WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2011; **11**:994–1010
Published online 8 June 2009 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.800

SPECIAL ISSUE PAPER

Localized access point selection in infrastructure wireless LANs with performance guarantee

Mingming Lu[1]* and Jie Wu[2]

[1] School of Information Science and Engineering, Central South University, Changsha, China
[2] Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, U.S.A.

ABSTRACT

Efficient network management can improve the network performance and reduce the cost of system maintenance and administration. One of the important duties assigned to the network management system in a infrastructure wireless LAN (or simply wireless LAN) is to assign users to appropriate accessible access points (APs). The current AP selection schemes in wireless LANs cause an unbalanced load, which in turn reduces the performance of individual users. This has motivated intensive studies attempting to determine efficient methods to balance loads among different APs. Existing works either provide heuristic solutions without performance guarantees or provide centralized solutions which are not desirable for the AP selection problem. In this paper, we study the localized solutions that can provide performance guarantees. We model the AP selection problem as a matching problem in bipartite graph. Our objective is to maximize 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 mathematical analysis. Simulations are conducted to verify our results. Copyright © 2009 John Wiley & Sons, Ltd.

KEYWORDS
access point selection; bipartite matching; network management; wireless LANs

***Correspondence**
Mingming Lu, Computer Science Building, Room 301, Central South University, Changsha, Hunan, Post code: 410083, China.
E-mail: ming.lu@gmail.com

1. INTRODUCTION

With the proliferation of wireless LANs, users can easily obtain access to multiple APs. One of the important duties assigned to the network management system in a wireless LAN is to assign users to appropriate accessible access points (APs). 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] 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] 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 [4], and the other is based on the simulated annealing technique [5].

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 for an environment with a highly dynamic user population. Therefore, we propose a set of purely localized algorithms that provide performance guarantees and do not require the propagation of global information.

In these localized algorithms, we consider 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 different users. We model the AP selection problem as a many-to-one matching problem in the bipartite graph, where each user can connect to only one AP, while an AP can be connected to multiple users. The objective of our problem is to maximize the network-wide throughput, i.e., the total load on all APs. Although it is well known that optimal bipartite matching can be computed in polynomial time, the polynomial-time algorithms require either a centralized node to execute, or the propagation of global information.

We first consider the case that all users have the same demand. In this case, the system objective can be reduced to the total number of users whose demands are satisfied. In the case of the homogenous user demand, 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 many-to-one 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 localized heuristic. Then, we extend the results from 1–1 matching model to the many-to-one matching model. We then consider the case that users have different demands. We consider our localized algorithms in two more realistic environments: (1) taking into account the effect of interference by incorporating an interference model and (2) adjusting the localized algorithm for use within a dynamical population where users can join and/or leave the network.

To sum up, the key contributions of our work are as follows: (1) We fill the gap between the existing centralized algorithms and the existing localized heuristics by proposing a localized algorithm with a constant average approximation ratio. (2) We study the effect of knowledge and the time of AP selection on the performance in terms of the total load. (3) We further improve the performance of our localized algorithm by proposing an iterative localized algorithm and prove the bound of iterations. (4) We extend our localized algorithm to handle various dynamic environments and more realistic network settings. (5) We evaluate our algorithms through extensive simulations.

The remainder of this paper is organized as follows. Section 2 introduces models and notations, and formalizes the problem. In Section 3, 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 4 extends our results to the many-to-one matching problem. Section 5 further extends our work to the heterogenous user demand. Section 6 empirically evaluates performance through our customized simulator. Section 7 presents related works. Finally, Section 8 concludes this work and outlines our future work. Proof of all results appear in the Appendix.

2. PRELIMINARIES

2.1. The network model

A set of APs and a set of wireless users compose a 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 [6,7].

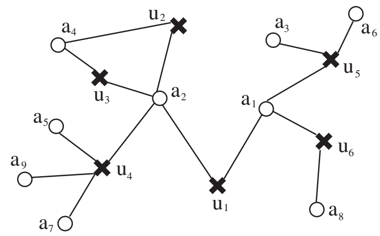

Figure 1. The graph constructed according to u_1's 1.5-hop information.

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 that two users are 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 knowledge that user has. The basic information for a user is the information about his neighbor AP. Without this information, a user can do nothing. 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 about 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 Figure 1, where users are represented by crossed and APs are represented by circles, 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 Figure 1.

The reason that we consider 1-hop knowledge and the 1.5-hop knowledge models is that the 0.5-hop knowledge alone is not useful enough to help users self-distribute the load among APs. To make the collection of the 1-hop knowledge practical, we can integrate the collection of the 1-hop knowledge into the AP selection process. In Section 5, we will see that it requires only a simple admission control mechanism at the AP side to implement the integration of the collection of the 1-hop knowledge. Although the collection of the 1.5-hop knowledge requires an additional round of information exchange, we adopt the 1.5-hop knowledge model as a comparison on how user knowledge can help improve performance.

2.2. 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. To simplify the model, we first assume that users are homogeneous, and hence, users have the same minimum requirement to the quality such as bandwidth, delay and etc. We will relax this assumption later. We also assume that there is a limitation on the capacity of each AP. 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 *many-to-one matching problem* in the bipartite graph. In the many-to-one 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 [8] 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.

3. SOLUTIONS TO THE 1–1 MATCHING PROBLEM

We first consider a special case of the many-to-one 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 mentioned in Section 2, we consider two different AP-selection orders: simultaneous connection and sequential connection.

3.1. 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.

Simultaneous connection. The schemes based on simultaneous connection 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 highest RSSI. In Algorithm 1, we formally present a generic user strategy for simultaneous connection in the 0.5-hop knowledge model. Without loss of generality, we assume user u has k neighbor APs denoted as $N_u = \{a_1, a_2, \ldots, a_k\}$, where u connects to a_j with probability $p(a_j)$.

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

Initialize the set of neighbor APs $N_u \leftarrow \{a_1, a_2, \cdots, a_k\}$
1: Calculate $p(a_j)$ for every $a_j \in N_u$
2: 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 highest RSSI 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 randomly with equal probability. In this generic scheme, there exists a conflict incurred by multiple users connecting to the same AP at the same time. In that case, those users can reduce their connection probabilities to reduce the conflict. For example, if the number of users that select an AP is l, each of those users can set his connecting probability to $\frac{1}{l}$. A trade-off exists between the conflict probability and the connection probability.

Sequential connection. To reduce the conflict mentioned previously, we consider a back-off-based sequential-connection scheme. In the sequential-connection scheme, users that compete for the same AP in the simultaneous connection can reduce the probability of conflict because they may connect to the AP in different orders and the users that connect later can be informed the unavailability of the AP through acknowledgement from AP, which cause the users to resign from competition. Note that this information also belongs to the category of 0.5-hop knowledge.

We assume that the time consists of multiple continuous time units, each of which consists of l slots. Initially, each user sets a back-off counter for each neighbor AP. The value of the counter (in terms of the number of time units) associated with each AP is reversely proportional to the RSSI of the AP. The counter of each neighbor AP will decrease by 1 for every time unit (l slots). At the beginning of every time unit, if any counter becomes 0, a user randomly picks one of the l slots within the time unit 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 that this AP is occupied. Those unconnected neighbor users will remove the AP from their neighbor AP sets. Note that we use l slots as the basic time unit in order to control the conflict. In the back-off scheme, it is still possible that multiple users will connect to the same AP. By setting l to a large value, the probability of conflict can be further reduced at the cost of extended delay. We will study the proper value of l in the experiment study.

The formal description of the back-off-based sequential-connection method employed by each user is presented in Algorithm 2, where we assume that each user initially has at least one neighbor AP, and $B(a_j)$ is used to denote the back-off counter for neighbor AP a_j. Another possible sequential connection scheme is to iteratively running the simultaneous connection scheme if there exist unconnected users or unsatisfied users. For those unsatisfied users (the

Wirel. Commun. Mob. Comput. 2011; **11**:994–1010 © 2009 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

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

Initialize the set of neighbor APs $N_u \leftarrow \{a_1, a_2, \cdots, a_k\}$
Initialize back-off counter $B(a_j)$ for every $a_j \in N_u$;
while u is unconnected and N_u is not empty **do**
  if Receive ConnAck from a_j **then**
    Remove a_j from N_u if u is not the intended user
  Quit the connection process if N_u is empty;
  if all $B(a_j) > 0$ at the beginning of a time unit **then**
    Reduce $B(a_j)$ by 1 for every $a_j \in N_u$;
  else if $B(a^*) = 0$ for an $a^* \in N_u$ **then**
    Randomly pick one slot in the current time unit;
    Connect to the a^* in the picked time slot;

users that connect to the same AP), we assume that it takes a relatively long time (compared with the time unit of the back-off-based scheme) to resolve the conflict, which involves the identification of the conflict, disconnection of all users connected to the same AP, and re-selection of AP.

3.2. 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 the 1-hop knowledge model. In the conflict-free heuristic, we assume that each user not only knows the RSSI of each neighbor AP, but also can obtain information whether it is the best user. The best user to an AP denotes the user that senses the highest RSSI among all neighbor users of that AP. Each user only considers the neighbor AP with the highest RSSI (the best AP), and each user connects to his best AP only if he is also the best user to the AP.

