

A Resource Allocation Method for Wireless Network with Multiple APs

Chunlin Li, Yuejiang Huang, Hong Wang and Changxi Niu

No.30 Institute of CETC, Chengdu 610041

Li.chl@foxmail.com

Abstract

When an amount of users are accessing the wireless network with multiple Access Points (APs), a challenge is how to allocate the wireless network resources effectively. A utility function is proposed to evaluate the effectiveness, and a Resource Allocation Algorithm (RAA) is introduced to solve this problem. The algorithm firstly calculates the utility values from a user's request to different available APs. Then it stores the results with request ID and AP ID in a pair. In each allocation time slot, request-AP-utility pairs are arranged in utility value from high to low. Then the available service resources are allocated to the requests properly by a Fast Resource Allocation. Compared with Fair-Method in the same simulation scene, RAA gains 200% performance in delay with only 15% losing of load balance.

Key words: *Wireless network, Network Delay, Resource Allocation, Utility Function*

1. Introduction

The available network band for mobile users via various access technologies has greatly increased in recent years. And protocols like MIP, RSVP are used to establish high quality wireless links. However, as more and more APs appearing in campuses, shopping malls and hotels, the problem of how to allocate the wireless network resources effectively with multiple APs arises.

To allocate the wireless network resources, Yingbin Liang [1] proposed a method for wireless relay channels to improve capacity of the wireless channel. And Rafeeq Ahmed [2] introduced fairness for bandwidth resource allocation. But these methods do not consider the case that a user can access multiple APs. Feiyi Huang [3] discussed the multi-access problem for mesh network, which mainly focus on solving routing and collision free. And a further discussion of handling newly deployed access points is proposed in Suhua TANG's [4] paper. Their solutions do not consider load balance of each AP. SHENG Jie [5] use a load balance access control strategy. His method used logic rules for accessing strategies, which would be quite complex with more factors in consideration. Lin Chen [6] and Museong Park [7] both proposed utility function to evaluate the link between the users and APs. However, their models do not consider the fairness for multiply users, which makes sure each user have chance to acquire the resource. Paolo Valente's [8] discussion of fairness based on a queue model for disk resource allocation is quite enlightening. Moreover, Yasunori Osana's Fair-Method [9] model with fairness and load balance in consideration for cloud computing is even closer to this paper's issue.

Concluding the above discussion, the goal of this paper is allocating the wireless network resources with multiple APs to multiple users. And the goal can be divided into three sub-purposes: fairness to users, load balance for multiple APs, and lower delay for applications. The work of this paper is introducing a utility function based on the Fair-Method model to evaluate these sub-purposes in a single formula, and proposing an algorithm called Resource Allocation Algorithm to allocate the wireless network resources with multiple APs. The following sections discuss the resource allocation

method in detail. In Section 2, a utility function is proposed based on the new model. And in Section 3 the principle of the resource allocation algorithm is introduced. The concrete algorithm is proposed in Section 4. Experiments are taken to verify the algorithm in Section 5. Finally, Section 6 gives the conclusion of this paper.

2. Resource Allocation Utility Function

As referred in Section 1, there are three factors in consideration of resource allocation. Let t represents the delay, and α represents the fairness factor, and β represents the load balance factor. Note that a user's request may time out when delay is too large and a user would not care the delay any more if the delay is smaller than a value. So we introduce ceil and floor threshold of delay. And Researches show that the effect of delay and load balance is nonlinear [11]. Then we give the utility function as following:

$$U(\alpha, t, \beta) = \begin{cases} \alpha e^{-\beta} & t < t_d \\ 0 & t > t_u \\ \alpha e^{-\beta \frac{t_u - t_d}{t_u - t}} & \text{others} \end{cases} \quad (1)$$

Where t_d is floor threshold of delay, t_u is ceil threshold of delay, $t \geq 0$, $\alpha \geq 1$, and $\beta \geq 1$.

The utility function is divided into three conditions. When the delay is small enough (less than t_d), then the effect of delay can be neglected. When the delay is time out (larger than t_u), the link between the user and AP is regard as broken and the utility value is 0. Otherwise, the effect of delay is exponential.

