国 立 交 通 大 學

電機資訊國際學位學程

碩 士 論 文

應用於 WiMAX 系統中配置二層需求之演算法

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

研 究 生：高亞蕾

指導教授：李程輝　博士

中 華 民 國 一 百 年 六 月

應用於 WiMAX 系統中配置二層需求之演算法

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

研 究 生：高亞蕾　　Student: Arleth Soleiy Garth Campbell

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

國 立 交 通 大 學

電機資訊國際學位學程

碩 士 論 文

A Thesis

Submitted to EECS International Graduate Program

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

June 2011

Hsinchu, Taiwan, Republic of China

中 華 民 國 一 百 年 六 月

應用於 WiMAX 系統中配置二層需求之演算法

研究生：**高亞蕾**　　　　　　　　　　　指導教授：**李程輝　博士**

電機資訊國際學位學程

國 立 交 通 大 學

摘　要

由於移動式無線網路高服務品質的要求快速成長，寬頻無線存取成為一個有趣且受歡迎的網路架構。正交分頻多工存取(OFDMA) 是 IEEE 802.16e 的物理層中最常被討論的傳輸技術。在 OFDMA 的架構下，媒介擷取控制(MAC)訊框被展開成為兩個維度來看，一個是時間的維度，單位為一個 OFDMA 符號區間；另一個是頻率的維度，單位為一個邏輯上的次通道。OFDMA 系統架構下，資源分配是很重要的一部分。一般來說 資源分配的模組包含兩部分： 排程和資料對映。排程部分，負責產生需求；資料對映部分，負責將需求放置在二維的 MAC 訊框內。由於所有的需求必需要以一個矩形的形狀被放置在訊框內，要找到一個下鏈路資料對映的最佳解為 NP 完全(NP-complete)問題。因此許多研究提出了多樣的啟發式演算法來達到低複雜且高效率的目的。

在本篇論文裡，我們提出了雙級需求資料對應(2L_DMA)演算法，主要目標有二：(1)提供雙級需求：必要(MUST)部分為高優先權的資料；期望(WISH)部分為低優先權的資料。(2) 退回最少的必要部分給排程器，同時維持資料對映機制的效率。我們將此想法實踐在當前擁有高效能的eOCSA 演算法上，我們將會驗證所提出來的 2L_DMA 演算法能夠提高資料對映的效能。模擬結果顯示我們所提出的演算法可藉由最大化系統吞吐量來達到比 eOCSA 更好的效能。

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

Student: Arleth Soleiy Garth Campbell **Advisor:** Dr. Lee Tsern-Huei

EECS International Graduate Program

National Chiao Tung University

ABSTRACT

Broadband wireless access has become a very interesting and popular networking infrastructure, because of the rapidly growing demands of high quality services over mobile wireless systems. Orthogonal Frequency Division Multiple Access (OFDMA) the physical transmission mode adopted by IEEE 802.16e WiMAX is one of the most intensively researched technologies. In OFDMA, the Medium Access Control (MAC) frame is extended over two dimensions; time in units of OFDMA symbol, and frequency in units of logical sub-channel. A very important component of OFDMA systems is resource allocation. In general, the resource allocation module consists of a scheduler and a data mapper. The scheduler generates requests while the data mapper maps those requests into the two-dimensional MAC frame. A constraint of downlink transmission is that every request has to be mapped as a rectangle. It was shown that due to this constraint finding an optimum solution for downlink data mapping is an NP-complete problem. Consequently, various heuristic algorithms were proposed to achieve high efficiency with acceptable complexity.

In this thesis, we propose a data mapping algorithm for two-level requests, which we called 2L-DMA algorithm and it consists of two main targets: (1) apply a Two-Level request: a MUST part, for high priority data (urgent data); and a WISH part, for low priority data (non-urgent data); and (2) return as less possible MUST part to the scheduler, while keeping the mapping scheme efficient. We have implemented our idea on an existing algorithm called eOCSA, a high-performance packing algorithm recently presented. The goal of our proposed data mapping algorithm is to achieve high efficiency. Furthermore, the performance of our proposed algorithm is compared with that of eOCSA, a low-complexity algorithm with satisfactory performance. Experimental results show that our proposed algorithm yields much better performance than eOCSA.

Acknowledgements

I am heartily thankful to my advisor Lee Tsern-Huei, for his support, supervision, guidance, patience, caring and encouragement during my Master study in NCTU. He provided me with an excellent atmosphere for doing research and valuable advices about my work. I would like to thank Shih Ju-Lin, who as a good friend was always willing to help and to give me his best suggestions. I also want to thank Yu Hui-Mei for her support, guidance, patience and assistance in my academic and personal needs. Many thanks to my entire lab mates for helping me through my academic year; they all played an important role during my studies her in Taiwan.

Last, but not the least; special thanks to my Mother and family for their support and encouragement with their best wishes; they have always trusted in me and be there for me, cheering me up through the good times and bad times.

Table of Contents

List of Tables

List of Figures

Acronyms

2L-DMA	2 Level-Data Mapping Algorithm
2L-DMA mv	2 Level-Data Mapping Algorithm modified version
AMC	Adaptive Modulation and Coding
BWA	Broadband Wireless Access
BS	Base Station
BE	Best Effort
CDMA	Code Division Multiplexing
CAC	Call Admission Control
CID	Connection Identifier
DL-MAP	Downlink MAP
eOCSA	enhanced One Column Striping with non-increasing Area first mapping
ertPS	extended real-time Polling Service
FDMA	Frequency Division Multiplexing
FUSC	Full Usage of Sub Channel
FDD	Frequency Division Duplexing
FCH	Frame Control Header
FER	Forward Error Control
IEEE	Institute of Electrical and Electronics Engineers
IE	Information Element
MAC	Medium Access Control
MS	Mobile Station
MSs	Mobile Stations
MCS	Modulation and Coding Scheme
nrtPS	non-real-time Polling Service

OFDMA	Orthogonal Frequency Division Multiple Access
OCSA	One Column Striping with non-increasing Area first mapping
PUSC	Partial Usage of Sub Channel
QoS	Quality of Service
rtPS	real-time Polling Service
TDMA	Time Division Multiplexing
TDD	Time Division Duplexing
UGS	Unsolicited Grant Service
UL-MAP	Uplink MAP
WiMAX	Worldwide Interoperability for Microwave Access

Chapter 1

Introduction

1.1 WiMAX System: An Overview

IEEE 802.16e, known as WiMAX (Worldwide Interoperability for Microwave Access) [1], provide fixed and mobile Internet access and has been deployed in some areas as a broadband wireless access (BWA) technology. According to the WiMAX Forum, a non-profit association formed to ensure the compatibility and interoperability of IEEE 802.16 devices; OFDMA is specified as the air interface because of its capability to reduce multi-path fading and achieve multi-user diversity, when compared with other alternative technologies such as Frequency Division Multiplexing Access (FDMA), Time Division Multiplexing Access (TDMA), and Code Division Multiplexing Access (CDMA).

In this thesis, we consider downlink transmission in a WiMAX system. The mobile WiMAX system consists of a base station (BS) and several mobile stations (MSs). The BS is responsible for performing most of the system decisions. To support quality of services (QoS), those decisions include Call Admission Control (CAC), Scheduling and Resources Allocation [2]. The CAC module, determines if a new connection is to be accepted or rejected based on the available system capacity. For accepted connections, the network traffic must be properly prioritized according to a certain scheduling policy. The scheduler decides the services order of the user's data waiting to be transmitted. For convenience, we call requests, the data of a user generated by the scheduler to be serviced. 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].

After the scheduler has generated the requests, the next step is to perform resources allocation (or requests allocation), also known as data mapping. In a WiMAX system, the minimum data transmission unit is a slot, which is constituted of one or more sub-channels by one or more OFDMA symbols. A constraint for downlink transmission is that every request must be logically mapped into a two-dimensional rectangle, called burst. With such a constraint, it is highly likely to have unused and over allocated slots after the data mapping process. An unused slot defines a slot which is not allocated to any user or left unused in a certain frame; and a slot is over allocated if it is assigned to some user but not being utilized for transmitting data. The efficiency is defined as the ratio of slots used to transmit data to the total number of slots. It was shown that, to find the optimal solution that achieves maximum efficiency, the complexity of the data mapping process in NP-complete [5]. As a result, various heuristic data mapping algorithms were proposed to achieve high efficiency with acceptable complexity.

According to the standard, it is allowed to assign multiple bursts to a user. However, the DL-MAP overhead increases for such an allocation because each burst has its own Information Element (IE). It is also possible to combine multiple requests in a single burst if the modulation and coding schemes of the MSs of those requests are identical. A user can use Connection Identifier (CID) to retrieve its data if its request is combined with other user's requests in the same burst. Note that a request which is not mapped and transmitted in the current downlink sub-frame is returned to the scheduler for its transmission in a later frame.

1.2 OFDMA Resources Allocation

In the OFDMA mode, the available spectrum bandwidth in a frame is divided into several orthogonal subcarriers and these subcarriers are grouped into logical sub-channels. The association between physical subcarriers and logical sub-channels is called permutation mode. According to the standard the permutation modes is defined into two sets: distributed permutation modes (PUSC, FUSC) and adjacent permutation modes (like band-AMC).

In distributed permutation mode logical sub-channels are built from physically distributed subcarriers along the available frequency spectrum; and in adjacent permutation mode sub-channels are built from physically adjacent subcarriers. Consequently, a certain resource allocation mechanism designed for distributed permutation mode will not perform well for an adjacent permutation mode, or vice versa. The distributed permutation mode are the one that have attracted more attention so far in the literature, while some research paper [6] [7] indicate that adjacent permutation mode can be an interesting alternative, though more challenging.

