

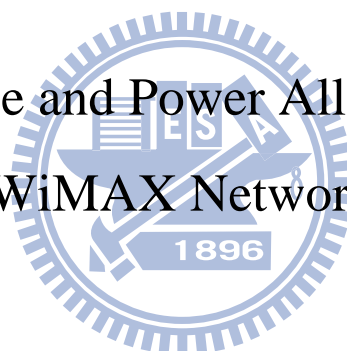
國立交通大學

資訊科學與工程研究所

博士論文

WiMAX 網路之資源與功率配置設計

Resource and Power Allocation in
WiMAX Networks



研究生：梁家銘

指導教授：曾煜棋 博士

林寶樹 博士

中華民國一百年六月

WiMAX 網路之資源與功率配置設計
Resource and Power Allocation in
WiMAX Networks

研究生：梁家銘

Student : Jia-Ming Liang


指導教授：曾煜棋

Advisors : Yu-Chee Tseng

林寶樹

Bao-Shuh Paul Lin

國立交通大學
資訊科學與工程研究所
博士論文



A Dissertation
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Computer Science

June 2011

Hsinchu, Taiwan, Republic of China

中華民國一〇一一年六月

WiMAX 網路之資源與功率配置設計

研究生：梁家銘

指導教授：曾煜棋 博士

林寶樹 博士

國立交通大學資訊科學與工程研究所

摘 要

IEEE 802.16 (WiMAX)是新一代遠距無線網路標準，提供低廉的最後一哩網路存取，支援更高速的傳輸及更多樣的媒體服務。根據網路承載量及覆蓋範圍，WiMAX 提供三種型態的網路架構，分別為：1)點對多點網路架構、2)中繼網路架構及 3)網狀網路架構。由於 WiMAX 網路下的行動用戶端較多、分佈較廣，因此 WiMAX 網路的資源由基地站統一負責管理，這使得 WiMAX 資源配置問題成為該網路上最重要的議題之一。本篇論文主要包含三個研究主題，分別探討 WiMAX 三種網路架構下的資源配置問題。第一個主題針對 WiMAX 點對多點網路，探討如何確保行動用戶端的服務品質問題。第二個主題考量在 WiMAX 中繼網路上，探討如何利用中繼站來協助行動用戶端傳輸，改善用戶端能源耗損的問題。最後，第三個主題針對 WiMAX 網狀網路，探討如何妥善配置資源以減少 multi-hop 傳遞造成的延遲問題。

在第一個研究主題中，我們觀察到 WiMAX 點對多點網路架構上的特殊二維訊框結構，該訊框結構下配置資源時會產生額外的資源浪費，因此容易造成網路效能降低。目前文獻中，主要的方法是利用單一的鏈結層排程單元或單一的實體層 burst 配置單元來減少資源的浪費，但兩者獨立運作下，不但運作複雜度無法降低外，實際可減少的資源浪費卻是有限的。因此，這個研究提出利用鏈結層的排程單元及實體層 burst 配置單元的協同設計，來有效降低運作的複雜度外，更減少訊框配置時造成的資源浪費。我們提出的跨層資源配置方法，包含一個二階權重的排程單元於鏈結層及廂型配置方法於實體層。此排程方法會根據用戶端的頻道品質、佇列資料、要求速率及容許延遲時間決定資料的優先權，再根據廂型配置法則將資料配置於二維訊框中。透過跨層的協同設計，可確保資料流的即時性及速率滿足度外，更可以公平分配行動用戶端的資源以及提升網路吞吐量。

在第二個研究主題中，我們指出在 WiMAX 中繼網路下，大部分研究採取提高行動用戶端的傳輸速率或增大平行傳送的個數以提升用戶端在中繼網路的吞吐量。但是，一旦用戶端使用更高的傳送速率或更多的平行傳送則會過度消耗他們的能源，這種設計對

