Dynamic Distributed Algorithm for AP Association under User Random Arrivals and Departures

Zhenwei Chen¹, Wenjie Zhang¹,*, Jingmin Yang¹, Shengyu Chen², Liwei Yang³ and Yeo Chai Kiat⁴

¹School of Computer Science, Minnan Normal University, China
²School of Informatics, Computing & Engineering, Indiana University, United State
³College of Information and Electrical Engineering, China Agricultural University
⁴School of Computer Engineering, Nanyang Technological University, Singapore

Abstract. In this paper, we study the novel problem of optimizing AP association by maximizing the network throughput, subject to the degree bound of AP. The formulated problem is a combinatorial optimization. We resort to the Markov Chain approximation technique to design a distributed algorithm. We first approximate our optimal objective via Log-Sum-Exp function. Thereafter, we construct a special class of Markov Chain with steady-state distribution specific to our problem to yield a distributed solution. Furthermore, we extend the static problem setting to a dynamic environment where the users can randomly leave or join the system. Our proposed algorithm has provable performance, achieving an approximation gap of $1 - \frac{1}{\log |F|}$. It is simple and can be implemented in a distributed manner. Our extensive simulation results show that the proposed algorithm can converge very fast, and achieve a close-to-optimal performance with a guaranteed loss bound.

1. Introduction

Wireless Local Area Networks (WLANs) are widely deployed to meet the growing demand on Internet access of users via Access Point (AP). By default, the user in a WLAN selects the AP with the strongest Received Signal Strength Indicator (RSSI). Due to non-uniform distribution of users, this simple RSSI based association approach often leads to the load imbalance among APs and degrades the performance of the entire network. To address this issue, many AP association algorithms are proposed by researchers. These algorithms can mainly be categorized into three classes: local-selfish association, optimized decentralized association and centralized association.

For local-selfish association scheme, users select the appropriate AP based on the collected load information of APs and their own requirements [1]. In [1], an appropriate AP is selected by user based on each individual device channel utilization. Although this approach can be implemented in a distributed manner, the selfish behaviour of the user makes the global optimization difficult to achieve. Optimized decentralized association algorithm tries to achieve the global optimum with the measurements by local users, such algorithm can optimize global network performance with the assumption that all users on a single AP have the same throughput. Centralized association can provide optimal AP association. It requires a control unit to collect the global load information [2], or collaboration among the APs from different networks [3]. However, systems running centralized algorithm, are often less adaptable to users joining and leaving the systems and are less robust to system/network dynamics.
Previous work often considers naive fixed association, such that one user is not allowed to change its associated AP during its lifetime. As a result, some certain APs may become a hot spot and have to handle a high traffic load, while other APs remain underutilized. Therefore, it is necessary to rebalance loads by user migration from hot AP to lightly loaded neighboring APs. Furthermore, due to the dynamic, unpredictable and spatially non-uniform distributed of users, it is desirable to optimize AP association over time. Therefore, it is challenge to study the scenario of dynamic network settings, which captures both the users’ mobility, arrival and departure [4]. To fill the gap between the existing centralized algorithm and suboptimal decentralized algorithm, we propose a distributed AP association algorithm that can provide performance guarantees without the propagation of global information.

2. Problem Formulation

2.1. Preliminaries

We consider a general WLAN, in which there are \( m \) APs and \( n \) users. Let \( A \) and \( U \) be the sets of APs and users, respectively, then we have \( |A| = m \) and \( |U| = n \). We model the network as a general bipartite graph \( G(A \cup U; l) \), where \( l \) represents the links between APs and users. A link exists between \((a, u)\), \( a \in A \) and \( u \in U \), if and only if the user \( u \) is within the transmission range of AP \( a \). We call the AP the user’s neighbor AP, denoted by \( A_u \). Figure 1(a) shows an example of 3 APs and 5 users, where \( A_{u1} = \{a1\}, A_{u2} = A_{u3} = \{a1, a2\}, A_{u4} = A_{u5} = \{a2, a3\} \). We assume that each user can connect to only one AP, whereas each AP can serve multiple users simultaneously. However, the number of users that can be served is further bounded due to the limited capacity of AP. Therefore, we consider the constraint that each AP has a degree bound \( B_a \). We allow different APs to have different degree bounds.

