An Approximation Downlink Bandwidth Allocation Scheme for IEEE 802.16 OFDMA System

Jang-Ping Sheu
Department of Computer Science
National Tsing Hua University
Hsinchu, 30013, Taiwan
sheujp@cs.nthu.edu.tw

Chen-Hao Ko
Department of Computer Science
National Tsing Hua University
Hsinchu, 30013, Taiwan
s9962532@m99.nthu.edu.tw

Chuang Ma
Department of Computer Science
National Tsing Hua University
Hsinchu, 30013, Taiwan
machuang@mx.nthu.edu.tw

***Abstract*—Recently, Orthogonal Frequency Division Multiple Access (OFDMA) transmission technique is applied widely in wireless networks because of its high transmission capacity. The IEEE 802.16 standard has also adopted the OFDMA as its access technique. However, the problem of bandwidth resource allocation in time and frequency is essential for efficient utilization of OFDMA system. In this paper, we proposed an approximation resource allocation scheme to improve the downlink bandwidth utilization. In the resource allocation scheme, we sort the requests of users and allocate bandwidth based on dynamic programming strategy which can save the calculation result of subproblems to reduce the executive time of algorithm. In simulations, it is shown that our scheme outperforms the greedy algorithm in bandwidth utilization. Also, we compare our scheme with optimal allocation method, which is implemented by brute force with branch and bound algorithm. The results show that our method outperforms optimal allocation method by bandwidth utilization, stability, and execution time.**

Keywords- IEEE 802.16, OFDMA, resource allocation, wireless networks

I. INTRODUCTION

Nowadays, the demands of broadband wireless access networks increase gradually in many application areas. However, the existing broadband wireless access networks are unable to satisfy fully requirements of most users who need long-distance and high-speed transmission. The bandwidth, transmission distance and number of users are often restricted in achievement of efficient applications and become the bottleneck of communications especially in the harsh outdoor environment. In order to improve the data transmission efficiency, Orthogonal Frequency Division Multiple Access (OFDMA) is introduced in IEEE 802.16 standard, which is a multiple access technique based on Orthogonal Frequency Division Multiplexing (OFDM) [1]. OFDM uses a large number of adjacent orthogonal subcarriers to transfer data symbols concurrently, and it has been evaluated and used efficiently in different transmission applications, due to its high-speed data transmission capacity and high efficient communication against the frequency selective fading [2]. As a combination of the frequency division and time division multiple access, OFDMA is more suitable and available in broadband wireless access networks because of its scalability and MIMO (Multiple-Input Multiple-Out-put) friendliness.

In data transmission, the Base Station (BS) takes charge of allocating available bandwidth resource to Mobile Stations (MSs) by downlink (DL) and uplink (UL). Based on IEEE 802.16 standard, time domain is segmented into frames with fixed duration, and a frame consists of a number of OFDM symbols. The frequency domain is partitioned into several sub-channels. We can use allocation of bandwidth resource to choose the OFDM symbols and sub-channels, which can be used by a DL request. However, the shape of request burst is limited to be rectangle in OFDMA DL frame, when BS determines rectangle regions of bandwidth resource for requests, the occurrence of over-allocated slots and unused slots will waste valuable bandwidth resource and reduce the efficiency of DL bandwidth utilization.

In this paper, we address the one-dimensional bandwidth resource allocation problem, because one-dimensional bandwidth allocation strategy does not allocate request bursts across different sub-channels and there will be no wastage of over-allocated slots. An approximation algorithm is proposed to solve the problem in polynomial time to improve the DL capacity utilization. We sort all requests of users and allocate requests to the sub-channels from the highest to the lowest transmission capacity. Every sub-channel is filled as fully as possible by adopting dynamic programming to reduce unused slots. To reduce the execution time of algorithm, we store the results of the allocated sub-problems in a table, which can be used by unsolved sub-problems. In performance evaluation, we compare our proposed approximation algorithm with a classic greedy method First Fit Decreasing (FFD) algorithm [3] at first. The simulation results show that the bandwidth utilization of our algorithm is better than FFD algorithm, and our approach reduces the wastage effectively without more control overhead. Specifically, we compare our algorithm with the optimal allocation solution, which implemented by brute force of branch and bound algorithm. Our algorithm outperforms it by high efficiency in execution time when the number of input requests is big enough and our bandwidth utilization is only 1% worse than the optimal allocation.

The rest of the paper is organized as follows. We review previous related work in Section II. Section III presents the approximation DL bandwidth resource allocation algorithm.

This work was supported by the NSC of Republic of China under grants NSC 100-2221-E-007-005-MY3.

Simulation results and comparisons are given in Section IV, and Section V concludes this paper.

