
    $C_a \leftarrow C_a + 1$;
else if $C_a = T_a$ then
    remove AP $a$ from $N(u_i)$;
```

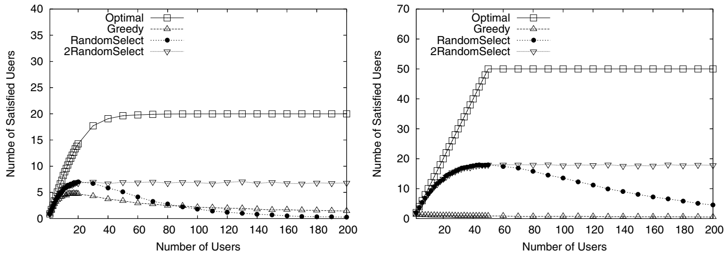
We also need to change the conflict condition to $C_a \geq T_a$, i.e., the number of users connected to AP a larger than its threshold. For the simultaneous AP connection in the 1-hop knowledge model, we need to modify the condition for a user to be able to establish a connection to a neighbor AP. In the $1-1$ bipartite matching problem, a user is able to establish a connection to a neighbor AP if and only if his priority is the highest among the neighbor users of the neighbor AP. In the $m-1$ bipartite matching problem, a user can connect t a neighbor AP if his rank is within the first T_a neighbor users of the neighbor AP. The modification for the simultaneous AP connection in the 1.5-hop knowledge model is the same as that for the simultaneous AP connection in the 1-hop knowledge model.

V. Simulation

In this evaluation, our concern focuses on two metrics: the total load and the number of rounds (for the sequentialized AP selection strategy only). We will consider the four localized heuristics for the $1-1$ matching model separately, and compare them with the optimal matching algorithms.

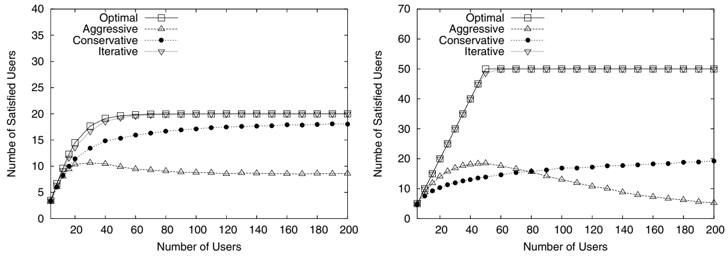
We simulate a stationary network with APs and users randomly located in a $100m \times 100m$ area. We assume all APs are of the same type, initially have the same transmission range, and can be deployed in this area arbitrarily. So are the users. In the simulation, we consider the following tunable parameters: (a) n, the number of users. (b) k, the number of APs, (c) r, transmission range of APs. (d) l, the number of slots per round in the back-off based heuristic.

One of the network configurations that affect the performance of the localized heuristics is the AP density, which can be reflected by parameters k and r. The greater the number of APs, or the larger the transmission ranges, the



(a) Networks with low AP density. (b) Networks with high AP density.

Fig. 3. The experiment on the localized heuristics for the simultaneous AP selection in the 0.5-knowledge model.



(a) Networks with low AP density. (b) Networks with high AP density.

Fig. 4. The experiment on the localized heuristics for the 1.5-hop user knowledge model.

higher the AP density. The AP density has a great impact on the performance of the localized heuristics. In the extreme case, if the transmission ranges are large enough so that each AP can cover the whole area, users can connect to any AP, and thus, the major concern is to avoid conflicts.

We consider two types of network configurations: low AP density and high AP density. The AP density can be reflected by parameters k and r. In the low AP density network, we set $k = 20$ and $r = 20m$. In the high AP density network, we set $k = 50$ and $r = 50m$. In any experiment, the performance of the heuristic will be compared with the optimal solution, which is from the augmentation-path based max-flow algorithm.

We first consider three variations of the localized heuristic for the simultaneous AP connection in the 0.5-hop knowledge model. The three variations are 1) the greedy method, where users connect to the neighbor AP with the highest priority, 2) the random-selection method, where users randomly connect to one of neighbor APs with equal probability, and 3) the double-random-selection method, where users first randomly determine whether they will be connecting to the network or not. If they determine that they will connect to the network, they will randomly determine which neighbor AP to connect to.

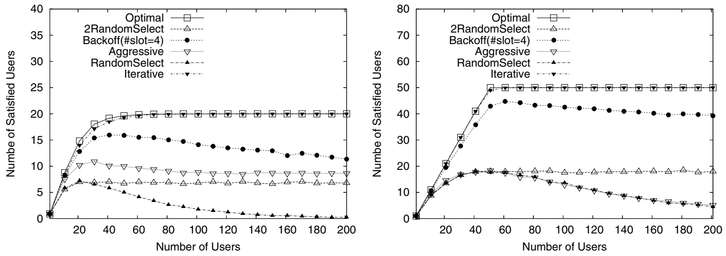
In the double-random-selection method, the probability to connect to the network depends on the ratio of the number of APs to the number of users. If the ratio is larger than 1, the probability of connection is 1; otherwise, the probability is equal to the ratio. Therefore, in the case where the number of APs is more than the number of users, the double-random-selection method is the same as the random-selection method.

Comparing the performance in the high AP density network (Fig. 3 (b)) and that in the low AP density network (Fig. 3 (a)), we find that it is easier for the greedy method to incur conflicts in the high AP density network. The reason is that each user has more choices in the high AP density network. All users tend to select the AP with the highest priority, which in turn causes the conflicts at the AP with the highest priority.