Then we consider the load balance factor β between different APs. The effect of β is also exponential, because the degree of imbalance also has upper and lower bound. For example, the load between two APs has two extremely cases. One is that they have the same load. And the other is that one AP is overload, while the other AP has no load. Obviously, an AP with heavier load has smaller utility. In the utility function, if the load is balanced ($\beta=1$), the balance factor has no effects on the utility. And the bigger β is, the more it affects the utility. The calculation function of β is given by formula (2).

$$\beta = 1 + (r_i - r_0)^2 \quad (2)$$

where $(r_i - r_0)^2$ is balance degree, which is defined in **Definition 1**.

Definition 1. If there are m APs, the i^{th} AP has T_i^P processing resources with C_i^P used, T_i^N network resources with C_i^N used. The balance degree of the cloud data center is $(r_i - r_0)^2$. And $r_i = \max(C_i^P / T_i^P, C_i^N / T_i^N)$, $r_0 = \sum_{i=1}^m r_i / m$. If any AP's load is heavier than others, the balance degree is bigger than others. Then the corresponding utility value is smaller than others. So the AP with smaller balance degree has more chance to allocate the resource. The fairness factor α is defined to ensure each user has the chance to get the resource. Each request's fairness factor is calculated by formula (3).

$$a_i = \begin{cases} 1 & i = 1 \\ a_{i-1} \times 2 & i > 1 \end{cases} \quad (3)$$

Suffix i is the counts of allocating time after a request is sent. When the request is allocated or time out, set i to 0. When a request is failed to be allocated, next allocating time α is doubled. This makes sure a request can be allocated sooner or later. The example in Figure 1 shows how fairness factor changes, where β is set to 1 and the delay is 1 second. Assume that the resources are allocated every second, and the user's request times out in the 4th second after sending the request. When T is 0, the utility value is 0.3679. If the request fails, the utility value doubles to be 0.7358 at $T=1$, which has much more chance to get the resource. And when T is 3, the request can get much higher chance to be allocated the resource.

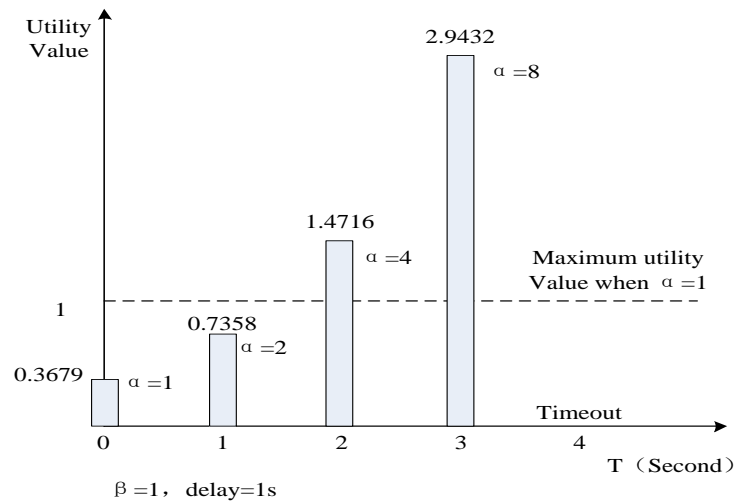


Figure 1. Example of How α Changes

3. Algorithm of the Resource Allocation

Based on the utility function, the goal of resource allocation becomes finding an allocation with the maxim utility. A simple way to allocate the resource is allocating the resources to users one by one in their requesting arriving orders. As mentioned in reference [9], such method may lead low efficiency of resource utility. A better way is combining the requests and available resource together, and then allocating the resources in a proper way. In fact, the requests are changing all the time. So the question becomes how many requests are needed to allocate in each allocating cycle. In this paper, we decide the counts of requests by time slots. In each time slot, we combine the received request and available resources together, and then allocate the resources in three steps.

1. Calculate the utility between each request to each AP, and combine the request ID, AP ID and utility value in a pair.
2. Arrange these pairs in utility value from high to low, and put the result in a queue.
3. Check the queue, and allocate the resources to the request of each pair. If a request has been allocated or the corresponding AP is overload, ignore the request.