II. RELATED WORK

To reduce bandwidth resource wastage is an important issue in bandwidth resource allocation of wireless networks. However, the IEEE 802.16 standard does not specify the method of bandwidth resource allocation with less bandwidth resource wastage. Thus, many research works devote to design efficient resource allocation algorithms to improve system DL throughput. In [4-6], the authors proposed DL resource allocation mechanisms based on IEEE 802.16 OFDMA system to reduce the resource wastage, increase the DL resource utilization, and improve QoS guarantee, fairness and throughput optimization. In [4], the authors focused on two dimension allocation problem, the strategy used request fragmentation to eliminate unused slots. But just as every coin has two sides, its control overhead will increase when the scheme execute request fragmentation. In [5], the authors formulated the optimization problem that subjected to the constraints on fairness and throughput requirement, and proposed a heuristic method to schedule users' data into the downlink subframe by exploiting multiuser multichannel diversity to guarantee the service and to utilize systems bandwidth wisely. However, the authors ignored the influence of interference and noise. In [6], the authors did not consider the overhead brought from request fragmentation. In addition, these studies ignored the resource wastage of padded bits which is created by assigning the requests to the inappropriate sub-channels. In order to reduce padded wastage, we have to allocate requests to appropriate sub-channels in the pattern of total allocated slots being close to the sum of requests size. Unfortunately, this bandwidth resource allocation problem in IEEE 802.16 OFDMA system is an NP-hard problem, so that many studies used heuristic method to achieve system throughput optimization.

Recently proposed request burst allocation schemes for throughput maximization can be classified into two categories: channel-unaware schemes and channel-aware schemes [7]. In channel-unaware schemes, the schedulers use no information of the channel state condition in making the scheduling decision. For BS, all the channels have the same bandwidth for every MS, thus they only need to consider how to allocate all of the bandwidth resource for quests of uses. Two-dimensional allocation strategies were proposed, which inclined to select the most appropriate set of requests based on frame length or width. In order to eliminate over-allocated slots, in [8], the authors proposed OBBP scheme to calculate the possible shapes for every request without producing over-allocated slots. For example, if a request needs 14 slots to transfer data where the rectangle shapes possibly are 1×14, 14×1, 2×7, and 7×2, the scheduler will construct a matrix to classify the requests according to their common number of symbols (vertical length) or sub-channels (horizontal length), and allocate resource group by group. This method limited the possibility of shape and the authors did not analyze the complexity of the algorithm. The scheme called by RTS in [9] is similar to OBBP, The main difference is that the possible rectangle shapes of requests are not restricted in RTS, and RTS uses recursion to compute the most suitable allocation for the requests. Although the frame utilization ratio of RTS can be over 95% in simulations, it generates a lot of over-allocated slots when the sizes of requests are randomly generated. In [10], the authors turned the packing of OFDMA frame into a one-dimensional packing problem and proposed a joint greedy scheduling algorithm for wireless network, which reduced the complexity of resource allocation.

In realistic environment, quality of sub-channels will change according to the change of time, locations, and the distance between BS and MS. Then BS checks every MS's sub-channel quality in the feedback information, and assigns the most appropriate sub-channel to the MS. So, the channel condition is taken into account in order to make the allocation decision optimal and efficient in channel-aware schemes. Typically, there are two kind of different strategies in channel-aware schemes. The first kind strategies determine the sequence of request bursts and assign sequentially the location to each burst, depending on the characteristics of each burst to adjust their positions. Efficient Downlink Bandwidth Allocation Scheme (EDBA) [11] adopted this strategy. It picked and sorted a group of requests in decreasing order, which can use the highest Modulation level and Coding rate Scheme (MCS), based on the method similar to OBBP to determine the possible shapes of burst. Then they found unused slots after allocating the current burst which fit the size of unused slots. Another kind of strategies of channel-aware scheduler allocates sub-channels to requests according to the sub-channels capacities of MSs. However, it is very complex to assign the suitable sub-channels for each MS. Assignment of bandwidth resource for the requests is an NP-hard problem. In [12] and [13], the authors proposed strategies to reduce the wastage, but the total of unused slots and over-allocated slots in their simulations still exceeds 20% of all bandwidth resource.

III. THE PROPOSED SCHEME

Instead of considering the two-dimensional allocation strategy mentioned in related works, we focus on one-dimension allocation problem, where there are no over-allocated slots. We adopt one-dimensional allocation strategy under the channel-aware environment in order to optimize system throughput. Although one-dimensional allocation strategy can decrease many allocations possibilities compared to two-dimensional ones, it can reduce the number of unused slots and eliminate over-allocated slots.