![Figure 1](image.png)

**Figure 1.** (a) Network architecture for 3 APs and 5 users. (b)-(d) Three feasible AP-user association under degree bound 2 for each AP.

2.2. AP Association Problem Formulation

We define \( y_{ij} \) to be a binary variable for associating user \( u_i \) to AP \( a_j \), such that

\[
y_{ij} = \begin{cases} 
1 & \text{if users } u_i \text{ is associated to AP } a_j \\
0 & \text{Otherwise}
\end{cases}
\]

Let \( r_{ij} \) be the transmission rate of user \( u_i \) associated to AP \( a_j \), then we formulate the AP-User association Problem (AUP) with the objective to maximize the total achieved rate on all APs as the following constrained optimization:

\[
\text{AUP1: } \max \sum_{i} \sum_{j} y_{ij} r_{ij} \\
\text{s.t. } \sum_{j=1}^{m} y_{ij} = 1, \forall u_i \in U
\]
\[ \sum_{i=1}^{n} y_{ij} \leq B_{aj}, \forall a_j \in A \quad (3) \]

\[ y_{ij} \in \{0,1\}, \forall i, j \quad (4) \]

3. The Proposed Distributed Algorithm

The formulated AUP problem is a combinatorial optimization, which is known to be NP-complete and it seems impossible to solve this problem in polynomial time even in a centralized manner. We resort to the Markov Chain approximation proposed in [5] to design a distributed algorithm in order to obtain an approximated solution. The distributed algorithm consists of the following two steps: log-sum-exp approximation and Markov Chain implementation.

3.1. Log-Sum-Exp Approximation

Let \( H \) denote all the subsets of \( U \), including the empty set and \( U \), and \( U_{aj} \) be the set of users associated to AP \( a_j \), then we start with the following definition of network configuration.

**Definition 1:** A configuration \( f = [U_{1f},...,U_{nf},...,U_{mf}] \) is feasible for AP-user association problem if

1. \( U_{1f} \cap U_{j'f} = \emptyset, \forall U_{j'f}, U_{j''f} \in H, \text{if } j \neq j' \);
2. \( |U_{jf}| \leq B_{aj}, \forall a_j \in A \).

where \( U_{jf} \) is the set of users associated to \( a_j \) under configuration \( f \). For ease of presentation, we use \( F \) to denote the set of all feasible configurations, and \( r_f \) be the objective function for configuration \( f \). The AP association problem can be formulated as:

**AUP2:** \( \max_{f \in F} r_f \quad (5) \)

We use the following Log-Sum-Exp function to approximate AUP2, that is

\[ \max_{f \in F} r_f \approx \frac{1}{\eta} \log \left[ \sum_{f \in F} \exp(\eta r_f) \right] \quad (6) \]

where \( \eta \) is a positive constant. The approximation gap is upper bounded by \( \frac{1}{\eta} \log |F| \) [5]. In order to provide a better understanding of this Log-Sum-Exp approximation, each configuration \( f \in F \) can be associated with a probability \( p_f \), which results in the following problem

**AUP3:** \( \max_{p_f \geq 0} \sum_{f \in F} p_f r_f \)

s.t. \( \sum_{f \in F} p_f = 1 \quad (7) \)

It can be easy to observe that the AUP2 and AUP3 are equivalent by setting the probability of the optimal configuration to be 1, and the other configurations to be 0. According to the Theorem 1 in [5], AUP3 can be rewritten in the following approximated form

**AUP4:** \( \max_{p_f \geq 0} \sum_{f \in F} p_f r_f - \frac{1}{\eta} \sum_{f \in F} p_f \log(p_f) \)

s.t. \( \sum_{f \in F} p_f = 1 \quad (9) \)