In the following, we integrate the collection of the 1-hop knowledge into the conflict-free localized heuristic: (1) each user selects the best AP; (2) each user sends a connection request to his best AP and attaches the corresponding RSSI; (3) after receiving connection requests within a timeout, each AP sends back a connection acknowledgement to the best user; (4) only the best user will connect to his best AP.

Unlike the greedy method in the 0.5-hop knowledge model, where each user directly connects to his best AP, in this localized heuristic, a user connects to his best AP only if he is the best user with respect to his best AP. The formal description of the heuristic is presented in Algorithm 3,

Algorithm 3 SimuConnect (u) for the basic 1-hop Model

Initialize $N_u \leftarrow \{a_1, a_2, \cdots, a_k\}$
Sense the RSSI $S_u(a_j)$ for every $a_j \in N_u$;
Find the best AP (the AP with the highest RSSI, denoted as a^*);
Send connection request to a^* about $S_u(a^*)$;
if Receive connection acknowledgement from a^* **then**
  if u is the best user with respect to a^* **then**
    Connect to a^*;

where $S_u(a)$ represent the RSSI of a sensed by user u. Although this heuristic is relatively straightforward, it can achieve a constant approximation ratio under the reasonable assumptions that the value of $S_u(a)$ is proportional to the distance between a and u, and all the APs and users are uniformly deployed in the coverage area of a wireless LAN. Theorem 1 presents this theoretical result on the average approximation ratio of this conflict-free localized heuristic, which is the ratio of the average number satisfied users by our heuristic to the maximum number of satisfied users by the optimal solution.

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

Based on the assumption that RSSI $S_u(a)$ is proportional to the distance between AP a and user u, we can partition the covered area into numerous Voronoi regions, within each of which exact one AP serves as the Voronoi point. Hence, there will be exactly one AP in each Voronoi region. Figure 2(a) gives an example of a Voronoi graph with six APs and eight users. Users in each Voronoi region will only select the AP in the same region because this AP has the highest RSSI among all neighbor APs. Figure 2(b) shows the AP-user matchings produced by Algorithm 3. No user connects to a_3 or a_6 because no user is within their regions.

Under the assumption that both users and APs are uniformly distributed, the probability of overload (occurring when the number of users within a Voronoi region is larger than the capacity of the AP in the same Voronoi region) is relatively small. Hence, our localized heuristic has a high average approximation ratio.

The above heuristic is conflict-free. Therefore, the performance of this heuristic can be further improved through iterative executions. Besides iterative executions, increasing user and AP knowledge can help improve performance. If we require each user to send RSSIs to every neighbor AP instead of the best AP alone, we can further improve the performance. The improved algorithm for the user side is formally described in Algorithm 4. The AP-side algorithm is the same as that of Algorithm 3, i.e., an AP simply replies an acknowledge message to his best user. Comparing Figure 2(b) and (c), we find that one more AP-user pair can be matched through Algorithm 4. The

Algorithm 4 SimuConnect (u) for improved 1-hop Model

Initialize $N_u \leftarrow \{a_1, a_2, \cdots, a_k\}$
Sense RSSI $S_u(a_j)$ for every $a_j \in N_u$;
Inform a_j about $S_u(a_j)$ for every $a_j \in N_u$;
repeat
  if Receive feedback from a_j **then**
    Put a_j into u's available AP set;
until Receive feedback from all a_j or timeout
if the available AP set is not empty **then**
  Connect to the AP with the highest RSSI among the available AP set

Wirel. Commun. Mob. Comput. 2011; **11**:994–1010 © 2009 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

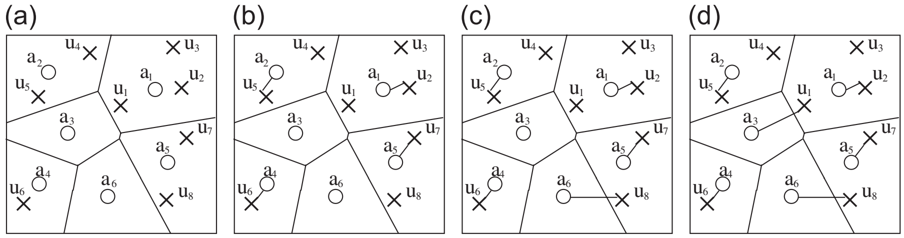

Figure 2. The example Voronoi graph with Voronoi regions centered at APs. There are six APs, represented by circles, and eight users, represented by crossings. The AP-user matchings generated by Algorithm 3, Algorithm 4, and the iterative execution of Algorithm 3 are shown in (b), (c), and (d), respectively. AP-user matching is represented by a link between the AP and the user.

reason for this is that some empty Voronoi regions (Voronoi regions without users) can borrow users from neighboring Voronoi regions. However, it is still possible that some AP-user pairs cannot be matched due to insufficient knowledge. For example, in Figure 2(c), a_3 and u_1 cannot be matched since the best user of a_3 is u_5, which is also the best user of a_2, but u_1 has no knowledge about this. This can be solved by iteratively running Algorithm 3.

3.3. 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 Figure 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, Figure 3 contains two local graphs for u_1's two neighbor APs, a_2 and a_1, as shown in Figure 3(a) and (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.

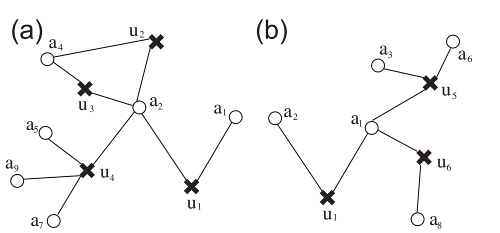

Figure 3. The graphs constructed according to the knowledge that user u_1 has, as shown in Figure 1. The local graph for a_2 is the common knowledge for users u_1, u_2, u_3, u_4, and u_5 (a), while the local graph for a_1 is the common knowledge of users u_1, u_2, u_5, and u_6 (b).

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 highest RSSI. If its first choice is not available (i.e., connected by other users), a user will consider its second choice, and so forth.

Simultaneous connection. We first consider simultaneous connection. The formal description of the heuristic is in Algorithm 5, where 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 1–3 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

Algorithm 5 Simu-Connection (u) in the 1.5-hop Model

1: Fetch and remove the AP with highest RSSI (denoted as a^*) from u's neighbor AP set;
2: Build local graph for a^* if a^* exists; otherwise, quit the AP connection process;
3: Fetch and remove the user (denoted as u^*) that senses the highest RSSI of a^* from a^*'s local graph;
4: **while** $u \neq u^*$ **do**
5: Mark u^*'s status as connected;
6: Do the same operations as lines 1-3;
7: **while** u^* status is connected **do**
8: Fetch and remove u^* from a^*'s local graph;
9: **if** $u = u^*$ (u has the highest priority) **then**
10: Connect to a^*;

 Wirel. Commun. Mob. Comput. 2011; **11**:994–1010 © 2009 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

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.

It should be noted that Algorithm 3 can be regarded as a special case of Algorithm 5 because users in Algorithm 3 consider the AP with the highest RSSI alone, while users in Algorithm 5 take into account all neighbor APs one by one in decreasing order of their RSSIs. Therefore, the performance of Algorithm 5 should be no worse than that of Algorithm 3, and hence, the approximation ratio of Algorithm 3 can also be regarded as a low bound for Algorithm 5.

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.

Sequential connection. The problem with the simultaneous AP connection can be mitigated by iteratively running Algorithm 5. We assume that, at the end of each iteration, the matched user-AP pairs are the common knowledge for their neighbor users.

4. EXTENSION TO THE MANY-TO-ONE MATCHING PROBLEM

With some modifications, the localized heuristics for the 1–1 matching problem can also be extended to the many-to-one matching problem. For the simultaneous AP connection in the 0.5-hop knowledge model, the users' strategies for the 1–1 matching and the many-to-one 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 many-to-one matching, a user is unsatisfied if and only if the number of users connecting to the same AP exceeds the AP's threshold.

For the sequential AP connection in the 0.5-hop knowledge model, we need to change the condition for a user to remove an AP from his neighbor AP set. In the 1–1 matching, the condition is that the AP is occupied by another user. In the many-to-one matching, it becomes that the number of users connecting to the AP is larger than the AP's threshold. We also need to change the conflict condition to state that the number of users connecting to the AP must be 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 many-to-one bipartite matching problem, a user can connect to a neighbor AP if his connection will not exceed the neighbor AP's threshold. In Theorem 1, we have proved that Algorithm 3 for the 1–1 bipartite matching has a constant approximation ratio. For the many-to-one bipartite matching problem, we also have a similar result.

Theorem 2. *If all APs and users are uniformly deployed in the coverage area, the approximation ratio of the one-round localized algorithm for the many-to-one bipartite matching is at least* $1 - \frac{1}{e}$.

The one-round localized algorithm for the 1-hop knowledge model can be further improved through iterative execution. For the iterative execution, a major concern is the convergence rate. Here, the convergence rate is defined as the number of rounds. The fewer the number of rounds, the higher the convergence rate. We have the following theorem concerning the upper bound for the convergence rate of our iterative localized algorithm.