As mentioned before, the minimum transmission unit in WiMAX system is a slot, which consists of one or more sub-channels by one or more OFDMA symbols. However, this depends on the set of sub-carrier permutation mode. In this thesis, we only consider PUSC. PUSC is the type of sub-carrier distribution in which the sub-channels are available in the downlink and uplink sub-frame. In the downlink sub-frame one slot is consisting of two OFDMA symbols by one sub-channel; and in the uplink sub-frame one slot is three OFDMA symbols by one sub-channel.

Transmission in OFDMA is done in a time frame basis. Each frame is of 5 ms duration [8]. In order to achieve bi-directional communication, WiMAX system can employ frequency division duplexing (FDD) in which downlink and uplink use different frequency bands; and time division duplexing (TDD) in which the downlink traffic follows the uplink traffic in the time domain.

In this thesis, we make use of TDD technology, where the same frequency can be use for downlink and uplink transmission. In other words, the TDD provides a flexible partitioning of the frame for data traffic. Dividing the frame into downlink and uplink sub-frames. Although we use the TDD system, the FDD technology can also be used.

The Mobile WiMAX downlink sub-frame starts with a downlink “preamble”, which is used for synchronization; a Frame Control Header (FCH), which describes the length of the DL-MAP message; followed by the DL MAP and UL MAP, these maps contain variable number of Information Element (IEs), which specifies each burst. As shown in fig. 1. The IE contains information of the burst start time and end time, a modulation type and a forward error control (FEC) if used. One IE occupies a slot and once occupied by an IE, such slot can’t be used to transmit traffic data. The remaining space is occupied by the actual traffic data.

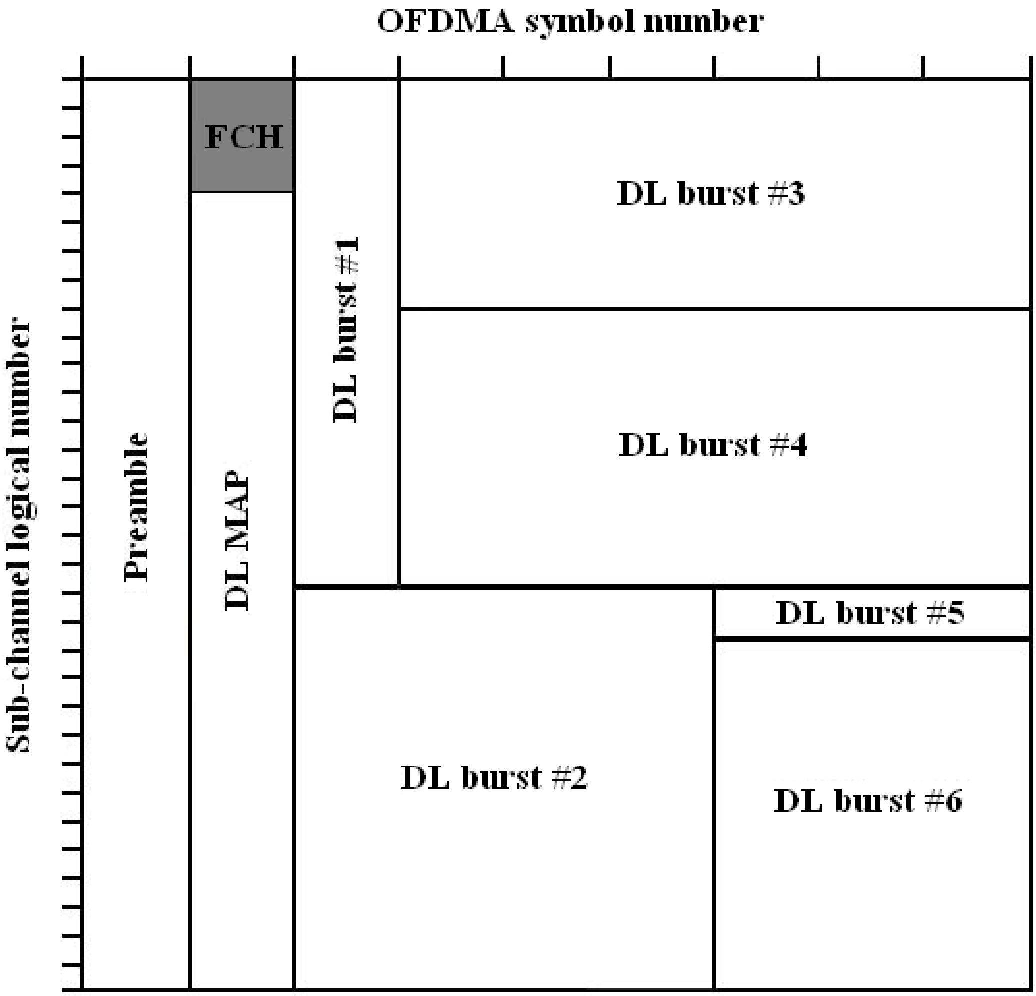


Fig. 1 OFDMA DL sub-frame structure

The IEEE 802.16 standard defines the mapping algorithm for the uplink traffic, which is the traffic from MSs to BS. But the downlink traffic mapping policy is not specified, allowing differentiation among designer. In order to design an efficient downlink mapping algorithm, some restriction are imposed:

- ✓ All data been sent in a burst must be transmitted using the same Modulation and Coding Scheme (MCS). A burst is an allocation for transmitting data aimed to one or more MSs. The MCS used by a burst is declared in its corresponding IE in the DL-MAP.

- ✓ The shape of a burst region is mandatory rectangular. This shape is specified in the burst IE as: starting symbol and sub-channel number, and width and height of the burst in symbols and sub-channel. Thus, a burst can't overlap each other.
- ✓ The dimensions of any allocated burst must be multiple of the minimal allocation unit, called a slot. The size of a slot depends on the permutation mode used.

Also, in order to design an efficient downlink mapping algorithm, some factor has to be taken into account: which are MAP overhead, unused slots, over allocated slots, and QoS preservation. When the data mapper inserts a new burst into the frame, also the respective IE must be added to the DL-MAP. The DL-MAP uses slots which could be used to transmit data. Meaning that, there is a certain MAP overhead directly proportional to the number of bursts in a frame. Reducing the number of burst may leave more slots for sending data. A way of reducing the number of burst is by grouping data from several MSs which use the same MCS in a certain frame, so it can be sent in the same burst.

Due to the specific mapping policy, it is possible that some slots of the frame didn't get finally assigned to any burst. These unused slots are considered as a waste of bandwidth, so the mapper should minimize its number. Also due to both the rectangular shaping and the slot restrictions, some space may be internally wasted in a burst; called as over allocated slots. For example, assume that the BS has to transmit seven slots of data to an MS using PUSC mode, the slot size is two symbols per one sub-channel. A possible allocation would be a burst of 2x4 or 4x2 slots. The remaining

1 slot will be wasted. These over allocated slots can severely impair the resource utilization and should also be minimized.

1.3 Motivation and Objective

In IEEE 802.16e the BS is responsible for mapping or allocating the requests into the MAC frame, for both downlink and uplink sub-frame. The MAC frame is extended over two dimensions: time in units of OFDMA symbol and frequency in units of logical sub-channel. The IEEE 802.16e Mobile WiMAX standard requires that all requests must be rectangular in shape when mapped into the downlink sub-frame. In WiMAX system this mapping process is a mapping problem; also called a bin packing problem; and it is known to be NP-complete [5], since the constraint require that all user's request has to be mapped as a rectangle into the downlink sub-frame. When fulfilling this constraint and after the mapping process is done, the MAC frame will result with over allocated slots and unused slots; which lower the system throughput. For this reason, the rectangular criterion requires an efficient two-dimensional mapping algorithm.

Furthermore, also due to this constraint, there are often requests that can't be mapped into the current frame, and will have to be returned to the scheduler for a later transmission in the next frame. An issue is that those unmapped request can contain urgent data that will need to be mapped into the current frame and can't wait for a later transmission in the next frame. To the best of our knowledge, there are no algorithm that consider requests with urgent and non-urgent data (two kind of data); the majority of the proposed algorithm only focus on the resources allocation [9] [10] [11] [12].

This motivated us to propose an algorithm that handles multi-level requests. In this thesis, we propose a data mapping algorithm for two-level requests, which we called 2L-DMA algorithm; and it consist of two main targets: (1) apply a Two-Level request: MUST part for high priority data, and WISH part for low priority data. In other words, all requests can consist of two parts: MUST part, for urgent data and WISH part, for non-urgent data. The main objective of implementing Two-Level request is that we can use it as a priority mechanism. (2) return as less possible MUST part to the scheduler, while keeping the mapping scheme efficient. Our algorithm focuses on minimizing the unused slots in order to maximize the efficiency, while accomplishing (2).

Our main targets have been implemented on an existing algorithm called eOCSA, a high-performance packing algorithm recently presented. Which is an enhanced version of the algorithm called OCSA or One Column Striping with non-increasing Area first mapping [11]. Similar to OCSA, the enhanced algorithm is also simple and fast o implement; however, eOCSA considers the allocation of an additional resource to ensure the QoS. Without this additional column's consideration; eOCSA can also roll the additional columns needed for the current frame to the next frame before beginning the next frame mapping. However, this may cause an extra delay. Moreover, without the extra columns a priority mechanism needs to be applied.

For example, the resource allocation with the highest priority is moved to the beginning of the mapping queue thus being mapped regardless of the largest size consideration. Therefore, this may lead to more unused slots. As mentioned before, our main targets are implemented on eOCSA algorithm; however, in our proposed 2L-DMA algorithm we do not take into consideration the additional columns.

Our proposed 2L-DMA algorithm is divided into two phases: First Phase and Second Phase. The first phase consists of mapping all requests according to eOCSA algorithm (eOCSA mapping). Therefore, taking into account that all requests can consist of two parts (two-level): MUST part and WISH part; and when mapping into a rectangle, all MUST part is allocated before the WISH part. The first phase is completed when there are no spaces left in the downlink sub-frame or there are no requests that can be fitted into the available spaces; we will call these requests "unmapped requests".