**Theorem 1.** The optimal solution to AUP4 problem is given by

\[ p_f^*(x) = \frac{\exp(\eta r_f)}{\sum_{f \in F} \exp(\eta r_f)}, \forall f \in F \quad (11) \]
The optimal solution can be obtained by solving the Karush-Kuhn-Tucker (KKT) conditions on the problem. The detailed proof can refer to our technical report in [6]. Therefore, by using Log-Sum-Exp approximation, we can obtain the optimal solution to the AUP1, off by an entropy term \( \frac{1}{\eta} \log |F| \). Then if we can design a Markov Chain with steady-state distribution in a distributed way, when this Markov Chain converges, the system will time share among all the feasible configurations. As such, we can obtain a close-to-optimal solution to the AP association problem, with approximation gap bound \( \frac{1}{\eta} \log |F| \).

3.2. Markov Chain Algorithm Design
In the following, we will design a time-reversible Markov Chain with the set of all feasible configurations \( F \) as its state spaces. Let \( f = [U'_1, \ldots, U'_n, \ldots, U''_n] \) and \( f' = [U'_1, \ldots, U'_n, \ldots, U''_n] \) represent two feasible configurations in \( F \). \( f \) and \( f' \) are also two states for the designed Markov Chain. We use \( p_f (r) \) and \( p_{f'} (r) \) to denote the corresponding stationary distribution for states \( f \) and \( f' \), and \( q_{gf} \) is the transition rate from \( f \) state to \( f' \). In order to make the designed Markov Chain time-reversible, the following balance equation should be satisfied, that is

\[
p_f (r) q_{f \rightarrow f'} = p_{f'} (r) q_{f' \rightarrow f}, \forall f, f' \in F
\]  

(12)

In this case, the constructed Markov Chain can have a stationary distribution \( p_f (r) \) in (11). As the Markov Chain traverses from one state to another state, the system will operate under different configurations. When the Markov Chain converges, the system will stay in the best configuration for most of the time, and then the AP association problem is solved approximately in a distributed way.

1) State-space-structure: As discussed before, we use \( A_u \) to denote the set of APs that cover user \( u \), let \( a'_{j} \) be the AP that is associated by user \( u \) under configuration \( f \), then the set of non-associated neighbor APs is denoted as \( A_u / a'_{j} \). To construct the Markov Chain, the transition rate \( q_{gf} \) is set to 0, unless the following conditions satisfy:

\[ C1: \forall a_j \in A \{ a'_{j}, a_{j_1} \}, U'_j = U''_j ; \quad C2: |U'_j - U''_j| = 1, \text{and} |U'_j - U''_j| = 1; \quad C3: \alpha_j, a_{j_2} \in A_u \]

In other words, we only allow direct transitions between two configurations \( f \) and \( f' \), if such transition corresponds to a single user \( u \) removes its connection with current AP \( a_{j} \) and associates with another neighbor AP \( a'_{j} \) in \( A_u / a'_{j} \). As such, the transition from configuration \( f \) in Figure.1(b) to configuration \( f' \) in Figure.1(c) is feasible, while the direct transition between configurations in Figure.1(b) and Figure.1(d) is not allowed.

2) Transition rate: In our paper, we set \( q_{gf} \) and \( q_{f' \rightarrow f} \) to be positively correlated to the difference of system performance between configuration \( f \) and \( f' \), that is

\[
q_{f' \rightarrow f} = \frac{1}{\exp(\alpha)} \exp(\frac{1}{2} \eta (r_f - r_{f'}))
\]

(13)

\[
q_{f \rightarrow f'} = \frac{1}{\exp(\alpha)} \exp(\frac{1}{2} \eta (r_f - r_{f'}))
\]

(14)

where \( \alpha \) is a constant. It can be easy to verify that the detailed balance equation (12) holds. Since the transition from configuration \( f \) to \( f' \) is scale-limited, only one user will change its transmission rate and leave remainders unchanged, thus the transition rate can be simplified as

\[
q_{f \rightarrow f'} = \frac{1}{\exp(\alpha)} \exp(\frac{1}{2} \eta (r'_f - r'_{f'}))
\]

(15)

Therefore, our choice of transition rate can be calculated in local way, it does not require any coordination in its implementation.