Theorem 3. *Our iterative selection executes at most* $\ln l$ *rounds, where* $l = \min\{m, n\}$*, and m and n are the number of users and APs, respectively.*

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.

Non-uniform user distribution. Although our theoretical results are based on the assumption of the uniform user distribution, these results also hold in stochastic distributions other than uniform distribution. Previous study [1] has shown that users tend to access network at certain popular spaces ('hot-spots') within the network. In our experiment study, we adopt normal distribution to model such hot-spots. We use the user location as random variable, and the location of those hot-spots as the mean value of the normal distribution. According to the guidelines [9,10], APs should be distributed based on the number of users, the physical aspects of locations and etc. Thus, we adopt the normal distribution that is identical to the user distribution to deploy APs so that the ratio of users to APs are approximately the same. Through experiment evaluation, we identify that both our one-round and iterative localized algorithms still have the property of constant approximation ratio, and the convergence rate is still very fast.

5. EXTENSION TO HETEROGENOUS USER DEMAND

In previous sections, we assumed that users have homogenous demands by normalizing the demand of each user to 1. In this section, we relax this assumption by allowing users to have heterogenous demands. We first describe the needed modifications that enable the above localized algorithms to adapt to the case of the heterogenous demand. Then, we evaluate our localized algorithms in the following practical environments: (1) APs can adjust their transmission power, (2) the effect of interference on the AP capacity is incorporated, (3) users randomly enter or leave the network.

5.1. Adaption to the heterogenous user demand

The modifications consist of two parts: the operations at the user side (Algorithm 6) and the operation at the AP side (Algorithm 7). The localized algorithms for the heterogenous-user-demand are different from those for the homogenous-user-demand in that users estimate the potential bandwidths of neighbor APs, and regard the neighbor APs with higher estimated bandwidth than their demands as their available APs. Each AP adds users into its priority queue, and push users in the order of their corresponding RSSIs until the demand of the last user cannot be satisfied. Unlike Algorithm 3, which requires the collection of the 1-hop knowledge through one round of information exchange, Algorithms 6 and 7 implement the same function by utilizing an admission control at the AP side, which reduces the information collection.

Algorithm 6 Bandwidth-based Operations at User Side (u)

neighbor APs $N_u \leftarrow \{a_1, a_2, \cdots, a_k\}$

APScan(u);
Find the best AP a^* (in terms of the highest RSSI);
Send connection request to a^*;
if Receive connection acknowledgement from a^* **then**
    Connect to a^*;

APScan(u)

for each $a_j \in N_u$ **do**
    Sense a_j's RSSI and estimate a_j's remaining capacity;
    Add a_j into the available AP set if a_j's remaining capacity is larger than u's request;

Algorithm 7 Bandwidth-based Operations at AP Side (a)

Order the users that sends connection request to a according to their RSSIs of a;
Fetch and remove the user u that has the highest RSSI;
while a's remaining capacity is larger than u's demand **do**
    Include u into the set of users whose connection requests are accepted;
    Reduce a's remaining capacity by the amount of u's demand;
    Fetch and remove the user u that has the highest RSSI from the remaining users;
Send connection acknowledge to the set of accepted users;

If the value of user demands are discrete and can be split, the theoretical analysis for the homogeneous demand can be applied to the heterogenous demand by splitting the demand from a user with D units of demand into D users of unit demand. Therefore, for the splittable heterogeneous demand, there also exists a constant approximation ratio. For the non-splittable heterogenous demand, we evaluate the performance of our localized algorithm through experiment study. Through simulation, we observe that our one-round localized algorithm still has a constant approximation ratio, and the iterative algorithm still converges fast, with the performance approximating the optimal solution.

5.2. Transmission power adjustment

In previous sections, we assumed that the transmission power of all APs are fixed, and thus, could not be adjusted. In this subsection, we relax this assumption. We assume that each AP initially set its transmission power to its maximum value. After users select their intended APs (the APs with the highest RSSIs), each AP can calculate the number of users that intend to connect to it and the total amount of demands. If the amount of demand is larger than an AP's capacity, the AP will decrease its transmission power to the extent that the amount of demands is just smaller than or equal to its capacity. Note that by decreasing the transmission power, an AP decreases the size of its associated Voronoi region, which in turn decreases the number of users that send the request to the AP.

The transmission power adjustment can be integrated into the AP selection process as follows:

1. Initially, each AP sets its transmission power to its maximum value, and each user sends connection request to the AP with the highest RSSI.
2. Repeat the follow procedure until the demands from all users are satisfied or all APs are completely utilized:
   (a) If an AP receives connection requests from users, it accepts the requests when its remaining capacity is larger than or equal to the total amount of the requests. Otherwise, it sorts the users that are connected or request to connect in the decreasing order of their RSSIs. The AP accepts the users as many as possible in the decreasing order and sets its transmission power accordingly.
   (b) If a user's demand cannot be satisfied, it marks the AP that it has sent request as *unavailable*, and tries to connect to the AP with the highest RSSI among the remaining APs.

To reduce the number of switch among users between different APs, we assume that a user will not switch to another AP unless its RSSI from a new AP is improved to a threshold value.

5.3. Incorporate interference

In this subsection, we consider the effect of the interference. We adopt the interference model proposed in Reference [11], where, for a user u_i that connects to an AP a_j, the signal to interference ratio (SIR) at u_i is as follows:

$$\text{SIR}_{u_i} = \frac{S_{u_i}(a_j)}{\sum_{k \neq j} S_{u_i}(a_k)}$$

where a_k is an AP other than a_j in the covered area, and $S_{u_i}(a_j)$ is independent, exponentially distributed random variable. The mean value of $S_{u_i}(a_j)$ is $c \cdot P_{a_j}/d(a_j, u_i)^\alpha$, where P_{a_j} is the transmission power of AP a_j, $d(a_j, u_i)$ is the distance between them, c is a constant, and $\alpha = 2$ or 4. We assume that reception can occur provided the SIR exceeds a given threshold SIRth.

The error probability of transmission from a_j to u_i is given by

$$\text{Prob}(\text{SIR}_{u_i} \leq \text{SIRth}) = \text{Prob}(S_{u_i}(a_j) \leq \text{SIRth} \sum_{k \neq j} S_{u_i}(a_k))$$

The transmission error probability can be expressed in analytical form

$$\text{error}_{u_i} = 1 - \prod_{k \neq j} \frac{1}{1 + \text{SIRth}\, d(a_j, u_i)^\alpha / d(a_k, u_i)^\alpha}$$

The above analytic expression for the transmission error probability is based on the assumption that all APs transmit at the same power, and is derived from the following result: Suppose $z_1, \ldots, z_n$ are independent exponentially distributed random variables with means $Ez_i = 1/\lambda_i$, Then we have

$$\text{Prob}\left(z_1 \leq \sum_{i=2}^{n} z_i\right) = 1 - \prod_{i=2}^{n} \left(\frac{1}{1 + \lambda_1/\lambda_i}\right)$$

Due to transmission error, not all packets transmitted from AP a_j can be received by user u_i. We assume that undelivered packets incur retransmissions. As proposed in Reference [12], for any packet, the expected number of transmissions from a_j to u_i is $\frac{1}{1-\text{error}_{u_i}}$. Since retransmissions consume bandwidth assigned by a_j, we should count the amount of bits delivered to u_i every second as the bandwidth. We called this bandwidth as *effective bandwidth*. If AP a_j assigns bandwidth B to user u_i, the effective bandwidth of u_i is $B(1 - \text{error}_{u_i})$. By taking interference into account, our objective turns into maximizing the total effective bandwidth.

Through simulations, we verify that the approximation ratio still holds even when the interference is taken into account.

5.4. Dynamic user population

In the previous sections, we have considered the localized algorithms for the static network topology. In this subsection, we study dynamic environments where users can join and/or leave the network. When a user leaves the system, the load at the corresponding AP decreases, potentially motivating other users to change their previous selection and associate with that AP. When a user joins, he automatically selects an AP to affiliate with.

Our objective in this subsection is to discuss the impact of dynamic user joining and/or leaving on the system performance. When users join or leave the network frequently, it is desirable to design a user selection scheme without requiring many users to handoff to different APs, because the overhead of handoff is non-negligible. We modify the existing localized algorithms for static user populations by requiring that a user will not switch to a different AP unless the RSSI from a new AP is improved by a threshold. In Section 6, we use simulation to investigate the behavior of the system under a dynamic user population. We adopt the Poisson process to simulate the inter-arrive time as well as the service time of users.

6. SIMULATION

6.1. Simulation setup

In this section, we give an experiment evaluation of our localized algorithms. Our concern focuses on two metrics: the total load and convergence rate (in terms of the number of rounds). We first consider the case of homogeneous user demand, and then consider the case of heterogeneous user demand. In both cases, we compare the performance of our one-round localized algorithm, the iterative localized algorithm, and the optimal solution. We adopt the Karp's MaxFlow algorithm [8] to compute the optimal bipartite matching. For the case of homogeneous user demand, we first consider the localized heuristics for the 1–1 matching model.