A. System Model

We consider an IEEE 802.16 OFDMA system including a BS and n MSs. BS manages bandwidth resource and builds connections to the MSs in backhaul networks. We assume that an MS only generates one request in one frame duration. Thus, there are n requests to be handled in a frame. The DL bandwidth resource of a frame extends in both time and frequency domain. The time domain is fragmented to s slots, and the frequency domain is partitioned to m sub-channels. For each sub-channel, we assume the channel quality is same for all MSs, but different sub-channels may have different channel capacities for MSs.

In DL bandwidth resource allocation problem, we need to determine in which sub-channels are utilized by requests and which slots are occupied in a frame. Because two-dimensional mapping strategies produce easily over-allocated slots, we only consider one-dimensional mapping in this paper, and we will not fragment requests. Based on the above assumptions, we can give a formal statement of DL bandwidth allocation problem as follows.

We assume that each frame of a sub-channel has s time slots, and there are m sub-channels Ch_1, Ch_2, …, Ch_m which are sorted in decreasing order by their slot capacities V_1, V_2, …, V_m (Unit: bit/slot). Given a set of requests R including n elements R_1, R_2, …, R_n with different requiring data size d_1, d_2,…, d_n (Unit: bit), respectively. The number of slots, which was allocated to request R_i on sub-channel Ch_j is denoted by $c_{i,j}$, for $1 \le i \le n$ and $1 \le j \le m$, where $c_{i,j} = \lceil d_i/V_j \rceil$. The E_j denotes the set of requests that are allocated to Ch_j and D_j denotes the data size summation of requests in E_j.

The one-dimensional DL bandwidth allocation problem is to find an allocation to maximize the total data transmission size $D = \sum_{j=1}^{m} D_j$, on the condition of $\sum_{i=1}^{n} c_{i,j}\, b_{i,j} \le s$, where $b_{i,j} = 1$ if R_i is assigned to sub-channel j, otherwise $b_{i,j} = 0$, for $1 \le j \le m$. For example, there are 6 slots in a frame and two available sub-channels Ch_1 and Ch_2, their bandwidth are 15 bits/slot and 10 bits/slot, respectively. There are three requests R_1, R_2, and R_3 for resource scheduling and their request sizes are 51, 35, and 27 bits, respectively. We can allocate R_1 and R_3 to Ch_1 because R_1 requires $\lceil 51/15 \rceil = 4$ slots and R_3 requires $\lceil 27/15 \rceil = 2$ slots, the summation of required slot is 6 which is equal to a frame length. As a result, we can obtain $E_1 = \{R_1, R_3\}$ and $D_1 = 51 + 27 = 78$ bits. Similarly, we have $E_2 = \{R_2\}$ and $D_2 = 35$ bits.

In one-dimensional DL bandwidth allocation problem, we must find an allocation strategy to maximize the total data transmission size. In order to discuss the complexity of one-dimensional DL bandwidth allocation problem, we will consider the Bin packing problem [3] firstly. In Bin packing problem, objects of different sizes have to be packed into the fixed-capacity bins and we try to minimize the number of used bins. Similarly, the sub-channels in bandwidth allocation problem correspond to the bins, and the requests correspond to the objects. Differently, the sub-channels have different channel capacities. If we simplify the resource allocation problem, assuming all sub-channels have same channel capacity, our resource allocation problem can reduce to the Bin packing problem. Because Bin packing problem is NP-Complete, one-dimensional DL bandwidth allocation problem is NP-hard for its more difficulty from different channel capacities than Bin packing problem.

B. Approximation DL Bandwidth Allocation Algorithm

The one-dimensional DL bandwidth allocation problem is a NP-hard problem, so it is difficult to find an optimal solution. As a practical scheme, an Approximation One-dimensional Downlink Bandwidth Allocation algorithm (AOBA) with polynomial time complexity is proposed to solve this problem in this paper. In order to increase the utilization efficiency of DL bandwidth and reduce wastage as much as possible, we allocate each sub-channel for requests one by one. We assume that the width of every burst is one, so there will be no redundant slots which are used to satisfy the shape constraint. To reduce the number of unused slots, we use the dynamic programming strategy to find individually the most suitable requests set for every sub-channel.

Initially, the scheduler will sort sub-channels in decreasing order of transmission capacities and calculate the slots number for each request in sub-channels. In order to maximize the transmission throughput, we allocate requests to the sub-channels from the highest to the lowest transmission capacity. Let R be the set of n requests. At first, we will use dynamic programming to allocate requests to the sub-channel which has the largest transmission capacity. After that, the requests set R will deduct the requests which have been allocated, and then we will allocate the remaining requests for next sub-channel with the second largest transmission capacity. The requests allocation procedure will be repeated until there is no unallocated request or available bandwidth resource.