電池供電的用戶端格外造成傷害。因此，我們進而探討如何配置用戶端的頻寬資源及功率，滿足他們在上鏈傳輸的需求外，同時使他們耗用的能源最少。在這個研究中，我們首先說明這個問題是 NP-complete，進而提出具節能的資源配置方法。首先，此方法會儘可能利用中繼站來協助用戶端傳輸，使更多的資源能有效的被利用。當有剩餘資源時，此方法會利用這些資源來調整用戶端的傳送速率、路徑以降低用戶端的能耗。根據我們的了解，這是第一個在 WiMAX 中繼網路上，考量能源與頻寬配置的研究。

在第三個研究主題中，我們考量在 WiMAX 網狀網路上的微型時槽排程設計。我們認為，一個有效的時槽排程設計需要考慮到實體層的傳遞耗費、排程的運作複雜度及排程結果的訊息發佈耗費。我們特別針對長鏈狀及格狀網狀網路等擁有許多應用的拓撲上作設計。和其他方法相比，我們的排程方法擁有較低的運作複雜度外，更能減少排程訊息發佈的耗費。其主要的貢獻在於利用一個簡易且具規律性的時槽配置規則，透過在傳送耗費及管線效應下取得平衡，使 multi-hop 傳遞產生的延遲得以減少，這個方法易於實作且效能媲美最佳結果。模擬結果說明了我們的方法能大幅的改善排程的運作時間，同時也維持較低的傳遞延遲。

關鍵字：IEEE 802.16、WiMAX、點對多點網路、中繼網路、網狀網路、跨層設計、公平排程、區塊配置、資源管理、功率配置、時槽排程、傳遞延遲。



Resource and Power Allocation in WiMAX Networks

Student: Jia-Ming Liang

Advisors: Dr. Yu-Chee Tseng

Dr. Bao-Shuh Paul Lin

Department of Computer Science
National Chiao Tung University

ABSTRACT

The IEEE 802.16 (WiMAX) is developed to provide broadband wireless access. The standard provides three network architectures to support the last mile wireless access: 1) the *point-to-multipoint (PMP)* architecture, 2) the *relay* architecture, and 3) the *mesh* architecture. However, the resource allocation of these architectures is left open to the implementation. Thus, we propose the resource and power allocation for WiMAX networks, which includes three works. The first work considers the cross-layer resource allocation in WiMAX PMP networks to guarantee the *quality of service (QoS)* requirements of *mobile subscriber stations (MSSs)*. The second work discusses the energy conservation issue in WiMAX relay networks. The third work focuses on the reduction of multi-hop transmission latency in WiMAX mesh networks.

In the first work, we observe that the WiMAX PMP downlink subframes have a special 2D channel-time structure. Allocation resources from such a 2D structure incurs extra control overheads that hurt network performance. Existing solutions try to improve network performance by designing solely either the scheduler in the MAC layer or the burst allocator in the physical layer, but the efficiency of overhead reduction is limited. In this work, we point out the necessity of ‘co-designing’ both the scheduler and the burst allocator to efficiently reduce overheads and improve network performance. We propose a cross-layer framework which contains a *two-tier, priority-based scheduler* in the MAC layer and a *bucket-based burst allocator* in the physical layer. The scheduler assigns priorities to MSSs’ traffics in a two-tier manner and allocates resources to these traffics based on their priorities. The burst allocator divides the free space of each downlink subframe into a special structure which consists of several ‘buckets’ and then arranges bursts in a bucket-by-bucket manner. Both the scheduler and the burst allocator are tightly coupled together and thus can significantly increase network throughput, maintain

long-term fairness, alleviate real-time traffic delays, and improve frame utilization.

In the second work, we point out that under WiMAX relay networks, existing studies only target at improving network throughput by increasing the transmission rates of MSSs or maximizing concurrent transmissions. However, using a higher transmission rate or allowing more concurrent transmissions could harm MSSs in terms of their energy consumption, especially when they are battery-powered. Therefore, we consider the energy-conserved resource allocation problem in the uplink direction of a WiMAX relay network. This problem asks how to arrange the frame usage with satisfying MSSs' demands as the constraint and minimizing their total energy consumption as the objective. We prove this problem to be NP-complete and develop an energy-efficient heuristic. The heuristic first exploits relay stations to allow more concurrent uplink communications to improve the transmission efficiency. When there are remaining resources, the heuristic lowers some MSSs' transmission power by adjusting their transmission rates and paths to save their energy. To the best of our knowledge, this is the first work considering both energy and bandwidth allocation in a WiMAX relay network.