3) Implementation of Markov Chain: The proposed implemented algorithm is described in Algorithm 1.
Algorithm 1: An Implemented Algorithm for Markov Chain

Require: \( U \) : User set, \( A \) : AP set, \( A_u \) : Neighboring APs for each user \( u \).

Initialization: Each user \( u_i \in U \) randomly chooses one AP \( a_j \) from \( A_{u_i} \), and flag = 0

1: Procedure Selection \((u_i)\)
2: Calculate transmission rate \( r_{ij}^f \) for configuration \( f \).
3: Estimate transmission rate \( r_{ij}^{f'} \) for targeting configuration \( f' \).
4: Generate a random exponentially distributed timer \( T_{u_i} \) for user \( u_i \) with a mean value equal to
   \[
   \frac{\exp\left(\frac{1}{2}n(r_{ij}^f - r_{ij}^{f'})\right)}{|A_{u_i}| - 1}
   \]
5: Begin to count down
6: while the timer \( T_{u_i} > 0 \) do
7: \textbf{if} other user is performing AP swapping \textbf{then}
8: \hspace{1em} flag = 1;
9: \hspace{1em} break;
10: \textbf{end if}
11: \textbf{end while}
12: \textbf{if} flag = 1 then
13: Terminate current countdown and invoke Section \((u_i)\)
14: \textbf{else}
15: flag = 0;
16: \textbf{User} \( u_i \) \textbf{swaps its associated AP from} \( a_{j_1} \) \textbf{to} \( a_{j_2} \);
17: \textbf{Announce} to its neighbors;
18: \textbf{end if}
19: \textbf{return rules}

Theorem 2. The proposed Algorithm 1 realizes a time-reversible Markov Chain with stationary distribution \( p_i^*(r) \) given in (11), \( \forall f \in F \).
The main idea is to verify that the balance equation holds. The proof is given in technical report in [6].

4. Dynamic Markov Chain with User Arrivals and Departures

In this section, we further extend the static problem setting to a dynamic environment where the number of users varies with time, which poses significant new implementation challenges. In particular, we assume that the users arrive according to a Poisson process with parameter \( \lambda \), and stay in the system for time that is exponentially distributed with parameter \( \mu \). Under this setting, the number of users in the system can be modeled as a \( M/M/m \) queue.

4.1. Log-Sum-Exp Approximation

Let \( \phi_n \) denote the steady-state probability of having \( n \) users in the system, then we have

\[
\phi_n = \frac{n! \sigma^n}{\sum_{k=0}^{\infty} k! \sigma^k} = \frac{1}{n!} \sigma^n e^{-\sigma}
\]
where \( \sigma = \lambda / \mu \). We use \( F_n \) to denote the set of all feasible configurations when there are \( n \) users in the system, and \( r^n_{\text{max}} \) be the objective function for configuration \( f_n \in F_n \), then the optimal performance for \( n \) users is \( r^n_{\text{max}} = \max_{f_n \in F_n} r^n_{\text{max}} \). Therefore, the long-term averaged system performance is \( R^* = \sum_{n=0}^{\infty} \phi_n r^n_{\text{max}} \).

By using the Log-Sum-Exp approximation, we have

\[
R^* \approx \frac{1}{\eta} \log \left( \sum_{f_n \in F_n} \exp(\eta r^n_{\text{max}}) \right)
\]

(17)

Let \( p_{f_n} \) be the probability associated to configuration \( f_n \in F_n \), apply the analysis in Section 3, we come to a conclusion in Theorem 3.