We define CA(R, i, k, Ch_j) as a function to compute the optimal solution of allocating requests in R to sub-channel Ch_j from the ith request to the last request if there are k remaining slots in Ch_j. $|R|$ is the number of requests in R. When $i > |R|$, it represents that all requests in R are checked whether can be allocated to channel Ch_j or not, so CA(R, i, k, Ch_j) = 0; when $k = 0$, it represents that there is no remaining resource, also CA(R, i, k, Ch_j) = 0. If the number of remaining slots k is less than the requirement of R_i, we will skip R_i and check next request R_{i+1}. If k is greater than the requirement of R_i, the optimal solution is equal to either allocate R_i to Ch_j or not. Thus, the optimal substructure of the resource allocation for a Ch_j can be represented as the following recursive formula.

$$\mathrm{CA}(R, i, k, Ch_j) = \begin{cases} 0 & \text{if } i > |R| \text{ or } k = 0 \\ \mathrm{CA}(R, i+1, k, Ch_j) & \text{if } k < c_{i,j} \\ \mathrm{MAX}\big(\mathrm{CA}(R, i+1, k, Ch_j), \mathrm{CA}(R, i+1, k - c_{i,j}, Ch_j) + d_i\big) & \text{if } k \ge c_{i,j} \end{cases} \quad (1)$$

AOBA algorithm will execute the CA(R, i, k, Ch_j) function m times for m sub-channels until there is no unallocated requests or available bandwidth resource.

For example, we assume that each frame has 6 slots and there are two available sub-channels Ch_1 and Ch_2 with capacities of 15 bits/slot and 10 bits/slot, respectively. There are three requests R_1, R_2, and R_3 with data sizes of 51, 35 and 27 bits, respectively. We can calculate the number of slots required to allocate to each request. On Ch_1, they are $c_{1,1} = 4$, $c_{2,1} = 3$, and $c_{3,1} = 2$, and on Ch_2, they are $c_{1,2} = 6$, $c_{2,2} = 4$, and $c_{3,2} = 3$. Firstly, we will compute CA($\{R_1, R_2, R_3\}$, 1, 6, Ch_1). Because the number of remaining slots in Ch_1 is 6 and larger than the require slots of R_1 which is 4, the optimal solution is equal to the maximum of CA($\{R_1, R_2, R_3\}$, 2, 6, Ch_1) and CA($\{R_1, R_2, R_3\}$, 2, 2, Ch_1) + 51. We will compute recursively CA($\{R_1, R_2, R_3\}$, 2, 6, Ch_1) and CA($\{R_1, R_2, R_3\}$, 2, 2, Ch_1), respectively. Similarly, the optimal solution of CA($\{R_1, R_2, R_3\}$, 2, 6, Ch_1) is equal to the maximum of CA($\{R_1, R_2, R_3\}$, 3, 6, Ch_1) and CA($\{R_1, R_2, R_3\}$, 3, 3, Ch_1) + 35. Because the optimal solutions of CA($\{R_1, R_2, R_3\}$, 3, 6, Ch_1) and CA($\{R_1, R_2, R_3\}$, 3, 3, Ch_1) are both equal to 27, the optimal solution of CA($\{R_1, R_2, R_3\}$, 2, 6, Ch_1) is CA($\{R_1, R_2, R_3\}$, 3, 3, Ch_1) + 35 = 27 + 35 = 62. Using the same method, we can find the optimal solution of CA($\{R_1, R_2, R_3\}$, 2, 2, Ch_1) is also 27. Therefore, the optimal solution of CA($\{R_1, R_2, R_3\}$, 1, 6, Ch_1) is equal to CA($\{R_1, R_2, R_3\}$, 2, 2, Ch_1) + 51, which is 27 + 51 = 78. So, AOBA will allocate R_1 and R_3 to Ch_1 which can transmit 78 bits in a frame. The recursive tree for the computation of CA($\{R_1, R_2, R_3\}$, 1, 6, Ch_1) is shown in Fig.1.

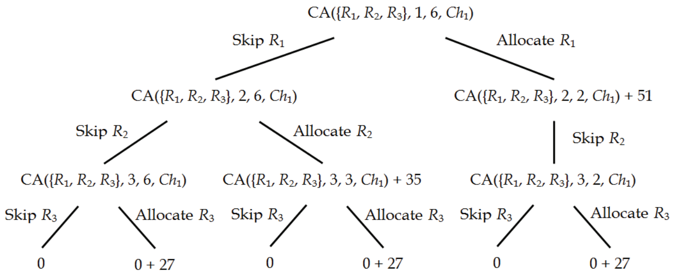

Figure 1: The recursive tree of CA($\{R_1, R_2, R_3\}$, 1, 6, Ch_1)

After the channel allocation of Ch_1, we will remove the requests allocated to Ch_1 from the set of requests R. Then, we will apply the same allocation procedure for $R = \{R_2\}$ on sub-channel Ch_2 by calculating CA($\{R_2\}$, 1, 6, Ch_2). Finally, we will allocate R_2 to Ch_2 which can transmit 35 bits in a frame.