In the third work, we consider the mini-slot scheduling problem in WiMAX mesh networks. An efficient mini-slot scheduling needs to take into account the transmission overhead, the scheduling complexity, and the signaling overhead to broadcast the scheduling results. We are interested in chain and grid-based WiMAX mesh networks, which are the basic topologies of many applications. We propose scheduling schemes that are featured by low complexity and low signaling overhead. Compared to existing works, this work contributes in developing low-cost schemes to find periodical and regular schedules that achieve near-optimal transmission latencies by balancing between transmission overhead and pipeline efficiency that are more practical and easier to implement. In this work, we show that our schemes can significantly improve over existing works in computational complexity while maintain similar or better transmission latencies.

Keywords: IEEE 802.16, WiMAX, PMP network, relay network, mesh network, cross-layer design, fair scheduling, burst allocation, resource management, power allocation, slot scheduling, transmission latency.

Acknowledgements

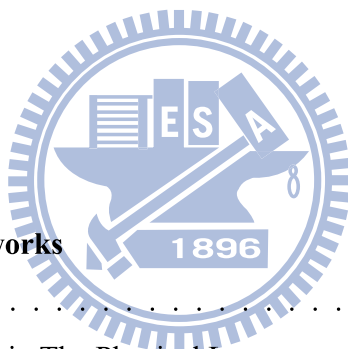
I would like to express my deepest and sincere gratitude to my advisors, Prof. Yu-Chee Tseng and Prof. Bao-Shuh Paul Lin, for their perspicacious advice and guidance throughout my graduate study. Their extensive knowledge and continuous support have been an invaluable help. Special thanks to my dissertation committee members, Prof. Rong-Hong Jan, Prof. Jyh-Cheng Chen, Prof. Jang-Ping Sheu, Prof. Wan-Jiun Liao, Prof. Cho-Li Wang, and Dr. Sheng-Lin Chou for their insightful comments and valuable suggestions. Moreover, I also thank to my co-workers and the colleagues in High-Speed Communication & Computing (HSCC) laboratory for their precious helps and friendship.

Finally, I am grateful to my family, my dear father, mother, sister, and my girlfriend for their unflinching love, firm support, and continuous encouragement in these years.



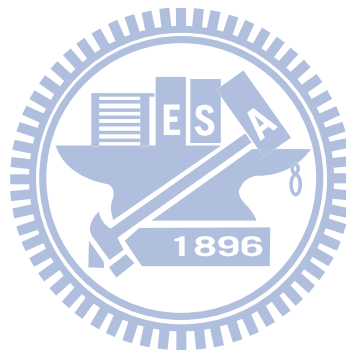
Contents

Abstract in Chinese	I
Abstract in English	III
Acknowledgements	V
Contents	VI
List of Figures	VIII
List of Tables	XI
1 Introduction	1
2 Overview of WiMAX networks	5
2.1 Network Architecture	5
2.2 Accessing Techniques in The Physical Layer	7
2.3 Frame Structures	8
2.4 QoS Service Classes	11
3 A Cross-Layer Resource Allocation in WiMAX PMP Networks	13
3.1 Motivations	13
3.2 Related Work	14
3.3 Problem Definition	16
3.4 The Proposed Cross-Layer Framework	18
3.4.1 Two-Tier, Priority-Based Scheduler	20
3.4.2 Bucket-Based Burst Allocator	23
3.5 Analysis of Network Throughput Loss by The Bucket-Based Scheme	26
3.5.1 Calculation of $E[\tilde{O}]$	27



