A Data Mapping Algorithm for Two-Level Requests in WiMAX Systems

Tsern-Huei Lee, Chi-Hsien Liu, Arleth Soleiy Garth Campbell
Institute of Communication Engineering
National Chiao Tung University
Hsin-Chu, Taiwan
{tsernhueilee, terence.am91g}@gmail.com

Yaw-Wen Kuo
Department of Electrical Engineering
National Chi-Nan University Taiwan
Puli, Nan-Tou, Taiwan
ywkuo@ncnu.edu.tw

Abstract—**The IEEE 802.16e standard known as Mobile WiMAX has recently been introduced. It is considered as one of the most promising wireless access technologies supporting high data throughput with low cost of deployment. Mobile WiMAX makes use of Orthogonal Frequency-Division Multiple Access (OFDMA) digital modulation scheme as the transmission method. With the constraint requires that all requests have to be mapped as a rectangle, it is shown that finding an optimum mapping solution is an NP-complete problem. Many burst mapping algorithms have been proposed, but none considered the case with prioritized requests. This paper presents an efficient packing algorithm for two-level requests with two targets: (1) map high (MUST part) priority data as much as possible; and (2) achieve high efficiency, reduces the number of unused slots, and minimizes the mapping information overhead. Simulation results show that the proposed algorithm achieves high efficiency.**

Keywords-OFDMA, WiMAX, Burst mapping, Two-level requests

I. INTRODUCTION

IEEE 802.16e Mobile WiMAX constitutes one of the most promising broadband wireless access technologies, supporting long-distance communications, high-speed mobile Internet access to the widest array of devices. As the fourth generation of wireless technology, WiMAX delivers low-cost, open networks and is the first all IP mobile Internet solution enabling efficient and scalable networks for data, video, and voice [1].

Mobile WiMAX makes use of an Orthogonal Frequency Division Multiple Access (OFDMA) which in order to achieve a higher MIMO spectral efficiency, shorter delay, and more efficient use of the bandwidth available [2]. In Mobile WiMAX systems, a base station (BS) has full control over resource allocations to various mobile stations (MSs) in both the downlink (DL) and the uplink (UL).

Mobile WiMAX uses a fixed frame based allocation. Each frame is typically of 5 ms duration and divided into the uplink and the downlink sub-frames. The bi-directional communication can be achieved by frequency division duplexing (FDD) or time division duplexing (TDD). In the FDD, UL and DL use different frequency bands, allowing the simultaneous transmission of both UL and DL sub-frames. The TDD provides a flexible partitioning of the frame into DL and UL sub-frames. This paper considers a TDD system, but the proposed algorithm can be used for both systems.

According to the IEEE 802.16e specification, a data burst is mapped to a two-dimensional data block consisting of a group of contiguous OFDMA slots. An OFDMA slot being the minimum possible data unit is defined in two dimensions of time and frequency (symbol-subchannel). The OFDMA frame starts with a downlink preamble and a frame control header (FCH) followed by the downlink map (DL-MAP) and the uplink map (UL-MAP). Both maps contain the informational elements (IEs) that detail the burst profile for each burst. A DL-MAP IE consists of four primitive parameters: OFDMA symbol offset, OFDMA subchannel offset, number of OFDMA symbols, and number of OFDMA subchannels.

In this paper, we consider the diversity sub-carrier permutation mode, e.g., the Partially Used Sub-Channelization (PUSC) mode, where the sub-carriers forming a subchannel are scattered uniformly over the entire frequency band. For the DL PUSC, it is assumed that the basic data mapping unit is a slot, which is a combination of one subchannel and two OFDMA symbols. According to WiMAX forum specified parameters, with 10 MHz spectrum, there are 30 subchannels each one consisting of 28 subcarriers.

The IEEE 802.16e standard has defined five scheduling services classes with different QoS requirements, including bandwidth, packet loss, delay and delay jitter: Unsolicited Grant Service (UGS), real-time Polling Service (rtPS), extended real-time Polling Service (ertPS), non-real-time Polling Service (nrtPS), and Best Effort (BE) [3] [4]. Each class has different QoS parameters. The system resource is controlled by a scheduler at, which allocates the number of slots to each MS frame by frame. The allocated resources to MSs form a set of requests to the mapping algorithm that maps each request into a rectangular region. With the constraint of rectangular mapping, it is possible that the actual data belonging to a burst cannot fill the whole rectangular region and vacant slots are considered wasted. This mapping process, or called a bin packing problem, is known to be NP-complete [5].