In the recursive function in (1), we do not need to count all sub-problems because many of them have same results. Actually, the total number of distinct sub-problems in (1) is a polynomial on input size n. We store the results of the solved sub-problems in a table, which can be used by unsolved sub-problems. The table size is $n \times s$, n and s are the number of requests and slots in a frame, respectively. Let *table*(i, k) be an array with three fields *table.ans*(i, k), *table.request*(i, k), and *table.reqslots*(i, k), where *table.ans*(i, k) stores the optimal result of CA(R, i, k, Ch_j), *table.request*(i, k) stores the set of requests allocated to Ch_j, and *table.reqslots*(i, k) stores the number of required slots for the requests in *table.request*(i, k), for $1 \leq i \leq n$ and $1 \leq k \leq s$. For example, when we find the solution of CA($\{R_1, R_2, R_3\}$, 3, 6, Ch_1) is 27, we have *table.ans*(3, 6) = 27, *table.request*(3, 6) = $\{R_3\}$, and *table.reqslots*(3, 6) = 2 as shown in TABLE 1.

TABLE 1: THE COMPUTATION RESULTS OF SUB-PROBLEMS

	1	2	3	4	5	6
$i = 1$	-	-	-	-	-	(78, $\{R_1, R_3\}$, 6)
$i = 2$	-	(27, $\{R_3\}$, 2)	-	-	(62, $\{R_2, R_3\}$, 5)	(62, $\{R_2, R_3\}$, 5)
$i = 3$	-	(27, $\{R_3\}$, 2)	(27, $\{R_3\}$, 2)	(27, $\{R_3\}$, 2)	(27, $\{R_3\}$, 2)	(27, $\{R_3\}$, 2)

- : null

It means that the maximal bandwidth allocation of CA($\{R_1, R_2, R_3\}$, 3, 6, Ch_1) is 27 bits/frame by allocating request R_3 to Ch_1, which required 2 slots. Because we only need 2 slots for R_3 in Ch_1, we can also copy *table*(3, 6) to *table*(3, 5), *table*(3, 4), *table*(3,3), and *table*(3, 2). Consequently, we can look up the table and obtain the result of sub-problems CA($\{R_1, R_2, R_3\}$, 3, 3, Ch_1) and CA($\{R_1, R_2, R_3\}$, 3, 2, Ch_1) from *table*(3, 3) and *table*(3, 2), respectively. After that, we can obtain the result of CA($\{R_1, R_2, R_3\}$, 2, 6, Ch_1) = 27 + 35 = 62, but without the calculation of CA($\{R_1, R_2, R_3\}$, 3, 3, Ch_1) and CA($\{R_1, R_2, R_3\}$, 3, 2, Ch_1). The number of required slots is 2 + 3 = 5. So, we can copy the content of *table*(2, 6) into *table*(2, 5). Finally, the optimal result of allocating requests to Ch_1 is 78. Similarly, we can solve the CA($\{R_2\}$, 1, 6, Ch_2) in the same way and have the optimal result 35. Therefore, the total throughput of Ch_1 and Ch_2 is 78 + 35 = 113 bits/frame.

IV. PERFORMANCE EVALUATION

We use the theoretical analysis and simulation to evaluate the performance of AOBA.

A. Throughput Analysis

We give a theorem to show the relationship between the throughput lower bound of AOBA and the transmission data size of optimal solution.

Theorem 1: The total transmission data size of AOBA, denoted by D, satisfies $D \geq 1/2\ OPT$, where OPT represents the transmission data size in optimal solution.

Proof: If AOBA can allocate all requests to m sub-channels, AOBA can achieve the same performance as the optimal solution. So, we only consider that AOBA cannot allocate all requests to m sub-channels. Let B_j denote the channel capacity of sub-channel j, where $B_j = s \times V_j$, for $1 \leq j \leq m$. Without loss of generality, we assume $B_1 \geq B_2 \geq \cdots \geq B_m$. Therefore, we allocate requests from the first sub-channel to the mth sub-channel. Because each sub-channel bandwidth allocation is optimal in our algorithm, we have $D_1 \geq D_2 \geq \ldots \geq D_m > 0$, where D_j is the total number of required data size allocated to sub-channel j. In the following, we will prove $D_j \geq 1/2\ B_j$, for $1 \leq j \leq m$.