3.5.2	Calculation of $E[\tilde{S}]$	28
3.6	Performance Evaluation	29
3.6.1	Network Throughput	32
3.6.2	IE Overheads and Subframe Utilization	34
3.6.3	Long-Term Fairness	35
3.6.4	Packet Dropping Ratios of Real-Time Traffics	35
3.6.5	Satisfaction Ratios of Non-Real-Time Traffics	37
3.6.6	Effects of System Parameters	38
3.6.7	Verification of Throughput Analysis	39
4	A Power and Bandwidth Allocation in WiMAX Relay Networks	41
4.1	Motivations	41
4.2	Preliminaries	42
4.2.1	Network Model	42
4.2.2	Energy Cost Model	43
4.2.3	Problem Definition	45
4.3	Two Heuristics to the ERA Problem	48
4.3.1	The Rationale of Our Designs	48
4.3.2	Demand-First Allocation (DFA) Scheme	52
4.3.2.1	Phase 1 — Burst and Path Assignment	52
4.3.2.2	Phase 2 — MCS, Path, and Group Adjustment	54
4.3.2.3	Phase 3 — Burst Allocation and Region Assignment	56
4.3.3	Energy-First Allocation (EFA) Scheme	57
4.3.4	Analysis of Time Complexity	60
4.4	Performance Evaluation	61
4.4.1	Energy Consumption	64
4.4.2	Satisfaction Ratio	65
5	A Regular Mini-Slot Allocation in WiMAX Mesh Networks	68
5.1	Motivations	68
5.2	Problem Definition	69
5.3	Scheduling Tree Construction Schemes	71
5.3.1	A Chain with A Single Request	71

5.3.2	A Chain with Multiple Requests	72
5.3.3	A Chain with Multiple Requests and BS in the Middle	74
5.3.4	A General Grid/Triangle Topology	76
5.4	Performance Evaluation	78
5.4.1	Impact of Network Size	79
5.4.2	Impact of Traffic Load	79
5.4.3	Impact of Transmission Overhead	79
5.4.4	Impact of BS Location	80
5.4.5	Computational Complexity	81
6	Conclusions and Future Directions	85
	Bibliography	87
	Curriculum Vitae	92
	Publication List	93



List of Figures

1.1	The organization of the dissertation.	2
2.1	The three network architectures supported by WiMAX.	6
2.2	Two accessing techniques adopted in the WiMAX physical layer, where the radio resource is distributed among five allocations.	7
2.3	The frame structures under different network architectures.	10
3.1	The structure of an IEEE 802.16 OFDMA downlink subframe under the TDD mode.	16
3.2	The system architecture of the proposed cross-layer framework, where $i = 1..n$	19
3.3	The flowchart of the two-tier, priority-based scheduler.	22
3.4	An example of the bucket-based burst allocation with three buckets and four resource assignments.	23
3.5	The flowchart of the bucket-based burst allocator.	25
3.6	The architecture of our C++ simulator.	30
3.7	A six-state Markov chain to model the channel condition.	32
3.8	Comparison on network throughput.	33
3.9	Comparison on IE overheads.	34
3.10	Comparison on subframe utilization.	35
3.11	Comparison on long-term fairness.	36
3.12	Comparison on real-time packet dropping ratios under different number of MSSs.	36
3.13	Comparison on real-time packet dropping ratios under different admitted non-real-time rates.	37
3.14	Comparison on non-real-time satisfaction ratios of the bottom 10% MSSs under the Markov scenario.	38
3.15	Effect of the number of buckets (B) on network throughput, subframe utilization, and IE overheads under the Markov scenario.	39