This work was supported in part by the National Science Council, Taiwan, R.O.C., under grant NSC 100-2221-E-009-112 and NSC100-2221-E-260-028.

978-1-4673-0990-5/12/$31.00 ©2012 IEEE

In addition to the waste in allocated bursts, in some cases the remaining slots in the DL sub-frame cannot form a rectangle to fit any unmapped requests, resulting in efficiency degradation. Moreover, it is possible that some urgent data are not mapped at the current frame. To the best of our knowledge, there are no algorithms that consider requests with urgent and non-urgent data (two kinds of data). The majority of the algorithms in the literature only focus on the resources allocation in a best effort manner [9]-[12].

The purpose of this paper is to present and evaluate an efficient downlink data mapping algorithm designed for an IEEE 802.16e single-cell OFDMA-based system which consists of one BS and *N* MSs. The rest of this paper is organized as follows. Section II briefly describes some of the related works in a context of Mobile WiMAX systems. Our proposed data mapping algorithm for the two-level requests in WiMAX systems is described in Section III. The performance evaluation is presented in Section IV. Finally, the conclusions are drawn in Section V.

II. Related Works

In this section, we briefly review some of the other burst mapping algorithms for Mobile WiMAX. Basically, the downlink scheduler generates a set of requests $\boldsymbol{A} = \{A_1, A_2, \cdots, A_N\}$ (in number of slots) to be mapped, which is determined based on current backlog, QoS parameters (delay and throughput), and available capacity. Then, a data mapping algorithm determines the shapes and locations in the sub-frame for these requests.

The simple heuristic mapping algorithm eOCSA [12], which is very similar to its predecessor (OCSA) [11], aims to keep the mapping operational complexity low. eOCSA mapping algorithm schedules each subscriber's request into the DL sub-frame from right to left and from bottom to top. As the first step, the algorithm sorts the set of requests in the descending order. In the second step, known as vertical mapping, the eOCSA algorithm allocates the biggest request, say j, to a rectangle with width W^* and height H^*, and $W^* = \lceil A_j/H \rceil$ and $H^* = \lceil A_j/W^* \rceil$ where H is the maximum height that can be used for allocation. The remaining unallocated space is handled in the third step, during which the horizontal mapping takes place, and where the algorithm looks for the next largest request to be allocated that can fit into the space left on top of the burst mapped in the previous step 2. But here, the widths of all following bursts allocated in the strip are fixed to that of the burst allocated in the previous step. This process is repeated until no space can be allocated or if there is no resource that can fit into this space. Next, the algorithm moves leftward to fill the remaining empty columns in the DL sub-frame and repeats from step 2. Figure 1. shows an example of allocation for eOCSA where the set of requests to be allocated is {92, 76, 65, 48, 25, 20, 17, 10, 5, 2}. Black represents over allocated slots, and gray represents unused slots. The eOCSA algorithm leaves 6 empty slots and over allocates 11 slots.

Note that most of the bursts allocated by the eOCSA algorithms have the least widths, which implies that the active time and consequently, the energy consumption, of each MS is

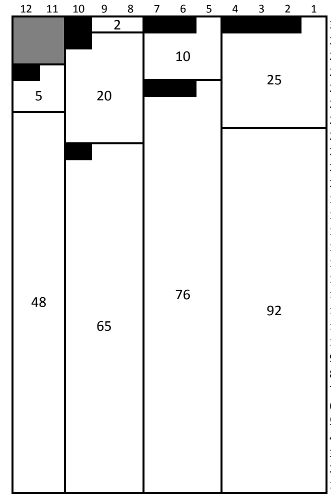

Figure 1. Example of resources allocated by the eOCSA algorithm.

minimized. Besides, by considering mapping the larger resources first, throughput can be maximized.