We assume that there exists j, it is satisfied $D_j < 1/2\ B_j$, for $1 \leq j \leq m - 1$. Because $D_1 \geq D_2 \geq \ldots \geq D_m$, we have $D_{j+1} \leq D_j < 1/2\ B_j$. Since AOBA is optimal for each sub-channel bandwidth allocation, the optimal solution for sub-channel j will satisfy $D_{j+1} + D_j \leq B_j$. So, the requests in the j+1th sub-channel will be filled to the jth sub-channel as shown in Fig. 2, this is a contradiction to our assumption. We assume $D_m < 1/2\ B_m$. If there exists any unallocated R_i ($R_i \in R$) with $d_i \geq 1/2\ B_m$, AOBA will use R_i to fill the sub-channel m instead of the original allocation. It will lead to $D_m \geq 1/2\ B_m$, which contradict with the assumption $D_m < 1/2\ B_m$. If for every unallocated request R_i satisfies $d_i < 1/2\ B_m$, AOBA can fill them to the Ch_m until it satisfies $D_m \geq 1/2\ B_m$. So, this also contradicts with the assumption $D_m < 1/2\ B_m$. As a result, we have $D = \sum_{j=1}^{m} D_j \geq \frac{1}{2}\sum_{j=1}^{m} B_j$, which is $D \geq 1/2\ OPT$. ■

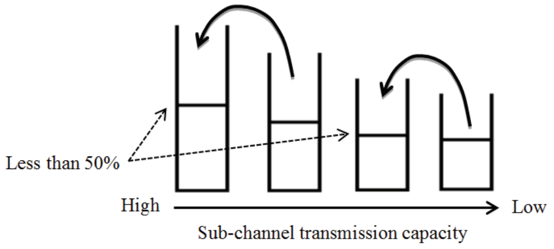

Figure 2: Combining two sub-channels into one sub-channel

B. Computational Complexity

The time complexity of AOBA for each sub-channel is $n \times$

s. Because there are *m* sub-channels, the time complexity of AOBA is *O*(*smn*). Moreover, the number of slots in a frame and the number of sub-channels are constants, so that the time complexity of AOBA can be simplified as *O*(*n*). In addition, we do not need to solve all of the distinct sub-problems as shown in TABLE 1, as a result, the execution time of AOBA is very efficient.

C. *Simulations and Comparison*

We compared AOBA with the First Fit Decreasing (FFD), a classic greedy algorithm for Bin packing problem in resource allocation of networks, and with the optimal solution calculated by brute force with branch and bound algorithm.

1) *Comparison with Greedy Algorithm*

To reduce the wastage caused by padded slots which is created by assigning the requests to the inappropriate sub-channels, we let the greedy algorithm adopted the same strategy as AOBA to allocate requests to the sub-channel with the best channel quality at first. In addition, the greedy algorithm will sort the input requests in decreasing order and allocate the first unallocated request to current sub-channel if the remaining sub-channel bandwidth is enough to meet the request size; otherwise, the scheduler will allocate the second unallocated request to current sub-channel until no request can be allocated. After that, the greedy algorithm will assign the unallocated request to the second best sub-channel with the same strategy until all requests are assigned or no sub-channel resource can be used. The complexity of the greedy algorithm depends on that of the fastest sorting, which is O(*n* log *n*), where *n* is the input number of requests. Since the complexity of AOBA is *O*(*n*), its executive time is better than the greedy algorithm. As a result, we will compare only the bandwidth throughput between greedy algorithm and AOBA.

There are 29 slots which can be used for DL transmission when we use TDD transmission mode. We assume the control message occupies 3 slots of a frame. Therefore, the DL data can use 26 slots for data transmission. We also assume the number of input requests is in the range of 50 to 100, and request sizes are randomly in the range of 1000 to 5000 bits. Other parameters used for simulation refers to the IEEE 802.16 standard. Frame length is 5ms; channel bandwidth is 10MHz; number of sub-channels is 16; DL/UP ratio is 29/18 slots. Number of input requests is in the range of 40 to 100 and simulation time is 10000 frames. Fig. 3 indicates the influence of the requests number on the utilization of bandwidth. It is shown that the bandwidth utilization of AOBA is better than the greedy algorithm by maximum, average, and minimum utilization ratio. Especially in average case, AOBA is 10% better than the greedy method because we sort all requests of users and adopt dynamic programming allocation to reduce unused slots in AOBA. Although the minimum bandwidth utilization of AOBA is low when there are a few requests, it increases rapidly and reaches 98% when the number of requests increases to large enough, such as 90. However, the average bandwidth utilization of the greedy algorithm cannot reach 90% because it has no mechanism to choose the requests which have smaller padded wastage for each sub-channel.

2) *Comparison with Optimal Solution*

The optimal solution is calculated by brute force which tries every possible combination of requests, and accumulated by the branch and bound algorithm. Branch and bound algorithm is a general algorithm for finding optimal solutions of various optimization problems, especially in combinatorial optimization. The scheduler estimates upper bounds for all of the possible sub-problems, and discards a large amount of fruitless candidates which upper bound is lower than the expected value. In simulations, the executive times of AOBA are always less than those of the optimal allocation. Therefore, we can use the result of AOBA as the expected value in optimal allocation, so that we can cut the worthless branch by the expected value, and once the result is greater than the expected value, the scheduler will replace the expected value with the new one.