3.16	Effect of γ on network throughput and real-time packet dropping ratios under the Markov scenario.	39
3.17	Effect of the number of buckets (B) on the throughput loss \mathcal{L} (by analysis) and the total network throughput (by simulation) under the Markov scenario.	40
4.1	The uplink communication of an 802.16j transparent-relay network.	42
4.2	The structure of the uplink subframe.	43
4.3	The example of a transparent-relay network with one BS, four RSs, and four MSSs.	49
4.4	The effects of rate, concurrent transmission, and receiver distance on energy consumption and resource usage.	50
4.5	The concepts of forward and backward searches by the gradient-like method.	51
4.6	Flowcharts of our proposed heuristics.	52
4.7	The energy consumption of MSSs under different numbers of MSSs, where there are 8 RSs.	65
4.8	The energy consumption of MSSs under different numbers of RSs, where there are 50 MSSs.	65
4.9	The satisfaction ratio of MSSs under different numbers of MSSs, where there are 32 RSs.	66
4.10	The satisfaction ratio of MSSs under different numbers of RSs, where there are 70 MSSs.	67
5.1	(a) A 5-node chain topology, (b) a 4×4 grid topology, and (c) a 4×4 triangle topology.	70
5.2	Transmission schedules for nodes in a chain network (idle state '0' is omitted in the drawing).	71
5.3	The arrangement of transmission-able groups for $H = 2, 3, 4, 5$, and 6 when BS is in the middle.	75
5.4	General collision-free proof for the cases of $H \geq 5$	76
5.5	(a) A fishbone-like tree on the 5×7 grid/triangle topology. (b) The grouping of branch chains when $H = 3$	77
5.6	The impact of network size on total latency in scenarios SN1, SN2, and SN3.	80
5.7	The impact of traffic load on total latency in scenarios SN1, SN2, and SN3.	81

5.8	The impact of transmission overhead (α) on total latency in scenarios SN1, SN2, and SN3.	82
5.9	The impact of locations of BS on total latency in scenarios SN2 and SN3. . . .	83
5.10	The impact of network size on computational complexity in scenarios SN2 and SN3.	84



List of Tables

2.1	The six MCSs supported by WiMAX.	9
3.1	Comparison of prior work and our cross-layer framework	16
3.2	Summary of notations	18
3.3	The amounts of data carried by each slot and the minimum required SNR thresholds of different MCSs	31
3.4	The simulation parameters used in the SUI scenario	31
4.1	MCSs supported by IEEE 802.16j.	44
4.2	Energy costs per bit for different MCSs.	55
4.3	The parameters in our simulator.	61
4.4	The traffic model used in our simulator.	62



Chapter 1

Introduction

The IEEE 802.16 standard (or well known as WiMAX) is an emerging wide-range wireless access technology for solving the last-mile communication problem, bridging the Internet and wireless local-area networks, and supporting broadband multimedia communication services [16, 43]. A series of IEEE 802.16 standards are defined to regulate WiMAX to support high-speed Internet access over long distances. Two types of accessing techniques, namely *orthogonal frequency division multiplexing (OFDM)* and *orthogonal frequency division multiple access (OFDMA)*, are employed in the WiMAX physical layer to realize the convergence of fixed and mobile broadband access through air interfaces. In a WiMAX network, the central *base station (BS)* is responsible for distributing the radio resource among *fixed/mobile subscriber stations (SS/MSSs)*. To manage the resource, the standards define a *scheduler* in the *media access control (MAC)* layer of the BS but leave its detailed implementation as an open issue to provide the flexibility for the hardware manufacturers and network operators.

Depending on the application requirements and the covered areas, WiMAX defines three types of network architectures: 1) The *point-to-multipoint (PMP)* architecture consists of one BS and multiple MSSs, where each MSS can directly communicate with the BS. Such an architecture could be applied in those areas with sparse MSSs such as suburbs. 2) Under the *relay* architecture, several *relay stations (RSs)* are deployed to help relay the data between the BS and MSSs. Each MSS can choose either one-hop or two-hop (via an RS) communication to reach the BS. The relay architecture could be adequate to those areas with dense MSSs such as downtowns. 3) Under the *mesh* architecture, all *subscribe stations (SSs)* are organized in an ad hoc fashion and each SS can reach the BS through a multihop manner. Compared to the above two architectures, the mesh architecture is usually adopted to cover a huge area such as metropolis or large islands. Explicitly, different architectures possess different network characteristics and