Finally, in [13], the algorithm applies horizon scheduling, permitting bursts to be scheduled efficiently and in a simple way. Initially, the algorithm detects the sensitive real-time requests with short deadlines, and gives the opportunity to large real-time requests to be accommodated firstly by applying a descending order sorting. Upon the completion of the real-time requests, the algorithm attempts to map the non real-time requests and ensure that large requests can find available space. However, this assumption, the request either the real-time data or the non real-time data, is not practical in real systems. In fact, each subscriber's request can has urgent and non-urgent data. For instance, rtPS, UGS, and ertVR regarded as urgent data, called the MUST data. nrtPS and BE are treated as non-urgent data, called the WISH data. In other words, the MUST data is a top priority to map. Then, the WISH data will be mapped at the same time, if there is available space in DL sub-frame.

III. The Proposed Resource Allocation algorithm

Assume that there are s slots and c subchannels in the downlink subframe. Let Ω be the set of all slots in the subframe and $\Omega = \{(x, y) \mid 1 \leq x \leq s, 1 \leq y \leq c\}$. Let v_i be the i^{th} column of Ω, in which $v_i = \{(i, y) \mid 1 \leq y \leq c\}, i = 1, 2, \cdots, s$. Let $K_{i,j}$ be a subset of Ω, where $K_{i,j} = \{v_i, v_{i+1}, \cdots, v_j\}$, $1 \leq i \leq j \leq s$. A rectangle R is denoted by $[(x_0, y_0); w, h]$, where (x_0, y_0), w, and h are its bottom-right corner, the width, and the height, respectively. Let the rectangle $R_{i,j}$ be the maximum available space in $K_{i,j}$. $|K_{i,j}|$ and $|R_{i,j}|$ are the cardinality of $K_{i,j}$ and $R_{i,j}$, respectively.

Let $\Phi = \{1, 2, \cdots, N\}$ be the set of users, and the scheduler determines the set of requests $\boldsymbol{A} = \{A_i\}_{i=1}^{N}$ for mapping. All requests consists of two parts: the MUST parts, M_i, and the WISH parts, W_i, i.e., $A_i = M_i + W_i$, $1 \leq i \leq N$. Given a data mapping space $K_{i,j}$, let $\Gamma_{i,j}$ be a subset of Φ such that $k \in \Gamma_{i,j}$ iff the request A_k is allocated in $K_{i,j}$.

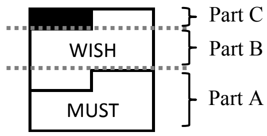

Figure 2. Example of a general burst.

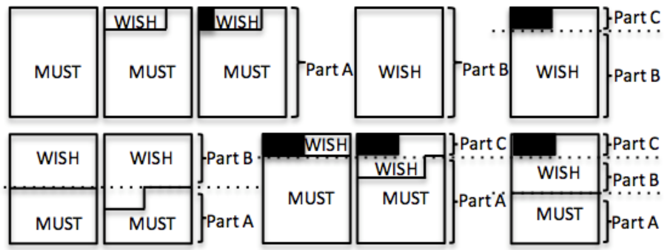

Figure 3. Bursts of every variety.

The proposed algorithm consists of two phases as described below. Phase 1 maps all requests with both MUST and WISH parts into the downlink subframe. Phase 2 is to return some allocated requests' WISH parts such that more MUST parts can be allocated.

The first phase consists of four steps as follows. The algorithm allocates slots to a request iteratively. As the algorithm proceeds, the subframe will be split into some vertical strips. Let Ψ be the set of all strips, and initially $\Psi = \{K_{1,s}\}$.

Step 1. Find a strip in Ψ, say K_{i^*,j^*}, such that it has the maximum available rectangle space which is denoted by $R_{i^*,j^*} = [(x', y'); w', h']$.

Step 2. Select the largest request in $\boldsymbol{A}$, say A_k, such that it can be allocated to R_{i^*,j^*} found in step 1. The first phase terminates if no such A_k exists.

Step 3. There are two cases for K_{i^*,j^*}.

i) If $|R_{i^*,j^*}| = |K_{i^*,j^*}|$, then K_{i^*,j^*} must be $K_{i^*,s}$.
  a) Calculate $w = \lceil A_k/h' \rceil$ and $h = \lceil A_k/w \rceil$. Allocate the rectangle $r = [(x', y'); w, h]$ for request A_k.
  b) If $w < (s - i^* + 1)$, then split the original K_{i^*,j^*} into K_{i^*,i^*+w-1} and $K_{i^*+w,s}$. Update $\Psi = (\Psi - \{K_{p,s}\}) \cup \{K_{p,p+w-1}, K_{p+w,s}\}$.

ii) Otherwise there are some requests allocated in K_{i^*,j^*}. Calculate $w = j^* - i^* + 1$ and $h = \lceil A_k/w \rceil$. Allocate the rectangle $r = [(x', y'); w, h]$ for request A_k.

Step 4. Update R_{i^*,j^*}, and remove A_k from $\boldsymbol{A}$.

In general, one burst consists of up to three parts as shown in Fig. 2. The black part represents the over allocated slots. Part A contains all the MUST data and possibly with some of the WISH data or even some of the over allocated slots. Part B contains only part of the rest WISH data. Part C is the row containing the over allocated slots along with the last part of WISH data, if over allocated slots exist. Figure 3. shows the bursts of every variety.

TABLE I. AN EXAMPLE TO ILLUSTRATE THE PROPOSED ALGORITHM

Resource (slots)	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
	38	8	86	32	41	3	115	7	17	13
Must part	14	6	48	22	15	2	50	7	11	5
Wish part	24	2	38	10	26	1	65	0	6	8
The rest of the Must data	0	0	0	0	0	0	0	0	0	0
Over Allocation	0	0	1	0	0	0	1	0	0	1

After completing the first phase, the unmapped requests might contain the MUST part. To serve as much urgent data as possible, the second phase selects the request according to the size of MUST part.

Let $\Psi = \{K_{1,p_1}, K_{p_1+1,p_2}, \cdots, K_{p_l+1,s}\}$, $1 \le p_1 \le p_2 \le \cdots \le p_l < s$, be the result after the first phase is done. The Second phase is described below.

Step 1. Select the request in $\boldsymbol{A}$, say A_k, such that its MUST part is the largest one. If $\boldsymbol{A} = \emptyset$, then Phase 2 is terminated.

Step 2. Now, we select a proper strip in Ψ to allocate M_k.

i) If $M_k \le R_{i,j}$ for some $K_{i,j}$ in Ψ, then select the strip with the smallest available space, say K_{i^*,j^*}, to fit M_k. Go to Step 3.

ii) Otherwise, check whether the M_k can be fit in a strip. Let $C_{i,j}$ and $B_{i,j}$ be the total numbers of slots belonging to Part C and Part B in strip $K_{i,j}$, respectively.
  a) If $M_k \le (R_{i,j}+C_{i,j})$ for some $K_{i,j}$ in Ψ, then select the strip with maximum $C_{i,j}$, say K_{i^*,j^*}. Remove row by row the slots belonging to Part C in K_{i^*,j^*} until M_k can be fit in the strip. When removing a row, choose the one which has the most number of over allocated slots. Shift up all empty rows to the top, and go to Step 3.
  b) Otherwise, select the strip with maximum $(C_{i,j} + B_{i,j})$, say K_{i^*,j^*}. If $M_k \le (C_{i^*,j^*} + B_{i^*,j^*})$, remove all rows the slots belonging to Part C in K_{i^*,j^*}. Remove row by row the slots belonging to Part B in K_{i^*,j^*} until M_k can be fit in the strip. Shift up all empty rows to the top, and go to Step 3. If $M_k > (C_{i^*,j^*} + B_{i^*,j^*})$, then remove A_k from $\boldsymbol{A}$ and return to Step 1.

Step 3. Allocate M_k in K_{i^*,j^*}. If there are unused slots in K_{i^*,j^*}, allocate the WISH part of A_k. In case that only partial WISH part is mapped, unmapped WISH part is returned to the scheduler. Remove A_k from $\boldsymbol{A}$.