Note that these unmapped requests can contain MUST part and WISH part. In the second phase of our algorithm will give higher priority to unmapped MUST part over unmapped WISH part. The second phase consists of mapping all the MUST part as possible for those unmapped requests (Unmapped MUST part mapping).

Consequently, to achieve this, some mapped WISH part in "eOCSA mapping" will have to be sacrificed (partially removed) in order to map as possible unmapped MUST part. The second phase is completed when there are not enough available spaces and no mapped WISH part that can be removed in order to map unmapped MUST part or there are no more unmapped MUST part that need to be map.

We proposed the data mapping algorithm for two-level requests to evaluate the system efficiency of a WiMAX system. The simulation result show that our proposed 2L-DMA algorithm can provide better performance compared to that of eOCSA as proposed in [12]. Furthermore, we also use an example of our 2L-DMA algorithm to demonstrate how it works. Above all, our 2L-DMA algorithm is proposed to improve in the data mapping process.

1.4 Organization of the Thesis

The rest of this thesis is organized as follow. In **chapter 2,** the system description of eOCSA algorithm is introduced and described in detail, since our proposed 2L-DMA algorithm is implemented on eOCSA "a low-complexity algorithm with satisfactory performance". In **chapter 3,** our Data Mapping Algorithm for Two-Level Requests in WiMAX systems, called 2L-DMA Algorithm is briefly introduced and described. In **chapter 4,** the information of the performance comparisons of our 2L-DMA algorithm when compared with that of eOCSA is provided. Finally, the conclusion of this thesis is drawn in **chapter 5.**

Chapter 2

Related Work

As mentioned earlier, in this thesis we propose a data mapping algorithm for two-level requests. Our proposed algorithm main targets are implemented on an existing heuristic algorithm called eOCSA. Since our 2L-DMA algorithm is implemented on eOCSA “a low-complexity algorithm with satisfactory performance”.

In this chapter we will briefly introduce and describe in detail the eOCSA algorithm system description. eOCSA is the enhanced version of One Column Striping with non-increasing Area first mapping, for two-dimensional downlink burst mapping in IEEE 802.16e Mobile WiMAX networks.

In order to maximize the efficiency, eOCSA consider the mapping in the descending order of the resources allocation size (largest first). Then, the allocations are mapped from bottom to top and from right to left into the downlink sub-frame; this allow the space for the variable portions of the DL-MAP and UP-MAP to be adjusted accordingly in the left part of the downlink sub-frame.

eOCSA algorithm does not consider all possible mapping pairs, it consider only one best mapping pair either the least width (vertical mapping) or height (horizontal mapping). The eOCSA algorithm consists of four steps, which we will briefly describe as follow.

2.1 eOCSA System Description

First step, when given a set of requests (resources allocation) $\{R_i,\ i=1,2,3...N\}$, sort the set of resources allocation in the descending order and select the largest element to map first. **Second step “vertical mapping”**, consists of mapping this resource allocation into the downlink sub-frame. Given an R_i , the algorithm maps the width-height pair (W_i, H_i) for the burst:

$$W_i = \lceil R_i / H \rceil \tag{1}$$

$$H_i = \lceil R_i / W_i \rceil \tag{2}$$

Where, $\lceil\ \rceil$ denotes the ceiling function, and H is the maximum available height in the downlink sub-frame. With 10 MHz Mobile WiMAX, H is 30 sub-channels. Note that this ensures that the mapped region is bigger than the required resource allocation $(W_i \times H_i \geq R_i)$ and that the rectangle has the maximum possible width (minimizing MS active time and energy).

After a resource allocation is mapped into the downlink sub-frame, some space may remain unallocated above the just mapped burst. **In the third step “horizontal mapping”**, the algorithm tries to assign this space (which they call strip) to the next largest element that can be fitted in; for example, j^{th} allocation. In this step the region width is fixed, and it is used to determine the required height for the next largest element that can be fitted into this available space.

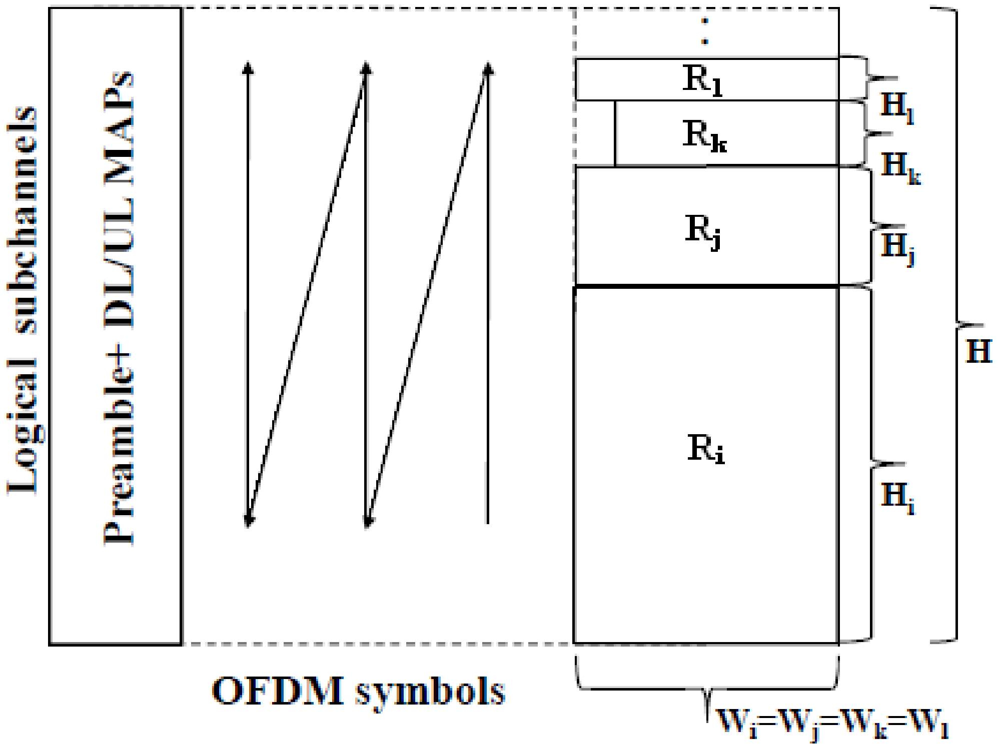


Fig. 2. An example of mapping downlink burst using eOCSA

Find largest R_j, such that $R_j < W_i \times H_0$,

$$H_j = \lceil R_j / W_i \rceil \tag{3}$$

$$W_j = \lceil R_j / H_j \rceil \tag{4}$$

Here, $H_0 = H - H_i$ is the maximum available height in the strip. This step is repeated until either no space is left vertically, or there is no allocation that can be fitted in the available space.

If no allocation can be found to fit, the algorithm moves leftward to fill the remaining empty column in the downlink sub-frame by moving back to step 2, and selecting the next largest element to map.

Fig. 2 shows the process of moving vertically and horizontally from right to left and bottom to top. **The fourth step,** would be adding an additional column; they consider this possibility of having extra resources to assure the QoS. From there simulations, they conclude that in order for the system to support all resources allocation, on average only one more slot column is needed. However, in our algorithm first phase we don't adopt the eOCSA fourth step (adding an additional column); we only consider **step 1, step 2** and **step 3.**

2.2 An eOCSA Example

In this section, we provide an example that helps explain eOCSA algorithm. In this example the scheduler makes an allocation decision for 10 mobile stations (MSs) in a Mobile WiMAX sub-frame.

Table I show the 10 MSs that were chosen and have been allocated R_1 through R_{10} by the scheduler. The sum of all resources allocation is 360 or 12×30.

Table I

Ten Random Resource Allocations: Example

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
Resource (slots)	85	65	61	35	34	33	31	12	2	2
Mapping: (Width, Height)	3, 29	3, 22	3, 21	2, 18	2, 17	2, 17	2, 16	3, 4	3, 1	3, 1
Over Allocated Slots	2	1	2	1	0	1	1	0	1	1

First, the algorithm sorts all resources allocation in the descending order of the request size and select the largest resource to map first (step 1). Note that in table I, we have already sorted the resources in the descending order. The largest resource allocation $R_1 = 85$ is chosen.

Applying step 2, we get a width of $\lceil 85/30 \rceil = 3$ columns and a height of $\lceil 85/3 \rceil = 29$ rows. The rectangle 3×29 results in an over allocated of 2 slots. The downlink sub-frame mapping is done from right to left and from bottom to top. The mapping of R_1 into the sub-frame leaves a strip of 3×1.

In step 3, the algorithm chooses the next largest resource allocation that can fit into the remaining strip, which is $R_{10} = 2$. And it is mapped as $3 \times \lceil 2/3 \rceil$ or 3×1, resulting in 1 over allocated slot. Since there is no left-over space within this strip, we repeat step 2 by moving horizontally to the left.

The next largest resource allocation is $R_2 = 65$, mapped into the downlink sub-frame in a rectangle of width $\lceil 65/30 \rceil = 3$ and height $\lceil 65/3 \rceil = 22$. The rectangular mapping of 3×22 results in an over allocated of 1 slot and a left-over strip of 3×8 on the top. We then move to step 3 to fill the 3×8 strip.

The next largest resource that can fit into this space is $R_8 = 12$; which is mapped to a rectangle of $3 \times \lceil 12/3 \rceil$ or 3×4 resulting in no over allocation and a left-over space of a 3×4 strip. We then repeat step 3 to find the next largest rectangle that can fit into this remaining strip.

At this time, $R_9 = 2$ being mapped to a rectangle of $3 \times \lceil 2/3 \rceil$ or 3×1; resulting in 1 over allocated slot and a left-over space of 3×3 on the top. Since there is no more resource that can be fitted into the remaining space, the algorithm repeat step 2 by moving horizontally to the left.