We simulate a stationary network with APs and users randomly located in a 100×100 area. We assume all APs are of the same type, have the same transmission range, and can be deployed in the area arbitrarily. So are the users. In the simulation, we consider the following tunable parameters: (a) m, the number of users, (b) n, the number of APs, (c) r, transmission range of APs, (d) l, the number of slots per round in the back-off-based heuristic, (e) c, the maximum capacity of APs (the capacity of each AP is a random number between 1 and c), (f) d, the user demand. In the case of homogenous user demand, $d = 1$. In the case of heterogenous user demand, d is the maximum user demands. User demand of each user is random number ranging from 1 to d.

6.2. Simulation results

One of the network configurations that affect the performance of the localized heuristics is the AP density, which can be reflected by parameters n and r. The greater the number of APs, or the larger the transmission ranges, the higher the AP density. The AP density has a great impact on the performance of the localized heuristics. In the extreme

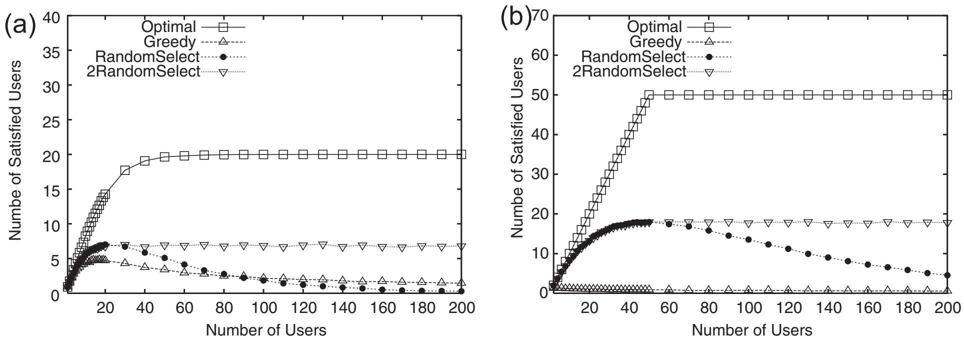

Figure 4. The experiment on the localized heuristics for the simultaneous AP selection in the 0.5-knowledge model. (a) Networks with low AP density. (b) Networks with high AP density.

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: the low AP density and the high AP density. The AP density can be reflected by parameters n and r. In the low AP density network, we set $n = 20$ and $r = 20$. In the high AP density network, we set $n = 50$ and $r = 50$. 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 (Figure 4(b)) and that in the low AP density network (Figure 4(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 Figure 5, 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 5, where a user connects to his second choice in the event that he cannot determine if it is safe to do so without introducing conflicts), the conservative simultaneous-AP-connection heuristic (Algorithm 5), 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, the conflicts caused by the aggressive heuristic offset 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 Figure 6, we select one or two representative heuristics

 Wirel. Commun. Mob. Comput. 2011; **11**:994–1010 © 2009 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

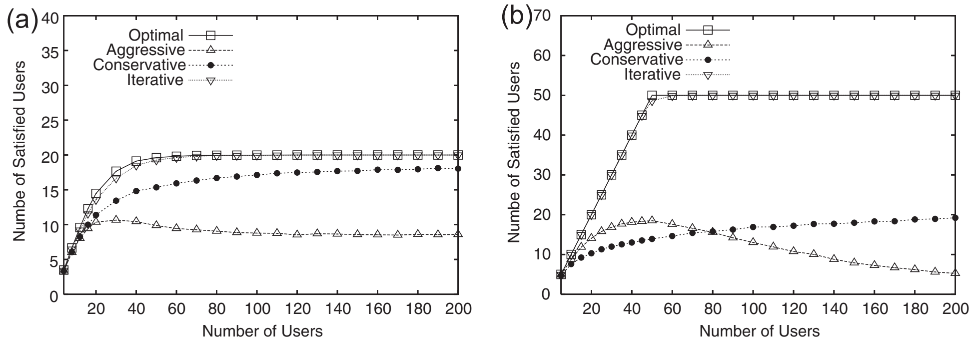

Figure 5. The experiment on the localized heuristics for the 1.5-hop user knowledge model.(a) Networks with low AP density. (b) Networks with high AP density.

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.

In the fourth experiment, we verify the approximation ratio of our one-round localized algorithm in the case of homogeneous user demand by testing four dimensions: the AP transmission range r, the number of users m, the number of APs n, and the maximum AP capacity c. The simulation result of the first dimension (r) is shown in Figure 7(a), where $m = 40$, $n = 30$, $c = 2$, and r increases from 10 to 100 in increments of 10. From this result, we observe that the performances of our one-round localized algorithm and the optimal solution increase quickly, as r increases from 10 to 30, and the ratio of the performances becomes approximately a constant if $r \geq 40$. The performance of the greedy scheme reaches its maximum at $r = 15$, and remains almost the same at $r \geq 30$. The reason for the short increments of performance is that the area is not fully covered when $10 \leq r \leq 40$, which in turn incurs uncovered users. Therefore, we can conclude that the AP transmission range has little effect on the performance of the three algorithms when it is greater than a threshold value, which is 40 in our experiment.

The simulation result of the second dimension (m) is shown in Figure 7(b), where $r = 40$, $n = 20$, $c = 3$, and m increases from 10 to 150 in increments of 10. From this result, we observe that the performance of the greedy scheme decreases dramatically as the number of users increases. This performance decrement occurs before the entire network overloads. The fundamental reason is that the greedy scheme easily causes locally unbalanced loads, which in turn decreases the performance of the entire network. On the contrary, the performance of our one-round localized algorithm becomes closer to that of the optimal solution as the number of users increases. The reason is that our one-round localized algorithm is conflict-free, and the number of the empty AP-centered Voronoi regions decreases as the number of users increases.

The simulation result of the third dimension (n) is shown in Figure 7(c), where $r = 40$, $m = 80$, $c = 1$, and n increases from 10 to 150 in increments of 10. From this result, we observe that the increment of the number of APs increases the performances of all three algorithms. That is because the increment of the number of APs increases the network capacity. The simulation result of the fourth dimension (c) is shown in Figure 7(d), where $r = 40$,

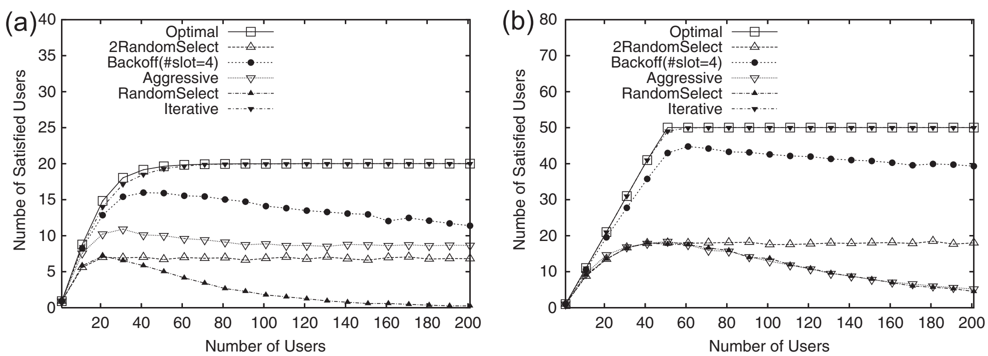

Figure 6. The experiment on all localized heuristics for both 0.5-knowledge model and the 1.5-hop user knowledge model.(a) Networks with low AP density. (b) Networks with high AP density.

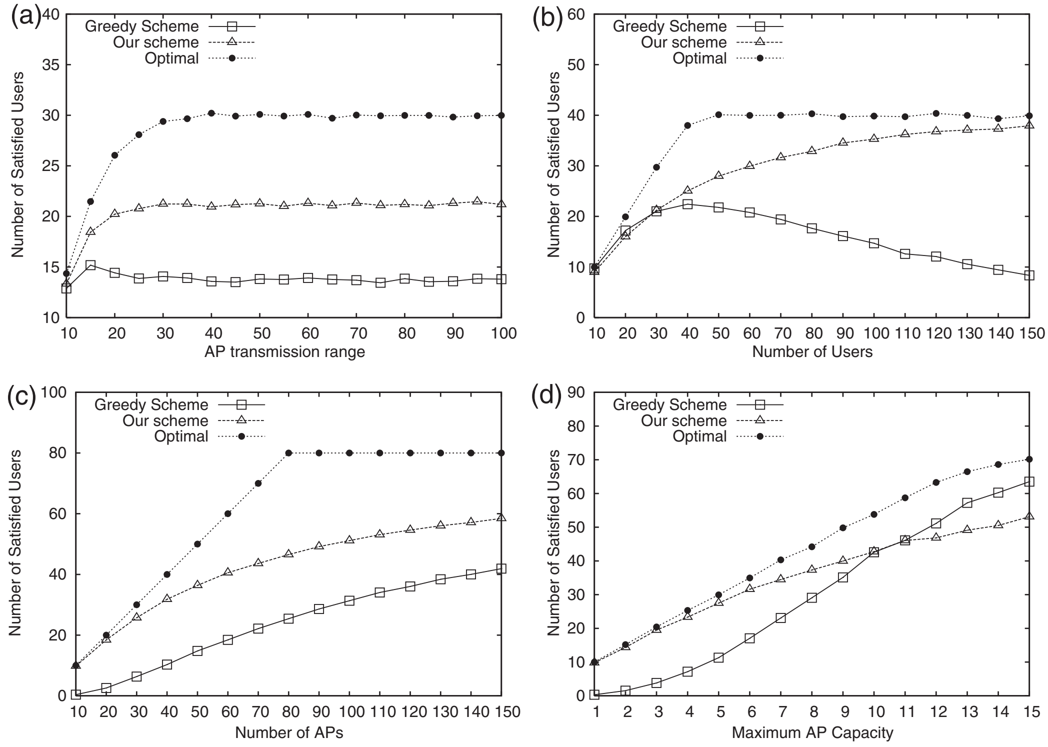

Figure 7. Comparison of the greedy scheme, one-round localized algorithm, and maximum flow- based optimal solution in terms of the total satisfied demands in the homogeneous user demand model. (a) AP transmission range dimension. (b) User number dimension. (c) AP number dimension. (d) AP capacity dimension.

