

OCSA: An Algorithm for Burst Mapping in IEEE 802.16e Mobile WiMAX Networks^{1,2}

Chakchai So-In, *Student Member, IEEE*, Raj Jain, *Fellow, IEEE*, Abdel-Karim Al Tamimi, *Member, IEEE*

Department of Computer Science & Engineering, Washington University in St. Louis
One Brookings Drive, Box 1045, St. Louis, MO 63130 USA

Most of IEEE 802.16e resource allocation proposals only focus on how to allocate the resources to meet QoS parameters such as throughput, delay, and delay-jitter. As described in the standard, the mapping from the allocation into downlink subframe for each burst needs to be in a rectangular shape. The rectangular mapping problem is a variation of a bin or strip packing problem, which is known to be NP complete. However, the mapping decision needs to be made within a few milliseconds for each Mobile WiMAX frame. In this paper, we introduce a heuristic algorithm, called One Column Striping with non-increasing Area first mapping (OCSA). The algorithm is fast and simple to implement and minimizes the unused slots in the frame.

Index Terms— *Burst Mapping, Two-dimensional downlink mapping, WiMAX, IEEE 802.16e, Mobile WiMAX, Scheduling, Resource Allocation, and QoS.*

I. INTRODUCTION

The IEEE 802.16e Mobile WiMAX standard [1] uses Orthogonal Frequency Division Multiple Access (OFDMA) in which the channel is divided into multiple subcarriers. The number of subcarriers is proportional to the channel spectral width. For example, for a 10 MHz channel, 1024 subcarriers are used. These subcarriers are grouped into a number of subchannels. Each user is allocated a certain number of subchannels for a certain amount of time. This is best described by a two dimensional diagram as shown in Fig. 1. The vertical axis in this figure represents the frequency or subcarrier and the horizontal axis represents time. The time is divided into frames (typically 5 ms duration) [2]. Bi-directional communication can be achieved by frequency division duplexing (FDD) in which uplink and downlink use different frequency bands or time division duplexing (TDD) in which the uplink (UL) traffic follows the downlink (DL) traffic in time domain. Fig. 1 shows the DL subframe and UL subframe of a TDD WiMAX system. In FDD, the two subframes are parallel in time. The mapping algorithm discussed in this paper can be used for both FDD and TDD systems. However, to keep the discussion focused, we use a TDD system.

Each DL subframe starts with a preamble and a frame control header (FCH) followed by the downlink map (DL-MAP) and uplink map (UL-MAP). These maps contain entries that define the burst profile for various bursts. The profile consists of the burst-start time, burst-end time, modulation type, and forward error control (FEC) used or to be used in the burst. Basically each burst belongs to a Mobile Station (MS). In the downlink direction a Base Station (BS) decides the burst size based on the packets waiting to be sent to various mobile stations. In the uplink direction, MSs send bandwidth

requests for each connection that they have set up. Each connection has a quality of service (QoS) requirement that is negotiated between the BS and MS at the time of connection setup. While the bandwidth requests are made separately for each connection, the grants (allocations) are made per MS in the sense that the MS can use its uplink allocation to transmit any packets. The standard defines there is that only one burst per mobile station for an uplink.

Fig. 1 shows ranging, channel quality indication (CQI), and acknowledgement regions in the uplink subframe. Ranging is used to determine the distance between the BS and MS so that the transmission start times at various stations can be properly synchronized. Ranging also helps to set the right transmit power level for each MS. Since the state of the wireless channel changes over time, the MS periodically sends the channel quality indications to the BS, which uses the information to set the modulation and coding parameters to be used in DL and UL bursts to that MS. Acknowledgement (ACK) region is used to send some feedback back to the BS.

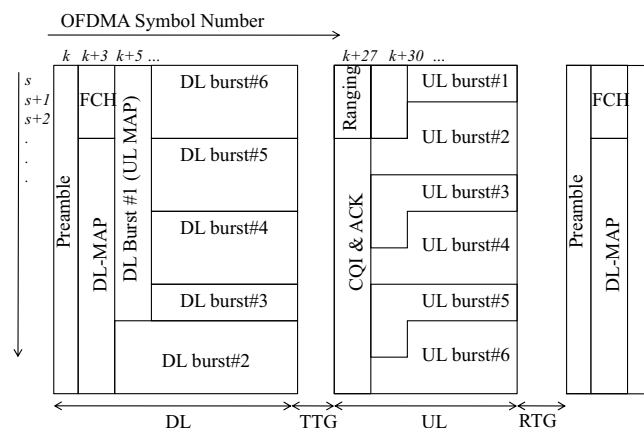


Fig. 1. A Sample IEEE 802.16e Frame Structure

Notice that all UL data bursts are allocated as horizontal strips, that is, the transmission starts at a particular slot and continues until the end of UL subframe. Then, it continues onto the next horizontal subchannel. This minimizes the number of subcarriers used by the MS and thus maximizes the

Manuscript received May 15, 2009, revised July 30, 2009. ¹ This work was sponsored in part by a grant from Application Working Group of WiMAX Forum. ² “WiMAX,” “Mobile WiMAX,” “Fixed WiMAX,” “WiMAX Forum,” “WiMAX Certified,” “WiMAX Forum Certified,” the WiMAX Forum logo and the WiMAX Forum Certified logo are trademarks of the WiMAX Forum.

Corresponding authors: C. So-In, R. Jain, and A. Tamimi (e-mail: cs5, jain, and aa7@cse.wustl.edu).

power per subcarrier and hence the signal to noise ratio.

In the downlink direction, the IEEE 802.16e standard requires that all DL data bursts be rectangular. Although the standard allows more than one burst per mobile station, it increases DL-MAP overhead. This particular case may be used when the mobile station really needs a different reliable channel such as different modulation and coding schemes (MCSs) for different connections, with an additional cost. The standard also allows more than one connections packing into one burst with the increase of the DL-MAP IE size [1]; however, the problem of rectangular mapping still remains. Our algorithm primarily maps the resource for each mobile station into a downlink burst in a rectangular fashion. We do not consider more than one burst per mobile station. However, it is possible to pack multiple mobile stations into one burst particularly if they are parts of the same physical mode. In this scenario, the unique connection identifier (CID) helps separate the mobile stations. Packing multiple mobile stations in one burst reduces DL-MAP overhead and the proposed algorithm can be applied directly to this combined resource allocation. The rectangular criterion requires an efficient two-dimensional mapping algorithm. This is the main focus of this paper.

Downlink resource scheduling can be done in two steps (Fig. 2). In the first step, the scheduler decides the allocation (number of slots to be allocated) to each MS. This can be done without any shape constraints and based solely on demand (the number of packets to be sent to a station); capacity (total available slots); and quality of service (QoS). We use the term “area” to denote the exact size and location of the rectangular burst. In other words, allocation is based on demand and then mapped to an area (second step).

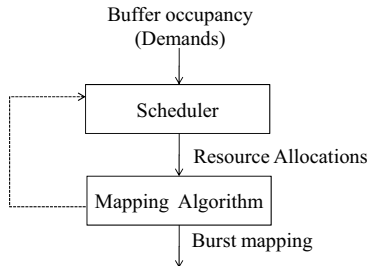


Fig. 2. Two steps downlink allocation

The two-dimensional rectangular mapping problem is a variation of the bin packing problem, in which one is given bins to be filled with objects. The bins may be constructed in two or more dimensions. If we restrict the bins to two dimensions, we have a “tiling” problem where the objective is to fill a given shaped bin with tiles of a given shape: for example, tiling circles in a circle, triangles in a circle, squares in a rectangle etc.

The bin packing problems are known to be NP complete. The complexity of the solution grows exponentially with the number of objects (or tiles) [11]. There have been many attempts to overcome these problems as stated in [8, 9, 10]. However, there is no easy way to achieve the optimality with simple computation. Examples of simple approaches are to apply first-fit, next-fit, best-fit, or bottom-left allocations [8].

With these simple approaches, many heuristic approaches have been introduced. For example, the level approach [8] is used to pack the bins with non-increasing height from bottom up, and then move to the next level when it reaches maximum allowable width W . Similar to the level approach, the reverse-fit approach [13] allocates the bins with decreasing height from the bottom up and when it reaches the width W , the bins are allocated in reverse order from top to bottom.

In this paper, we propose a heuristic algorithm for two-dimensional rectangular mapping for downlink burst subframe in IEEE 802.16e Mobile WiMAX. We apply the concept of the largest area first and level mapping approach together. The paper is organized as follows: the two-dimensional rectangular mapping problem is stated in Section II. Section III briefly describes some of the related work. Our heuristic algorithm for the two-dimensional downlink burst mapping is described in Section IV. Then, the performance evaluation is presented in Section V. Finally, the conclusions and future work are discussed.

II. DOWNLINK BURST MAPPING PROBLEM STATEMENT

The two-dimensional downlink burst mapping in IEEE 802.16e Mobile WiMAX problem is different from the original bin packing. A precise statement of the problem is as follows:

- 1) We are given a fixed rectangular bin B of width W and height H . The bin B has an area A equal to $W \times H$.
- 2) We are also given a set of n items $\{b_1, b_2, \dots, b_n\}$. The i th item b_i has an area A_i .
- 3) Determine a rectangular shape for the i th item with width W_i and height H_i such that $A_i \leq W_i \times H_i$.
- 4) Width $W_i \leq W$ for all i . Similarly, height $H_i \leq H$ for all i .
- 5) W_i, H_i, W , and H are all integers.
- 6) Since the mapped region is more than the desired allocation A_i , the extra resource is wasted and so, $W_i \times H_i - A_i$, should be minimized.
- 7) Due to the rectangular considerations, all n items may not fit the bin B , the goal is to minimize the additional resource width W that is required to fit all n items.

For each item, there are many possible W_i and H_i . For example, consider an item with an area of A_i of 20. It can be mapped as $1 \times 20, 2 \times 10, 4 \times 5, 5 \times 4, 10 \times 2$, and 20×1 . Thus there are at least 6 possible combinations of $W_{ij} \times H_{ij}$ for this item.

The unit of allocation in WiMAX is “slot”. The slot definition depends upon the subchannelization mode. In this paper, we present only the Partially Used Sub-Channelization (PUSC) mode, which is the most commonly used mode. The analysis is applicable to other modes as well. Consider downlink PUSC with a 10 MHz channel. With WiMAX forum specified parameters, a 10 MHz channel requires 1024 subcarriers. In the DL, these 1024 subcarriers are grouped as 30 subchannels with each subchannel consisting of 28 subcarriers.

On the time axis, each OFDM symbol is 102.8 μ s. A 5ms frame (which is the only value allowed by the WiMAX forum profiles) results in a frame duration of 48.6 OFDM symbols.

Allowing 1.6 symbol durations for TTG (Transmit-Transmit Gap) and RTG (Receive-Transmit Gap), gives us 47 symbols per frame for communication. Some of these symbols are used for the DL and the remaining for the UL. Assuming a 2:1 DL:UL ratio, a good division of these 47 symbols is 29 symbols for the DL (including 1 symbol for preamble) and 18 symbols for the UL. As shown in Fig. 3, a cluster is defined as 14 subcarriers over two symbol-periods and a slot consists of 2 clusters or one subchannel over two symbols. In the burst mapping problem as defined above, all heights are in units of subchannels, all widths are in units of 2 OFDM symbols, and all areas are in units of slots [1, 2].

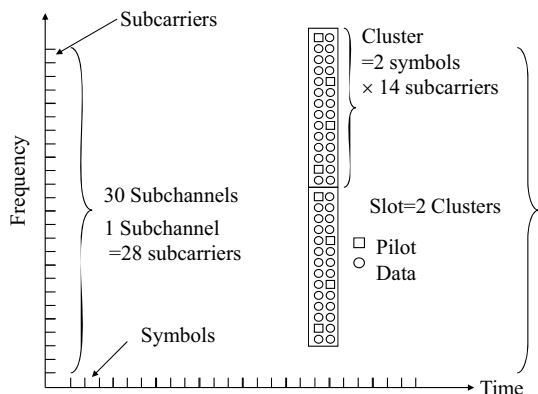


Fig. 3. Symbols, Clusters, and Slots [12]

Assuming the first two slot columns in the DL subframe are used for FCH, DL-MAP, and UL-MAP, we are left with 12 slot columns for data bursts. In terms of the bin packing problem, we have a bin of 30 subchannels height and 12 slots in width with a total area of 360 slots. However, the proposed algorithm can be used for any other combination of subchannels and time symbols.

Design Factors

As stated earlier, bin packing problem and therefore burst mapping problem is the NP complete. A heuristic algorithm is proposed in this paper. There were four considerations in designing this algorithm as discussed below.

First, the resource mapping should maximize the throughput and minimize the over allocation and unused spaces.

Second, the mapping algorithm should be simple and fast so that large number of users and bursts can be handled efficiently.

Third, the algorithm should be aware of variable components of the DL-MAP and UL-MAP. This variable portion consists of a burst profile for each burst, and therefore it depends upon the number of bursts [7].

Finally, the resource mapping should be such that the energy consumption of the MS is minimized. The MS needs to be active during the burst and so this duration width should be minimized.

III. RELATED WORK

Two-dimensional mapping for Orthogonal Frequency Division Multiple Access (OFDMA) was introduced by

Yehuda Ben-Shimol and his colleagues [3]. The algorithm simply assigns the resource allocation row by row with largest resource allocation first. There is no detailed explanation on how to map the resources of an unused space in a frame when their sizes span over multiple rows.

Another rectangular mapping algorithm was introduced by Takeo Ohseki and his colleagues [4]. They basically allocate in the time domain first and then the frequency domain (left to right and top to bottom). The algorithm is similar to the algorithm in [3] but it allows a burst compaction if there is more than one burst that belongs to the same physical mode. The algorithm does not consider an unused space. Without this consideration, the algorithm results in reduced throughput.

Claude Desset and his colleagues [5] showed that only 8 users at a maximum can be supported with a binary-tree full search algorithm for burst mapping in a rectangular shape. However, this work does not consider the variations of possible mapping pairs for each particular burst. An optimization was introduced, but the purpose was to reduce only the number of allocated OFDM symbols.

Bacioccola et al. [6] presented an algorithm that allocates from right to left and bottom to top. They map a single allocation into multiple rectangular areas that may result in an increased DL MAP element overhead. In our assumption, we have only one burst per mobile station.

In this paper, we propose a two-dimensional downlink burst mapping algorithm called One Column Striping with non-increasing Area first (OCSA). This algorithm generates rectangular mapping by considering all possible mapping pairs for each particular burst. The four considerations in its design have already been mentioned in Section II. The algorithm is simple to implement and provides very good throughput efficiency. In the performance evaluation section, we show that the usefully allocated space is approximately 95% of the optimal (full search) mapping.

IV. OCSA ALGORITHM

Our two-dimensional rectangular burst mapping algorithm is described in this section.

We divide the resource scheduling problem in two steps. In the first step, the scheduler computes the allocation of each user based solely on its demand, quality of service (throughput and delay) guarantee, and available capacity. The rectangular mapping constraint is not considered in this first step and is the main task in the second part.

To maximize the throughput, the resource allocations are sorted in a descending order (largest first). To allow space for variable parts of downlink and uplink maps, the resource allocations are mapped from right to left and from bottom to top of the DL subframe. Given a burst area, there are many possible height and width combinations that may be contained in that area; we chose the pair that is smallest in width. This allows the receiving MS to shut down its electronic circuit for most of the remainder of the DL subframe, thereby, saving energy.

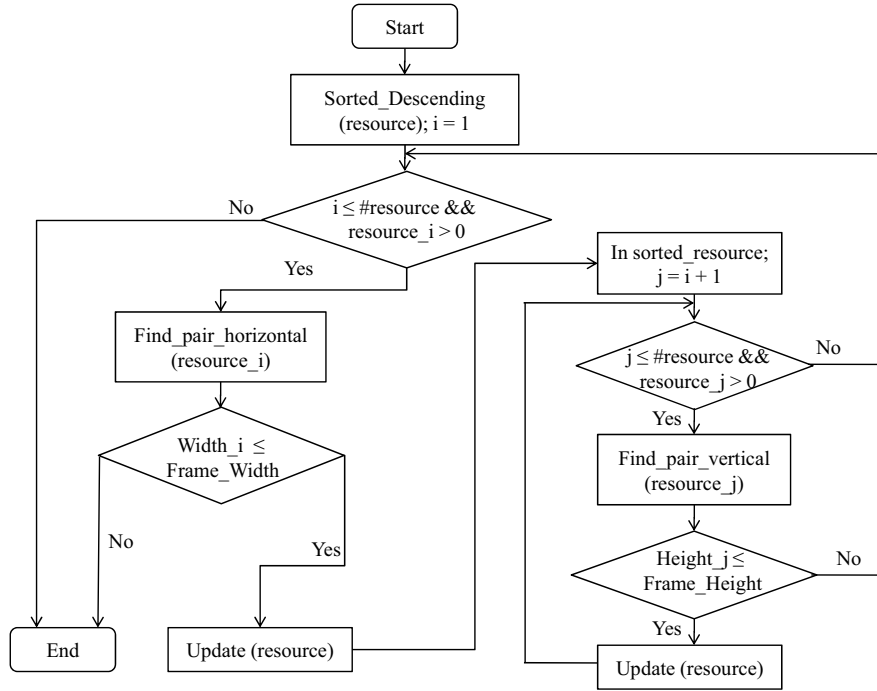


Fig. 5. Flow chart of OCSA burst mapping algorithm

A. Algorithm Description

Our algorithm has three steps. First, given a set of resource allocations $\{A_i\}$, we sort the set in a descending order.

The second step consists of mapping this allocation to the DL subframe (horizontal mapping). The algorithm selects the largest element to map that can fit in the unused space. Given area A_i , the algorithm determines the possible j th width-height pair (W_{ij}, H_{ij}) for the burst, such that $W_{ij} \times H_{ij} = A_i$. The only constraint is that each W_{ij} is less than W (width of the remaining DL subframe) and H_{ij} is less than H (height of the DL subframe). For example, if the resource allocation is 40, the possible mapping pairs are 2×20 , 4×10 , 5×8 , 8×5 , and 10×4 . Note that 1×40 , 20×2 , or 40×1 are not feasible since their height or width exceeds the DL subframe size. Again, in this example the DL subframe height is 30 and 12 in width. Given several feasible pairs, we select the pair with the smallest width. This helps save energy consumption at the MS.

However, instead of 40 as an example, consider the resource allocation 37. There are no feasible mapping pairs. In this step, the algorithm chooses the mapping pair that results in the least over allocation. In this example, we try all possible widths (from 2 to 12) resulting in 2×19 , 3×13 , 4×10 , 5×8 , 6×7 , 7×6 , 8×5 , 9×5 , 10×4 , 11×4 , 12×4 . These mappings will result in the over allocation of 1, 2, 3, 3, 5, 5, 3, 8, 3, 7, and 11, respectively. The constraints of the maximum allowable width and height still apply.

The size of the available space dictates the mapping. For example, the resource allocation of 35 can be mapped as 5×7 or 7×5 . Neither of these two mappings can be used if the unused space of 90 is available as 3×30 . In all cases, the algorithm chooses the mapping that results in the smallest

over allocation. In this example, the pair of 2×18 would be chosen with one slot over allocation.

After an allocation is mapped to the DL subframe, some space may remain unallocated above the mapped bursts. In the third step (vertical mapping), we try to assign this space before moving on to the next DL subframe mapping. Consider the vertical strip over the mapped bursts. Our goal is to fill in this strip before moving on to the remaining empty columns in a frame. The algorithm searches for the largest allocation that will fit in the left-over area in this strip. However, *smaller height and larger width* are preferred for the utilization of the space within this current width. If a match is found, the mapping is made. This process of filling the vertical strip is repeated until no further allocation will fit vertically in the unused space with this DL subframe width.

At this point we go back to Step 2 and select the next largest allocation for horizontally mapping to the DL subframe. The process of moving vertically and then horizontally to the left is shown in Fig. 4.

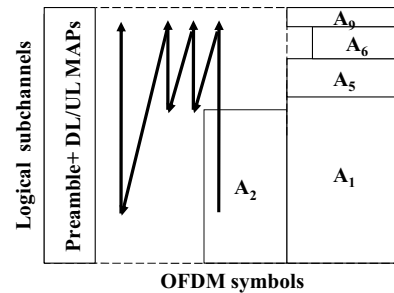


Fig. 4. An example of mapping downlink burst using OCSA

Fig. 5 shows a flow chart of an OCSA algorithm. A pseudo code showing the nesting of various steps is presented in Fig. 6. Notice that the computational complexity in the worst case

of OCSA is in the order of $O((rn)^2)$, where n is the number of resource allocations within a frame and r is the resource allocation (for mapping-pairs enumeration purpose).

$$Complexity = O(sorting) + O(allocations) = O(n \log n) + O((rn)^2)$$

```

Sorted_allocations = Sort(resource_allocations) //1st step
FOR each unmapped element in sorted_allocations //2nd step
  Map_to_Subframe (&start_strip_i, &end_strip_i, &height_i)
  FOR each unmapped element in sorted_allocations //3rd step
    Map_to_Strip_Vertically (start_strip_i, end_strip_i, height_i,
    &sub_height_j)
  END FOR
END FOR
    
```

Fig. 6. Steps in OCSA Algorithm

One possible extension of OCSA as described above is that before terminating, the algorithm may try to map all remaining allocations to the unused spaces in reverse order. This will introduce one more round of searching and hence increase complexity. The allocations may be only partially satisfied. The remainder is allocated in future frames. If partial allocations are allowed, the algorithm can be optimized to consider not just the next largest resource allocation, but the allocation that will result in the least unused space. Again, the remainder is kept for future frames.

OCSA not only optimizes the frame utilization but also maximizes the resource allocation; however, in case there is a strict QoS requirement for some particular mobile station or to avoid starvation of smaller resource allocation, the other possible extension of OCSA is to apply the priority mechanism. For example, the resource allocation with strict priority will override the next largest resource allocation criterion. In other words, this resource allocation is moved to the beginning of the mapping queue and this resource allocation mapping will be mapped regardless of the largest size consideration. This is a trade-off because the strict priority may lead to more unused space.

B. OCSA Examples

In this section, we provide two examples that help explain our algorithm. Once again, the DL subframe is assumed to be 12×30 resulting in 360 slots. Table I shows a simple example for five MSs. These MSs have been allocated A_1 through A_5 by the scheduler. The sum of all resource allocations is equal to 360.

TABLE I
EXAMPLE I: FIVE RESOURCE ALLOCATIONS

	A_1	A_2	A_3	A_4	A_5
Allocations	150	14	28	110	56
Possible Mapping Pairs	5×30 6×25 10×15	1×14 2×7 7×2	1×28 2×14 4×7 7×5	5×22 10×11 11×10	2×28 4×14 7×8 8×7

First, the algorithm sorts all resource allocations in descending order of area. That results in $A_1, A_4, A_5, A_3,$ and A_2 , respectively. The DL subframe area mapping is done from right to left and bottom to top. Resource allocation A_1 is chosen first, all possible factors of 150 are listed, and the pair 5×30 is selected and mapped into the DL subframe. This pair has the smallest width. This mapping results in no unused space over the allocated burst.

Second, we move on to the next largest resource allocation, which is A_4 (110). This time the pair 5×22 is chosen. This mapping results in an unused space of 5×8 in the strip over the A_4 burst. The algorithm moves horizontally and searches for all possible mapping pairs that can fit within this space. The resource allocation A_3 is selected and the mapping 4×7 provides the best match to the width. Though there is unused space within this width, none of the remaining unmapped resource allocation can fit.

Whenever the algorithm reaches the height limitation, the algorithm moves leftward to fill the remaining DL subframe. The unused space in this DL subframe is 60 (2×30) therefore A_5 , next largest resource allocation, is mapped with the pair of (2×28) and thus leaves four unused slots. At this stage, although there is unmapped resource allocation, there is no unused space where A_2 can fit. The mapping algorithm informs the scheduler to reschedule A_2 in a future frame and the algorithm terminates.

In this particular example, the total unused space is 16 slots as shown by the shaded area in Fig. 7. The optional searching round can attempt to map A_2 partially in the seven unused slots (7×1), reducing the unused space to 9 slots.

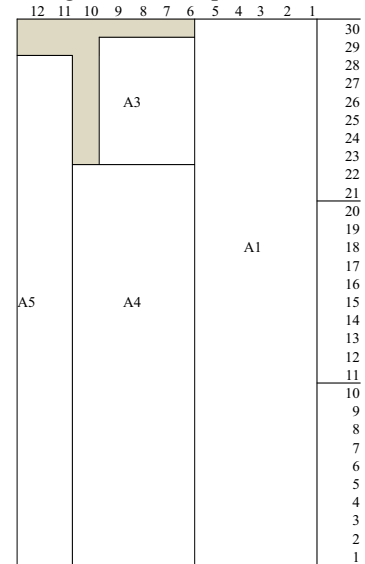


Fig. 7. Example I: Two-dimensional downlink burst mapping

For the second example, we generated the resource allocations randomly as shown in Table II. The resulting mapping is shown in Fig. 8. Notice that for $A_3 = 89$, there is no exact mapping possible. The algorithm, therefore, chooses the feasible mapping that results in the smallest over allocation. For A_3 , the pair of 3×30 is chosen with one slot over allocation.

TABLE II
EXAMPLE II: SEVEN RESOURCE ALLOCATIONS

	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Allocations	55	120	89	72	11	9	4
Possible Mapping Pairs	5×11 11×5	4×30 5×24 6×20 8×15 10×12 12×10		3×24 4×18 6×12 8×9 9×8 12×6	1×11 11×1	1×9 3×3 9×1	1×4 2×2 4×2 8×1

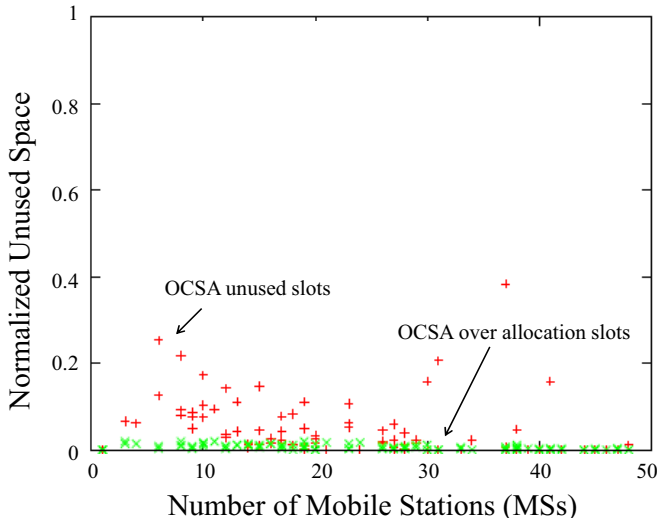


Fig. 9. Normalized unused space vs. number of MSs (OCSA)

It should be noted that when mapping the burst in the DL subframe, the mapping with the least width is selected. However, when mapping the burst in the vertical strip covering this burst, the mapping with the least height is selected. For example, to map A_5 , the algorithm chooses 3×4 instead of 2×6 . This reduces unused space. However, both pairs result in one slot over allocation.

For this example, the total unused space is 6 slots and the total over allocation is 3 slots as shown by the light and dark shaded areas in Fig. 8. The efficiency of the algorithm (percentage of space used) is 97.50% with over allocation being counted as wasted.

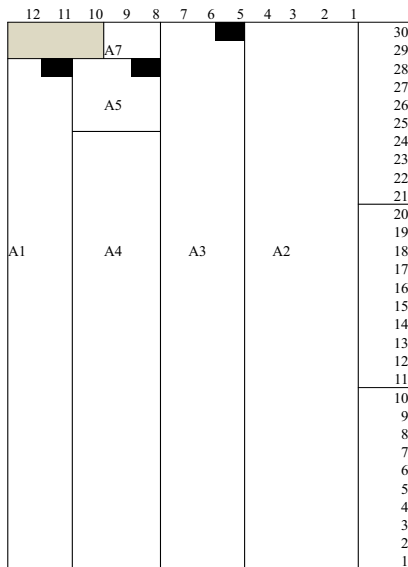


Fig. 8. Example II: Two-dimensional downlink burst mapping

V. PERFORMANCE EVALUATION

In this section, we present numerical results comparing OCSA with the ideal (full-search) algorithm.

In this performance evaluation, again a DL subframe of 12×30 is used. We assume each MS needs only one burst. The

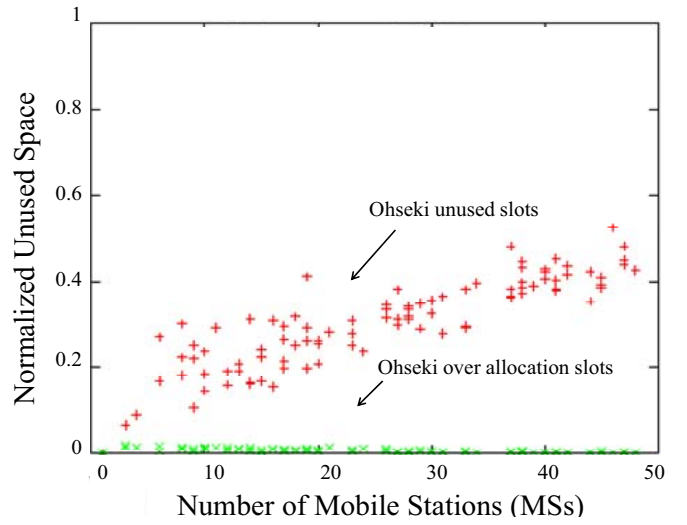


Fig. 10. Normalized unused space vs. number of MSs (Takeo Ohseki et al.)

number of MSs is randomly chosen from 1 to 49. The resource allocation for each MS is also randomly assigned in the range of 1 to 360 slots. The over allocations and unused slots are averaged and normalized over 100 trials. The results are shown in Fig. 9 in terms of the normalized over allocations and unused space versus the number of MSs. On average, the normalized unused slots and over allocation slots are 0.0422 and 0.0059 compared to the ideal mapping (full search).

In addition to comparing OCSA with the ideal mapping, we also chose to compare OCSA with the mapping algorithm by Takeo Ohseki et al. [4]. We could not compare OCSA with other algorithms for various reasons. There are no details as to how to map the resources of an unused space if their sizes are over multiple rows with the algorithm by Yehuda Ben-Shimol et al. [3]. Bacioccola et al [6], assume that it is possible to have more than one burst per mobile station. This violates our goal of minimizing burst overhead. In our analysis we show OCSA can support more than 30 mobile stations for the case where binary-tree full search supports only 8 mobile stations [5].

With the same configuration as used for Fig. 9, the results for the algorithm by Takeo Ohseki et al are shown in Fig. 10. We compared their algorithm to the ideal mapping. The normalized unused slots and over allocation slots are 0.2954 and 0.0051 on average because they left the space above the bursts unused. Note that each resource allocation within a single frame is treated as a separate burst and so there is no burst compaction since the burst compaction may lead to the decoding delay which requires further investigation.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we introduce a new heuristic algorithm for the two-dimensional downlink burst mapping for IEEE 802.16e Mobile WiMAX networks. The algorithm meets the rectangular shape allocation constraint and achieves high throughput by minimizing an unused space, and optimizes the energy consumption at MS by minimizing the receive time for the MS.

The algorithm consists of ordering the allocations in a descending order and the allocations are made from right to left and bottom to top. This allows space for variable portions of the DL-MAP and UL-MAP.

OCSA is a heuristic algorithm. There is a tradeoff between the throughput optimality and the computation complexity. We can make it more efficient by introducing further complexity such as by making a recursive for both vertical and horizontal mapping.

We presented the performance of OCSA with ideal full-search algorithm and found that OCSA provides 95% throughput compared to a full search algorithm.

OCSA algorithm can be applied to all QoS service classes. We assumed that prior to mapping, the scheduler has made resource allocations based on QoS parameters and capacity. Further study is needed to make the scheduler aware of the unused space and unmapped resource allocation. Moreover, the admission control algorithms should allow for some unused space.

REFERENCES

- [1] IEEE P802.16Rev2/D2, "DRAFT Standard for Local and metropolitan area networks," Part 16: Air Interface for Broadband Wireless Access Systems, Dec. 2007, 2094 pp.
- [2] WiMAX Forum, "WiMAX System Evaluation Methodology V2.1," Jul. 2008, 230 pp.
URL=[<http://www.wimaxforum.org/technology/documents>]
- [3] Y. Ben-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.
- [4] T. Ohseki, M. Morita, and T. Inoue, "Burst Construction and Packet Mapping Scheme for OFDMA Downlinks in IEEE 802.16 Systems," in *Proc. IEEE Global Telecommunications Conf.*, 2007, pp. 4307-4311.
- [5] C. Desset, E.B.d.L. Filho, and G. Lenoir, "WiMAX Downlink OFDMA Burst Placement for Optimized Receiver Duty-Cycling," in *Proc. Inter. Conf. on Commun.*, 2007, pp. 5149-5154.
- [6] A. Baccioccola, C. Cicconetti, L. Lenzini, E.A.M.E. Mingozzi, and A.A.E.A. Erta, "A downlink data region allocation algorithm for IEEE 802.16e OFDMA," in *Proc. Int. Conf. on Information, Commun. & Signal Processing*, 2007, pp. 1-5.
- [7] B. Kim, C. So-In, J. Yun, R. Jain, Y. Hur, and A. Al-Tamimi, "Capacity Estimation and TCP Performance Enhancement over Mobile WiMAX," *IEEE Wireless Comm. Mag.*, vol. 47, no. 6, pp. 132-141, Jun. 2009.
- [8] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *European Journal of Operational Research.*, vol. 141, no. 2, pp. 241-252, Sept. 2002.
- [9] F. Clautiaux, J. Carlier, and A. Moukrim, "A new exact method for the two-dimensional orthogonal packing problem," *European Journal of Operational Research.*, vol. 127, no. 3, pp. 1196-1121, Dec. 2007.
- [10] E. Hopper and B.C.H. Turton, "A Review of the Application of Meta-Heuristic Algorithms to 2D Strip Packing Problems," *Artif. Intell. Rev. Journal.*, vol. 16, no. 4, pp. 257-300, Dec. 2001.
- [11] M-R. Garey and D-S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," W.H. Freeman, 340pp, Jan. 1979.
- [12] 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.
- [13] I. Schiermeyer, "Reverse-Fit: A 2-Optimal Algorithm for Packing Rectangles," in *Proc. European Symp. on Algorithms.*, 1994, pp. 290-299.