The next largest resource allocation $R_3 = 61$ is mapped to a rectangle of 3×21, with 2 over allocated slots and a 3×9 left-over space. We then move again to step 3, to look for the largest resource that can fit into the remaining space, but this time there are no resource that can fit into the remaining space, so me move back to step 2, and select the next largest resource to map $R_4 = 35$; mapped to a rectangle of 2×18, results in 1 over allocated slot and a 2×12 left-over space.

Unfortunately, there are no resources that can fit into this left-over space. Furthermore, the algorithm has not yet reaches the maximum frame width; there is still a 1×30 available space.

But at this time, there are no resources that can fit into this available space. $R_5 = 34$, $R_6 = 33$ and $R_7 = 31$ are the only three unmapped resources, and finally the algorithm terminates.

In this particular example, the total of over allocated slots is 2+1+1+1+2+1= 8, and the total of unused slots is 3×3 + 3×9 + 2×12 + 1×30 = 90; the efficiency of the algorithm (percentage of space used) is 72.78%, with over allocated slots and unused slot being counted as waste space. The final results of this example are shown in Table II.

Table II

eOCSA Example Results

Over Allocated Slots	**8**
Unused Slots	**90**
Unmapped Resources	**3**
Efficiency	**72.78%**

The downlink burst mapping results of this example are shown in Fig. 3. Where the dark shaded (black) represent the over allocated slots and the light shaded (gray) represent the unused slots; areas in white are all the mapped resources. This example is one of the worse cases in eOCSA algorithm, because of its low efficiency.

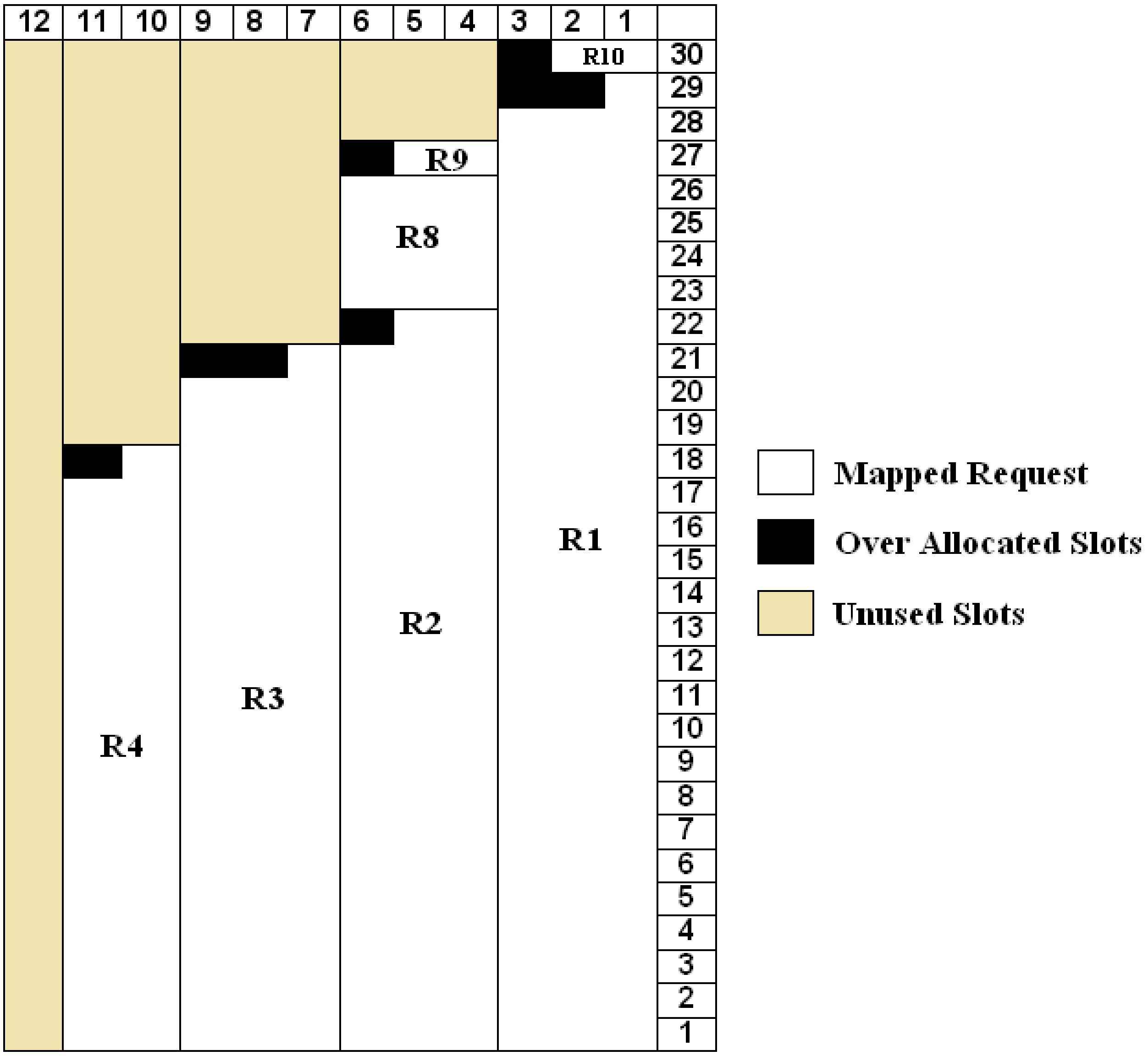


Fig. 3. Resources allocation by "eOCSA algorithm"

Chapter 3

Our Proposed 2L-DMA Algorithm

In this chapter, we describe our proposed data mapping algorithm for two-level requests which we called 2L-DMA algorithm. The proposed 2L-DMA algorithm uses a modified version of eOCSA algorithm. In order to maximize bandwidth utilization, eOCSA sorts the requests (or resources allocation) in the descending order (largest first) and starts by mapping the largest one. The requests are mapped from bottom to top and from right to left to allow the space for the variable portions of the DL-MAP.

Our proposed 2L-DMA algorithm is divided into two phases. **The first phase,** called "eOCSA mapping" consists of mapping all requests according to a modified eOCSA algorithm, which takes into account that all request can consist of two parts (or two levels): MUST part and WISH part. The MUST part is urgent data which will be dropped if it is not mapped and the WISH part represents non-urgent data that can be transmitted in later frames.

Usually there are some remaining requests that were not mapped in the first phase. We call these requests "unmapped requests". Since these unmapped requests can contain MUST part and WISH part, **the second phase** consists of mapping as much MUST part as possible for those unmapped requests. Therefore, we call the second phase "Unmapped MUST part mapping".

There are 5 steps in the "Unmapped MUST part mapping" phase. Some mapped WISH parts in "eOCSA mapping" will have to be sacrificed (partially removed) in order to map as much unmapped MUST parts as possible in the second phase.

To evaluate our proposed 2L-DMA algorithm, in this thesis, we consider various MUST proportions for the requests. We assume that either all requests have the same MUST proportions or different requests have different MUST proportions randomly selected from a range. Our proposed 2L-DMA algorithm is described in more detail as follow:

We assume in this section that the data mapper receivers a set of requests $\{R_i,\ i=1,2,...,N\}$ from the scheduler such that $\sum_{i=1}^{N} R_i = C$, the capacity of a downlink sub-frame. Let $R_i = R_i^m + R_i^w$; where R_i^m and R_i^w are respectively, the sizes of the MUST part and the WISH part of the i^{th} request.

3.1 2L-DMA System Description

- ✓ **First Phase (eOCSA mapping)**

In the first phase, we basically perform the eOCSA algorithm, but with two-level requests. However, for two-level requests, we require that when mapping the requests into rectangles, the MUST part is allocated first, followed by the WISH part, (see Fig. 4). In this figure the blue color represent the MUST part and the white color represent the WISH part. Note that the blue color is allocated first then followed by the white color if any. In case there are no MUST part we just allocate the WISH part and vice versa. The black color in this figure represent the over allocated slots and the gray color represent the unused slots.