$m = 80$, $n = 10$, and c increases from 1 to 15 in increments of 1. From this result, we observe that the increment of the maximum AP capacity can improve the performance of all three algorithms.

In the fifth experiment, we compare the performance of our one-round localized algorithm, our iterative localized algorithm, and the optimal algorithm. The simulation result is illustrated in Figure 8(a), where $n = 40$, $r = 100$, $c = 3$, and m ranges from 10 to 150 in increments of 10. From this result, we can conclude that the total load by our iterative algorithm is very close to the optimal solution. From Figure 8(b), where the parameters are the same as that of Figure 8(a), except that $c = 1$, 2, and 3, we observe that the converge rate of the iterative solution arrives its minimum (the maximum number of rounds) when the total capacity is equal to the total demand (the sum of demands of all users) no matter what the value of AP capacity is. The bigger the AP capacity, the slower the minimum convergence rate associated with that AP capacity. The reason is that fewer users can establish connection when the total demand is equal to the total capacity. We also evaluate the approximation ratio of the one-round localized algorithm. The simulation result is illustrated in Figure 8(c), where the parameters are the same as that of Figure 8(b), we observe that all the approximation ratios for the three different capacities reach their minimum values when the total capacity is equal to the total demand for different AP capacity. The smaller the AP capacity, the lower the minimum approximation ratio associated with that AP capacity. Therefore, when $c = 1$ and the total capacity is equal to the total demand, the approximation ratio is minimum, which is consistent with Theorem 1.

In the sixth experiment, we consider the case where users have heterogenous demands. We first compare the performance of our one-round localized algorithm, our iterative localized algorithm, and the upper bound of the optimal solution, which is the minimum of the total capacity and the total demand because the optimal solution cannot be larger than the total capacity or the total demand. The simulation result is illustrated in Figure 8(d), where $n = 20$, $r = 40$, $c = 6$, maximum user demand $d = 3$, and m ranges from 10 to 150 in increments of 10. This result testifies that our localized algorithms maintain a good approximation in the case of heterogenous demand. We then evaluate the convergence rate of the iterative localized algorithm. Figure 9(a) shows the simulation result, which illustrates the fast convergence rate in terms of the number of rounds. In the same parameter setting as that of Figure 9(a), we also conduct the experiment on the dynamic user population. We adopt the Poisson process to simulate the inter-arrive time of users. The simulation result shown in Figure 9(b) illustrates that our result holds for dynamic user population either. In the same parameter setting, we conduct the experiment on the normal distribution on user deployment. The simulation

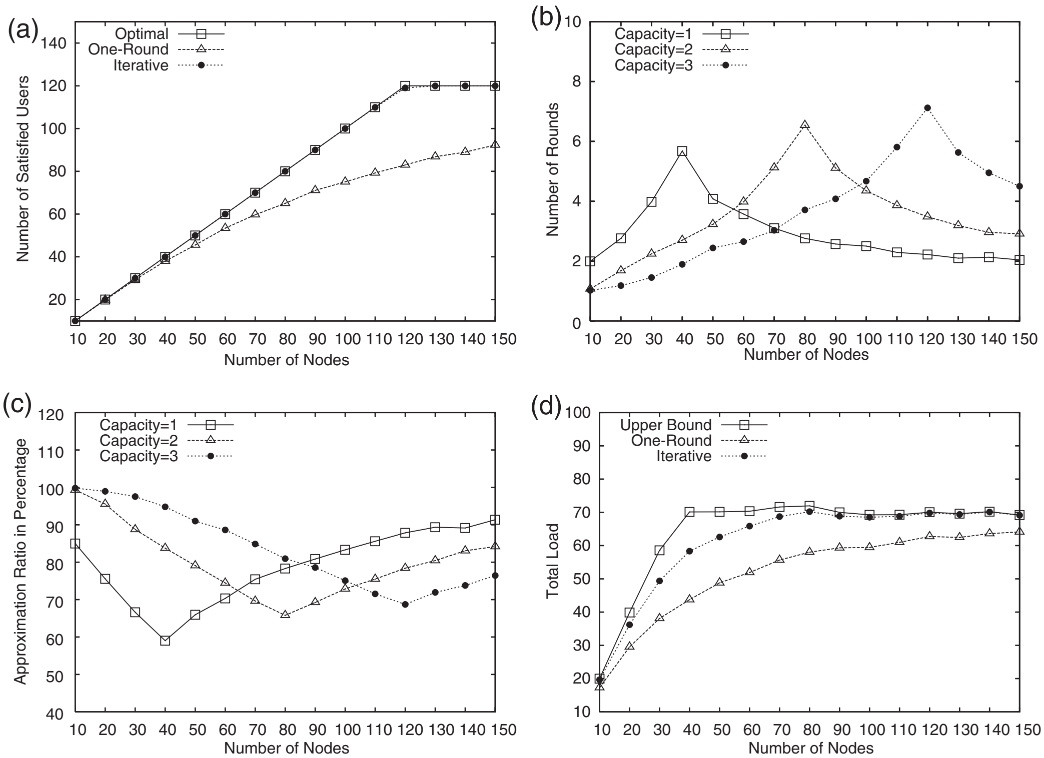

Figure 8. Simulations on the performance comparison, approximation ratio for both homogenous demand and heterogenous demand. (a) Performance comparison (homogenous demand). (b) Converge rate (homogenous demand). (c) Approximation ratio of the one-round localized algorithm. (d) Performance comparison (heterogenous demand).

result shown in Figure 9(c) illustrates that the constant approximation ratio property of our localized algorithms is not limited to the uniform user distribution.

We also evaluate the performance under the environment where interference is taken into account. The simulation result is shown in Figure 9(d), where the area size is 1000×1000, $n = 10$, $r = 250$, $c = 11$ Mbps, maximum user demand $d = 1$ Mbps, and m ranges from 10 to 150 in increments of 10. The simulation result illustrates that our localized algorithm approximates to the optimal solution in performance when the interference is considered.

7. RELATED WORK

Currently, most of the IEEE 802.11 protocols adopt the RSSI-based greedy approach to associate users with APs. However, previous works [1–3] have shown that the RSSI-based approach can lead to poor performance through theoretical analysis and experiment evaluations. The fundamental reason of the poor performance is that the user load is often distributed unevenly among APs by applying the current AP selection strategy [6,7,13].

To address this problem, existing works [1–3,14,15] have proposed numerous approaches. For example, Bejerano, Han, and Li [2] took both load balance and fairness into account by proposing a user-AP association model. They proved the NP-completeness of the proposed user-AP association problem, and modeled the proposed problem as an instance of LP. By utilizing the existing LP-based methods, they designed approximation algorithms with a constant approximation ratio. However, their model assumed that a central node exists to execute the algorithms.

Hajiaghayi *et al*. [4] proposed a load balancing mechanism, which is based on cell breathing, a well-known concept in cellular telephony, to handle client congestion in wireless LANs. They modeled the AP association problem as a weighted matching problem in bipartite graph, and adopted the power control at the AP side to balance the loads among APs. Their major contributions are their solid analysis of the worst case bounds, and the avoiding of the modification at the user side. However, their algorithms are centralized, which are suitable only for the type of wireless LANs, where all APs are owned by the same entity.

