A Decentralized Scheme for Load Balancing in IEEE 802.11 Wireless Networks

Madyan Alsenwi and Choong Seon Hong
Department of Computer Science and Engineering, Kyung Hee University, South Korea
malsenwi@khu.ac.kr, cshong@khu.ac.kr

Abstract

In this paper, a decentralized method for Load Balancing in IEEE 802.11 wireless LANs is proposed. In this proposed method, users autonomously select the most appropriate Access Point (AP) by sensing available APs, and selecting the best one. The Hopfield Neural Networks are used which is an autonomous and decentralized optimization technique. The Hopfield Neural Networks can be used in optimization by investing the decreasing property of energy function of the Hopfield Networks. We run the proposed method on mobile terminal to select the AP that maximizes the throughput and of the device and keeps a load balancing between all users. Numerical simulations show the validity of the proposed algorithm.

1. Introduction

IEEE 802.11 wireless LANs are widely deployed these days and mobile users are most times located in places where many APs are available such as university campuses and airports. The throughput that the user can get may vary from Ap to other and this depends on the traffic load and propagation environment. as a consequence, users want to select the AP which will give them the highest throughput.
The most common approach is selecting the AP based only on Received Signal Strength (RSS). The main drawback of this scheme is the AP which is the closest to the most of the users becomes overloaded, and the others stay underutilized. Therefore, selecting an AP based only on RSS can cause imbalanced between APs. Many ideas have been proposed to overcome the problem of AP selection Wireless LANs [1].
In [2] an AP selection scheme by exploiting the Retry field in the MAC header as feedback about channel conditions is proposed. Each mobile user can select an AP in terms of expected throughput.
In [3] the mobile user observes and estimates the frame delays and uses these results to estimate the traffic loads. Depending on the traffic loads, the users can determine which AP will select.
In [4] the mobile user learns how to select an AP from his past experience. In this approach, measurements based on conditions of the past link are collected. Based on a Neural Networks trained, the mobile user selects the AP.
In this paper, a decentralized algorithm for AP selection in wireless LANs, IEEE 802.11, is proposed. In this proposed algorithm, a user can select the best AP autonomously. The Hopfield Neural Networks are used which is a distributed optimization technique. In this approach, a user selects the AP which gives high throughput and keeps load balancing in the network. Our contribution in this paper is to use real time AP selection algorithm.

2. Proposed Scheme

A mathematical model for the AP selection problem is introduced in this section. In addition, an introduction to the Hopfield Neural Networks and how we can use it to solve the AP selection problem.

2.1 Optimization Problem

We assume that there are number of APs connected to the Internet as shown in Figure (1). we can achieve the optimum state of balancing if all users have equal throughput. The total capacity is constant and we assume that all communicating users utilize all resources:

$$\sum_{j=1}^{n} C_j = \sum_{i=1}^{m} T_i = const. \qquad (1)$$

Where C_j is the j^{th} AP capacity, T_i is the throughput of the i^{th} mobile user, n is the number of APs, and m is the number of the communicating users. Therefore, we can achieve the optimum state by minimizing the following equation:

$$F_{obj}(T) = \sum_{i=1}^{m} \frac{1}{T_i} \qquad (2)$$

While satisfying $\sum_{i \in S_j} T_i < C_j$ for all j, where S_j is the set of mobile users connected to the j^{th} AP. If all $T_i (i = 1, ..., m)$ become equal, we can achieve the smallest value of $F_{obj}(T)$ [5]. So, differences between T_i will be minimized by minimizing $F_{obj}(T)$, under the condition $\sum_{i=1}^{m} T_i = const.$

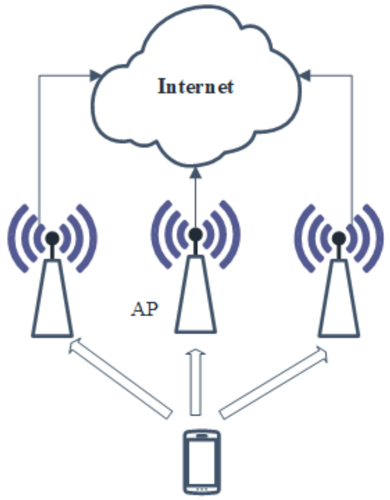

Figure 1: The proposed model

Also, we assume all the mobile users connected to the same AP have equal throughput. Therefore, we can write the throughput of each user as $T_i = \frac{C_{L(i)}}{S_{L(i)}}$, where S_j is the number of mobile users connected to j^{th} AP, and $L(i)$ is the AP which the mobile user i is connected. According to the previous assumptions, we can write the traffic load balancing problem as a minimization of the following equation:

$$F_{obj}(L) = \sum_{i=1}^{m} \frac{S_{L(i)}}{C_{L(i)}} \tag{3}$$

Equation 3 depends on the vector L, and this vector contains the list of APs connected to each mobile user. The number of combinations in this combinatorial optimization problem becomes n^m.

2.2 Hopfield Neural Networks

Each neuron in Hopfield Networks updates its state by the following equation [6]:

$$x_{ij}(t+1) = \begin{cases} 1 & for \sum_{k=1}^{m} \sum_{l=1}^{m} w_{ijkl} x_{ij}(t) > \theta_{ij} \\ 0 & Otherwise \end{cases} \tag{4}$$

Where w_{ijkl} is the connection weight between (k,l) and (i,j) neurons, θ_{ij} is the threshold of the (i,j) neuron, and $x_{ij}(t)$ is the state of the (i,j) neuron.
If we put $w_{ijkl} = w_{klij}$, $w_{ijij} = 0$, and updating asynchronously for each neuron the network will reach to steady state which meets local minimum of its energy function $E(X)$ [6].

$$E(X) = -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{n} w_{ijkl} x_{ij} x_{kl} + \sum_{i=1}^{m} \sum_{j=1}^{n} \theta_{ij} x_{ij} \tag{5}$$

This energy function always decreases and converges to local minimum by updating the neurons state [6]. Therefore, by updating the neurons state we can achieve state X which correspond to the minimum of $E(X)$.
We can use this minimization property to solve the optimization problems by defining the state of the optimization problem by the neurons state x_{ij}, and expressing the objective function in terms of neuron states. We can easily determine the connection weights w_{ijkl} and threshold θ_{ij} by comparing the energy function with the objective function.
By using n neurons for each mobile user i, we can express the state of AP selection for each user by the firing pattern of neural networks where n is the number of APs. Firing the j^{th} neuron in mobile user i means that the mobile user i is connected to the j^{th} AP.
For constructing the network that solve the load balancing problem, we need to define the objective function in term of $x_{ij}(t)$.
We can express $\frac{S_{L(i)}}{C_{L(i)}}$ using $x_{ij}(t)$ as follows:

$$\frac{S_{L(i)}}{C_{L(i)}} = \sum_{j=1}^{n} \frac{S_j}{C_j} x_{ij}(t) \tag{6}$$

Also, S_j can be expressed with $x_{ij}(t)$,

$$S_j(t) = \sum_{k=1}^{m} x_{kj}(t) \tag{7}$$

By substituting these two equation in equation (3), we can express the objective function as a function of neuron states x_{ij}:

$$F_{obj}(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{n} \frac{\delta_{jl}}{C_j} x_{ij}(t) x_{kl}(t) \tag{8}$$

Where $\delta_{ij} = 1$ when $i = j$, and $\delta_{ij} = 0$ otherwise.
Under the constraint that $x_{ij} = 0$ for $j \notin A_i$. By comparing $F_{obj}(X)$ in equation (8) with the energy function $E(X)$ in equation (5) , the connection weights W_{ijkl} and the threshold θ_{ij} can be obtained as follows:

$$W_{ijkl} = -2 \times \frac{\delta_{jl}}{C_j} \tag{9}$$

$$\theta_{ij} = 0 \tag{10}$$

3. Simulation and Results

The proposed model is studied using MATLAB. A network composed from multiple APs is assumed.
First, every incoming user discovers the available APs and then makes a connection with an AP randomly. The incoming user need to get the states of all other neurons (i.e state of all connected users) in order to run the proposed algorithm. In this algorithm, the incoming user can get the state

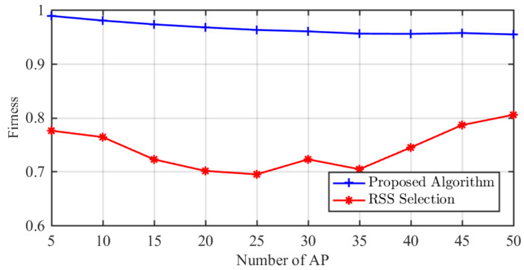

Figure 2: Throughput fairness index with number of APs in case of number of mobile users=100