Before leaving this subsection, we present an example to explain the operation of the proposed algorithm. Assume that $s = 12$ and $c = 30$ and there are 360 slots in total. There are ten MSs with the request set $\boldsymbol{A}$ as shown in TABLE I. and the sum of all requests is equal to 360, in which the sum of the MUST part of all requests is equal to 180.

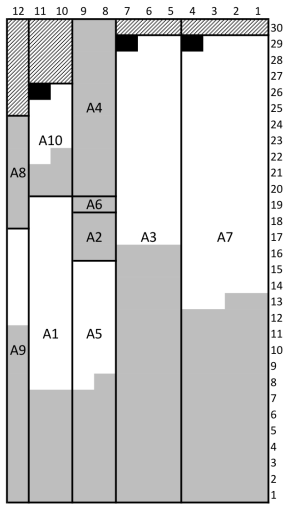

Figure 4. Example of resources allocated by the proposed two-level requests allocation algorithm.

TABLE II. SYSTEM SIMULATION PARAMETERS.

Parameter	Value
Frame length	5 ms
Channel BW	10 MHz
Permutation scheme	PUSC
Number of subchannels	30
DL subframe	12 slot columns
Total number of slots per DL subframe	30 × 12 slots
Total Must data : Total Wish data	1:1
Simulation time	5000 frames

Initially, we have $\Psi = \{K_{1,12}\}$. In the first phase, consider the first iteration. The largest rectangle in $K_{1,12}$ is $[(1,1); 12, 30]$ and the largest request smaller than $12 \times 30 = 360$ is $A_7 = 115$. According to Step 3, the rectangle $[(1,1); 4, 29]$ is allocated to A_7. The over allocation is only 1 slot. The set $K_{1,12}$ is split into $K_{1,4}$ and $K_{5,12}$. The set Ψ is updated as $\Psi = \{K_{1,4}, K_{5,12}\}$. In the second iteration, the maximum rectangle is $[(5,1); 8, 30]$ in $K_{5,12}$ and the largest one in the updated request set smaller than $8 \times 30 = 240$ is $A_3 = 86$. According to Step 3, the rectangle $[(5,1); 3,29]$ is allocated to A_3 with 1 slot of over allocation. The set $K_{5,12}$ is split into $K_{5,7}$ and $K_{8,12}$. The set Ψ is updated as $\Psi = \{K_{1,4}, K_{5,7}, K_{8,12}\}$. In the third iteration, $[(8,1); 5, 30]$ is the maximum rectangle in $K_{8,12}$ and is used for allocation. The rectangle $[(8,1); 2, 21]$ is allocated to A_5 with 1 over allocated slot. After the allocation, Ψ is updated as $\Psi = \{K_{1,4}, K_{5,7}, K_{8,9}, K_{10,12}\}$. The first phase is executed for six more iterations to sequentially allocate rectangles to requests A_1, A_9, A_{10}, A_2, A_8, and A_6. After completing the first phase, the unmapped request is A_4. In the second phase, we have $\Psi = \{K_{1,4}, K_{5,7}, K_{8,9}, K_{10,11}, K_{12,12}\}$. At the beginning, we try to map M_4 into $K_{i,j}$, $\forall K_{i,j} \in \Psi$. According to Step 2 ii) and Step3, we choose the $K_{8,9}$ with the largest value of the removed over allocated slots (two) to fit M_4. The result is shown in Figure 4.

For this example, the unused slots, the MUST parts, and the WISH parts are shown in diagonal, gray, and white, respectively. The total number of over allocated slots (shown in black) is 3. For all requests, sum of the rest of the MUST parts is zero. The efficiency (percentage of space used) of the proposed algorithm is 94.2% with over allocated slots and unused slots being counted as wasted.

IV. PERFORMANCE EVALUATION

In this section, we compare the performance of the proposed algorithm with that of eOCSA. TABLE II. shows the parameters used in the simulation, which are from the suggestions in the WiMAX forum. Resource requests are generated randomly with the constraint that sum of all requests is 12×30 slots. Two cases are considered in the simulations. In case 1, we can map or remove some WISH data of the request as the procedures presented in the previous section. Case 2 is used to simulate the case that all WISH parts cannot be partially transmitted. Slightly modification is required in Phase 2. In order to handle the two-level requests, we modified the final step of the eOCSA. When eOCSA finishes the final step, it selects the request, say A_k, whose MUST part is the largest and smaller or equal to the unused slots of current strip. After mapping the M_k, the remaining unused slots can be allocated to the WISH data of the A_k. Finally, remove A_k from $\boldsymbol{A}$.