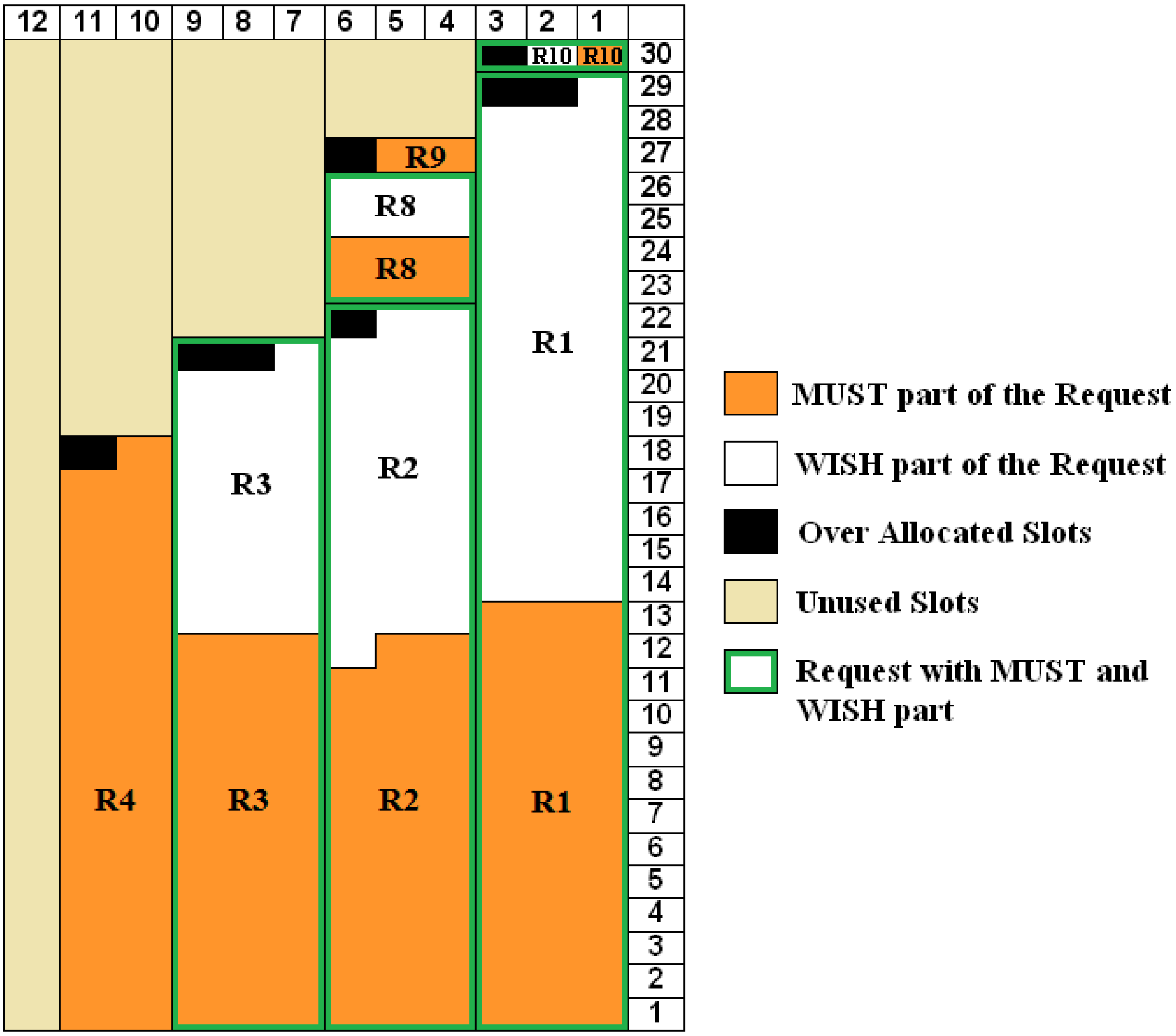


Fig. 4. Resources allocation after "eOCSA mapping"

As mentioned before, there are usually unmapped requests after the first phase is completed. It is likely for these unmapped requests to contain MUST parts and Wish parts. So, in the second phase we try to map as much MUST parts as possible of them unmapped requests.

After the first phase is completed "eOCSA mapping", we know exactly which requests is covering the bin base (frame width), these is known as vertical mapping in the original eOCSA algorithm.

We also know which others requests are mapped on the top of each vertical mapping; these are known as horizontal mapping in the original eOCSA algorithm.

Based on the width of each vertical mapping (or request) that cover the bin base and the complete height of the frame, we defined **Vertical Blocks** (V_i). The total possible amount of V_i are 12, since the downlink sub-frame is 12×30.

In other words, for each **vertical block** we consider the width of each request that is mapped on the bin base after the first phase and the total height of the frame (30 sub-channels); in case they are empty columns, we define V_i base on the width of the empty column and the frame total height.

- ✓ **Second Phase (Unmapped MUST part mapping)**

Before we actually start describing the steps of our proposed 2L-DMA algorithm second phase, we will illustrate a figure of resources allocation by eOCSA algorithm which contain some notations that is needed for our second phase.

The Fig. 5, show the results of our proposed algorithm first phase and some notations that are required for our second phase. These notations are explain and described in detail later on.

Resources Allocation (Requests):

	R1	R2	R3	R4	R8	R9	R10
Resource (slots)	85	65	61	35	12	2	2
MUST part	39	35	36	35	6	2	1
WISH part	46	30	25	0	6	0	1

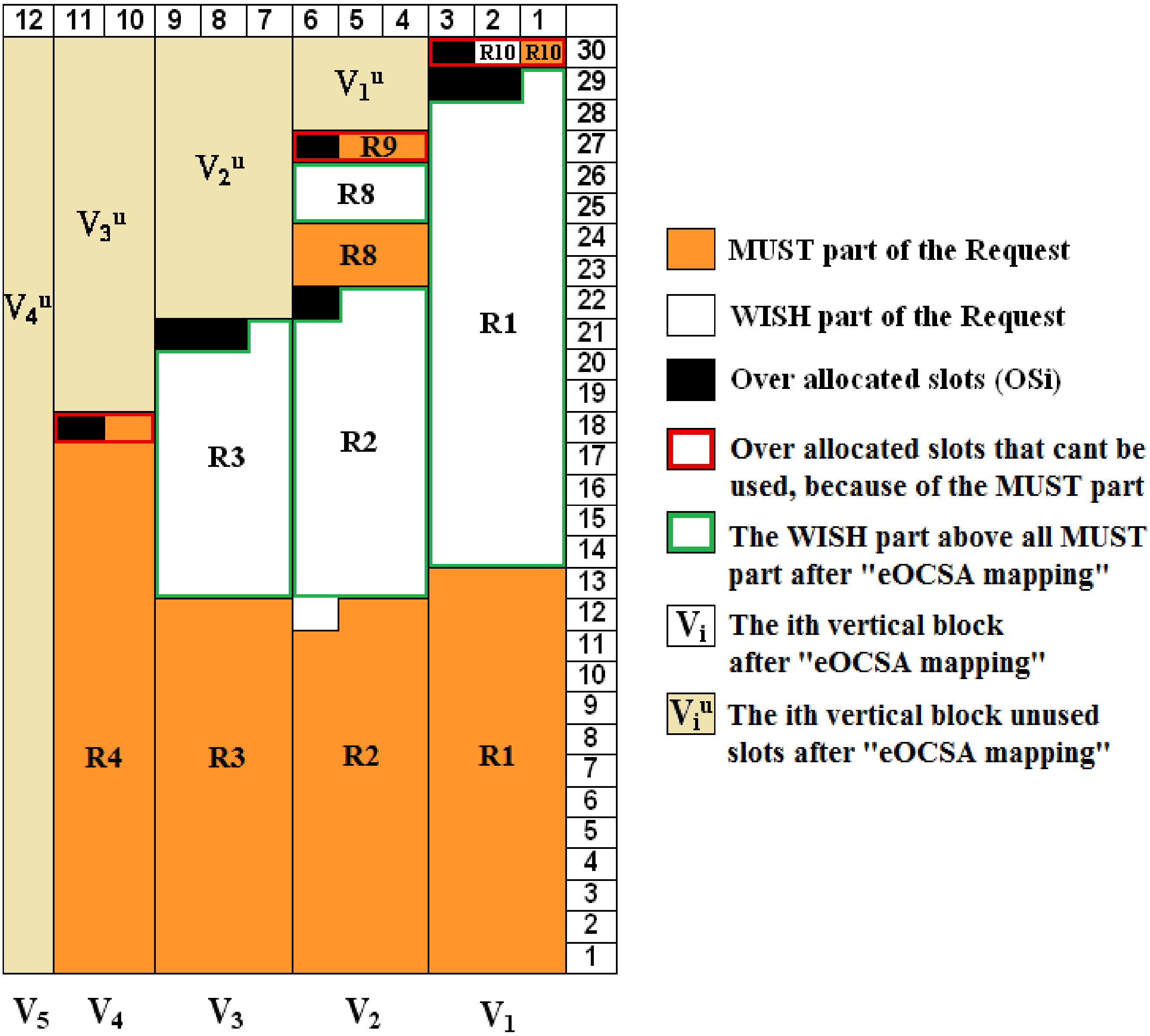


Fig. 5. Resources allocation after "eOCSA mapping"

Notations & Description

R_i : The size of the i^{th} request.

R_i^m : The MUST part of the i^{th} request.

R_i^w : The WISH part of the i^{th} request.

V_i : The i^{th} vertical block after "eOCSA mapping".

V_i^u : The i^{th} vertical block unused slots after "eOCSA mapping".

V_i^w : The i^{th} vertical block width.

$R_i^{w'}$: The WISH part above all MUST part after "eOCSA mapping".

OS_i : Over allocated slots of each mapped requests.

O_i^a : Over allocated slots that can be used; only if $R_i^{w'} \geq (V_i^w - OS_i)$

MW : Over allocated slots of each R_i^m . $MW = \left(\left(\lceil R_i^m / V_i^w \rceil\right)\ V_i^w\right) - R_i^m$

Our proposed 2L-DMA algorithm second phase consist of the following 5 steps:

Step 1: For unmapped requests R_i with $R_i^m > 0$; sort them in the descending order and select the largest R_i^m to map first.

Step 2: Get $\left(V_i^u, O_i^a, R_i^{w'}\right)$ for each V_i; and then sort the set of V_i by the $V_i^u + \sum_{i=1}^{n} O_i^a$ in the descending order. $\{O_i^a,\ i = 1, 2, \ldots, n\}$

Step 3: In this step the V_i width is fixed or known, and it is used to determinate the required height for the R_i^m; such as $H_k = \lceil R_i^m / V_i^w \rceil$.

Since $\lceil \; \rceil$, denotes the ceiling function, this ensures that the mapped region is bigger or equal to the required resource allocation $V_i^w \times H_k \geq R_i^m$.

Select the V_i ; having the largest $V_i^u + \sum_{i=1}^{n} O_i^a$; which satisfies $V_i^u + \sum_{k \in V_i} (R_k^{w'} + O_k^a) > R_i^m$.

If $V_i^w \times H_k > R_i^m$ then fill MW partially with slots from R_i^w if any.

Step 4: Map R_i^m into V_i, first filling V_i^u; second filling O_i^a in the descending order, and then fill with $R_i^{w'}$ if needed.

In case there are still V_i^u in the V_i after mapping R_i^m into the V_i; then fill V_i^u with R_i^w if any; and when finish recalculate $O_k^a, V_i^u, R_k^{w'}$ *for* $k \in V_i$.

Step 5: Update the V_i by accommodating R_i^m according to $V_i^w \times H_k$.

Repeat **step 3, step 4** and **step 5** until all R_i^m are mapped or there are no available spaces in V_i.