Haidar *et al.* [16] proposed an AP association scheme based on power management in order to minimize the maximum congested AP. However, their scheme needs the iterative execution of the Integer LP, the time complexity of which is very high. Besides, they did not analyze the convergence rate of their heuristic. Vasudevan *et al*. [15] proposed a simple and easy-to-implement method for AP selection in IEEE 802.11-based wireless networks. In their

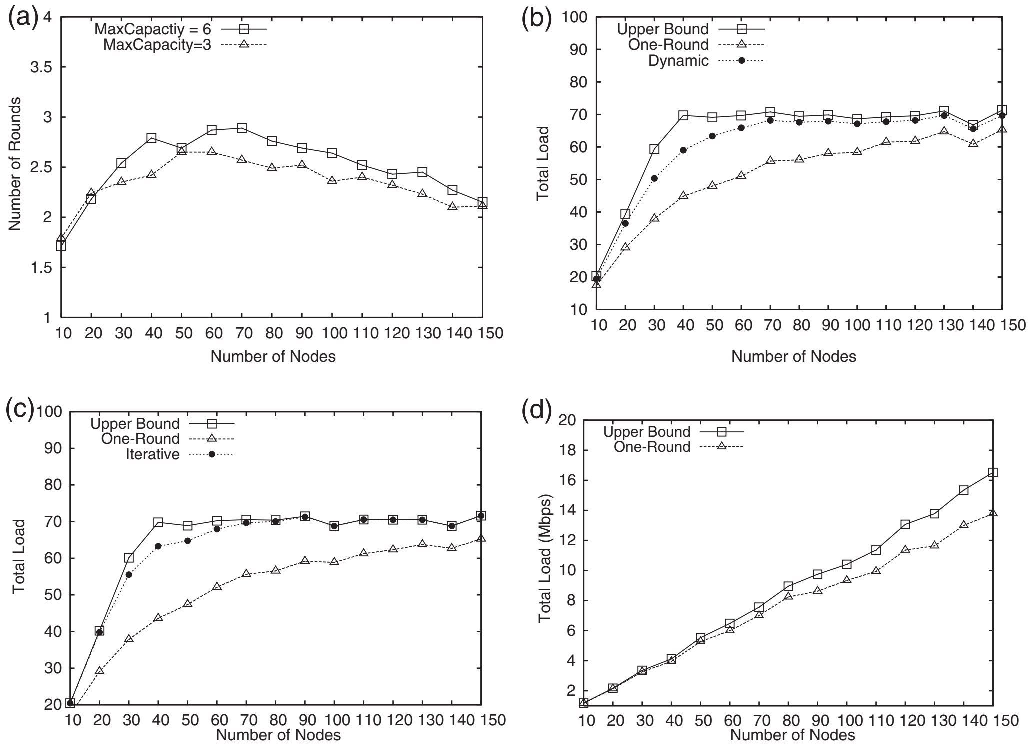

Figure 9. Simulation on the effect of dynamic user population, normal user distribution, and interference. (a) Converge rate (heterogenous demand). (b) Dynamic user population. (c) Normal distribution. (d) Heterogenous user demand with interference.

scheme, end-users make decisions as opposed to the APs. Each user estimates the potential bandwidth of neighbor APs by passive measurement and selects the AP with the highest potential bandwidth to connect. Their scheme is localized and does not require assistance from the AP, but their scheme is only evaluated by simulation. Papanikos and Logothetis [17] proposed a load balancing heuristic that helps users to select the AP to associate with based on the number of users already associated with the AP and the mean RSSI. However, they did not provide any analysis to their heuristic. Mishra *et al*. [18] proposed an client-based approach for channel assignment and load balancing in 802.11-based WLANs that lead to better usage of the wireless spectrum. Mishra *et al*. [19] also considered distributed channel management in uncoordinated wireless environments. Chen *et al*. [20] proposed a solution to select APs during the handoff process to achieve overall load balance.

All of solutions in the above works 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 LP approach [4], and the other is based on the simulated annealing technique [5]. 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 purely localized algorithms, parts of which can provide performance guarantees and do not require the propagation of global information. Our previous works [21,22] presented the preliminary idea of the proposed model and solutions. In this work, we improve our work by providing mathematical analysis and additional simulations.

8. CONCLUSIONS

Existing solutions to the AP selection problem in wireless LANs are either centralized algorithms or localized heuristics without any performance guarantees. In this work, we proposed a set of localized algorithms to the AP selection problem, and proved that several localized algorithms have performance guarantees. We modeled the AP selection problem as the many-to-one matching problem in the bipartite graph. We first considered a special case, the one-to-one bipartite matching, i.e., each AP can support access for one user. In the one-to-one bipartite matching model, we analyzed the effect of the user knowledge and the timing of the AP selection on the performance. We considered three user knowledge models (0.5-hop knowledge, 1-hop knowledge, and 1.5-hop knowledge) and two timing strategies (simultaneous AP selection and sequential AP selection). For each combination of the knowledge model and the timing of selection, we proposed one or two localized algorithms. Through mathematical analysis, we proved that a constant approximation ratio, in terms of average total load, exists if users have knowledge on whether they are the best users of their best APs. We also extended the localized algorithm through iterative executions, and proved that a bound exists for the convergence rate. We also considered various practical issues such as interference, power adjustment, dynamic user population, as well as non-uniform AP and user distributions. Through extensive simulations, we verified that our localized algorithms display good performance under various conditions. In the future, we will conduct an analytical study on our localized algorithms in non-uniform distributions. We will also extend our work to the case in which each AP has multiple channels.

ACKNOWLEDGEMENT

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

Appendix

Proof of theorem 1

Proof. Without loss of generality, we construct the deployment of users and APs as follows: we first randomly deploy all APs within a coverage area, construct the Voronoi graph with all APs as Voronoi points, and then randomly deploy users one by one. Since APs are identical (in terms of the location distribution in the coverage area), the expected area sizes of all Voronoi regions should be the same. Note that the expected area size of an AP is independent of its capacity. Therefore, the probability for a user to be deployed in any Voronoi region should be the same, which is equal to $\frac{1}{n}$, where n is the number of APs. Based on this deployment of users and APs, we have the following lemma.

Lemma 1. *If $m \leq n$, the deployment of the ith AP can on average increase the non-empty Voronoi regions by* $(1 - \frac{1}{n})^{i-1}$.

Proof. We can prove the lemma by induction over the number of APs. For the basis, the first AP can add one non-empty Voronoi region, which satisfies $(1 - \frac{1}{n})^{1-1} = 1$. The second AP can additionally add $(1 - \frac{1}{n})$ non-empty Voronoi region because the probability, with which the second AP is not in the same Voronoi region as the first AP is $(1 - \frac{1}{n})$.

For the inductive step, we assume that the kth AP additionally adds $(1 - \frac{1}{n})^{k-1}$ non-empty Voronoi region. We use $X_n^i(k)(1 \leq i \leq k)$ to represent the probability that k APs are in i different Voronoi regions. If $k - 1$ APs have already been deployed in i different Voronoi regions, the conditional probability that the kth AP is deployed in one of the empty Voronoi regions is $1 - \frac{i}{n}$. Note that it is the only way that the kth AP can introduce the new non-empty Voronoi region under that condition. Therefore, the additional non-empty Voronoi region added by the kth AP can be represented by

$$\sum_{i=1}^{k-1} X_n^i(k-1)\left(1 - \frac{i}{n}\right) = \left(1 - \frac{1}{n}\right)^{k-1} \quad (1)$$

We also observe that the case of k APs being in i different Voronoi regions comes from two sources: (1) $k - 1$ APs have already been deployed in $i - 1$ different Voronoi regions, and the kth AP is deployed in an empty Voronoi region (a Voronoi region other than those $i - 1$ Voronoi regions); (2) $k - 1$ APs have already been deployed in i different Voronoi regions, the kth AP is deployed in one of the non-empty Voronoi regions (those i Voronoi regions). Thus, we can derive the following equation.

$$X_n^i(k) = \begin{cases} X_n^i(k-1) \cdot \frac{i}{n} & i = 1 \\ X_n^{i-1}(k-1) \cdot (1 - \frac{i-1}{n}) + X_n^i(k-1) \cdot \frac{i}{n}, & 1 < i < k \\ X_n^{i-1}(k-1) \cdot (1 - \frac{i-1}{n}) & i = k \end{cases}$$

With the same argument, the additional non-empty Voronoi region introduced by the kth AP is

$$\begin{aligned} &\sum_{i=1}^{k} X_n^i(k)\left(1 - \frac{i}{n}\right) \\ &= \sum_{i=2}^{k}\left[X_n^{i-1}(k-1) \cdot \frac{i-1}{n} \cdot \left(1 - \frac{i-1}{n}\right)\right. \\ &\quad \left. + X_n^{i-1}(k-1)\left(1 - \frac{i-1}{n}\right)\left(1 - \frac{i}{n}\right)\right] \\ &= \left(1 - \frac{1}{n}\right) \cdot \sum_{i=1}^{k-1} X_n^i(k-1)\left(1 - \frac{i}{n}\right) \end{aligned}$$

By combining the above equation with Equation (1), we derive

$$\sum_{i=1}^{k} X_n^i(k)\left(1 - \frac{i}{n}\right) = \left(1 - \frac{1}{n}\right)^k$$

Therefore, the $k + 1$th AP additionally adds $(1 - \frac{1}{n})^k$ non-empty Voronoi regions, which verifies our induction assumption. ■

Based on this lemma, if $m \leq n$, the expected number of satisfied users by our heuristic is exactly $H = \sum_{i=1}^{m}(1 -$

$\frac{1}{n})^{i-1} = n \times (1 - (1 - \frac{1}{n})^m)$. Since the maximum number of satisfied users by the optimal solution is at most m in the case of $m \leq n$, the average approximation ratio of our heuristic to the optimal solution in the case of $m \leq n$ is at least $\frac{n}{m} \times (1 - (1 - \frac{1}{n})^m)$.