An example in Figure 2 illustrates the principle. Assume there are both three resource requests and available APs. Each request has three utility values corresponding to the APs. Following step 1, we get nine utility pairs. Then we sort these pairs in a queue. At last, each request is handled based on the queue. In this example, request 1 and request 2, which are contained in the first two pairs, are allocated to C1 with the maxim utility. The 3rd pair shows that request 3 has the maxim utility with AP C2. But C2 is overload, so the pair is dropped. The 4th and 5th pairs are dropped because request 1 and 2 are already allocated. At the 6th pair, request 3 is allocated to C3. While all the requests are allocated, the left request pairs are dropped.

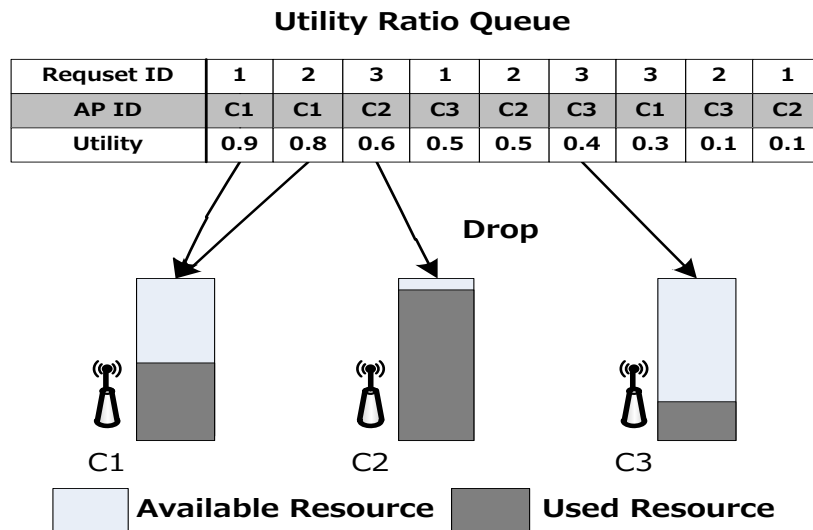


Figure 2. Example of the Resource Allocation Principle

Following the three steps, we give the resource allocation algorithm pseudo-code in **Table 1**.

Table 1. Resource Allocation Algorithm

Input: The user's request User_Apply, the APs AP_Src, the delay between user and the AP Delay Output: allocated resources Src_alloate and unallocated resources Un_Src_allocate 1: $n \leftarrow$ number of the requests, $m \leftarrow$ number of the Aps 2: if User_Apply arrives then 3: for $i \leftarrow 1, n$ 4: for $j \leftarrow 1, m$ 4: Utility(i, j) \leftarrow calculate the utility U(Delay) 5: end for 6: end for 7: Sorted_Utility \leftarrow sort the Utility(i, j) from max to min 8: for $i \leftarrow 1, n * m$ 9: if User_Apply(Sorted_Utility(i)) is not allocated then 10: if AP_Src(Sorted_Utility(i)) is not overload then 11: Src_alloate \leftarrow User_Apply(i) 12: update AP_Src(Sorted_Utility(i)) 13: update User_Apply(Sorted_Utility(i)) 14: else 15: if the request fails to all APs Miss_allocate(Sorted_Utility(i)) $> n - 1$ then 16: Un_Src_allocate \leftarrow User_Apply(i), User_Apply(i).fairness_factor*2 17: else 18: Miss_allocate(Sorted_Utility(i)) \leftarrow Miss_allocate(Sorted_Utility(i))+1 19: end if 20: end if 21: end if 22: end for

The first step is accomplished from line 1 to 6, which monitors the coming requests and calculates the utility values between the request and APs. Line 7 sorts

the utility values from high to low, which is the second step. And the last step of allocating the APs to each request is done from line 8 to 22.

4. Experiment

In the experiment, we propose a simplified mobile network with two servers and three APs. The network topology of the experiment is shown in Figure 3. The delay threshold t_d is 10ms and t_u is 1000ms. The experiment compares RAA's results with Fair-Method to verify its validity.

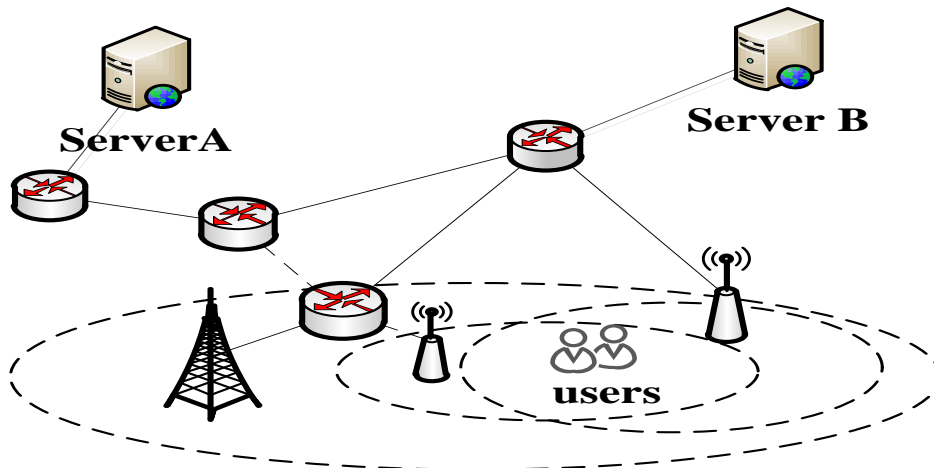


Figure 3. The Network Topology of the Experiment

We use the Jain's measurement of fairness ^[13] to evaluate the fairness. The experiment result is shown in Figure 4. RAA and Fair-Method uses different strategies to ensure the fairness of each user. RAA double the utility value between the unallocated users and APs. So the user can get the AP sooner or later, unless the user cancels the request. This strategy ensures the users' fairness of get the wireless network resources. And the experiment proves it has better achieves better fairness than Fair-Method.

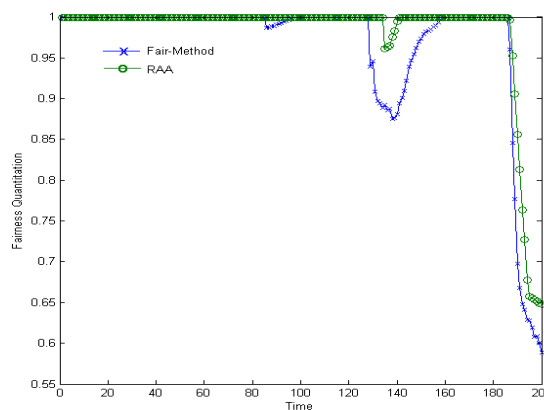


Figure 4. Fairness Quantitation Comparison

The result of the average utility value and delay is shown in Figure 5. The RAA's average delay is about 200ms, while the Fair-Method's is about 600ms. So, the RAA has much better delay performance than Fair-Method. The result correctly reflects the difference between RAA and Fair-Method. While the former algorithm chooses

the AP with the delay as small as possible, and the later algorithm uses a random selecting method.