**Theorem 3.** The optimal solution of the following optimization problem

\[
\max_{f_n \in F_n} \sum_{n=0}^{\infty} \sum_{f_n' \in F_n} p_{f_n} r^n_{f_n'} - \frac{1}{\eta} \sum_{n=0}^{\infty} \sum_{f_n \in F_n} p_{f_n} \log p_{f_n}
\]

(18)

\[
\sum_{f_n \in F_n} p_{f_n} = \phi_n, n = 1, 2, ...
\]

(19)

is given by

\[
p^*_{f_n} = \frac{\exp(\eta r^n_{\text{max}})}{\sum_{f_n \in F_n} \exp(\eta r^n_{\text{max}})} \phi_n, \forall f_n \in F_n, n = 1, 2, ...
\]

(20)

4.2. Markov Chain Algorithm Design

Next, we construct a Markov Chain to achieve close-to-optimal performance in the long run. The key idea is that we only allow the system to jump from one state to another state when there are user arrivals or departures. In particular, when a user arrives, it will select one possible AP to associate, leading a transition from state \( f_n \) to \( f_{n+1} \). When a user departs, one of the users will remove its connection with the AP, making the system transmit from state \( f_n \) to \( f_{n+1} \). Otherwise, the system configuration will not be changed when there is no user arrival and departure.

When a new user arrives, the system will move from state \( f_n \) to \( f_{n+1} \). Let \( q_{f_n \rightarrow f_{n+1}} \) denote this transition rate, which is defined as

\[
q_{f_n \rightarrow f_{n+1}} = \mu \frac{\exp(\eta r^{n+1}_{\text{max}})}{\sum_{f_n \in F_n} \exp(\eta r^n_{\text{max}})}
\]

(21)

where \( C \) is the set of local configurations that are available for the new user, and \( f_{n+1} = f_n \cup c \).

When a user departs, one of the \( n+1 \) users will remove its connection with the AP, and transmit from state \( f_{n+1} \) to \( f_n \), this transition rate is designed as:

\[
q_{f_{n+1} \rightarrow f_n} = (n+1) \mu \frac{\exp(\eta r^n_{\text{max}})}{\sum_{f_n \in F_n} \exp(\eta r^n_{\text{max}})}
\]

(22)

Next we will show how to implement the Markov Chain to make sure the constructed Markov Chain is time-reversible and its stationary distribution is indeed (20).

**Step 1:** Let \( f_0 \) denote the current configuration and \( f_{n+1} = f_n \cup c \) be the targeting configuration when a new user arrives. This user will estimate the received transmission rate for all feasible local configurations \( c \in C \), and \( f_{n+1} = f_n \cup c \).

**Step 2:** The new user assigns to configuration \( c \in C \) with a probability:

\[
\frac{\exp(\eta r^{n+1}_{\text{max}})}{\sum_{c \in C, f_n \cup c \in F_n} \exp(\eta r^{n+1}_{\text{max}})}
\]
Step 3: Suppose the system is on state $f_{n+1}$ on a user departure, one of the users is chosen to remove with a probability

$$
\frac{\sum_{f_n \in f_{n+1}} \exp(\eta r_{n+1}^f) \exp(\eta r_n^f)}{\sum_{f_n \in f_{n+1}} \sum_{c \in C_{f_n+1}, c \in C_{f_n}} \exp(\eta r_{n+1}^f)}
$$

The calculation of $q_{f_n \rightarrow f_{n+1}}$ requires no knowledge of global information. However, the transition rate $q_{f_{n+1} \rightarrow f_n}$ cannot be estimated locally without information from all targeting configurations $f_{n+1} \in F_{n+1}$ which presents significant distributed implementation challenge.

**Theorem 4.** The implementation realizes a time-reversible Markov Chain with stationary distribution $p_n$ given in (22), $f_n \in F_n$, $n=1,2,...$.