If the number of users is much more than the number of APs, the performance of the random-selection method decreases dramatically, and can be even worse than that of the greedy method. The reason is that if each user insists on connecting to an AP in the case where there are more users than the APs, it is highly possible users will connect to the same APs. In the greedy method, the conflicts occur at the APs with higher priority, while in the random-selection method, the conflicts can be anywhere.

Furthermore, we can see that the performance of the optimal method increases dramatically in the high AP density environment because each user can connect to almost every AP. We can also find that the performance of the double-random-selection method is better than that of the random-selection method when the number of users is larger than the number of APs because some users do not join the competition for APs, which mitigates the conflicts. Although our double-random-selection implementation requires global information, the number of users and APs, the estimation of the bandwidth of neighbor APs can be applied to determine the probability to connect the network.

In the second experiment, the results of which are shown in Fig. 4, we consider the heuristics for the 1.5-hop knowledge model. We compare three heuristics: the aggressive simultaneous-AP-connection heuristic (a variation of Algorithm 4, where a user connects to his second choice in the event that he cannot determine if it is safe to connect to neighbor APs other than the best AP without introducing conflicts), the conservative simultaneous-AP-connection heuristic (Algorithm 4), and the iterative heuristic based on the conservative simultaneous-AP-connection heuristic. When the number of users is relatively small, the aggressive heuristic has better performance than that of the conservative one because the probability of connecting an unoccupied AP is relatively high. However, when the number of users is much more than the number of APs,



(a) Networks with low AP density. (b) Networks with high AP density.

Fig. 5. The experiment on all localized heuristics for both 0.5-knowledge model and the 1.5-hop user knowledge model.

the conflicts caused by the aggressive heuristic cancel its benefit, i.e., enabling more users (either satisfied or unsatisfied) to connect to APs. We also observe that the number of satisfied users in the iterative heuristic is almost the same as that of the optimal solutions. The reason is that the conservative heuristic does not incur conflicts, and by repeatedly running the conservative heuristic, it is highly possible that each unoccupied AP will be connected by one user, especially when the number of users is large.

In the third experiment, the results of which are shown in Fig. 5, we select one or two representative heuristics from each model and compare their performances. Among the five heuristics, the iterative heuristic in the 1.5-hop knowledge model has the best performance, while the back-off heuristic in the 0.5-hop knowledge model has the second to the best performance. The performance difference between these two heuristics is expected because the former heuristic has more knowledge.

VI. Related Work

Currently, most of the IEEE 802.11 protocols adopt the received signal strength indicator (RSSI) based approach in order to select an AP to affiliate with. Previous works [1], [3], [6] have shown that the RSSI-based approach can lead to poor performance in terms of the total load. To address this problem, existing works [1], [3], [6], [9], [10] have proposed numerous approaches. All of them are essentially various AP access control schemes. Most of these approaches were evaluated through simulations or experiments on test beds. To the best of our knowledge, there are only two types of algorithms that provide performance guarantees. One is based on the linear programming (LP) approach [5], and the other is based on the simulated annealing technique [7].

However, the LP-based approach is centralized, which cannot adapt to the self-organized wireless LANs. Although the simulated annealing technique and the primal-dual scheme (derived from the LP) can be implemented in a localized manner, both of them require the propagation of global information and a large number of rounds in order to converge. This is not desirable in an environment with a highly dynamic user population. Therefore, we propose a class of pure, localized algorithms, parts of which can provide performance guarantees and do not require the propagation of global information.

VII. Conclusions

In this work, we model the AP selection in 802.11 wireless networks as the $m-1$ matching problem in the bipartite graph. We analyze the user knowledge and the timing of the AP selection on the performance, and propose several localized heuristics adaptable to the existing 802.11 wireless networks. We adopt the 0-1 function to classify satisfied users and unsatisfied users. That is, when the number of users connected to an AP exceeds the some limit, all users become unsatisfied. In the future, we will explore other possible functions to characterize the satisfaction of wireless users.

Acknowledgement

This work was supported in part by NSF grants ANI 0073736, EIA 0130806, CCR 0329741, CNS 0422762, CNS 0434533, CNS 0531410, and CNS 0626240.

References

[1] A. Balachandran, P. Bahl, and G. Voelker. Hot-spot congestion relief and service guarantees in public-area wireless networks. *SIGCOMM Computer Communication Review*, 32(1), 2002.

[2] 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 ACM SIGMETRICS'02*, pages 195–205, 2002.

[3] Y. Bejerano, S. Han, and L. Li. Fairness and load balancing in wireless lans using association control. In *Proceedings of ACM MobiCom'04*, 2004.

[4] T. Cormen, C. Stein, R. Rivest, and C. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

[5] M. T. Hajiaghayi, S. V. Mirrokni, A. Saberi, P. Bahl, K. Jain, and L. Qiu. Cell breathing in wireless LANs: Algorithms and evaluation. *IEEE Transactions on Mobile Computing*, 6(2):164–178, 2007.

[6] G. Judd and P. Steenkiste. Fixing 801.11 access point selection. In *Poster in Proceedings of ACM MobiCom'02*, 2002.

[7] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot. Measurement-based self organization of interfering 802.11 wireless access networks. In *Proceedings of INFOCOM'07*, 2007.

[8] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. *Wireless Network*, 11(1-2):115–133, 2005.

[9] R. Murty, E. Qi, and M. Hazra. An adaptive approach to wireless network performance optimization. Technical report, Cororate Technology Group (CTG), Intel Corporation, 2004.

[10] S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley. Facilitating access point selection in IEEE 802.11 wireless networks. In *Proceedings of the 5th ACM SIGCOMM conference on Internet measurement (IMC'05)*, pages 293–298, 2005.