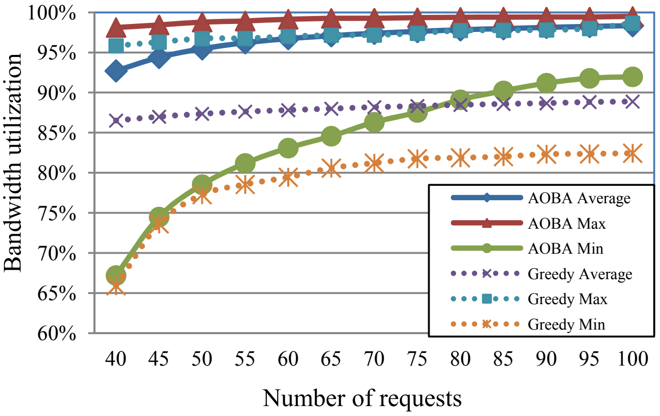

Figure 3: Bandwidth utilization by increasing number of requests

In despite of using the method of branch and bound, the complexity of optimal solution is still very high. Because the brute force try to allocate all combinations of requests to sub-channels, its worst-case complexity is O(m^n) where *m* is the number of sub-channels and *n* is the number of requests. In order to reduce the simulation time to a reasonable range, we change three parameters above for the following simulations: the bandwidth reduces to 2.5 MHz and the input packet size is in the range of 300-2000 bits. Also, the simulation time is decreased to 2000 frames.

In Fig. 4, we compare the execution time between AOBA and the optimal solution. It is shown that AOBA has a great advantage than the optimal allocation in execution time, which is in millisecond. Moreover, when the number of input requests increases, the execution time of optimal solution is exponential growth. By contrast, the polynomial time complexity algorithm AOBA shows its stable and approximate linear complexity in simulations against the exponential complexity of the optimal solution.

In Fig. 5, we compared AOBA with the optimal solution by the bandwidth utilization. It is shown that the maximal bandwidth utilization of AOBA almost coincides with it of the optimal allocation. In average case, the performance of AOBA is very close to optimal allocation. When the number of input requests is small, the combination diversity of the requests will be low. As a result, when we use brute force to try all the possible cases, its result cannot be better than AOBA because fewer requests combinations are chosen.

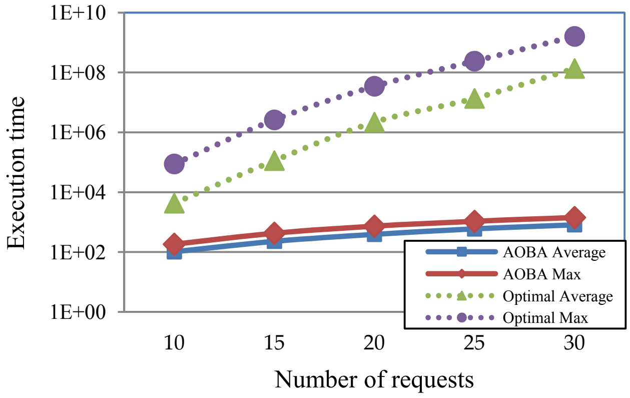

Figure 4: Execution time by increasing number of requests

With increasing of the requests number, all bandwidth utilizations in six lines of Fig. 5 are increased noticeably except the maximum utilizations which are almost unchanged. The average bandwidth utilization of AOBA can reach 95% when the number of input requests is big enough, such as over 25. Noticing that, the average utilization of AOBA is approximate 1% worse than the optimal allocation. Although the optimal solution outperforms AOBA in minimum bandwidth utilization when there are a few input requests, the gap reduce rapidly with the increasing of number of requests.