The proof is relegated to our technical report in [6].

5. Simulation Results
In this section, the simulation is conducted to evaluate the distributed AP association algorithm.

5.1. Evaluation of Proposed Distributed Algorithm
The following two aspects will be investigated for different degree bound of AP:

1) Does the distributed algorithm converge to optimal solution?
2) How fast does the algorithm converge?

We randomly choose a feasible configuration and run our algorithm over it. The results are illustrated in Figure 2. From Figure 2 we can see that as the iteration increases the transmission rate also increases. This because that our proposed algorithm is likely to make the system transfer to a configuration with better performance. It converges about 70 iterations in the small size network as shown in Figure 2(a), and 390 iterations in the large size network as shown in Figure 2(b). Furthermore, we can also observe that the converged transmission rate is very close to the optimal value, which verifies the effectiveness of our proposed algorithm.

![Figure 2](image_url)

**(a)** The achieved transmission rate with $n=10$ and $m=10$  
**(b)** The achieved transmission rate with $n=30$ and $m=30

**Figure 2.** Performance comparison between optimum and Algorithm 1.

5.2. Evaluation of Dynamic Algorithm
In this subsection, we will evaluate our proposed algorithm in dynamic environment where the number of users varies with time.

From Figure 3(a), we can see that the achievable transmission rate decreases at time point $t=[5,15,25]$ and keeps invariant at point $t=[10,20,30]$. And also we can see from Figure 3(b), the achievable transmission rate increases at time point $t=[6,12,18,24]$, decreases at $t=[10,20]$, and keeps invariant at point $t=30$. This can be explained as follows, when a user arrives, it will select one available AP to associate, then the achievable transmission rate will increase. On the contrary, when a user departs, it will remove its connection with AP, and the achievable transmission rate will decrease.
6. Conclusion
In this paper, we focus on the AP association problem by taking the users arrival and departure into consideration, which brings great challenges. We first model the AP association problem as a 0-1 integer optimization problem, which is a combinatorial optimization problem. Then, we design a distributed algorithm based on Markov chain to solve it. Furthermore, we extend our distributed algorithm to a dynamic scenario where the user randomly arrives or departs. Evaluation results corroborate that our proposed distributed algorithm can converge very fast to the optimal solution.

7. Acknowledgments
This work is supported by Natural Science Funds of China (Nos. 61701213, 61705260), Natural Science Funds of Fujian (No.2018J01546), the Special Research Fund for Higher Education of Fujian (No. JK2017031), Zhangzhou Municipal Natural Science Foundation (No. ZZ2018J21).

8. References
[1] Sun, T., Zhang, Y., and Trappe, W., Improving Access Point Association Protocols Through Channel Utilization and Adaptive Switching, IEEE Educational Activities Department, 15(5), pp.1157-1167, 2016.
[2] FRD Sa, AMDCunha, and CDAC Cesar, Effective AP Association in SDWN Based on Signal Strength and Occupancy Rate, Local Computer Networks, pp.159-162, 2017.
[3] B. Karimi, J. Liu, and J. Rexford, Optimal Collaborative Access Point Association in Wireless Networks, in Proc. IEEE Infocom, pp.1141-1149, 2014.
[4] R Zhang, Y Cui, H Clausen, H Haas, L Hanzo, Anticipatory Association for Indoor Visible Light Communications: Light, Follow Me!, IEEE Transactions on Wireless Communications, Vol.17, issue 4, pp.2499-2510, Apr. 2018.
[5] M. Chen, S. Liew, Z. Shao, and C. Kai, Markov Approximation for Combinatorial Network Optimization, IEEE Transactions on Information Theory, vol.59, no.10, pp.6301-6327, Oct. 2013.
[6] https://pan.baidu.com/s/1hi78X26waN6YRNN1e6VqJw
[7] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, Joint VM Placement and Routing for Data Center Traffic Engineering, in Proc. IEEE INFOCOM, pp.2876-2880, 2012.