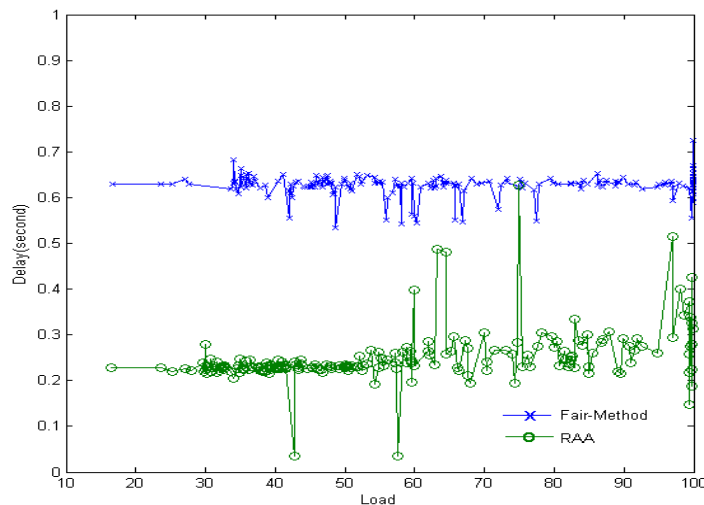


Figure 5. Average Delay under Different APs' Load

Moreover, the result of balance degree is shown in Figure 6. In Fair-Method, each server's load is almost the same, so the balance degree is almost zero. In RAA, the balance degree is wobbling around 0.15. So the RAA's load balance is about 15% worse than Fair-Method. This is because Fair-Method's random selecting strategy ensures the balance of each AP. However, the RAA tries to supply the user an AP with low delay even with heavier load. So, the result is rational.

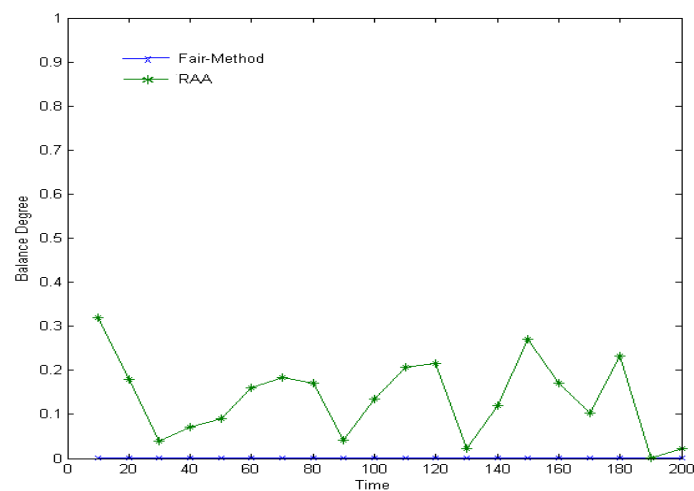


Figure 6. Balance Degree Corporation

Figure 7 show the utility value calculated by formula (1) with RAA and Fair-Method. The result shows that the utility value of RAA is stable at about 0.25, while the utility value of Fair-Method seems random distributed from 0 to 0.4. Such a result is corresponding to comparison of fairness, delay and load balance between the two algorithms. So, we can assert that the utility value function is rational.

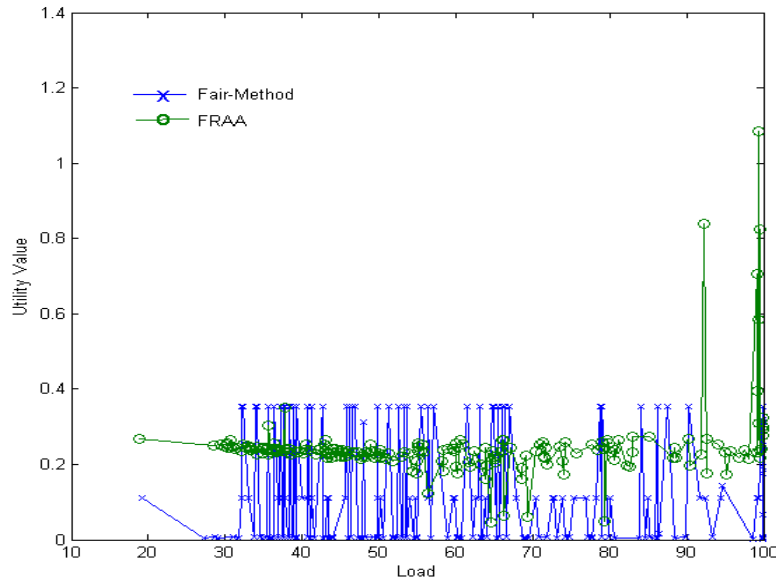


Figure 7. Utility Value Comparison

From these results, we can conclude that the RAA is better than Fair-Method in fairness and delay, but worse in load balance. However, the losing of about 15% load balance performance gains about 200% better delay performance. So, we can draw a conclusion that RAA algorithm achieves the goal of this paper.