Figure 5. shows the average number of slots allocated to the MUST parts. Because the sum of all requests' MUST part is 180 slots, the proposed algorithm achieves the first design goal of traffic differentiation. In fact, all MUST data can be served by the proposed algorithm. Figure 6. illustrates the impact of the number of MSs on average unused slots per DL subframe. The average number of unused slots for the proposed algorithms is smaller than that for eOCSA. That is because the proposed algorithm wastes less unused slots in its second phase.

Figure 7. indicates the average over allocated slots per frame for all schemes to different number of MSs. In case 2, because Phase 2 of the proposed algorithm removes some allocated WISH parts to fit the MUST part of a unmapped request, sometimes the remaining unused slots cannot serve the WISH part of the selected request. That is the reason why the proposed algorithm in case 2 performs the worst.

Nevertheless, the proposed algorithm still outperforms eOCSA in terms of the average efficiency as shown in Figure 8. The average efficiency is defined as allocated slots divided by total slots per frame. In addition, the efficiency increases with the number of stations. Because the sum of total requests is a constant, the request decreases with the number of stations. It is easier to map requests with small sizes.

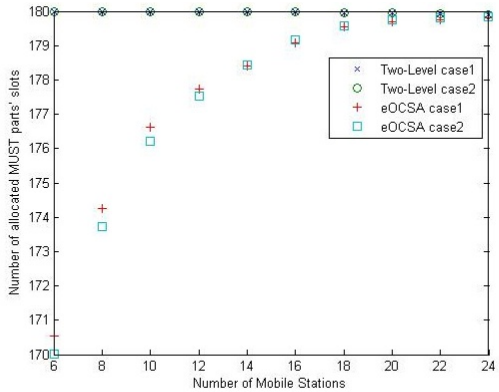

Figure 5. Average allocated slots of the MUST part vs. number of MSs

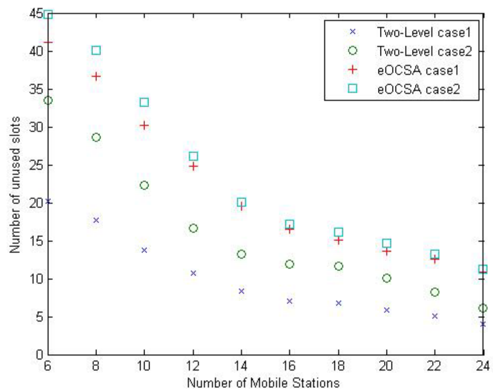

Figure 6. Average unused slots vs. number of MSs

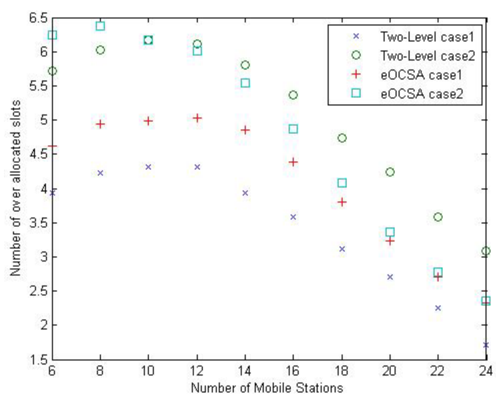

Figure 7. Average over allocation slots vs. number of MSs

V. CONCLUSION

This paper presented a novel downlink burst allocation algorithm for two-level requests in WiMAX networks. The proposed algorithm introduced data mapping algorithm (the first phase and the second phase), it can effectively prioritizes the sensitive real-time traffic. Similar to eOCSA, the proposed algorithm meets the rectangle shape allocation constraint, achieves high throughput by considering mapping the larger resources first. The basic idea of our proposed algorithm is to remove less the WISH data and fit in the MUST data as much as possible. The performance of the proposed algorithm is compared with that of eOCSA. Simulation results show that the proposed two-level requests algorithm outperforms eOCSA.

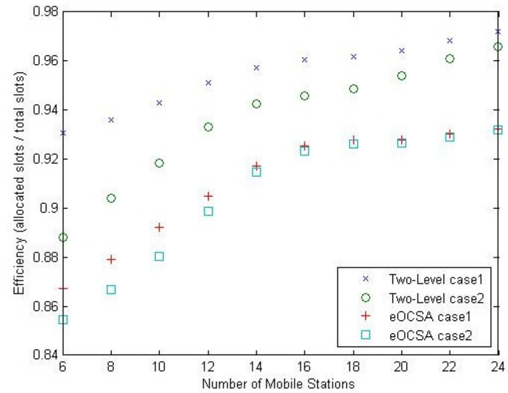

Figure 8. Average efficiency vs. number of MS

REFERENCES

[1] IEEE 802.16e-2005, “IEEE Standard for Local and Metropolitan Area Networks – Part 16: Air interface for Fixed Broadband Wireless Access systems – Amendment 2: Physical and Medium Access Control layers for combined fixed and mobile operation in licensed bands and Corrigendum 1,” February 2006.

[2] Juan I. del-Castillo, Francisco M. Delicado, Jesús Delicado and Jose M. Villalón “OFDMA Resource Allocation in IEEE 802.16 Networks: A Performance Comparative” in Wireless and Mobile Networking Conference (WMNC), 2010 Third Joint IFIP.

[3] Yanqun Le, Yi Wu, Dongmei Zhang “An Improved Scheduling Algorithm for rtPS Services in IEEE 802.16” 2009 IEEE.

[4] C. So-In, R. Jain, and A. Al-Tamimi, “Scheduling in IEEE 802.16e Mobile WiMAX Networks: Key Issues and a Survey,” in IEEE Journal on Selected Areas in Comm., vol. 27, no. 2, pp. 156-171, Feb. 2009

[5] M-R. Garey and D-S Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness,” W.H. Freeman, 340 pp., Jan. 1979.

[6] I. Gutiérrez; F. Bader; R. Aquilué; J. Pijoan, “Contiguous Frequency-Time Resource Allocation and Scheduling for Wireless OFDMA Systems with QoS Support”, EURASIP Journal on Wireless Communications and Networking. Volume 2009, Article ID 134579, pages 12.

[7] T. Ali-Yahiya; A. Beylot; G. Pujole; “Downlink Resource Allocation Strategies for OFDMA based Mobile WiMAX”. Telecommunication System;Vol 44, no.1-2, June 2010. pp. 29-37.

[8] WiMAX Forum, “WiMAX System Evaluation Methodology V2.1,” Jul. 2008, 230 pp. URL=http://www.wimaxforum.org/technology/documents

[9] A. Bacioccola; C. Cicconetti; L. Lenzini; E. Mingozzi, A. Erta, “A Downlink Data Region Allocation Algorithm for IEEE 802.16e OFDMA”, 6th International Conference on Information, Communication and Signal Processing, pp. 1-5, December 2007.

[10] T. Ohseki, M. Morita, and T.Inoue, “Burst Construction and Packet Mapping Scheme for OFDMA Downlinks in IEEE 802.16 System,” Proceeding of IEEE Global Communications Conference (IEEE GLOBECOM), 2007.

[11] C. So-In, R. Jain, and A. Al-Tamimi, “OCSA: An algorithm for Burst Mapping in IEEE 802.16e Mobile WiMAX Networks,” To appear in the 15th Asia-Pacific Conference on Comm. (APCC 2009), Oct. 2009.

[12] Chakchai So-In, Raj Jain and Abdel-Karim Al Tamimi “eOCSA: An Algorithm for Burst Mapping with Strict QoS Requirements in IEEE 802.16e Mobile WiMAX Networks” Singapore, February 26-28, 2010.

[13] Sarigiannidis, P.G., Louta, M., Stratogiannis, D.G., Tsiropoulos, G.I., “Towards a QoS-Aware IEEE 802.16 Downlink Sub- Frame Mapping Scheme,” To appear in IEEE GLOBECOM Workshops, Dec. 2010.