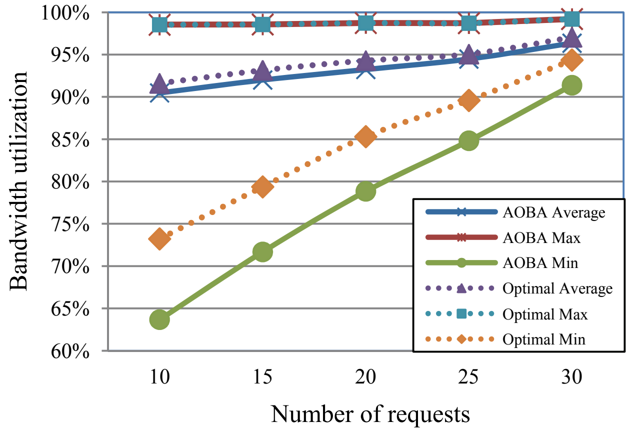

Figure 5: Bandwidth utilization by increasing number of requests

V. CONCLUSION

In broadband wireless access networks, the bandwidth resource allocation problem involves network throughput optimization, wastage reducing, and time complexity. The tradeoff of these three factors in bandwidth resource allocation scheme becomes an essential issue in applications of OFDMA systems. In this paper, we address the resource allocation problem in OFDMA system based on IEEE 802.16 standard. To improve the utilization efficiency of bandwidth resource and balance the throughput, wastage and time complexity, we propose a one-dimensional approximation downlink resource allocation scheme AOBA to optimize communication bandwidth utilization in wireless networks. AOBA uses dynamic programming to reduce unused slots and stores the results of sub-problems to reduce execution time complexity. Moreover, we prove that the throughput lower bound of AOBA is greater than or equal to half of optimal solution. The simulation results show that our proposed scheme can allocate bandwidth resource more efficiently, and the average bandwidth utilization of AOBA is approximate equal to the optimal allocation method. Especially, the average bandwidth utilization of AOBA is greater than 95% with much lower time complexity than the optimal allocation method.

REFERENCES

[1] L. Hanzo, T. Keller, S. Ng, and W. Webb, “Quadrature Amplitude Modulation: From Basics to Adaptive Trellis-Coded, Turbo-Equalised and Space-Time Coded OFDM, CDMA and MC-CDMA Systems,” Wiley-IEEE Press, USA, 2004.

[2] K. S. Sastry, and D. Babu, “Adaptive Modulation for OFDM System Using Fuzzy Logic Interface,” *in Proceedings of IEEE International Conference on Software Engineering and Service Sciences*, pp. 368-371, Beijing, China, July 2010.

[3] D. S. Johnson, “Near-Optimal Bin-Packing Algorithms,” Doctoral Thesis, M.I.T., Cambridge, Mass., UK, 1973.

[4] Y. B. Shimol, I. Kitroser, and Y. Dinitz, “Two-Dimensional Mapping for Wireless OFDMA Systems,” *IEEE Transactions on Broadcasting*, Vol. 52, No. 3, pp. 388-396, Sept. 2006.

[5] N. Ruangchaijatupon, and Y. Ji, “Resource Allocation for Guaranteed Service in OFDMA based Systems,” *in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pp.1-6, Budapest, Hungary, April 2009.

[6] X. Jin, J. Zhou, J. Hu, J. Shi, and Y. Sun, “An Efficient Downlink Data Mapping Algorithm for IEEE802.16e OFDMA Systems,” *in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1-5, New Orleans, LA, USA, Dec. 2008.

[7] C. S. In, R. Jain, and A. K. A. Tamimi, “Resource Allocation in IEEE 802.16 Mobile WiMAX,” *Orthogonal Frequency Division Multiple Access (OFDMA)*, Auerbach Publications CRC Press, April 2010.

[8] O. M. Eshanta, M. Ismail, K. Jumari, “An Efficient Burst Packing Algorithm for OFDMA Systems,” *in Proceedings of Computer Technology and Development Conference (ICCTD)*, pp.456-461, Cairo, Egypt, Nov. 2010.

[9] C. Cicconetti, L. Lenzini, A. Lodi, S. Martello, E. Mingozzi, and M. Monaci, “Efficient Two-dimensional Data Allocation in IEEE 802.16 OFDMA,” *in Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pp. 1-9, San Diego, CA, USA, March 2010.

[10] A. Zubow, J, Marotzke, D. Camps-Mur, and X. Perez-Costa, “sGSA: An SDFMA-OFDMA Scheduling Solution,” *in Proceedings of the 18th European Wireless Conference*, Poznań, Poland, April 2012.

[11] H. C. Chen, K. P. Shih, S. S. Wang, and C. T. Chiang, “An Efficient Downlink Bandwidth Allocation Scheme for Improving Subchannel Utilization in IEEE 802.16e WiMAX Networks,” *in Proceedings of IEEE Vehicular Technology Conference (VTC)*, pp. 1-5, Taipei, Taiwan, May 2010.

[12] K. P. Shih, H. C. Chen, C. T. Chiang, and T. H. Hsieh, “Channel-Aware Subchannel Renumbering and Downlink Burst Allocation for IEEE 802.16 OFDMA Systems,” *in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6, Sydney, Australia, April 2010.

[13] H. C. Chen, K. P. Shih, C. T. Chiang, and C. L. Chen, “A Subchannel-aware Burst Fragmentation, Packing and Scheduling (BFPS) Algorithm for Downlink Traffic in IEEE 802.16 OFDMA Systems,” *in Proceedings of IEEE Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1141-1146, Istanbul, Turkey, July, 2011.