5. Conclusion

This paper proposes a method for wireless network resource allocation, which can be used to optimize the utility of wireless network for multiple APs and users. The RAA algorithm is based on a utility function composed of three parameters of delay, load balance and fairness. Experiments verify the validation of this method. And the results are better than the known algorithm Fair-Method.

However, there are other factors that are not considered in this paper, like power controlling, error bit rate and so on. Combining those purposes makes the problem much more complex. The further job of this paper is developing a simple and intelligent way to achieve these purposes.

References

- [1] L. Yingbin and V. V. Veeravalli, "Resource Allocation for Wireless Relay Channels", Signals, SYSTEMS, and Computers 2004, Conference Record of The Thirty-Eight Asilomar Conference, (2004), pp. 1902-1906.
- [2] R. Ahmed, M. A Qadeer and M. Ahmad, "Bandwidth Resource Allocation", 2009 International Conference on Future Computer and Communication, (2009), pp. 344-348.
- [3] F. Huang, Y. Yang and X. Zhang, "Receiver Sense Multiple Access Protocol for Wireless Mesh Access Networks", IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings, (2007), pp. 3764-3769.
- [4] S. Tang, A. Hasegawa and R. Nagareda, *et al.* "Improving Throughput of Wireless LANs with Transmit Power Control and Slotted Channel Access", 2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, (2011), pp. 834-838.
- [5] J. Sheng, L.-r. Tang and J.-h. Hao, "Hybrid Load Balancing Algorithm Based on Service Transformation and Admission Control in Heterogeneous Wireless Networks", ACTA ELECTRONICA SINICA, vol. 41, no. 2, (2013), pp. 321-328.
- [6] L. Chen, J. Huang and D. Liu, "A Location Related Wireless Channel Allocation Algorithm", International Journal of Multimedia and Ubiquitous Engineering, vol. 7, no. 2, (2012), pp. 265-272.

- [7] M. Park and W. Yoon, "Joint Distributed Rate Control and Resource Allocation in Multiradio Multichannel Multihop Wireless Networks with Network Coding", International Journal of Multimedia and Ubiquitous Engineering, vol. 8, no. 3, (2013), pp. 93-102.
- [8] P. Valente and F. Checconi, "High Throughput Disk Scheduling with Fair Bandwidth Distribution", IEEE TRANSACTIONS ON COMPUTERS, vol. 59, no. 9, (2010), pp. 1172-1186.
- [9] T. Tomita and S.-i. Kuribayashi, "Congestion control method with fair resource allocation for cloud computing environments, Communications, Computers and Signal Processing (PacRim), (2011), pp.1-6.
- [10] Y. Osana and S.-i. Kuribayashi, "Enhanced fair joint multiple resource allocation method in all-IP networks", IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, (2010), pp. 163-168.
- [11] Yusuke, S. Zhou and I. SasaseRui, "Packet Scheduling Algorithm with Modified Time-Utility Function to Improve NRT Throughput in OFDMA System, (2007), pp. 545-548.
- [12] L. Tang, J. Sheng and Qi Bing, "Dynamic Load Balancing in Heterogeneous Integrated Communication Networks Oriented to Smart Distribution Grid", Proceedings of the CSEE, vol. 33, no.1, (2013), pp. 39-49.
- [13] R. Jain, D. Chiu and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems, DEC Research Report TR-301, (1984), <http://www.cs.wustl.edu/~jain/papers/ftp/fairness.pdf>.

论文题目	A Resource Allocation Method for Wireless Network with Multiple APs		
所属主题	移动网络		
第一作者			
姓名	李春林	职称/学位	博士研究生
单位	中国电子科集团公司第三十研究所	邮编	610041
地址	成都市高新区创业路6号22分箱		
电话	028-85169625	手机	15928841773
Email	Li.chl@foxmail.com		
第二作者			
姓名	黄月江	职称/学位	研究员, 博导
单位	中国电子科集团公司第三十研究所	邮编	610041
地址	成都市高新区创业路6号		
电话	028-85169001	手机	13908013872
Email	Huangyuejiang@30wish.com		