Additionally, we present a modified version of our proposed 2L-DMA algorithm, called 2L-DMA mv. The objective of this idea is to basically obtain it results and analyzes it performance when compared to our proposed 2L-DMA algorithm. 2L-DMA mv is similar to 2L-DMA algorithm, the only difference is given in the first step of the “eOCSA mapping”; the remaining of the steps are the same, no modification was made.

In our 2L-DMA "eOCSA mapping", the first step consist of sorting all the requests (or resources allocation) in the descending order of the request size and selecting the largest request to map first.

Therefore, taking into account that all request can consist of two parts (or two levels): MUST part and WISH part. We require that when mapping the requests into rectangles, the MUST part is allocated first, followed by the WISH part.

Furthermore, in 2L-DMA mv "eOCSA mapping", the first step consist of sorting all the requests in the descending order of the MUST part size and selecting the request with the largest MUST part to map first. We also require that when mapping the requests into rectangles, the MUST part is allocated first, followed by the WISH part.

In order to evaluate the performance of the modified version of our proposed 2L-DMA algorithm "2L-DMA mv", we assume that all requests have random MUST proportions from in a range.

If we set all the requests to have the same MUST proportions then it will be performing same as our 2L-DMA algorithm, for this reason we set all requests have random MUST proportions from in a range. The performance of this algorithm will be given later in chapter 4.

Chapter 4

Performance Comparisons

In this thesis, we present the simulation results to show the performance comparisons of our proposed 2L-DMA algorithm with that of eOCSA algorithm. Furthermore, we will also present the performance comparisons of our proposed 2L-DMA algorithm with a modified version of our proposed 2L-DMA algorithm, (2L-DMA mv). In our simulations, we assume that either all requests have the same MUST proportions or all requests have random MUST proportions from in a range. We also assume that the request for each MS is randomly generated; with the constraint that the sum of all requests is 12×30 slots. We consider one burst for each MS.

As mention before, for those unmapped requests after our 2L-DMA algorithm first phase, which can contain MUST part and WISH part; in our 2L-DMA algorithm second phase we consider that there burst can consist of only the MUST part or the MUST part with some WISH slots only; the burst does not consist of the complete request size, due to the objective of the second phase, which is to map as much possible MUST part, for this reason we do not consider the burst having the total amount of the request size.

The number of MSs is randomly chosen from 1 to 40. The over allocated slots, unused slots and efficiency are average over 1000 trials. For our simulation, the physical bandwidth is 10 MHz and the frame duration is 5 ms. The duplexing techniques is TDD, and the permutation mode is PUSC, which is the most commonly used mode. Also as mentioned before, the selected algorithm to compare our 2L-DMA algorithm with is that of eOCSA. We couldn't compare 2L-DMA with other published algorithms for various reasons.

Our proposed 2L-DMA algorithm first phase consist of a modified eOCSA algorithm, which make eOCSA the main comparison algorithm. After the first phase, when mapping the remaining unmapped MUST part we consider vertical blocks; with others algorithm we can't obtain what we denotes vertical blocks.

To compare our proposed 2L-DMA algorithm with that of eOCSA algorithm, we assume that all requests have the same MUST proportions. We start by evaluating the impact of the unused slots overhead, which is define as the fraction of the downlink sub-frame taken by the slots that are left unused in a certain frame.

Figure 6. Unused Slots Overhead for 10 Mobile Stations (MSs)

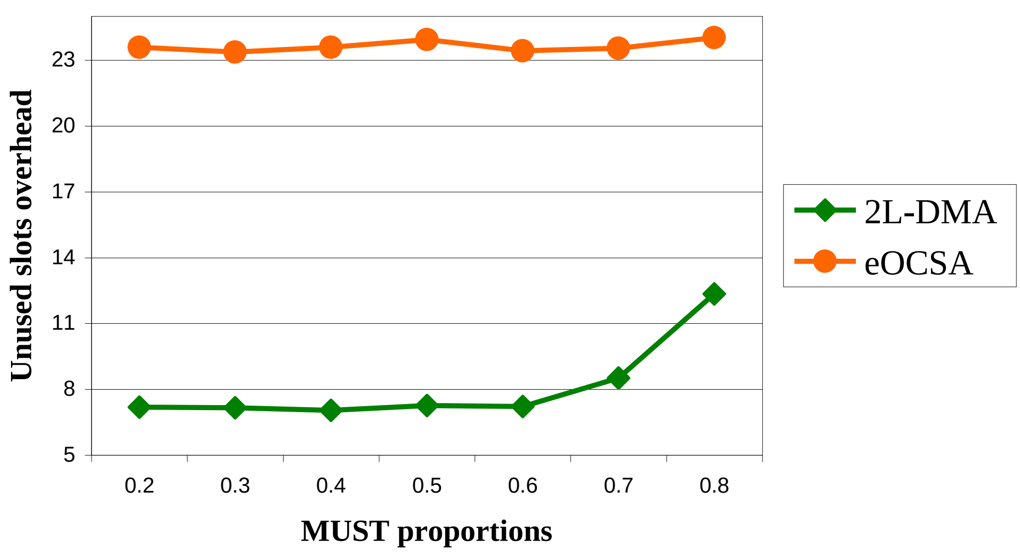


Figure 6, illustrate that under the low traffic load (10 MSs) the eOCSA algorithm generate much more unused slots than our 2L-DMA algorithm. Our algorithm reduces the unused slots, since it start mapping the unmapped MUST part into the vertical block by filling the unused slots. However, after mapping into the vertical block, this can still contain unused slots, so our algorithms fill this space with unmapped WISH part partially in order to minimize the unused slots generated by eOCSA. Furthermore, in figure 6 it is shown that when the MUST proportions is 0.7 our proposed algorithm increase the unused slots, due that there are not much unmapped WISH part to fill into the unused slots; and when the MUST proportions is 0.8 our proposed algorithm increase the unused slots, due that the unmapped MUST part is too large and sometime can’t be mapped. On average, the unused slots of eOCSA are 23.931 and 7.257 with our 2L-DMA algorithm.

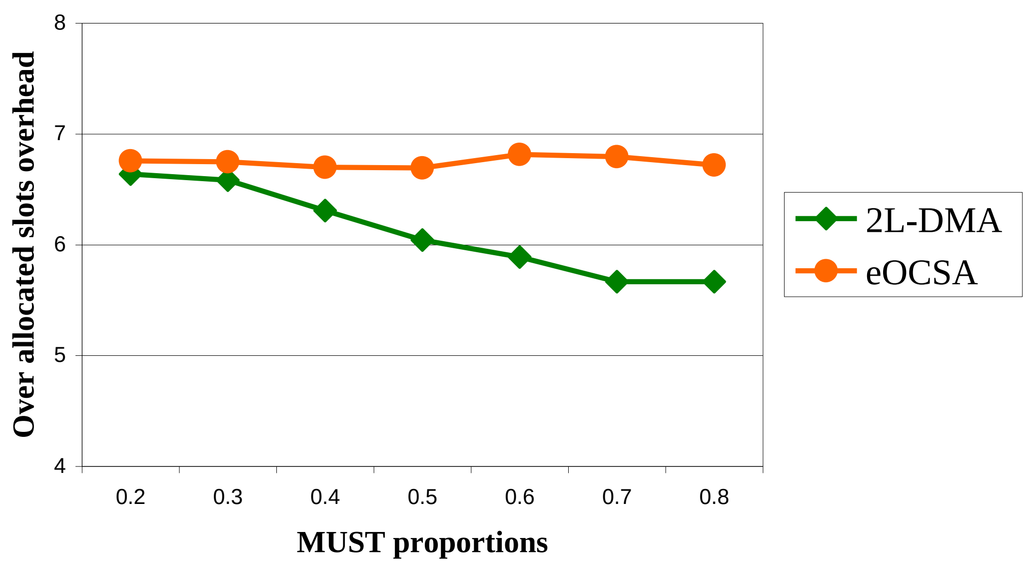
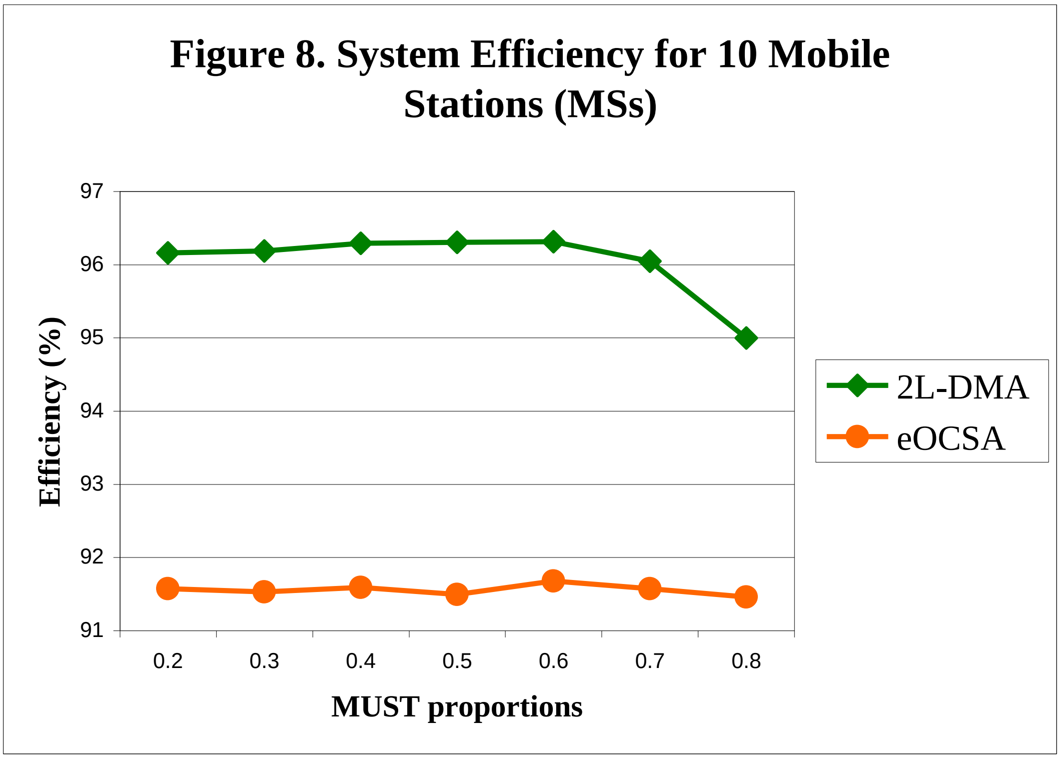


Figure 7. Over Allocated Slots Overhead for 10 Mobile Stations (MSs)

In figure 7, we compare the over allocated slots of eOCSA algorithm with our proposed algorithm. The over allocated slots overhead is define as the fraction of the downlink sub-frame taken by the portions of the burst that are not actually being utilized for sending data. Fig. 7 shows that our proposed 2L-DMA algorithm is generating less over allocated slots when the MUST proportions are increasing, this is because after filling the unused slots our algorithm will fill some over allocated slots while also removing some mapped WISH part in this row so it can also be filled.