If $m > n$, the average increment of the non-empty Voronoi regions through the ith AP ($i > n$) is less than $\frac{1}{(1-n)^{i-1}}$, but it is still increasing. Hence, the expected satisfied users in the case of $m > n$ is more than $\sum_{i=1}^{n} \frac{1}{(1-n)^{i-1}} = n \times (1 - (1 - \frac{1}{n})^n)$. In this case, the maximum number of satisfied users is at most n. Therefore, the expected approximation ratio is also more than $1 - (1 - \frac{1}{n})^n$. We can prove $\frac{n}{m} \times (1 - (1 - \frac{1}{n})^m) \geq 1 - (1 - \frac{1}{n})^n$ for any $m \leq n$. Therefore, combining the above two cases, we can conclude that the approximation ratio of our heuristic is $1 - (1 - \frac{1}{n})^n$. Since $(1 - \frac{1}{n})^n$ is strictly increasing, and $(1 - \frac{1}{n})^n \to \frac{1}{e}$ as $n \to \infty$, the minimum value of the approximation ratio is $1 - \frac{1}{e}$. Hence, the lower bound of the approximation ratio is $1 - \frac{1}{e}$. This theorem is thusly proved. ■

Proof of theorem 2

Proof. The basic idea of this proof stems from our experimental study on the approximation ratio. As shown in Figure 8(c), the approximation ratio reaches its minimum in all the three capacities when the total demand (the number of users) is equal to the total capacity. Moreover, among the three minimum approximation ratios, the smaller the capacity, the smaller the corresponding approximation ratio. If the above properties hold, the approximation ratio should reach its minimum when all APs have unit capacity and the number of users equals to the total capacity.

To prove the above conjecture, we first need to calculate the expected number of users whose demands are satisfied. To do so, we first consider the expected number of users $E_u(a)$ that are deployed in a given Voronoi region $R(a)$, where a represents the AP in that region. To derive the expression of $E_u(a)$, we need to introduce two notations.

First, we use $p_n(i, j)$ $(0 \leq i \leq m)$ to represent the probability that j units of a's capacity are occupied (exactly j users in $R(a)$) after the deployment of the ith user, where m is the number of users. $p_n(i, j)$ comes from two sources: (1) j of the first $i-1$ users have already been deployed in $R(a)$, and the ith user is not deployed in $R(a)$; (2) $j-1$ of the first $i-1$ users have already been deployed in $R(a)$, and the ith user is deployed in $R(a)$. Hence, we can derive the following recursive expression:

$$p_n(i, j) = \left(1 - \frac{1}{n}\right) p_n(i-1, j) + \frac{1}{n} p_n(i-1, j-1) \tag{2}$$

with $p_n(0, 0) = 1$ and $p_n(i, j) = 0$ if $j > i$ or $j < 0$.

Second, we use $y(i, j)$ to represent the event that the ith user is deployed in $R(a)$, given the conditional probability $p_n(i-1, j-1)$, i.e., $y(i, j) = \frac{1}{n} p_n(i-1, j-1)$ $(1 \leq i \leq m)$. Combining this definition and Equation (2), the following equation is straightforward

Table I. The illustration of $y(i, j)$.

1					$y(1, 1)$
2				$y(2, 2)$	$y(2, 1)$
3			$y(3, 3)$	$y(3, 2)$	$y(3, 1)$
⋮		⋯	⋮	⋮	⋮
m	$y(m, m)$	⋯	⋯	⋯	$y(m, 1)$

$$y(i, j) = \left(1 - \frac{1}{n}\right) y(i-1, j) + \frac{1}{n} y(i-1, j-1) \tag{3}$$

with $y(1, 1) = \frac{1}{n}$ and $y(i, j) = 0$ if $j > i$ or $j < 1$. $y(i, j)$ reflects the probability that the ith user deployment fills jth unit of a's capacity, as illustrated in Table I, where the ith row (counted from top to bottom) represents the ith user deployment, and the jth column (counted from right to left) represents the jth unit of a's capacity. $\sum_{j=1}^{i} y(i, j)$, the sum of the ith row in Table I, represents the probability that the ith user is deployed in $R(a)$. Hence, $E_u(a) = \sum_{i=1}^{m} \sum_{j=1}^{i} y(i, j)$, the sum of all rows, represents the expected number of users deployed in $R(a)$.

Through different orders of sum of $E_u(a)$, we can obtain

$$E_u(a) = \sum_{i=1}^{m} \sum_{j=1}^{i} y(i, j) = \sum_{j=1}^{m} \sum_{i=j}^{m} y(i, j) = \sum_{j=1}^{m} L_j(m)$$

where we define $L_j(m) = \sum_{i=j}^{m} y(i, j)$, i.e., the sum of jth column in Table I. Since each AP has a capacity constraint $C(a)$, the expected number of users that can connect to a is $\sum_{j=1}^{C(a)} L_j(m)$. Before we prove the main properties, we need the following lemmas.

Lemma 2. *$y(i, j) = \frac{1}{n^j}(1 - \frac{1}{n})^{i-j} C_{i-1}^{j-1}$ where C_{i-1}^{j-1} is binomial coefficient.*

Proof. This lemma can be proved through induction on j. For the inductive basis, $y(1, 1) = \frac{1}{n}$ satisfies the above equation. For the inductive step, we assume that $y(i-1, j) = \frac{1}{n^j}(1 - \frac{1}{n})^{i-j-1} C_{i-2}^{j-1}$ and $y(i-1, j-1) = \frac{1}{n^{j-1}}(1 - \frac{1}{n})^{i-j} C_{i-2}^{j-2}$. By applying Formula (3), and exploring the fact that $C_r^k = C_{r-1}^k + C_{r-1}^{k-1}$, we can complete the inductive step. The lemma is thusly proved. ■

Lemma 3. *$L_j(m) = L_{j-1}(m) - \frac{(1-\frac{1}{n})^{m-j+1}}{n^{j-1}} C_m^{j-1}$ where $2 \leq j \leq m$, and $L_1(m) = 1 - (1 - \frac{1}{n})^m$.*

Proof. From the definition of $L_j(m)$ and Lemma 2, we have $L_j(m) = \frac{1}{n^j} \sum_{i=j}^{m} (1 - \frac{1}{n})^{i-j} C_{i-1}^{j-1}$. Therefore, $\frac{L_j(m)}{n} = L_j(m) - (1 - \frac{1}{n}) L_j(m) = \frac{1}{n^j} [n^{j-1} L_{j-1}(m) - (1 - \frac{1}{n})^{m-j+1} C_m^{j-1}]$. By multiplying n on both sides of the above equation, we derive the recursive expression of $L_j(m)$. Since

$y(m+1, j) = \frac{(1-\frac{1}{n})^{m-j+1}}{n^j} C_m^{j-1}$. The recursive expression can also be written as $L_j(m) = L_{j-1}(m) - ny(m+1, j)$. We also have $L_1(m) = \sum_{i=1}^{m} \frac{1}{n}(1 - \frac{1}{n})^{i-1} = 1 - (1 - \frac{1}{n})^m$. This lemma is thusly proved. ■

Lemma 4. $\sum_{j=1}^{l} y(i, j) \leq \sum_{j=1}^{l} y(i-1, j)$.

Proof. By applying Equation (3), we have $\sum_{j=1}^{l} y(i, j) = \sum_{j=1}^{l} (\frac{1}{n} y(i-1, j-1) + (1 - \frac{1}{n}) y(i-1, j))$. Because $y(i-1, 0) = 0$ according to the definition, we have $\sum_{j=1}^{l} y(i, j) = \sum_{j=1}^{l-1} y(i-1, j) + (1 - \frac{1}{n}) y(i-1, l) \leq \sum_{j=1}^{l} y(i-1, j)$. ■

As a direct application of Lemma 4, we have

$$\sum_{j=1}^{C} y(i, j) \leq \sum_{j=1}^{C} y(q, j), \quad 1 \leq q < i \tag{4}$$

Lemma 5. $\sum_{j=1}^{l} L_j(m) \geq m \sum_{j=1}^{l} y(m+1, j)$.

Proof. Because $\sum_{j=1}^{l} L_j(m) = \sum_{j=1}^{l} \sum_{i=1}^{m} y(i, j) = \sum_{i=1}^{m} \sum_{j=1}^{l} y(i, j) \leq \sum_{i=1}^{m} \sum_{j=1}^{l} y(m+1, j) = m \sum_{j=1}^{l} y(m+1, j)$ where the inequality holds according to Inequality (4), we can derive this lemma directly. ■

Without loss of generality, we assume that the maximum AP capacity is C, and there are k_l APs with capacity l. Hence, $\sum_{l=1}^{C} k_l = n$ and the total capacity $X = \sum_{l=1}^{C} lk_l$. The expected number of users that can connect to an AP with capacity l is $\sum_{j=1}^{l} L_j(m)$, which is also the expected load on the AP since all users have unit demand. Since the expected loads on any two APs are independent, the total expected load is $\sum_{l=1}^{C} k_l \sum_{j=1}^{l} L_j(m)$. Because the optimal solution cannot exceed the total capacity or the total demand (the total number of users in the case of homogenous demand), the upper bound of the optimal solution is the minimum of the total capacity and the total demand. Therefore, the lower bound of the approximation ratio can be defined as a function

$$f(m) = \begin{cases} \frac{\sum_{l=1}^{C} k_l \sum_{j=1}^{l} L_j(m)}{m}, & m \leq X \\ \frac{\sum_{l=1}^{C} k_l \sum_{j=1}^{l} L_j(m)}{X}, & m \geq X \end{cases}$$

Lemma 6. $f(m) \geq f(m+1)$, $m+1 \leq X$.