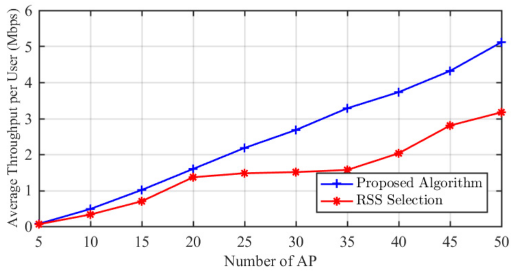

Figure 3: The average throughput per user with number of APs in case of number of mobile users=100

of each neuron from the available APs. Each user obtains the available resources in each AP using the collected information [7, 8] and updates their neurons based on the available resources in each AP. Finally, the user selects the best AP based on the updated states of the neurons, and hands over his connection to the selected AP.
The fairness of throughput and the average throughput per mobile user are evaluated to check the validity of the proposed algorithm. Jain's fairness index [9] is used to evaluate the fairness which is given by the following equation:

$$f(T_1, ..., T_m) = \frac{(\sum_{i=1}^{m} T_i)^2}{m \sum_{i=1}^{m} T_i^2} \qquad (11)$$

The fairness index takes values between 0 and 1 (i.e 0.9 means that it is fair to 90% of the mobile users). The proposed scheme is compared with the RSS selection.

In RSS selection scheme, a mobile user selects the AP which has the strongest signal. Figure (2) shows that the value of fairness index in case of the proposed algorithm is about 0.95 and in case of RSS selection is about 0.75. This means that in case of the proposed algorithm all users have approximately the same throughput and there is a very good traffic load balancing in the system. By using the proposed algorithm all resources in the system are distributed equally among mobile users and there is less traffic congestion compared with RSS scheme. So, the average throughput per mobile user in the proposed scheme is better than that in the RSS selection, as shown in figure (3).
Simulation results show that the proposed scheme has higher throughput and higher fairness, and it's a decentralized implementation. So, we can apply it in the case of a centralized server is not available.

4. Conclusion

In this paper, an optimization algorithm is proposed for AP selection in wireless LANs based on Hopfield Neural Networks.
First, a load balancing problem is defrined to avoid the traffic congestion in the network. A distributed algorithm based on Hopfield Neural Network is introduced. By mapping the objective function of load balancing problem into the form of the energy function of Hopfield Networks the user throughput is optimized directly. The proposed algorithm is compared with RSS selection in order to check the effectiveness of the proposed algorithm.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (NRF-2017R1A2A2A05000995). *Dr. CS Hong is the corresponding author.

References

[1] J. B. Ernst, S. Kremer, and J. J. P. C. Rodrigues, "A utility based access point selection method for IEEE 802.11 wireless networks with enhanced quality of experience," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, jun 2014.

[2] J.-w. Jang, Y.-s. Lim, and C.-k. Kim, "Traffic-aware decentralized ap selection for multi-rate in wlans," in *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, 2010.

[3] J.-C. Chen, T.-C. Chen, T. Zhang, and E. van den Berg, "WLC19-4: Effective AP selection and load balancing in IEEE 802.11 wireless LANs," in *IEEE Globecom 2006*. IEEE, nov 2006.

[4] B. Bojovic, N. Baldo, and P. Dini, "A neural network based cognitive engine for IEEE 802.11 WLAN access point selection," in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, jan 2012.

[5] M. Alsenwi, H. Elsayed, and M. Elghoneimy, "Optimization of channel selection in cognitive heterogeneous wireless networks," in *2015 11th International Computer Engineering Conference (ICENCO)*. IEEE, dec 2015.

[6] R. Rojas, *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[7] Y. Tay and K. C. Chua, "A capacity analysis for the ieee 802.11 mac protocol," *Wireless networks*, vol. 7, no. 2, pp. 159–171, 2001.

[8] Q. Ni, I. Aad, C. Barakat, and T. Turletti, "Modeling and analysis of slow CW decrease for IEEE 802.11 WLAN," in *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003.* IEEE.

[9] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984, vol. 38.