In figure 8, the efficiency of our proposed 2L-DMA algorithm compare to eOCSA is shown. As shown, eOCSA is not properly performing mainly because of its problem with the unused slots; and due to the amount of unused slots generated by eOCSA, it leads to a low performance. Fig. 8 illustrates how our proposed 2L-DMA algorithm outperforms eOCSA.

Our proposed algorithm has the best performance because it actually reduces the problem with the unused slots generated by eCOSA. However, we can also observe from the result given in fig. 8 that our proposed algorithm is very steady when the MUST proportions is from 0.2 to 0.6, and starting from 0.7 the efficiency began to decrease, due to the size of the MUST proportions. When the MUST proportions is too large, the unmapped MUST part can't be mapped, since there are not enough space in order to map these.

For example, the sum of the unused slots and the mapped WISH part that can be partially removed is not enough in order to map the unmapped MUST part. Our proposed 2L-DMA algorithm perform better when the MUST proportions is 0.6; at this proportion the efficiency of our proposed algorithm is 96.313% with over allocated slots and unused slots counted as waste.

So far, we have seen in Fig. 6, 7 and 8 the results of the over allocated slots, the unused slots and the efficiency under 10 MSs (or 10 requests). Furthermore, since our proposed algorithm performs better when the MUST proportions is 0.6, we will illustrate it's results of the over allocated slots, the unused slots and the efficiency when increasing the amount of MSs.

Fig. 9, 10 and 11; show the results of eOCSA algorithm again compared to our proposed algorithm when we increase the network traffic and with MUST proportions 0.6; note that as we increase the network traffic our algorithm perform differently. Under heavy traffic load (40 MSs) our proposed algorithm perform almost similar to eOCSA, this is because eOCSA is at its best performance, so there are not much unmapped request to deal with in our proposed algorithm second phase.

Figure 9. Unused Slots Overhead with MUST proportions 0.6

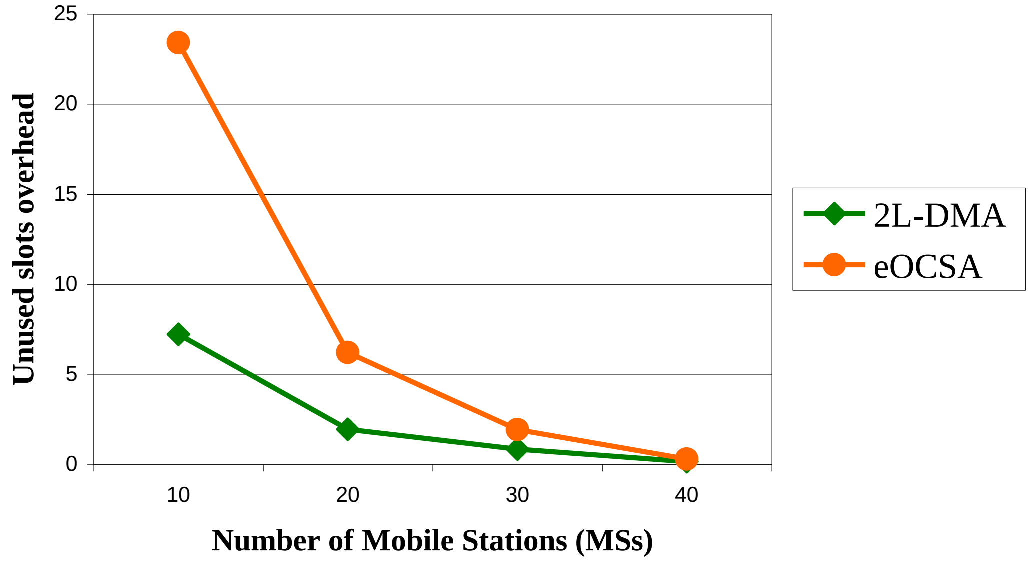
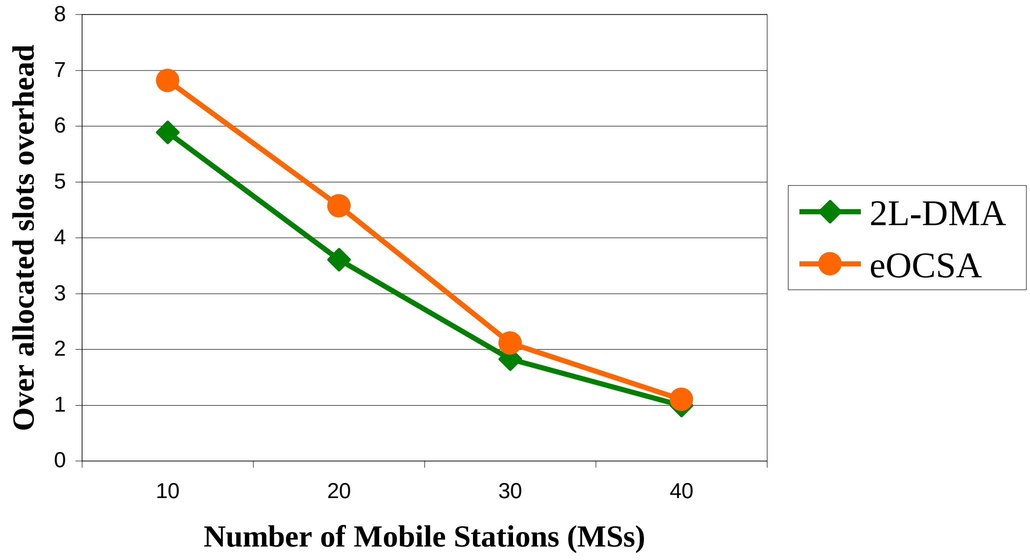
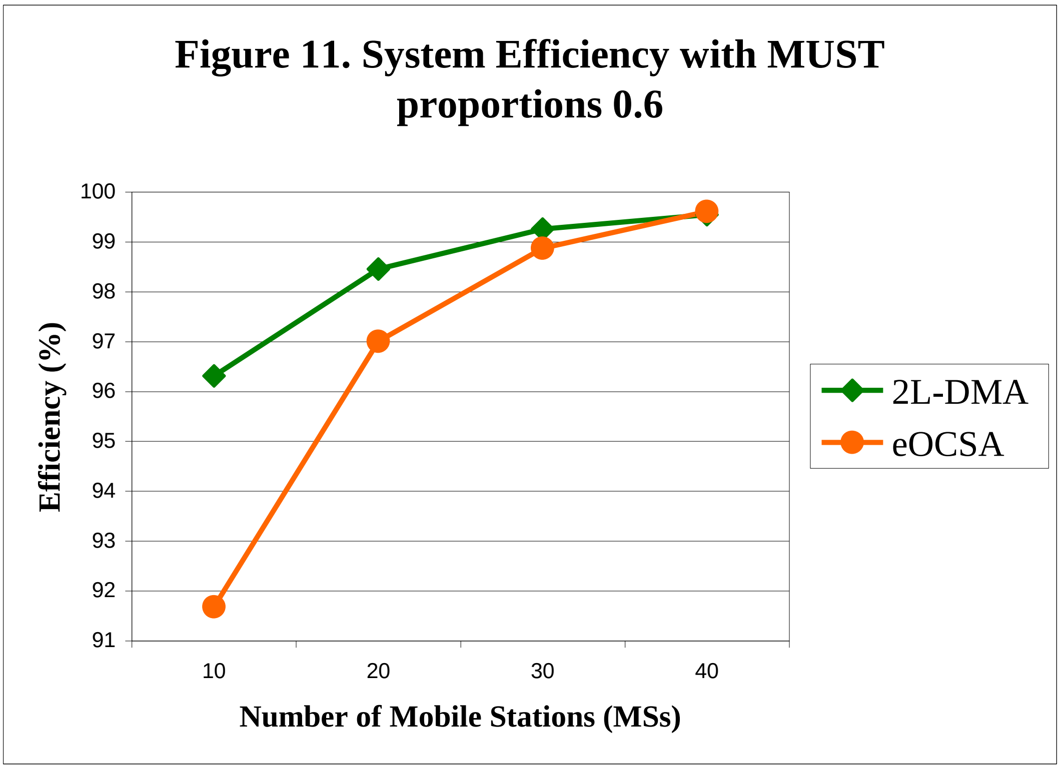


Figure 10. Over Allocated Slots Overhead with MUST proportions 0.6





Finally, in order the compare our proposed 2L-DMA algorithm with the modified version of our proposed 2L-DMA algorithm, (2L-DMA mv). We assume that all requests have random MUST proportions from in a range. As mentioned before, in the modified version of our proposed 2L-DMA algorithm all requests are mapped according to the size of the MUST parts in the descending order (largest MUST part first) in "eOCSA mapping". Fig. 12 shows the results of the unused slots overhead between the two comparative algorithms when the MUST proportions are from 0.2 to 0.8. The fig. 12 illustrates that the modified version of our proposed 2L-DMA algorithm is generating more unused slots overhead when compare to our proposed 2L-DMA algorithm, due that 2L-DMA mv map all request according to the size of the MUST parts, it results in more unused slots when compare to 2L-DMA.

Figure 12. Unused Slots Overhead with MUST proportions from 0.2~0.8

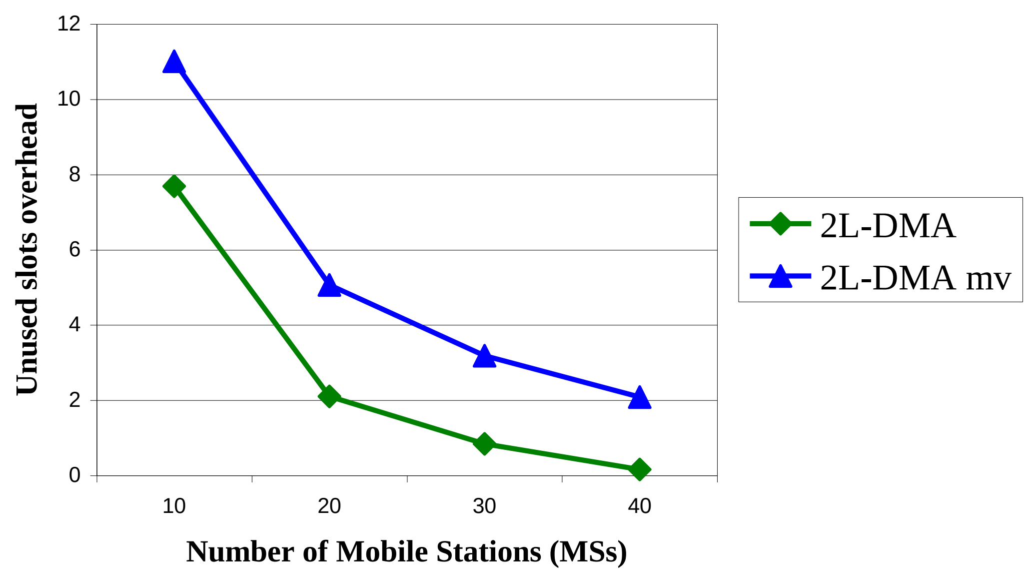
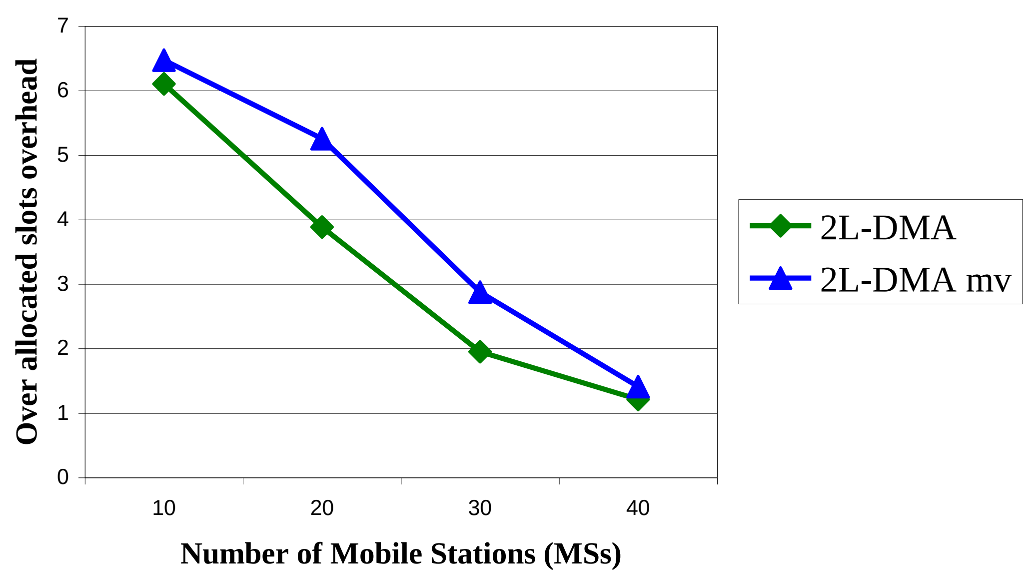


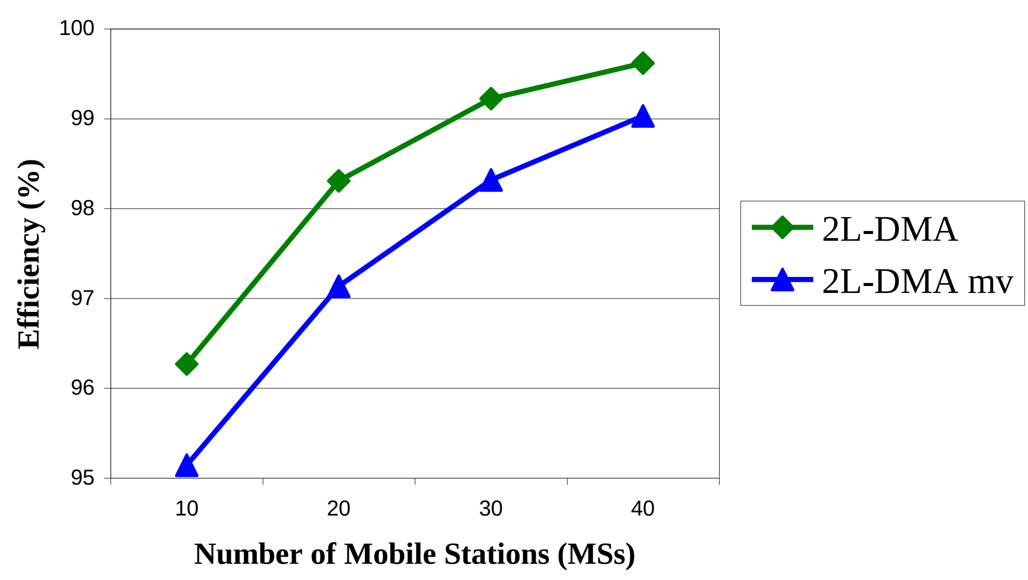
Figure 13. Over Allocated Slots Overhead with MUST proportions from 0.2~0.8



In figure 13, we can observe the over allocated slots overhead generated with MUST proportions from 0.2 to 0.8. This fig. illustrate that 2L-DMA mv generate more over allocated slots overhead compare to our 2L-DMA algorithm, due to the fact that when mapping all the request in the descending order of the MUST part size, the mapping is more disorderly, resulting in more over allocated slots.

Fig. 14 describes the efficiency of the two comparative algorithms. Note that our proposed 2L-DMA algorithm performs better when compare to the modified version of our proposed algorithm, this is because 2L-DMA mv generate much more over allocated slots and unused slots by mapping the requests in the descending order of the MUST part size. On average, the efficiency of 2L-DMA mv are 95.142% and 96.267% with our 2L-DMA algorithm for 10 MSs.

Figure 14. System Efficiency with MUST proportions from 0.2~0.8



Chapter 5

Conclusion

In this thesis, the resource allocation problem which is also known as the bin packing problem in WiMAX system is concerned. First, the resource allocation problem in WiMAX system is briefly introduced. Then we introduce our proposed data mapping algorithm for two-level request called 2L-DMA algorithm. The basic idea of our proposed algorithm is to apply a two-level request; MUST part, for urgent data and WISH part for non-urgent data; and to return as less possible MUST part to the scheduler.

In chapter 3, our proposed 2L-DMA is described. Our proposed 2L-DMA algorithm is divided into two phases. **The first phase,** called "eOCSA mapping", consists of mapping all requests according to a modified eOCSA algorithm which takes into account that all request can consist of two parts (or two levels): MUST part and WISH part. **The second phase,** called "Unmapped MUST part mapping", consists of mapping as much unmapped MUST part as possible for those unmapped requests.

The objective is to improve the resources allocation by utilizing our 2L-DMA algorithm. The simulation results are presented in chapter 4. From the simulation results we conclude that our proposed 2L-DMA algorithm can improve the performance compared to that of eOCSA, due to the fact that it minimizes the unused slots generated in "eOCSA mapping" by first filling this waste space with unmapped

MUST part and then with unmapped WISH partially. On average, the efficiency of eOCSA is 91.443% and 96.267% with our 2L-DMA algorithm for 10 MSs.

In addition, we present a modified version of our proposed 2L-DMA algorithm (2L-DMA mv) which consist of mapping all the requests in the descending order of the MUST part size (largest MUST part first) and starts by mapping the largest one regardless the size of the request in the "eOCSA mapping ".

In conclusion when compared to our 2L-DMA algorithm the simulations results show that our 2L-DMA algorithm outperform 2L-DMA mv, due to the fact that when mapping all requests according to the size of the MUST part in the "eOCSA mapping " it maximize the unused slots. On average, the efficiency of 2L-DMA mv are 95.142% with different MUST proportions and 96.267% for our 2L-DMA algorithm also with different MUST proportions, under 10 MSs for both algorithms.

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.

Autobiography

My name is Arleth Soleiy Garth Campbell, I was born in Nicaragua, in 1987. Am bilingual, I speak both Spanish and English very fluently. I received my B.S. degree in Computer System from the Universidad Cristiana Autónoma de Nicaragua, in 2008. After I graduated from my college I decided to come to Taiwan for further education. I have been living here in Taiwan for almost 3 years now. Upon the first year, I studied Mandarin Chinese language at National Taiwan Normal University; then I enrolled at this prestigious university named National Chiao Tung University, where I received my M.S degree in Telecommunication & Networking from EECS department in the year 2011.