Proof. To prove the lemma, it is equal to prove $\frac{k_l \sum_{j=1}^{l} L_j(m)}{m} \geq \frac{k_l \sum_{j=1}^{l} L_j(m+1)}{m+1}$ for $l = 1, \ldots, C$, which is equal to $(m+1) \sum_{j=1}^{l} L_j(m) \geq m \sum_{j=1}^{l} L_j(m+1)$. By applying Lemma 5 and the equation $L_j(m+1) = L_j(m) + y(m+1, j)$, we can prove this lemma. ■

Since $L_j(m)$ is non-decreasing as m increases, the numerator of $f(m)$ in the case of $m \geq X$ is non-decreasing as m increases. Hence, $f(m)$ is non-decreasing when $m \geq X$ because the denominator of $f(m)$ in the case of $m \geq X$ does not change as m increases. From Lemma 6, we can conclude that $f(m)$ is non-increasing when $m \leq X$. Therefore, $\min_m f(m) = f(X)$ if the total capacity is X.

Lemma 7. $L_1(n) \leq f(X)$.

Proof. To prove the above inequality, it is equal to prove $XL_1(n) \leq \sum_{l=1}^{C} k_l \sum_{j=1}^{l} L_j(x)$. Since $X = \sum_{l=1}^{C} k_l l$, the above inequality is equal to $k_l l L_1(n) \leq k_l \sum_{j=1}^{l} L_j(X)$ $(l = 1, \ldots, C)$, which is equal to $lL_1(n) \leq \sum_{j=1}^{l} L_j(X)$ $(l = 1, \ldots, C)$. The last inequality can be derived by applying Lemma 4 recursively. Thus, this lemma is proved. ■

Note that $L_1(n) = \frac{nL_1(n)}{n}$ is the minimum approximation ratio in the case that all APs have unit capacity. Thus, Lemma 7 implies that the lowest approximation ratio is $L_1(n)$, which is $1 - (1 - \frac{1}{n})^n$ by Lemma 3. Since $(1 - \frac{1}{n})^n$ is strictly increasing, and $(1 - \frac{1}{n})^n \to \frac{1}{e}$ as $n \to \infty$, the minimum value of the approximation ratio is $1 - \frac{1}{e}$. Hence, the lower bound of the approximation ratio is $1 - \frac{1}{e}$. This theorem is thusly proved. ■

Proof of theorem 3

Proof. We first consider the case that $m \leq n$. Based on the analysis of Theorem 1, we observe that at least $(1 - \frac{1}{e}) \times m$ users are connected after the first round. Hence, the number of unconnected users is at most $\frac{1}{e} \times m$. Among the unconnected users, we use d_u^i to denote the number of unconnected users that cannot be covered by any AP after the ith round, and r_u^i to denote the number of the unconnected users except those uncovered users. Assume that the number of rounds is k. We have the following inequality:

$$r_u^i \leq r_u^i + d_u^i \leq \frac{1}{e} \times r_u^{i-1}, \quad 1 \leq i \leq k \tag{5}$$

Combining $r_u^k \geq 1$ and Formula (5), we can conclude that $1 \leq (\frac{1}{e})^k \times m$. Note that $r_u^0 = m$. Therefore, we have $k \leq \ln m$. For the case where $m \geq n$, at least $(1 - \frac{1}{e}) \times n$ users are connected after the first round. Similarly, we can derive $k \leq \ln n$. Therefore, we have $k \leq \ln l$. ■

REFERENCES

1. Balachandran A, Bahl P, Voelker G. Hot-spot congestion relief and service guarantees in public-area wireless networks. *SIGCOMM Computer Communication Review* 2002; **32**(1): 59.
2. Bejerano Y, Han S, Li L. Fairness and load balancing in wireless lans using association control. In *Proceedings of ACM MobiCom'04*, 2004.

3. Judd G, Steenkiste P. Fixing 801.11 access point selection. In *Poster in Proceedings of ACM MobiCom'02*, 2002.
4. Hajiaghayi MT, Mirrokni SV, Saberi A, Bahl P, Jain K, Qiu L. Cell breathing in wireless LANs: algorithms and evaluation. *IEEE Transactions on Mobile Computing* 2007; **6**(2): 164–178.
5. Kauffmann B, Baccelli F, Chaintreau A, Mhatre V, Papagiannaki K, Diot C. Measurement-based self organization of interfering 802.11 wireless access networks. In *Proceedings of INFOCOM'07*, 2007.
6. Balachandran A, Voelker GM, Bahl P, Rangan PV. Characterizing user behavior and network performance in a public wireless LAN. In *Proceedings of the ACM SIGMETRICS'02*, 2002; 195–205.
7. Kotz D, Essien K. Analysis of a campus-wide wireless network. *Wireless Network* 2005; **11**(1–2): 115–133.
8. Cormen T, Leiserson C, Rivest R, Stein C. Introduction to Algorithms (2nd edn). The MIT Press: Cambridge, Massachusetts, 2001.
9. Hills A. Wireless Andrew. *IEEE Spectrum* 1999; **36**(6): 49–53.
10. Alexander B, Snow S. Preparing for wireless LANs: secrets to successful wireless deployment. *Packet Magazine*. Cisco Systems. http://www.cisco.com/warp/public/784/packet/apr02/ p36cover.html.
11. Kandukuri S, Boyd S. Optimal power control in interference limited wireless channels with outage probability fading specications. *IEEE Transactions on Wireless Communications* 2002; **1**(1): 46–55.
12. Biswas S, Morris R. ExOR: opportunistic multi-hop routing for wireless networks. In *Proceedings of ACM SIGCOMM'05*, 2005; 133–144.
13. Balazinska M, Castro P. Characterizing mobility and network usage in a corporate wireless local-area network. In *Proceedings of MobiSys'03*, 2003.
14. Murty R, Qi E, Hazra M. An adaptive approach to wireless network performance optimization. *Technical Report*, Cororate Technology Group (CTG), Intel Corporation, 2004.
15. Vasudevan S, Papagiannaki K, Diot C, Kurose J, Towsley D. Facilitating access point selection in IEEE 802.11 wireless networks. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC'05)*, 2005; 293–298.
16. Haidar M, Akl R, Al-Rizzo H, Chan Y, Adada R. Optimal load distribution in large scale WLAN networks utilizing a power management algorithm. In *Proceedings of IEEE Sarnoff Symposium*, 2007.
17. Papanikos I, Logothetis M. A study on dynamic load balance for IEEE 802.11b wireless LAN. In *Proceedings of the 8th International Conference on Advances in Communications and Control (COMCON'01)*, 2001.
18. Mishra A, Brik V, Banerjee S, Srinivasan A, Arbaugh WA. A client-driven approach for channel management in LANs. In *Proceedings of INFOCOM'06*, 2006.
19. Mishra A, Shrivastava V, Agrawal D, Banerjee S, Ganguly S. Distributed channel management in uncoordinated wireless environments. In *Proceedings of MOBICOM'06*, 2006; 170–181.
20. Chen J, Chen T, Zhang T, Berg E. Effective ap selection and load balancing in ieee 802.11 wireless lans. In *Proceedings of GLOBECOM'06*, 2006.
21. Lu M, Wu J. Localized access point selection in infrastructure wireless lan. In *Proceedings of Military Communications Conference (MilCOM'07)*, 2007; 1–7.
22. Lu M, Wu J. Localized access point association in wireless lans with bounded approximation ratio. In *Proceedings of the 17th Computer Communications and Networks (ICCCN'08)*, 2008; 1–8.

AUTHORS' BIOGRAPHIES

Mingming Lu is an Assistant Professor in the Department of Computer Science at Central South University. He received his Ph.D. from the Department of Computer Science and Engineering, Florida Atlantic University in August, 2008. He has published over 20 papers in various journal and conference proceedings. His current research includes various topics in wireless computing, such as opportunistic routing, utility-based routing, routing algebra, game theory, access control in wireless LAN, and discrete target coverage in wireless sensor networks. He is a member of IEEE.

Jie Wu is a Distinguished Professor at Florida Atlantic University (FAU), and a Program Director at National Science Foundation. He has published more than 450 papers in journals and conferences in the areas of wireless networks and routing protocols. He is the author of the text Distributed System Design. Currently, he serves on the Editorial Board of the *IEEE Transactions on Mobile Computing*. He was a recipient of the 1996–1997, 2001–2002, and 2006–2007 Researcher of the Year Award from FAU. He served as an IEEE Computer Society distinguished visitor and is the chairman of the IEEE Technical Committee on Distributed Processing. He is an IEEE fellow.

Wirel. Commun. Mob. Comput. 2011; **11**:994–1010 © 2009 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm