國立交通大學

電信工程研究所

碩 士 論 文

WiMAX 系統之雙階層需求下的下行鏈路資源分配演算法

A Downlink Resource Allocation Algorithm of Two-Level Request in WiMAX Systems

研 究 生：洪瑞良

指導教授：李程輝　教授

中 華 民 國 一 零 二 年 七 月

WiMAX 系統之雙階層需求下的下行鏈路資源分配演算法

A Downlink Resource Allocation Algorithm of Two-Level Request in WiMAX Systems

研 究 生：洪瑞良　　Student：Rui-Liang Hong

指導教授：李程輝　　Advisor：Prof. Tsern-Huei Lee

國 立 交 通 大 學

電信工程研究所

碩 士 論 文

A Thesis

Submitted to Institute of Communications Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Communications Engineering

July 2013

Hsinchu, Taiwan, Republic of China

中華民國一零二年七月

WiMAX 系統之雙層需求下的下行鏈路資源分配演算法

研 究 生：洪瑞良　　　　　　　　　　指導教授：李程輝

國 立 交 通 大 學

電信工程研究所碩士班

摘要

IEEE 802.16e 是一項著名的行動通訊規格，並使用正交分頻多工存取(OFDMA)作為實體層的傳輸方式。在 OFDMA 系統架構下，資源分配是很重要的部分。若能有效率分配二維資源(時間和頻率)給行動用戶就能提升通道使用率和資料傳輸量。本論文提出一個雙階級需求下的下行鏈路資源分配演算法，有 3 個主要的目的(1)考慮資源分配的單位為封包等級，(2)將需求分為兩種優先權資料。MUST 資料為高優先權的資料;WISH 資料為低優先權資料，(3)考慮需求使用的調變與編碼機制。我們將提出的演算法與雙階級需求對應演算法(2L-DMA)作比較，由模擬結果可知本論文提出的演算法可以提升 21.9%的資料傳輸量，多服務 6%~7%的高優先權資料，並減少約一半的資源分配所需的額外資訊。

關鍵字： 正交分頻多重存取，IEEE 802.16e，資源分配， 雙階級需求, 調變與編碼機制

A Downlink Resource Allocation Algorithm of Two-Level Request in WiMAX Systems

Student：Rui-Liang Hong Advisor：Prof. Tsern-Huei Lee

Institute of Communications Engineering

National Chiao Tung University

ABSTRACT

IEEE 802.16e is a popular wireless communication technology, which use the OFDMA technology to achieve high data rate and reliability of physical layer (PHY) for wireless transmission. Resource allocation is an important part in the WiMAX system. It would improve channel utilization and throughput by allocating two-dimension (time and frequency) burst efficiently to each mobile user. In this thesis, we propose a downlink resource allocation algorithm for two-level request and it consists of three targets: (1) consider packet-level resource allocation; (2) classify urgent data as MUST data; classify non-urgent data as WISH data; and (3) consider MCS for each request. The goal of our proposed algorithm is to achieve high throughput, channel utilization, and reduce map overhead. We compare the performance of our proposed algorithm to 2L-DMA. Simulation results show that our proposed algorithm can enhance the throughput more 1.3Mbps than the throughput of 2L-DMA. Allocate the high priority data which is 6%-7% more than 2L-DMA. Besides, reduce almost half of the map overhead comparing to the one of 2L-DMA.

Keywords: OFDMA, IEEE 802.16e, resource allocation, two-level requests, MCS

TABLE OF CONTENTS

LIST OF FIGURES

LIST Of TABLES

Chapter 1

INTRODUCTION

IEEE 802.16e [7] mobile WiMAX is a broadband wireless access technology for high data transmission. In order to achieve high data rate, high spectral efficiency, short delay, and efficient operation with fading channel, mobile WiMAX use Orthogonal Frequency Division Multiple Access (OFDMA) technology to transmit data in PHY layer. Both downlink (DL) and uplink (UL) data transmission that the base station (BS) allocates the resource to multiple mobile stations (MSs) in mobile WiMAX. An efficient resource allocation algorithm is a necessary component to achieve high system throughput.

1.1 WiMAX System: An Overview

In Moblie WiMAX system, the bandwidth is divided into multiple subcarriers, which are grouped into a number of subchannels by distributed or adjacent subcarrier permutation. In the distributed subcarrier permutation, the subcarriers are randomly distributed across the bandwidth that it has great frequency diversity. Frequency diversity is used to reduce the effect of fast fading on mobile environments. The subchannelizations based on distributed permutation include PUSC (partial usage of subchannelization), and FUSC (full usage of subchannelization). In the adjacent permutation, each subchannel consists of some adjacent subcarriers using band Adaptive Modulation and Coding (AMC) mode. AMC is adaptive for fixed and low mobility device .For a particular user, the channel qualities under random subchannels are the same with PUSC but different with AMC. In this thesis, we consider that the DL subframe data mapping under PUSC mode.

Mobile WiMAX uses a fixed frame base allocation .Each of frame are 5 ms long and divided into downlink (DL) and uplink (UL) subframes. WiMAX supports frequency division duplexing (FDD) and time division duplexing (TDD). In the FDD, DL and UL use the different frequency band. TDD uses a single frequency band for DL and UL subframe as shown in Fig. 1. DL and UL subframes are separated by a Transmit to Transmit Gap (TTG) and BS switches from receive to transmit mode during the Receive to Transmit Gap (RTG).

In mobile WiMAX PHY layer, OFDMA slot is the minimum unit composed of one OFDMA subchannel and certain number of OFDMA symbols, where the subchannel and OFDMA symbol are the units in the frequency and time domain. The first symbol of the DL subframe is preamble, which is used for time synchronization. The frame control header (FCH) describes the length of the DL-MAP message, the repetition encoding, the modulation and coding scheme (MCS) applied to the DL-MAP. Where DL-MAP and UL-MAP contain the total number of burst, location of burst (offset and lengths in time-frequency domain) and burst profile. During the number of burst increases the length of DL-MAP and UL-MAP will grow up.

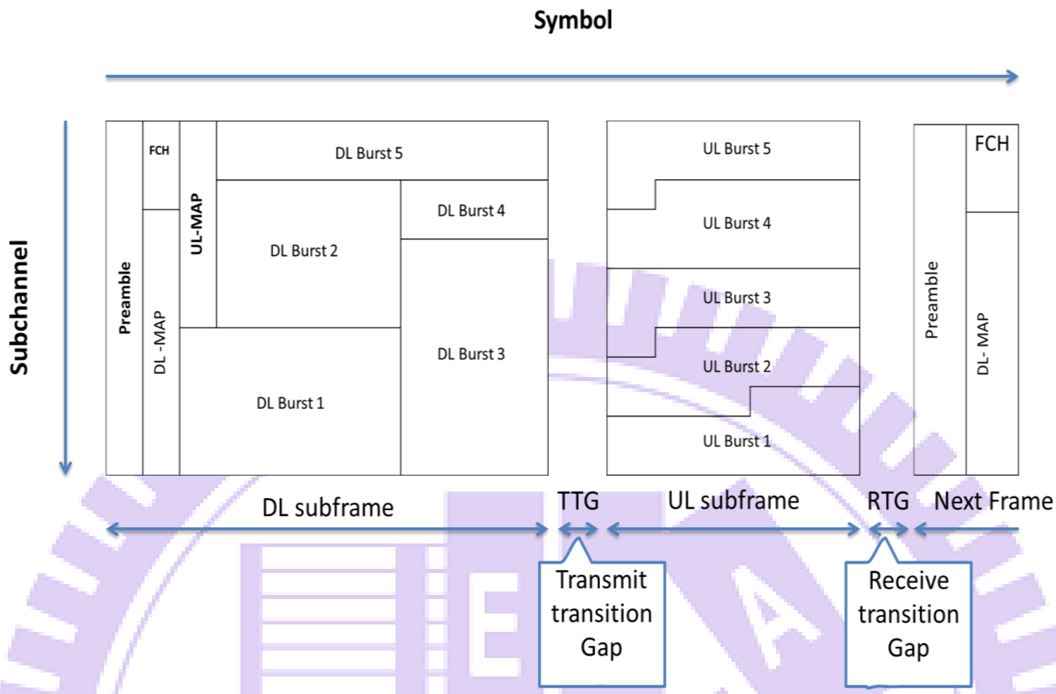

Figure 1. Example of an OFDMA frame structure in TDD mode

1.2 OFDMA Downlink Mapping

In mobile WiMAX DL subframe, each MS feedback channel quality message to BS which has full control over resource allocations to various MSs with adaptive MCS. A DL burst of a MS is mapped into a rectangle region of contiguous OFDMA slots. In PUSC mode, an OFDMA slot is composed of a subchannel and two OFDMA symbols. In case, one slot with higher MCS can carry more data to transmit than one with lower MCS.

Because a burst must be a rectangle block in DL subframe, there would cause some wasted slots inside or outside the bursts as shown in Fig. 2. The wasted slots in bursts are called over-allocated waste (OAW) and the wasted slots outside the bursts are called unallocated waste (UAW). It is clear that more wasted slots in DL subframe would reduce channel utilization, where the channel utilization is defined as the percentage of slots carrying data of total slots.

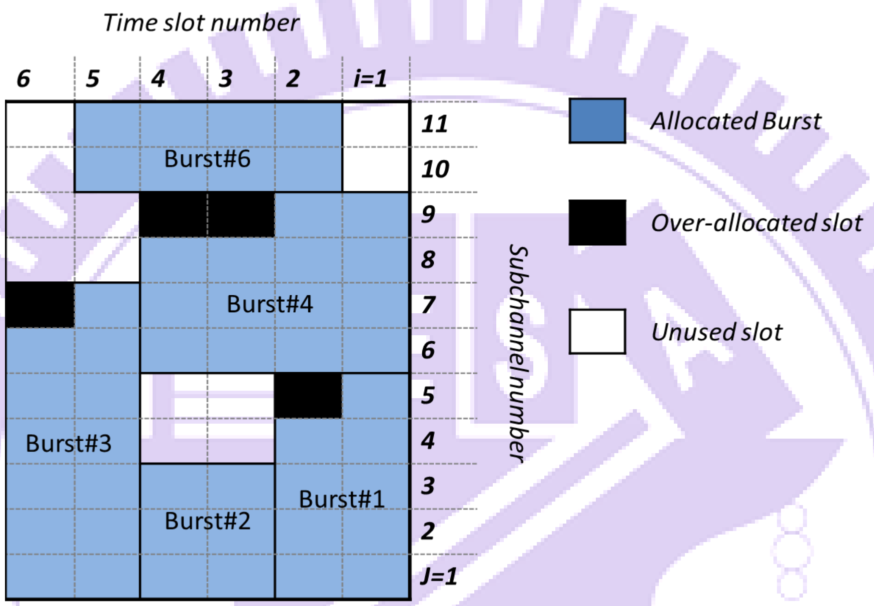

Figure 2. An example of data mapping

To maximize channel utilization and throughput, it is difficult to find an optimal solution of the two-dimensional packing problem, which is known as a NP-complete problem. WiMAX system has a constraint for data mapping executed within a frame time. It is impossible to find an optimal solution during the frame time. So, we design a heuristic data mapping algorithm to achieve high throughput and channel utilization.

Chapter 2

SYSTEM MODEL AND PROBLEM FORMULATION

According to bandwidth, packet lose, delay and delay jitter, IEEE 802.16e standard class five scheduling services with different Quality of Service (QoS) requirement. There has 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). The UGS scheduling service type is designed to support real time data streams that consist of fixed size data packets issued at periodic intervals for VoIP. The rtPS scheduling service type is designed to support real time data streams consisting of variable sized data packets that are issued at periodic intervals for MPEG video transmission. ertPS is a scheduling mechanism that builds on the efficiency of both UGS and rtPS. The nrtPS is designed to support delay tolerant data streams consisting of variable size data packets that a minimum data rate is required for FTP transmission. The BE service is designed to support data streams for which no minimum service guarantees are required.

In this thesis, we classify UGS, rtPS, and ertPS as urgent data, said MUST data. nrtPS and BE are classified as non-urgent data, said WISH data as shown in Fig. 3.

Generally, user's request can have MUST and WISH data both. In this thesis, return some WISH part packets to fit MUST part data as much as possible.

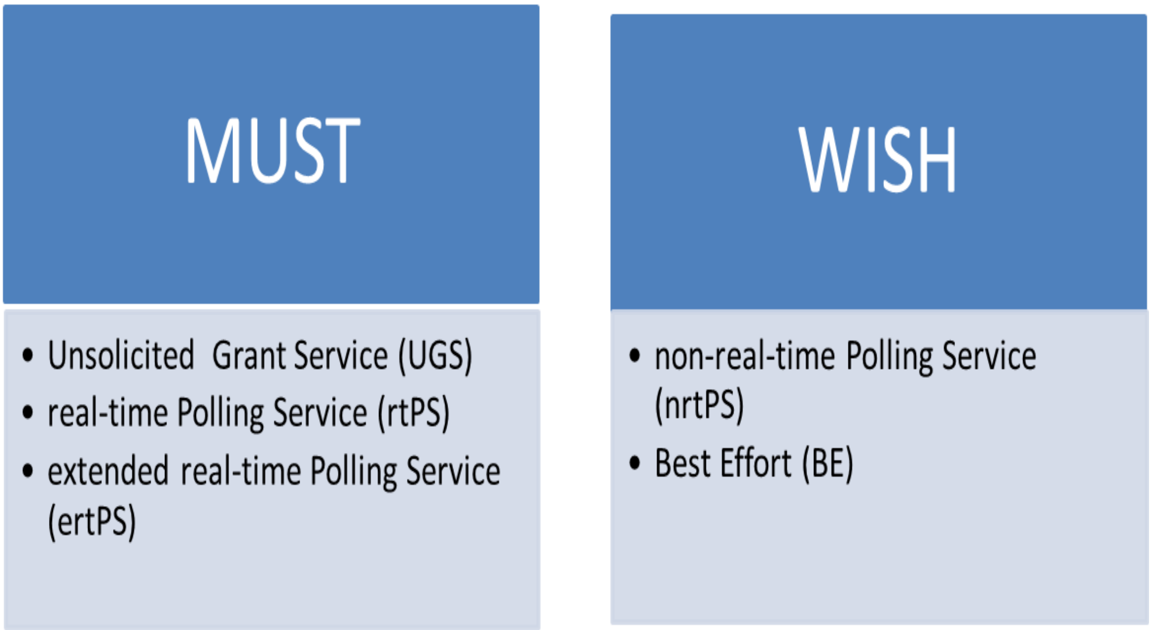

Figure 3. An example of MUST and WISH data

In WiMAX system, a complete resource allocation is constructed of scheduler and data mapper in the BS as shown in Fig. 4. In DL, scheduler determines the request size in slots for each user (MS) according to user's QoS, MCS, and then send request to data mapper. Data mapper determines the request's rectangular shape (width and height) and location, then map into DL subframe. The rectangular region is used for transmission data which is called burst. With constrain of rectangular shape, we have to return some packets back to scheduler if the request is larger than unused slots rectangle space. In addition, the request that cannot fill up the rectangular region will have some over-allocated slots in the burst considered as waste.

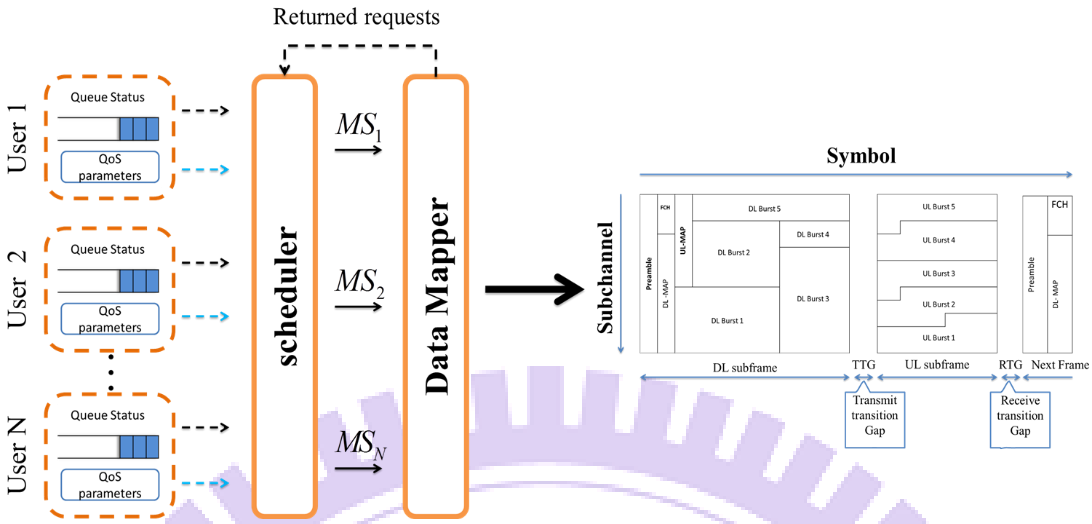

Figure 4. The entire resource allocation process

In IEEE 802.16e standard, an information element (IE) describes a burst in DL MAP. DL MAP massage is broadcast to each MS with robust (low) modulation and repetition coding for low error probability. Besides, each IE occupies five slots and there is less space to allocate request if the number of IE is grown. We consider IEs as overhead that influence system throughput significantly. In order to reduce overhead, we could combine requests into a burst with the same MCS.

There are many papers research data mapping problem for IEEE 802.16e, but most of researches didn't consider packet level problem. SDRA [1] and "A Data Mapping Algorithm for Two-Level Requests in WiMAX System" [5] assume that data mapper can fragment burst arbitrarily without considering packet size. In fact, packet is the minimal unit of request and data mapper map request to DL subframe packet by packet but not row by row or column by column. In this thesis, we analyze the impact on throughput and channel utilization under packet level mapping algorithm.

Chapter 3

RELATED WORK

In this section, we briefly review several burst mapping algorithms for Mobile WiMAX. In addition, we describe the drawback that we improve in the proposed mapping algorithm. The heuristic mapping algorithms have been proposed in [1] – [5] and we compare our proposed algorithm with [5]. All of them are aimed to achieve high channel utilization, but seldom works consider of throughput. However, we consider both channel utilization and throughput. SDRA [1] is a simple algorithm demonstrates the burst can be fragmented into smaller bursts to fill up DL subframe. [3] and [4] are similar simple heuristic mapping algorithms. [5] considers the case with two priority data for data mapping which is related to our work.

3.1 Sample Data Region Allocation Algorithm (SDRA)

A Sample Data Region Allocation (SDRA) algorithm for a DL subframe has been proposed in [1], which is a simple algorithm. SDRA allocate data region in column-wise order, and burst can be fragmented into smaller data regions. Because data regions are allocated in column-wise order, each data region can be fragmented into at most three smaller data regions as shown as Fig. 5. Obviously, SDRA bring more map overhead in DL MAP, because it split data region into multiple bursts.

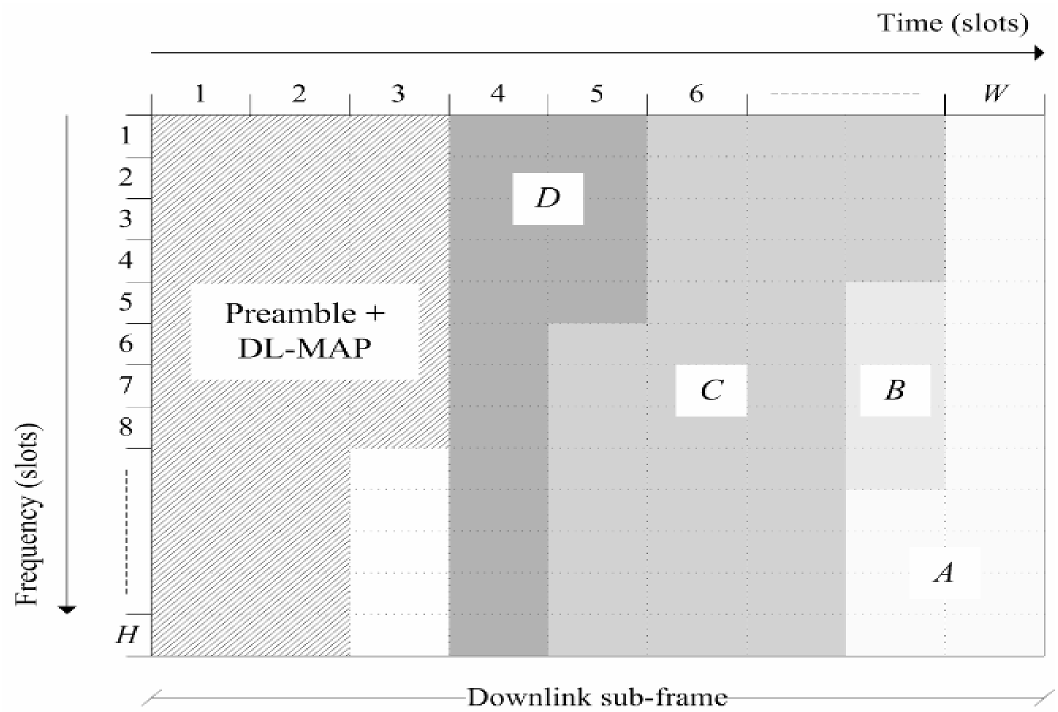

Figure 5. SDRA downlink burst mapping example [1]

3.2 Orientation-Based Burst Packing (OBBP)

The principle of OBBP **[3]** algorithm is to group up the resources based on the orientation factor (OF) and pack the bursts of each group in column-wise or row-wise in the DL subframe. This algorithm divided into three stages: The first stage is a prepacking stage to prepare and classify the bursts, the second stage using the common OF to packing the bursts, and the third stage packing remaining bursts from second stage by using the conventional best fit packing algorithm.

In first stage, a process called OF calculation will give the possible dimension for each rectangle. For example, there is a set of resource request B = {10, 6, 8, 5}. The orientation factors for B will be {10x1, 1x10, 2x5, 5x2}, {6x1, 1x6, 3x2, 2x3}, {8x1, 1x8, 4x2, 2x4}, and {5x1, 1x5}, respectively. The orientation factors are allocated in the matrix, and OBBP use the matrix to allocate bursts as shown as Fig. 6.

$$OF\_Matrix = \begin{pmatrix} 0 & 0 & 0 & 0 & 5 & 6 & 0 & 8 & 0 & 10 \\ 0 & 0 & 6 & 8 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 6. Example of an orientation factor matrix by OBBP algorithm [2]

In the second stage, the resources are classified according to their common number of slots (vertically) or subchannels (horizontally) in a matrix generated from the OFs obtained on the first stage. In vertical scenario the packing sets are selected by calculating the maximum sum of the elements in each column of OF matrix, and sort the packing sets in descending order. Afterward, the algorithm allocate the burst packing sets from the bottom-right corner and leave the unallocated slots at the top-left corner as shown in Fig. 7.

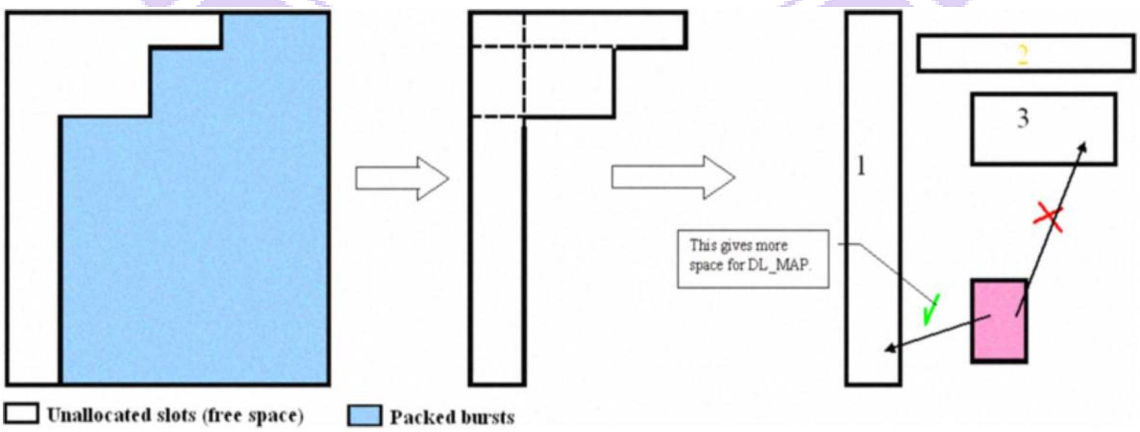

Figure 7. OBBP free space rectangles selection [2]

In the final stage, overlapped rectangles with its maximum possible dimensions are made with the unallocated slots as shown in Fig. 7. In this step, the remaining set of bursts is sort in descending order. Then allocate the first burst into an optimum rectangle which has the minimum difference with the size of packing burst.

3.3 Enhanced OCSA (eOCSA)

eOCSA is a heuristic data mapping algorithm, and is aimed to maximize the channel utilization. In the first step, a set of request are sorted in a descendinging order and select the largest request to map. The second step is called vertical mapping, eOCSA algorithm selects the largest request to be allocated, say *A*, and calculates the rectangle shape with $W* = \lceil A/H \rceil$ and $H* = \lceil A/W* \rceil$. H is the maximum height that can be used for allocation, and $W*$ and $H*$ are the width and height of the allocated request.

In the third step, 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. In addition, it follows the principle of allocating the largest request first. In this step, the region width is fixed, and width is used to determine the required height. This process is repeated until no space can be allocated or no request that can fit into this space. After the third step is complete, the algorithm return to the second step and select the next largest request to allocate.

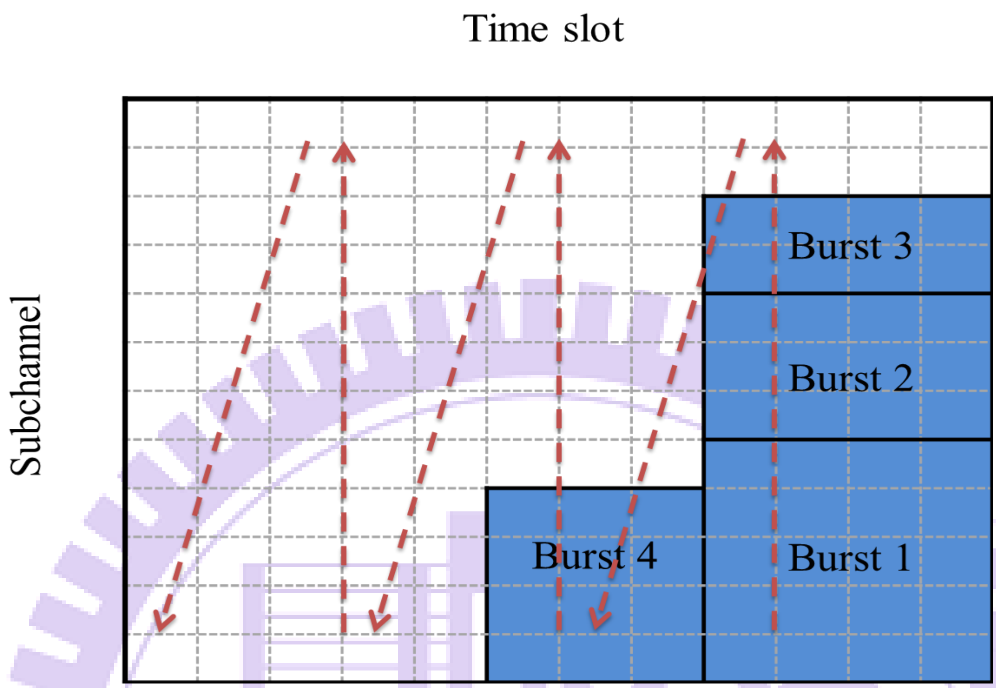

Figure 8. An example of mapping downlink burst using eOCSA

3.4 Maximum Rectangle-Based DL Burst Allocation (MaxRectangle)

MaxRectangle is a simple data mapping algorithm, and the main idea is find the maximum rectangle region to allocate the request. In the first step, data mapper will find the rectangle with maximum area in DL subframe as shown in Fig. 9. In the second step, data mapper selects the largest request, such that the size of packing request is equal or smaller than the maximum rectangle region. In the third step, compute the width and height of packing request and allocate to the bottom-right corner of maximum rectangle region, and repeat first step until there is no request or space can be allocated.

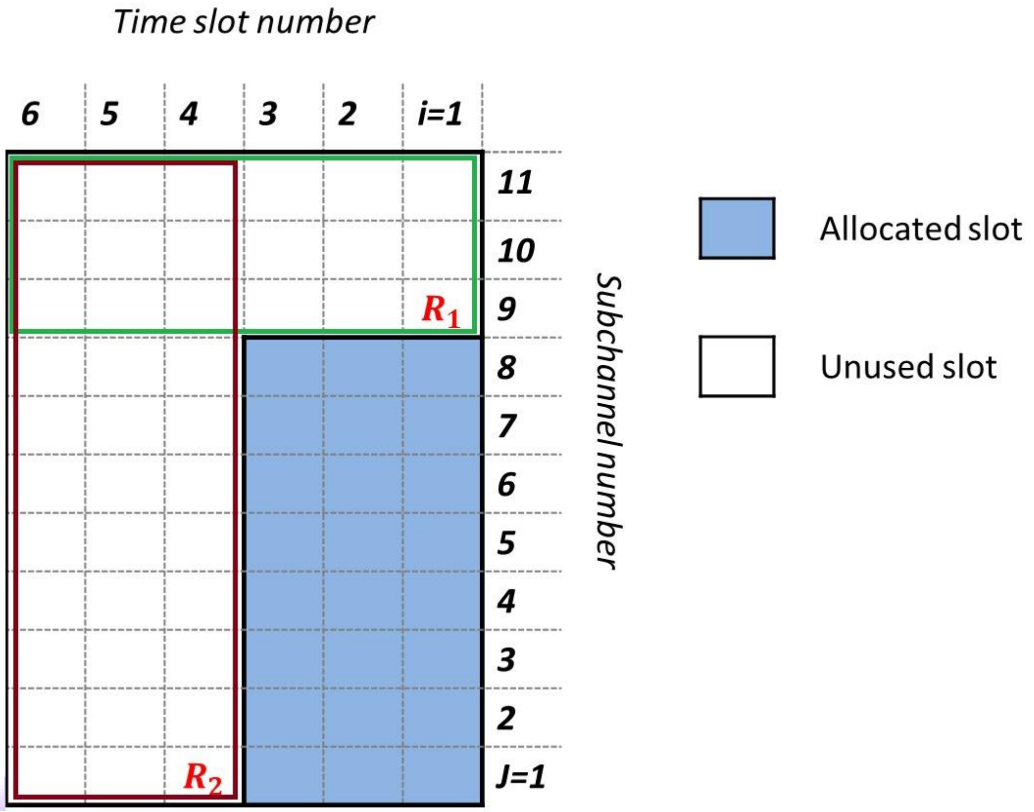

Figure 9. Process for finding maximum rectangle area

3.5 A Data Mapping Algorithm for Two-Level Request (2L-DMA)

2L-DMA [5] also considers the data mapping for two priority requests. In addition, the algorithm regards high priority data as MUST data and low priority data as WISH data. The main purpose of this algorithm is that allocate MUST data as more as possible and reduce the waste in DL subframe.

The algorithm consists of two phases. In Phase 1, data mapper maps both MUST part and WISH part of the request into DL subframe, which will be split into some vertical strips as shown in Fig. 10. In Fig. 10, yellow part is MUST data and blue part is WISH data. In Phase 2, data mapper returns some WISH data to allocate MUST data. There are four steps in Phase 1. In Step 1, find a strip that has maximum available rectangle space, say R_{i^*,j^*}. In Step 2, select the largest request, say A_k, such that it can be allocated into R_{i^*,j^*}. In Step 3, compute the width and height of packing request and allocate to the bottom-right corner of R_{i^*,j^*}. However, there are two cases in Step 3 as follow: 1) One case is that there is no burst in the strip, we have to calculate the width w and height h of the rectangle by $w = \lceil A_k / h' \rceil$ and $h = \lceil A_k / w \rceil$ where h' is the maximum height that can be used for allocation. If the width w is smaller than the width of strip, then split the strip into two strips; 2) in the other case, there are some requests allocated in the strip. Let the width w equal to the width of strip. Then calculate the height $h = \lceil A_k / w \rceil$. Repeat the three steps until there is no request or space can be allocated.

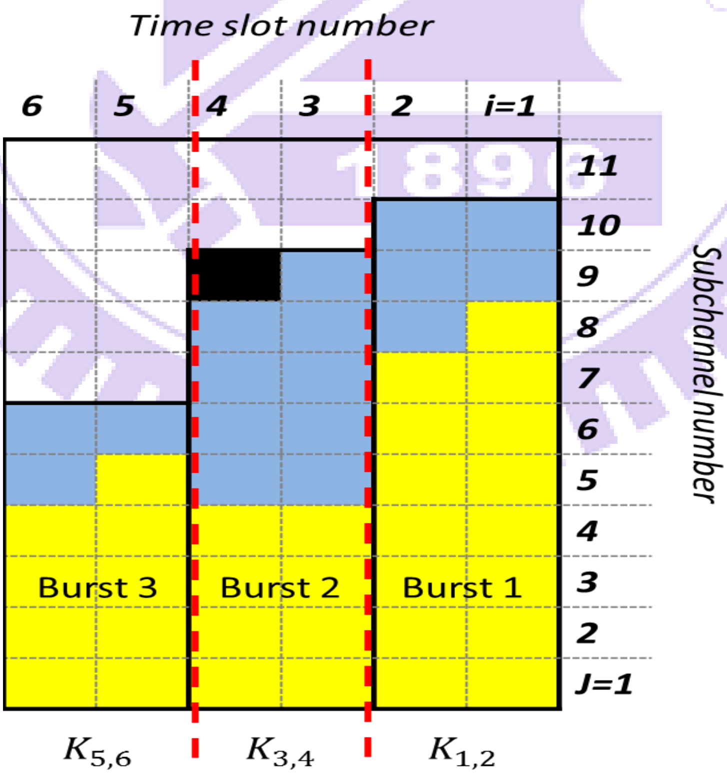

Figure 10. Example of resource allocation

After Phase 1 terminate, there might have remaining requests with MUST data. Thus, the algorithm will allocate MUST data as more as possible in Phase 2 by return some WISH data. There are three steps in Phase 2. In Step 1, data mapper selects the request with largest MUST part from remaining requests, say A_k. In Step 2, find a proper strip to map MUST part of A_k, say M_k. In addition, one burst can be split up to three parts as shown in Fig. 11. One case is that M_k can be allocated into the strip with the smallest available rectangle space. In another case, there is no available rectangle space to allocate M_k, so find the strip with maximum Part C, and remove row by row the slots belonging to Part C. Otherwise, if M_k still cannot be allocated, the algorithm will find the strip with maximum Part B and Part C, then remove all the Part C in the strip. Remove slots of Part B row by row until M_k can be fit in the strip. In the final Step, allocate M_k in the strip that found in Step 2.

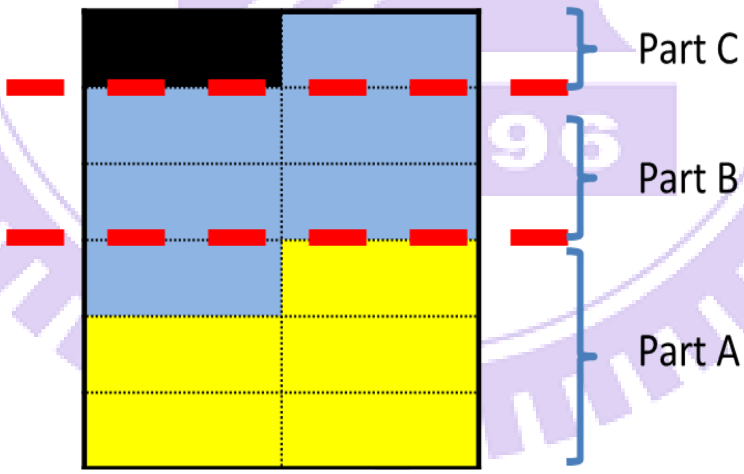

Figure 11. Example of a general burst

Chapter 4

PROPOSED RESOURCE ALLOCATION ALGORITHM

This section presents the proposed two-dimensional rectangular burst mapping algorithm. Assume that there are *s* time slots and *c* subchannels in the DL subframe. Let Ω be the set of all slots in the DL subframe and $\Omega = \{(x, y) | 1 \le x \le s,\ 1 \le y \le c\}$. Therefore, we split time slots into strips, say v_i, which is the $i-th$ column of Ω and $v_i = \{(i, y) | 1 \le y \le c\}, i = 1, 2, ... s$ as shown in Fig. 12. Let $K_{i,j}$ be a subset of Ω, which is consist of adjacent column from column v_i to v_j. Let the rectangle $R_{i,j}$ be the maximum unused slots rectangle space in $K_{i,j}$, and is denoted by $[(x_0, y_0); w, h]$, where (x_0, y_0), w and h are the bottom-right corner, the width and the height.

Let Φ be the set of users and $\Phi = \{1, 2, .., N\}$, $A = \{A_k\}_{k=1}^{N}$ be the set of user requests, M_k be the MUST part data, and W_k be the WISH part data for A_k. In other words, $A_k = M_k + W_k$, $1 \le k \le N$. The proposed algorithm consists of two phases. In Phase 1, to enhance the system throughput, we map requests with high MCS first.

After Phase 1 completed, there might be some remaining requests with MUST data and many unused slots in DL subframe. In order to reduce the number of unused slots and to serve more urgent data, we design an efficient subroutine to return WISH part packets for more MUST data allocation.

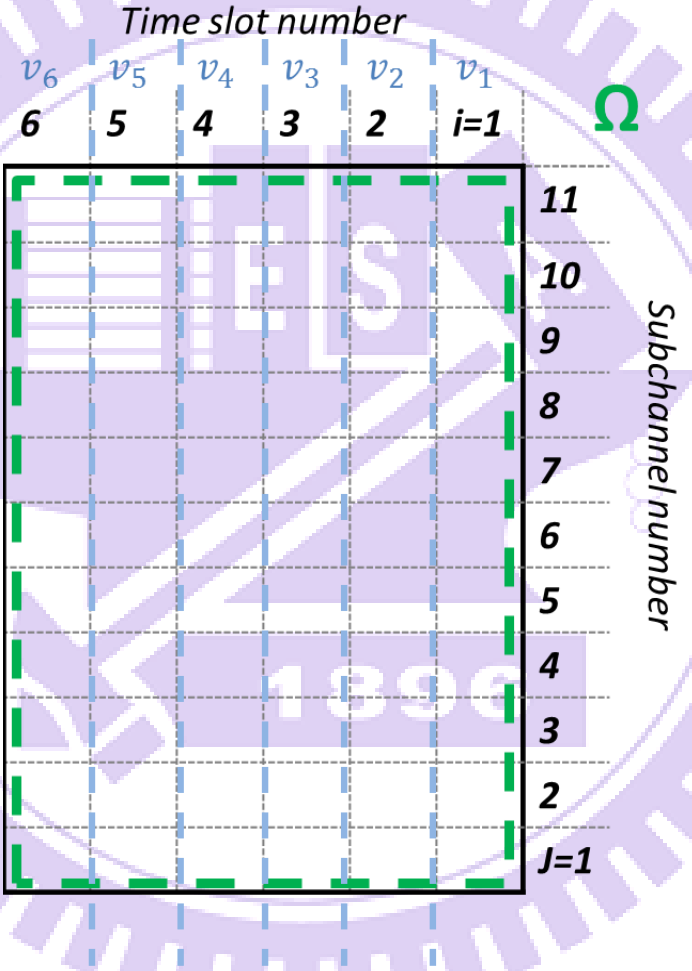

Figure 12. Example of frame structure

In the first phase, it consists of four steps as follow. First, let G_m be the set of MSs which support modulation level *m*, $1 \le m \le M$, and *M* be the highest modulation level. As the algorithm proceeds, the DL subframe will be split into some vertical strips. Let Ψ be the set of all strips, initially $\Psi = \{K_{1,S}\}$ and $m*$ be the target group index, initially $m* = M$.

Step 1: Find a strip has the maximum unused slots rectangle space $R_{i*,j*}$ in Ψ , say $K_{i*,j*}$. And $R_{i*,j*}$ is denoted by $[(x_0, y_0), W, H]$, where (x_0, y_0), W and H are the bottom-right corner, the width and the height.

i. If $|R_{i*,j*}|=0$ or G_1 is empty, the first phase terminates and goes to second phase. Note that $|R_{i*,j*}|$ means the area of $R_{i*,j*}$.

ii. Else if $G_{m*} = \varnothing$, reduce $m*$ by1 and repeat Step 1.

iii. Otherwise, we sort the requests in descending order from G_{m*}. Then, we map requests to DL subframe in descending order. Go to Step 2.

Step 2: we allocate the requests of G_{m*} in descending order iteratively. Let A_k be the target request for allocation and r be the summation of allocated requests, initially, $r = 0$. If all requests have been scanned by following processes, go to Step3.

i. If $r + A_k > |R_{i*,j*}|$, return A_k back to G_{m*}, then find the next one from G_{m*}, repeat Step 2.

ii. If $r + A_k \le |R_{i^*,j^*}|$, we replace r_m by $r + A_k$ and remove A_k from G_{m^*}. Then find the next one from G_{m^*}, repeat Step 2.

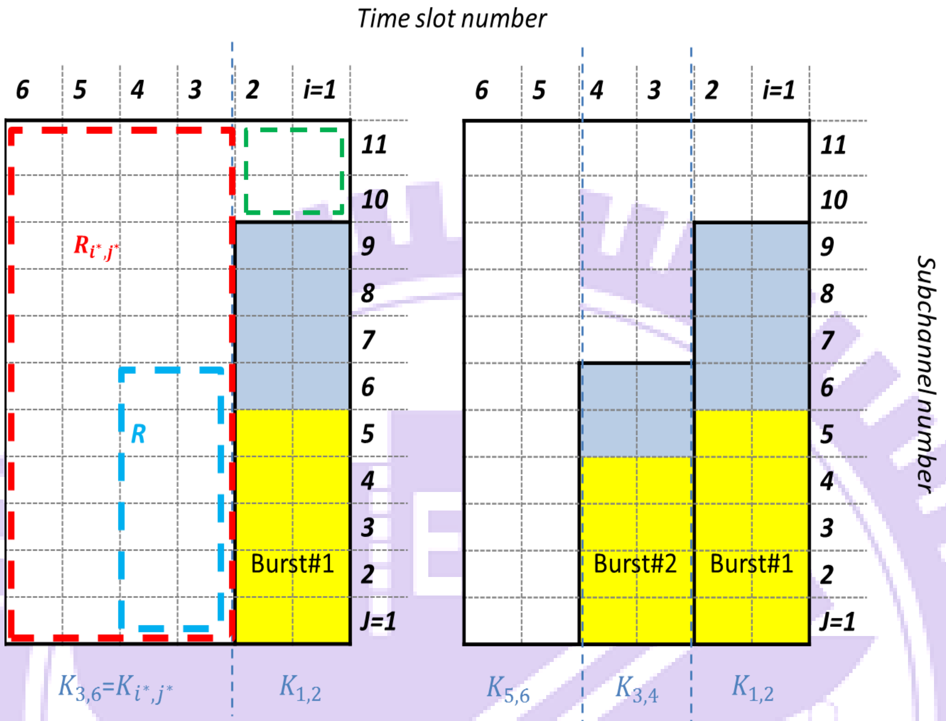

Figure 13. Example of found R_{i^*,j^*} to allocate requests

Step 3: Determine the shape of requests, and there are two cases for K_{i^*,j^*}.

i. In this case, there is no request allocated in the strip K_{i^*,j^*}. Calculate the width $w = \lceil r / H \rceil$, the height $h = \lceil r / w \rceil$, and denote $R = [(x_0, y_0); w, h]$ where R is a rectangle area allocated r. Allocate R into the bottom-right corner of strip K_{i^*,j^*}. If the width of R is smaller than W then split the strip K_{i^*,j^*} into K_{i^*,i^*+w-1} and K_{i^*+w,j^*}. Update $\Psi = (\Psi - \{K_{i^*,j^*}\}) \cup \{K_{i^*,i^*+w-1}, K_{i^*+w,j^*}\}$ as shown in Fig. 13.

ii. There are some requests allocated in $K_{i*,j*}$, we fix the width w equal to strip's width W and calculate the height $h = \lceil r/w \rceil$. Denote $R = [(x_0, y_0); w, h]$ then map it into $K_{i*,j*}$.

Step 4: Update $R_{i*,j*}$, reset r and reduce $m*$ by 1. Return to Step 1.

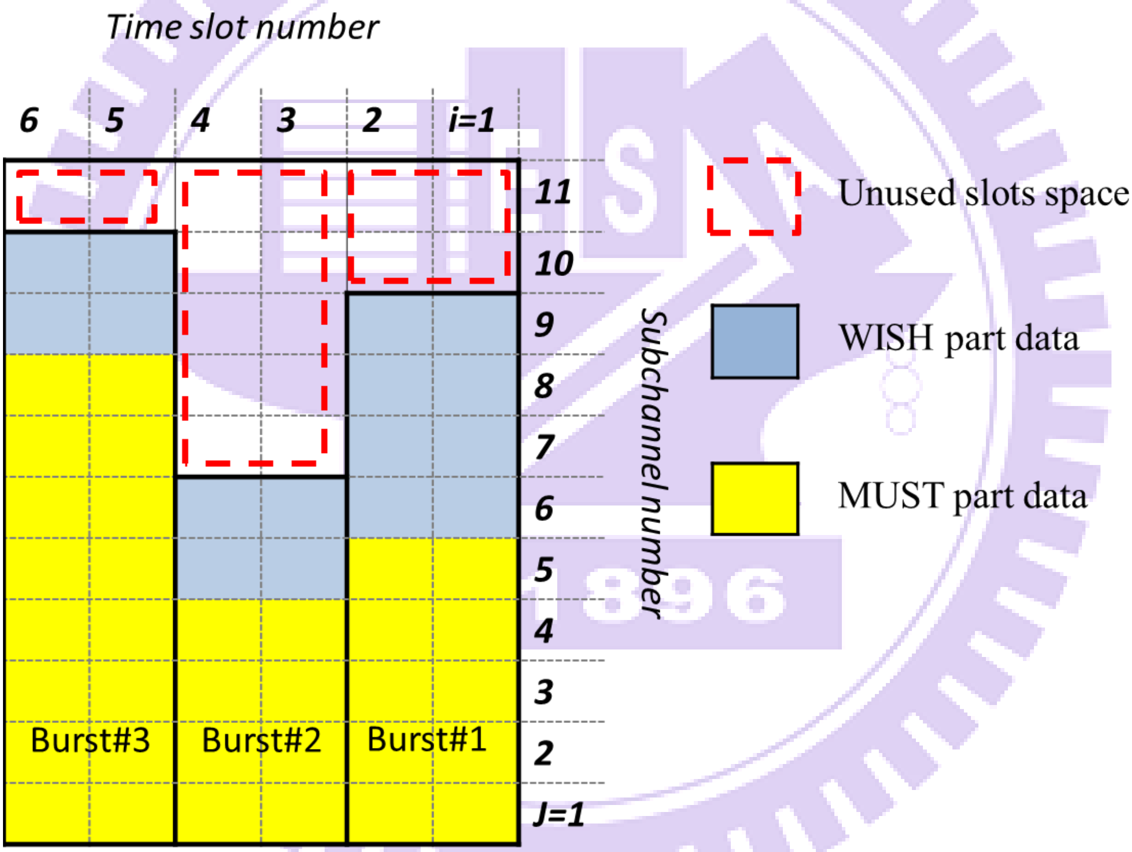

Figure 14. Example of phase 1 has finished

After Phase 1 is completed, there might have remaining requests and many unused slots in DL subframe as shown in Fig. 14. In Phase 2, we allocate high MCS MUST part data as much as possible to enhance throughput and channel utilization. Let $\Psi = \{K_{1,p_1}, K_{p_1+1,p_2}, \ldots, K_{p_{l+1},s}\}$, $1 \le p_1 \le p_2 \le \ldots \le p_l \le s$, be the result after phase 1 is terminated and $m*$ be the target group index, initially $m* = M$. In the second phase, it consists of three steps as follow.

Step 1: Select a MUST data of request, say M_k, which has the largest MUST part data in G_{m*}.And there are three conditions in this step.

i. If $G_1 = \varnothing$, the algorithm is terminated.

ii. Else if $G_{m*} = \varnothing$, there is no request in G_{m*}, we have to select request from the other groups and reduce $m*$ by1. Repeat Step 1.

iii. Otherwise, select the largest MUST part from G_{m*}, say M_k, and go to Step 2.

Step 2: In this step, first, we find the strip has maximum unused rectangle space $R_{i*,j*}$ to allocate M_k. And there are two cases in this step as follow.

case 1. If $M_k \le |R_{i*,j*}|$, we map M_k into $R_{i*,j*}$. Calculate the width $w = W$ and the height $h = \lceil M_k / w \rceil$. Denote $R = [(x_0, y_0); w, h]$ then map it into $K_{i*,j*}$. After allocating M_k in $K_{i*,j*}$, if there are unused slots in $K_{i*,j*}$, allocate the WISH part packets of A_k in descending order. In addition, if there have burst with same MSC as M_k in the strip, combine them into a new burst to reduce map overhead.

case 2. Otherwise, $M_k > |R_{i*,j*}|$, check whether the M_k can be fit in a strip. First, we calculate unused slots rectangle space and whole WISH part data in each strip and scan if there have a strip can be fit.

i. If there is no such strip that can fit M_k. Go to Step 3.

ii. Otherwise, calculate the number of bytes of WISH part packets that are returned for each strip. For a single strip, we return whole WISH part data from a burst which use lowest MCS in the strip, then check whether the M_k can be fit. Repeat previous process until there have sufficient space to fit M_k, and then allocate it into the strip. If there are some unused slots in the strip, we allocate highest MCS WISH part packets back to original burst in descending order. Finally, we calculate the number of bytes of returned WISH part packets, say $b_{K_{i,j}}$. After calculating $b_{K_{i,j}}$ for each strip, we select the strip with minimum $b_{K_{i,j}}$, say $K_{i*,j*}$, to allocate M_k and the WISH data of the other strips will put back to original bursts. Remove A_k from G_{m*} and repeat Step 1.

Step 3: In this step, first, we find the strip has maximum unused rectangle space $R_{i*,j*}$ to allocate partial MUST part data, say $M_k^{'}$. Then allocate MUST part packets of A_k in descending order iteratively until the strip is full. Remove A_k from G_{m*} and repeat Step 1.

In Fig. 15, it shows a simple example for case 1. In this case, we assume that M_k need 4 slots. In our proposed algorithm, we find the maximum unused slots rectangle space $R_{i*,j*}$ first, then map M_k into the strip. After M_k is allocated, we can observe that there have some unused slots in this strip. In order to reduce UAW, the algorithm will allocate WISH part data packet by packet in descending order until this strip is full or all WISH part data are fit in.

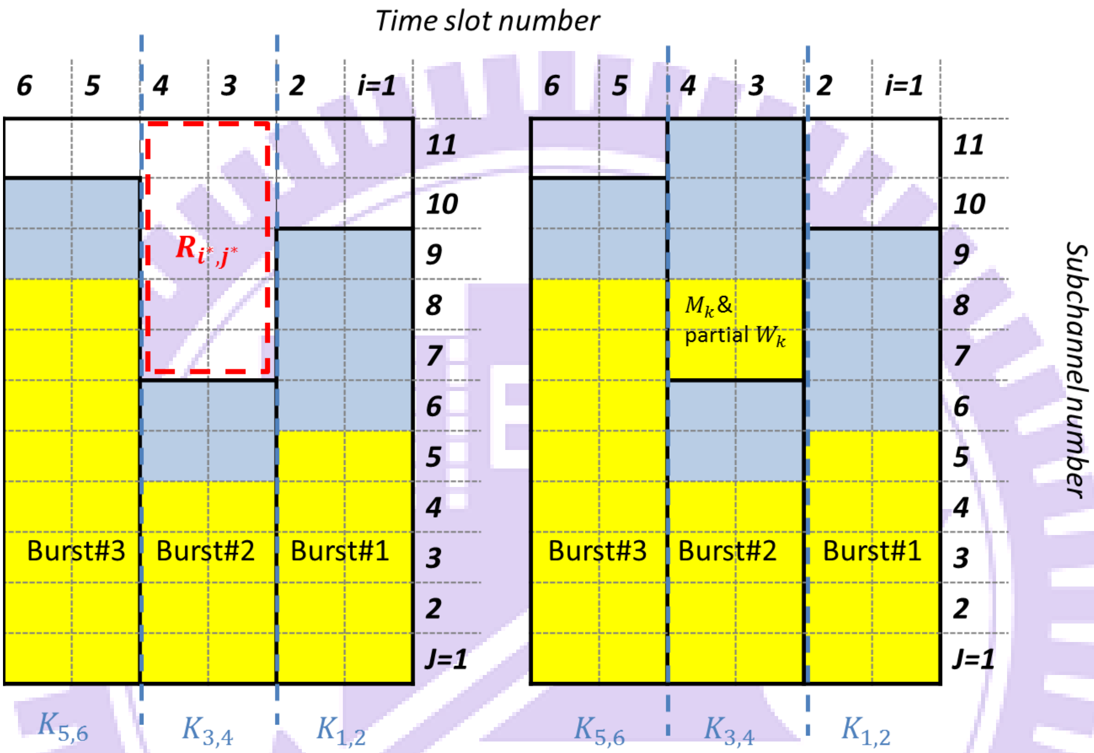

Figure 15. An example of case 1

In Fig. 16, it shows a simple example for case 2. In this case, we assume that M_k need 12 slots to allocate as green dashed line region. WISH data would be returned as brown dashed line region. First, we find that strips $K_{1,2}$ and $K_{3,4}$ have sufficient space to fit M_k. In next step, we calculate $b_{K_{1,2}}$ and $b_{K_{3,4}}$, then select the strip with minimum $b_{K_{i,j}}$. In table 1, u_m denotes the number of bytes which a slot can carry with modulation level *m*. Therefore, we calculate that $b_{K_{1,2}}$ is 216 and $b_{K_{3,4}}$ is 96. Thus, we select strip $K_{3.4}$ to fit M_k.

	Burst#1	Burst#2	Burst#3
m	1	2	3
u_m (bytes/slot)	27	24	18
$b_{k_{i,j}}$ (bytes)	216	96	0

Table 1. An example of u_m and $b_{K_{i,j}}$ for each burst

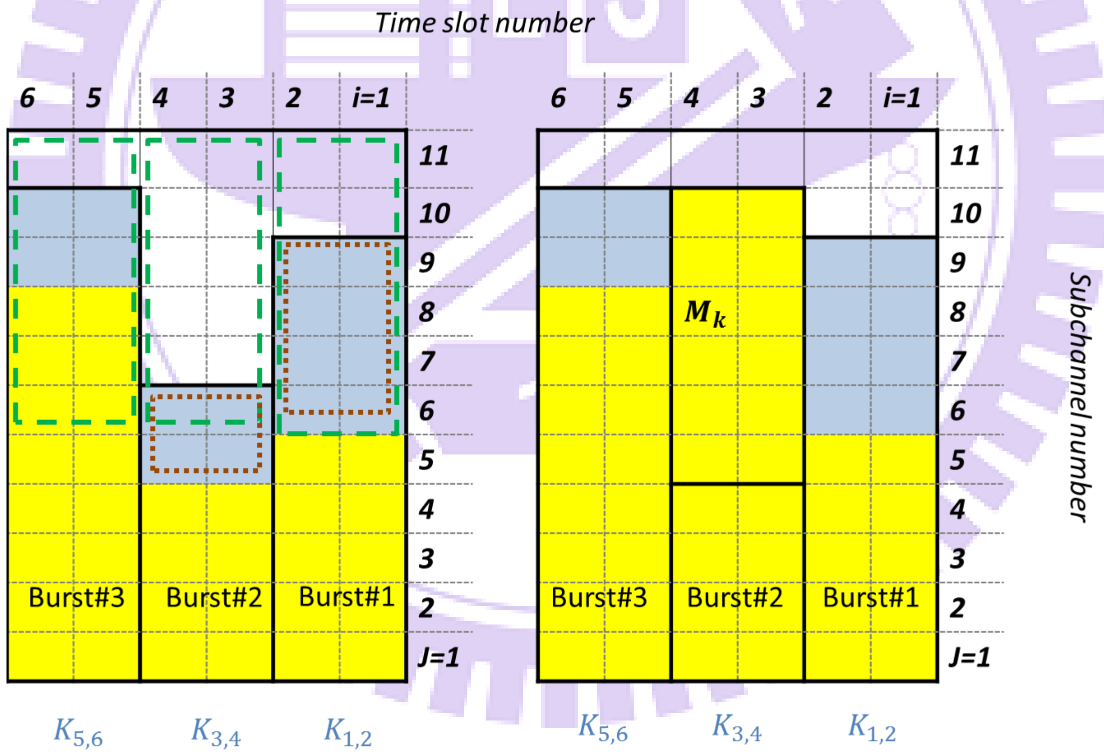

Figure 16. An example for case 2

For the example of Step 3, we assume that M_k needs 22 slots to allocate and consists of three packets [12, 7, 3] in descending order. First, we select the strip $K_{3,4}$

which has maximum unused slots rectangle space. Next step, we select the largest packet to map, but packet size 12 is larger than the number of unused slots 10. Therefore, we select the next one to map. Repeat this process until the strip is full. The result is shown in Fig. 17.

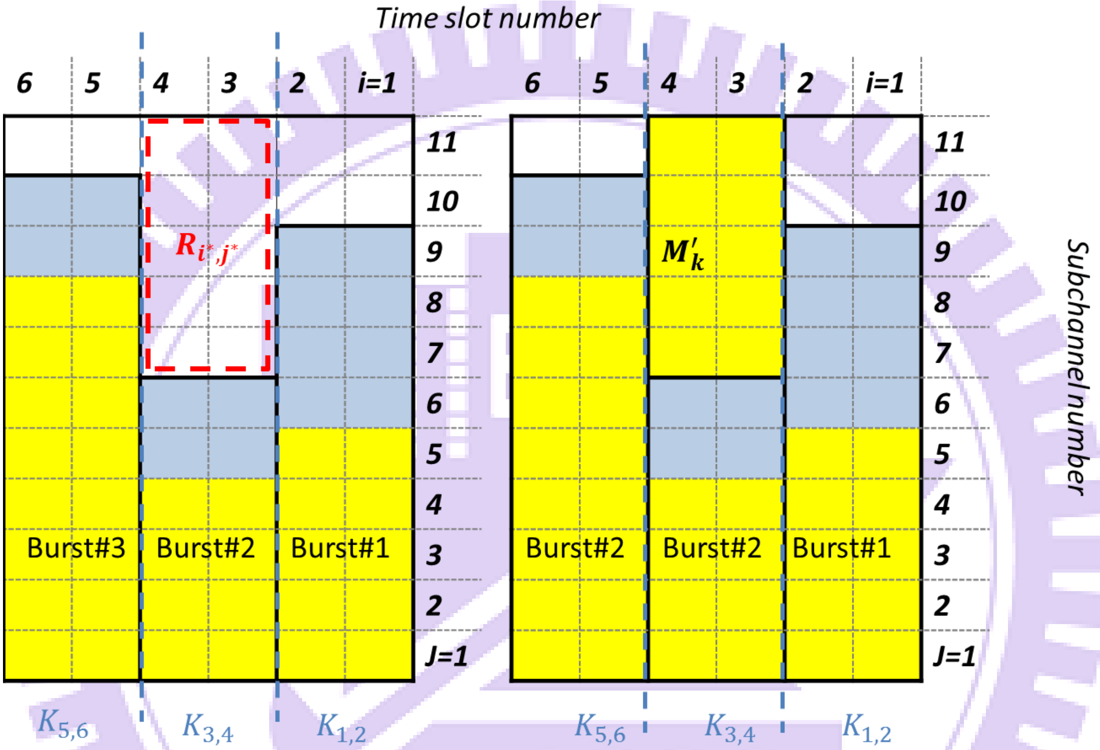

Figure 17. An example for Phase 2 Step 3

Chapter 5

SIMULATION RESULTS

In this section, we use simulations to evaluate the performances of the 2L-DMA and the proposed algorithm.

Parameter	Value
Frame length	5 ms
Channel BW	10 MHz
Permutation scheme	PUSC
Number of subchannels	30
DL subframe	12 slot columns
Modulation and Coding scheme (bytes/slot)	QPSK 1/2(μ_1=6) QPSK 3/4(μ_2=9) 16QAM 1/2(μ_3=12) 16QAM 3/4(μ_4=18) 64QAM 1/2(μ_5=18) 64QAM 2/3(μ_6=24) 64QAM 3/4(μ_7=27)
Must Part Packet size	0-200Bytes
Wish Part Packet size	64-200Bytes
Simulation time	20000 frames

Table 2. System simulation parameters

Table 2 shows that 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 MUST data within 12×30 slots. The channel bandwidth is 10 MHz, frame duration is 5 ms, and run 20000 frames.

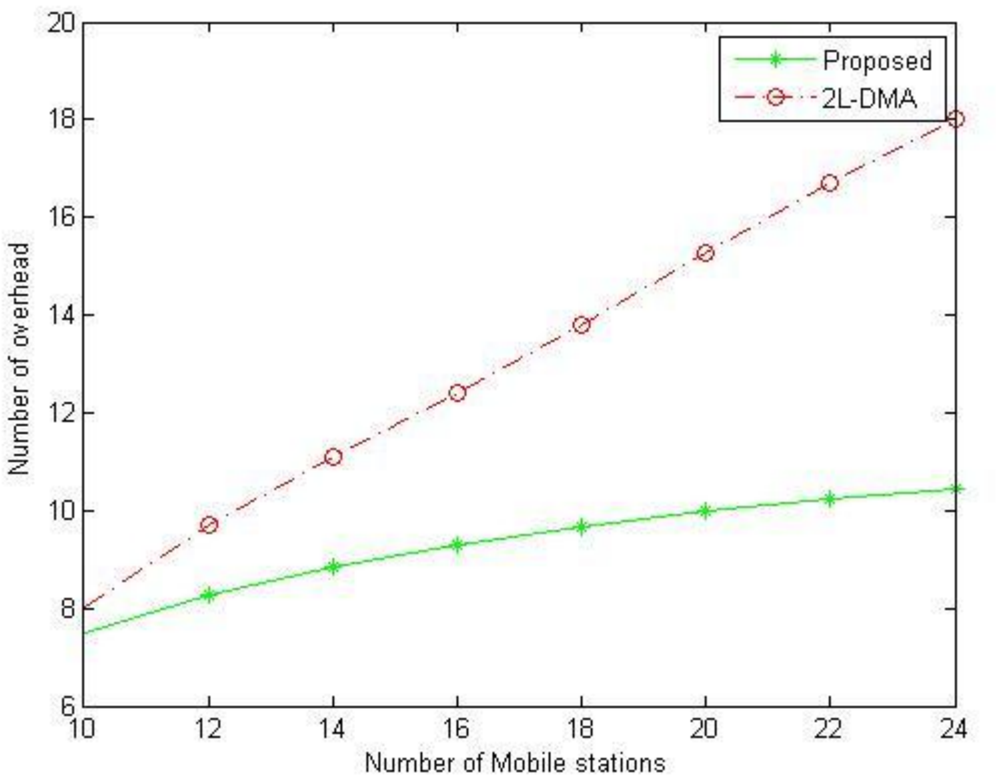

Figure 18. Average overhead

Fig. 18 illustrates the impact of the number of MSs on the average overhead per DL subframe. The average overhead for the proposed algorithm is smaller than that for 2L-DMA. Where 2L-DMA has more overhead because of it allocate one burst for a single MS. However, the requests could be grouped into a burst with the same MCS in our proposed algorithm. In addition, one overhead occupies 5 slots. Overhead would influence on system throughput and channel utilization, when the number of overhead increases.

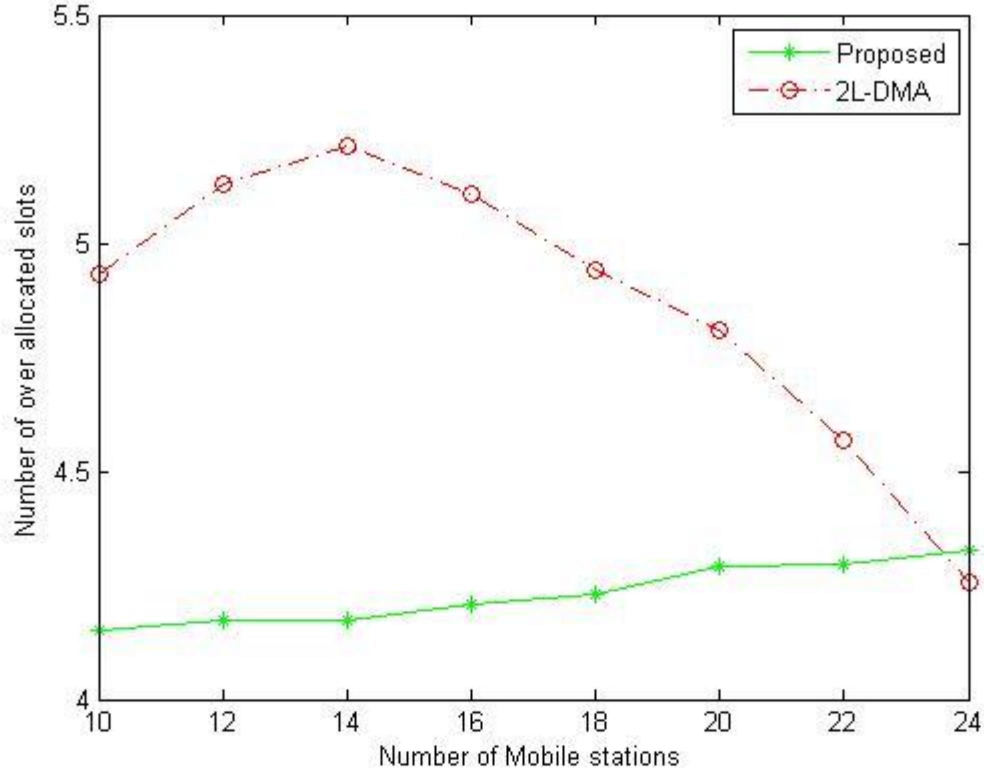

Figure 19. Average over-allocated slots.

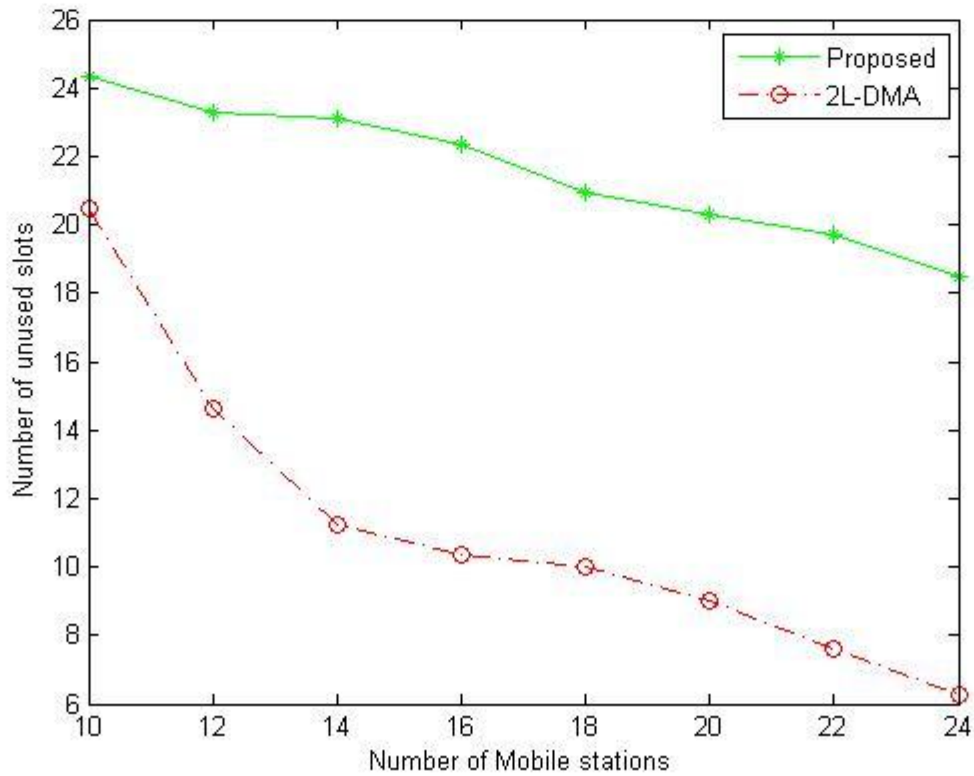

Figure 20. Average unused slots

Fig. 19 illustrates the impact of the number of MSs on the average over-allocated slots and Fig. 20 illustrates the impact of the number of MSs on the average unused slots. In Fig. 19, it is observed that the proposed algorithm has fewer over-allocated slots than 2L-DMA when the number of MSs under 24. The proposed algorithm has better performance because we group requests to allocate into DL subframe. Both the proposed algorithm and 2L-DMA might have more requests cannot be allocated in Phase 1 when the number of MSs increases. In the proposed algorithm, we assume that a request consists of several packets, and we return WISH part data packet by packet to allocate MUST data might produce over-allocated slots. However, 2L-DMA returns WISH part data row by row and selects the row with maximum over-allocated slots to remove data. Therefore, 2L-DMA can reduce more over-allocated slots when the number of MSs increases.

In Fig. 20, it is observed that 2L-DMA has fewer unused slots than the proposed algorithm. The reason is that 2L-DMA didn't consider about packet-level problem. In phase 2, both 2L-DMA and the proposed algorithm have remaining requests, which usually with lower MCS. For 2L-DMA, after allocating the MUST data, it can allocate WISH data row by row until the strip is full. In our algorithm, the size of packet is larger than the unused slots space, we cannot split packet to fit the space. Therefore, 2L-DMA can reduce more unused slots when the number of MSs increases.

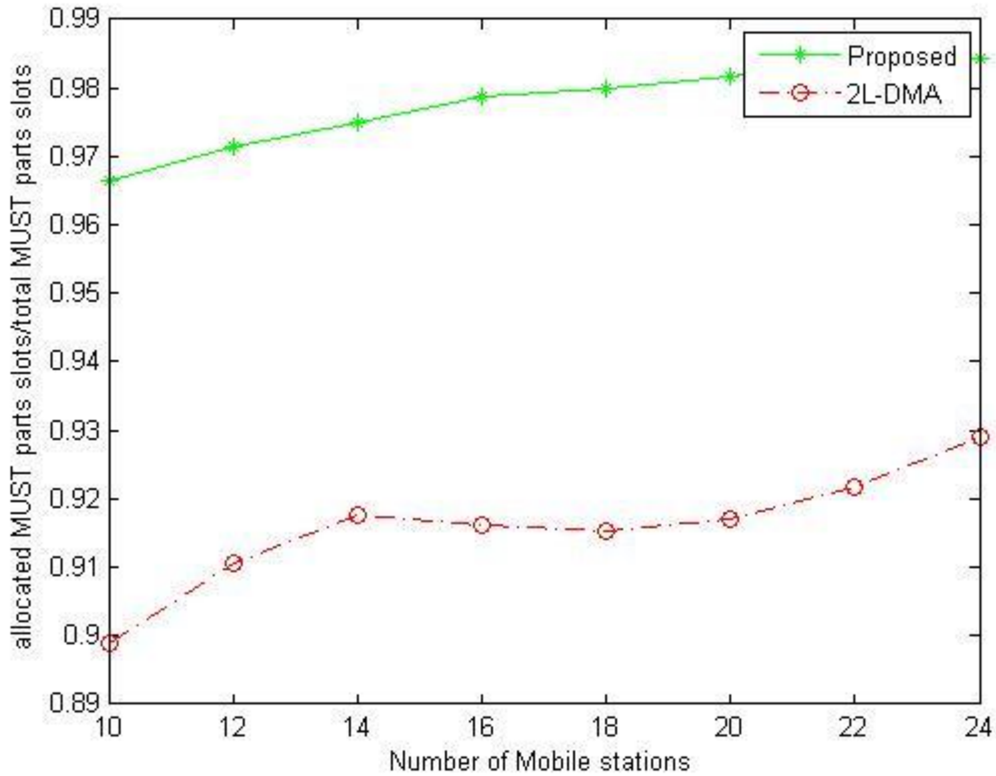

Figure 21. Average MUST part ratio

Fig. 21 illustrates the impact of the number of MSs on the average MUST part ratio. The MUST part ratio can be calculated by using the equation (1). We observe that the proposed algorithm can allocate more MUST data than 2L-DMA. In a common case, there might have some large bursts with low MCS. In our proposed algorithm Step 3 of Phase 2, we could allocate some packets of MUST data to fit the strip. However, 2L-DMA has to allocate whole MUST data to DL subframe. If there is no strip can allocate whole MUST data, the request will be returned to the scheduler.

$$MUST\ ratio = \frac{Allocated\ MUST\ part\ slots\ in\ DL\ subframe}{Total\ MUST\ part\ slots} \qquad (1)$$

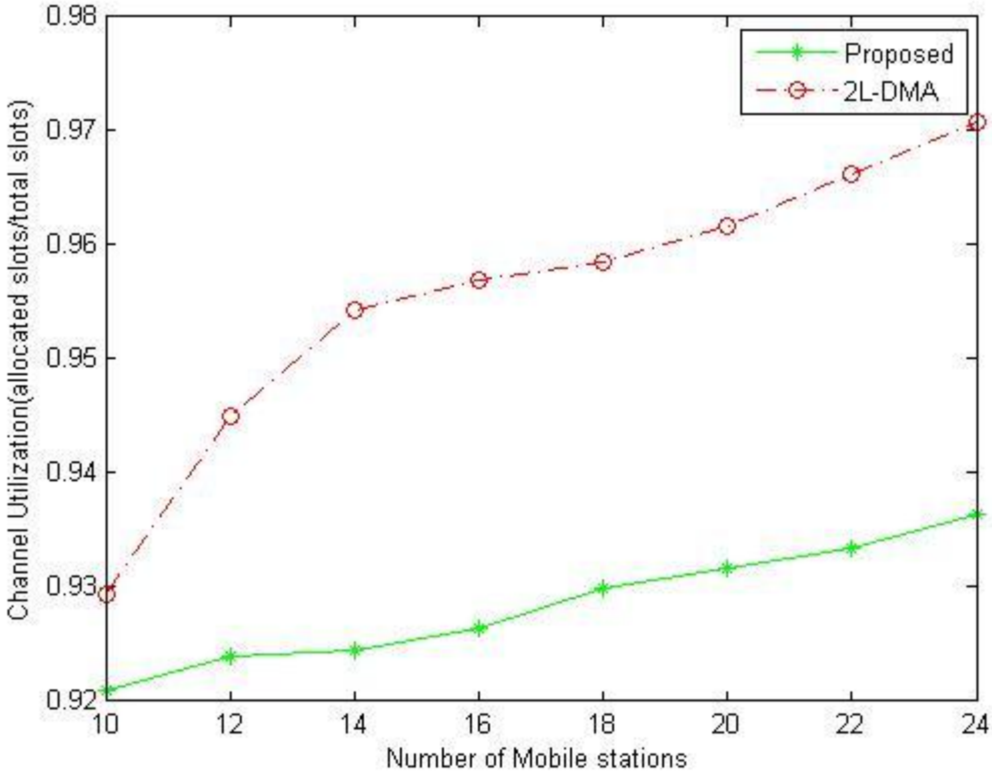

Figure 22. Average channel utilization

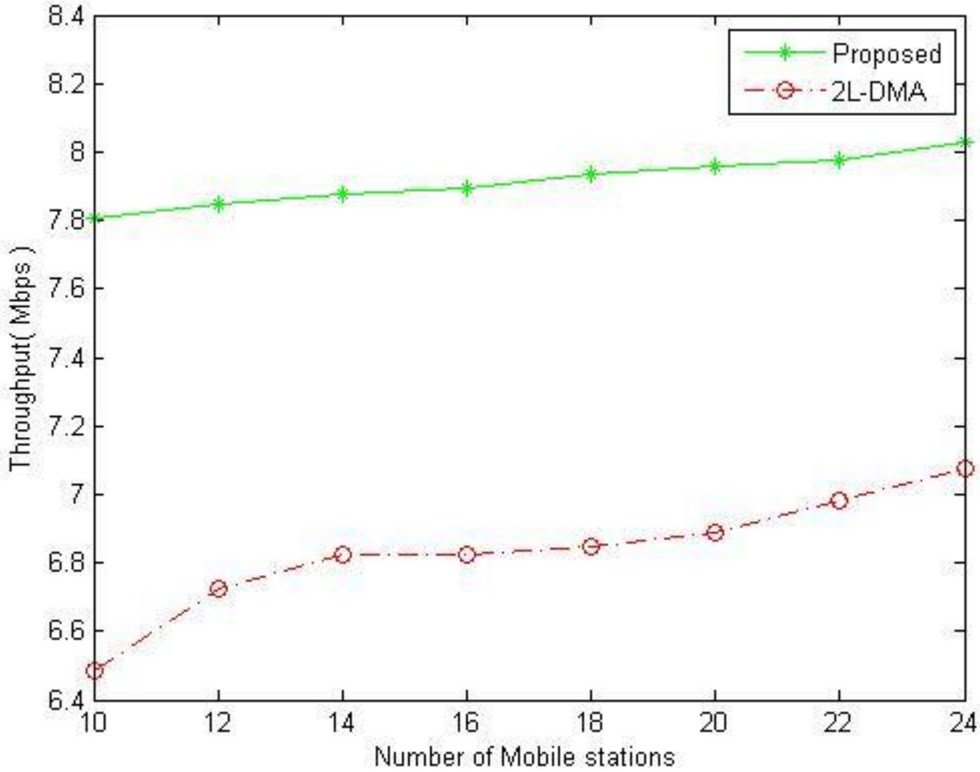

Figure 23. Average throughput

Fig. 22 illustrates the impact of the number of MSs on the channel utilization. The channel utilization can be calculated using the equation (2). Fig. 23 illustrates the impact of the number of MSs on the average throughput. We have lower channel utilization than 2L-DMA because we produce more unused slots. Because we allocate bursts according to MCS but not the size of request that the proposed algorithm achieve higher throughput than 2L-DMA.

$$Channel\ utilization = \frac{Total\ number\ of\ slots\ per\ DL\ subframe - unused\ slots - overallocated\ slots}{Total\ number\ of\ slots\ per\ DL\ subframe} \quad (2)$$

Chapter 6

CONCLUSION

This thesis presents an efficient packet-level mapping algorithm for the allocation of downlink resources for WiMAX systems that operate in PUSC mode. The proposed algorithm meets the rectangle shape allocation constrain, reduces map overheads, achieves high throughput by considering mapping the requests with high MSC first, and considering the bursts allocation in packet-level. In order to reduce over-allocated slots and map overhead, the idea that we group requests to map is from OBBP [3]. In addition, we select the strip with maximum unused slots rectangle space to allocate requests that could improve channel utilization and system complexity as MaxRectangle [4].

The basic idea of the proposed algorithm is to serve MUST data as much as possible by returning less low MCS WISH data. The performance of the proposed algorithm is compared with 2L-DMA. Furthermore, the algorithm is enhanced to improve the ratio of allocated urgent data, map overhead and throughput. Simulation results show that the proposed algorithm outperforms 2L-DMA.

REFERENCES

[1] Erta, C. Cicconetti, and L. Lenzini, "A Downlink Data Region Allocation Algorithm for IEEE 802.16e OFDMA," IEEE ICICS'07, Dec. 2007.

[2] C. So-In, R. Jain, and A. Al-Tamimi, "eOCSA: An Algorithm for Burst Mapping with Strict QoS Requirements in IEEE 802.16e Mobile WiMAX Networks," Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE 2010), Feb. 2010.

[3] Eshanta, M. Ismail, K. Jumari,"An efficient burst packing algorithm for OFDMA systems,"Proceedings of the 2nd International Conference on Computer Technology and Development (ICCTD), Nov. 2010.

[4] T. H. Lee, C. H. Liu, J. Yau, Y. W. Kuo, "Maximum Rectangle-Based Down-Link Burst Allocation Algorithm for WiMAX Systems," TENCON, Nov. 2011.

[5] T. H. Lee, C. H. Liu, A. S. G. Campbell, "A Data Mapping Algorithm for Two-Level Requests inWiMAX Systems,"Vehicular Technology Conference (VTC 2012-Spring), 2012 IEEE 75th.

[6] 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," Vehicular Technology Conference (VTC 2010-Spring), May. 2010.

[7] IEEE 802.16-2009, "IEEE Standard for Local and Metropolitan Area NetworksPart 16: Air Interface for BroadbandWireless Access Systems,"May 2009.

[8] M. Ergen, "Mobile Broadband : Including WiMAX and LTE," Springer Science+Business Media, 2009

[9] X. Zhu, J. Huo, C. Xu and W. Ding,"QoS-guaranteed scheduling and resource allocation algorithm for IEEE 802.16 OFDMA system," in Proc. IEEE ICC'08, May 2008.

[10] Shrivastava, S. ; Vannithamby, R, "Performance analysis of persistent scheduling for VoIP in WiMAX networks," Wireless and Microwave technology Conference,2009.

[11] Satrya, G.B. ; Agung, I.W.P. ; Cahyani, N.D.W,"Performance analysis of packet scheduling with QoS in IEEE 802.16e networks" Telecommunication Systems,

Services, and Applications (TSSA), 2012 7th International Conference on Telecommuncation Systems, Services, and Applications (TSSA).

[12] Xin Jin ; Jihua Zhou ; Jinlong Hu ; Jinglin Shi ; Yi Sun ;Dutkiewicz, E, “An Efficient Downlink Data Mapping Algorithm for IEEE802.16e OFDMA Systems,” Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE.

[13] Ahmed M Husein Shabanni; Prof.M.T.Beg and Ammar Abdul-Hamed Khader,”Survey of Down Link Data Allocation Algorithms in IEEE 802.16 WiMAX” International Journa.