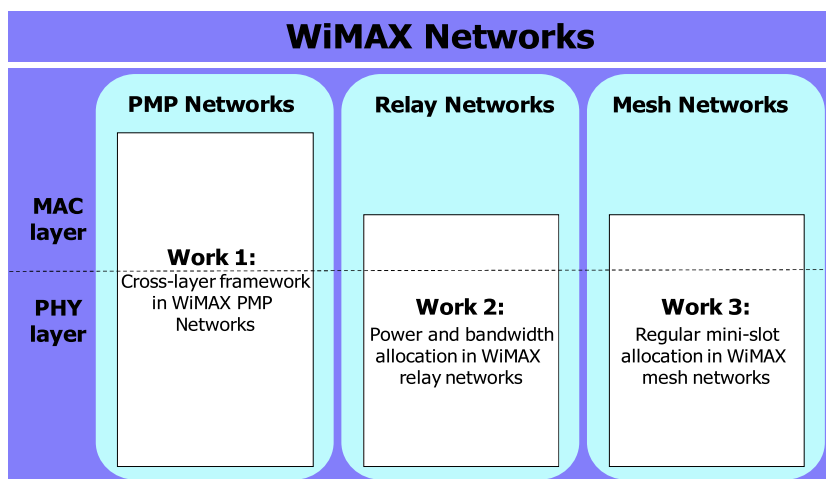


Figure 1.1: The organization of the dissertation.

constraints, which makes the resource allocation problem in WiMAX more challenging and interesting.

In the dissertation, we propose the resource and power allocation in WiMAX networks, which is composed of three works. Fig. 1.1 shows the organization of the dissertation. The first work discusses the problem of traffic scheduling, burst allocation, and overhead reduction in WiMAX PMP networks. The second work discusses the energy consumption issue in WiMAX relay networks, especially for uplink transmissions. The third work discusses the problem of reducing multi-hop transmission latency incurred by WiMAX mesh networks.

In the first work, we propose a cross-layer framework that contains a *two-tier, priority-based scheduler* and a *bucket-based burst allocator* for WiMAX PMP networks. The scheduler assigns priorities to MSSs' traffics in a two-tier manner and allocates resources to these traffics based on their priorities. In the first tier, traffics are differentiated by their types. Then, a γ ratio ($0 < \gamma < 1$) of non-urgent real-time traffics are assigned with level-2 priority and non-real-time traffics are given with level-3 priority. Finally, the scheduler assigns resources to traffics according to the burst arrangement manner to significantly reduce IE overheads. Unlike traditional priority-based solutions that are partial to real-time traffics, our novel two-tier priority scheduling scheme not only prevents urgent real-time traffics from incurring packet dropping (through the first tier) but also maintains long-term fairness (through the second tier). On the other hand, the burst allocator divides the free space of each downlink subframe into a special structure which consists of several 'buckets' and then arranges bursts in a bucket-by-bucket manner. Given k requests to be filled in a subframe, we show that this burst allocation scheme

generates at most k plus a small constant number of overheads. The above bucket-based design incurs very low computation complexity and can be implemented on most low-cost WiMAX chips [2]. Explicitly, in our cross-layer framework, both the scheduler and the burst allocator are tightly coupled together to solve the problems of overhead reduction, real-time and non-real-time traffics scheduling, and burst allocation.

The second work considers the uplink communications in a WiMAX relay network with the transparent RSs. Given the traffic demand of each MSS per frame, we consider an energy and resource allocation problem with satisfying MSSs' demands as the constraint and minimizing their total energy consumption as the objective. Minimizing energy consumption of MSSs is critical since they are usually battery-powered. Existing works have not well addressed the energy consumption issues in WiMAX relay networks. Our "power and bandwidth" optimization problem tries to satisfy MSSs' traffic demands and minimize their energy consumption by selecting proper RSs, rates, and transmission power for them. We show this problem to be NP-complete and propose two energy-efficient heuristics, called *demand-first allocation (DFA)* and *energy-first allocation (EFA)* schemes. These two schemes try to find the suboptimal solutions by exploiting the gradient-like search method. The rationale of DFA is to first find a feasible solution which uses the minimal frame space as the start point. This implies that MSSs will transmit at their maximum power levels. Then, DFA tries to lower their total energy consumption by exploiting the free frame space. On the other hand, EFA first relaxes the frame space constraint to start from a low energy solution where each MSS transmits at a lower rate with no concurrent transmission. However, this may not meet all MSSs' demands. Therefore, EFA tries to increase their rates/power to pack all demands into one frame. Both DFA and EFA have an iterative process to gradually improve their solutions to approximate the optimal one. To the best of our knowledge, this is the first work considering both energy and bandwidth allocation for a WiMAX relay network. Extensive simulation results show that our heuristics can significantly reduce the energy consumption of MSSs while satisfy their traffic demands.

In the third work, we consider the problem of scheduling SSs' traffics on WiMAX mesh networks such that the total latency to transmit all data to the BS is minimized and the scheduling complexity and signaling overhead are as low as possible. We first observe that when the actual transmission size is small, the pipeline could be full for the most of the time, but the transmission overhead could occupy too much time. On the other hand, when the actual transmission size is too large, the above problem may be fixed, but the pipelines may not be filled with suf-

efficient concurrent transmissions, thus hurting spatial reuse. Thus, the proposed approach first finds the optimal transmission size for the given loads to strike a balance between the ratio of transmission overhead and the pipeline efficiency. We then assign the transmissions of each link in a periodic and regular manner with a proper transmission size. Since our scheduling is periodical, the signaling overhead to inform each SS is also quite low. To the best of our knowledge, our work is the first one with these properties. Our scheme incurs low complexity and the result is applicable to most regular topologies, such as chain, grid, and triangle networks, which have been proved to have many applications, such as the mesh networks deployed in rural areas in South Africa to provide Internet access [33], the VoIP testbed [5], and the mobility testbed developed in [34]. These topologies outperform random topologies in terms of their achievable network capacity, connectivity maintenance capability, and coverage-to-node ratios (about two times that of random topologies) [55, 54]. We remark that the chain topology is a special case of grid topologies, which is the most suitable for long-thin areas, such as railways and highways [32]. Simulation results are provided to verify our claims on these topologies.

This dissertation is organized as follows. Chapter 2 overviews the network architectures, frame structures, accessing techniques, and QoS service classes of WiMAX networks. Chapter 3 presents a cross-layer framework for downlink transmissions in WiMAX PMP networks. In Chapter 4, two energy-efficient schedulings are proposed for WiMAX relay networks. Chapter 5 presents a simple and regular mini-slot scheduling for the grid-based WiMAX mesh network. Finally, conclusions and future directions are drawn in Chapter 6.

Chapter 2

Overview of WiMAX networks

Below, we give an overview of WiMAX networks, which covers the topics of network architectures, accessing techniques in the physical layer, frame structures, and QoS service classes.

2.1 Network Architecture

To make the deployed networks be able to meet the application requirements or constraints imposed by the covered areas, WiMAX supports three types of network architectures, which are specified in different versions of IEEE 802.16 standards.

PMP architecture: Specified in the IEEE 802.16d and 802.16e standards [28, 29], PMP is a fundamental network architecture to support the wireless backhaul that enables high-speed Internet access (up to 70 Mbps) over long distances (up to 30 miles). Under the PMP architecture, the central BS can directly communicate with MSSs within its signal coverage, as shown in Fig. 2.1(a). In this case, the network will form a star topology centered at the BS. Those MSSs near the BS can receive stronger signals so that they could enjoy higher communication rates. On the other hand, those MSSs near the coverage boundary (such as MSS_1 and MSS_4) may receive weak signal power from the BS. Thus, they are asked to transmitted/received using lower communication rates so that more radio resource will be wasted. In addition, interfered by obstacles such as high buildings, trees, and mountains, the communication signal between the BS and an MSS would be weakened or even obstructed. This is called a *shadowing effect*. In this case, there could exist some *coverage holes* inside the BS's signal coverage and MSSs could not be able to communicate with the BS when they move into these coverage holes. Fig. 2.1(a) gives an example, where there is a coverage hole caused by the shadowing effect from the tree. MSS_5 may not receive the signal from the BS when it moves into the coverage hole.

Relay architecture: To improve network performance and solve the shadowing problem

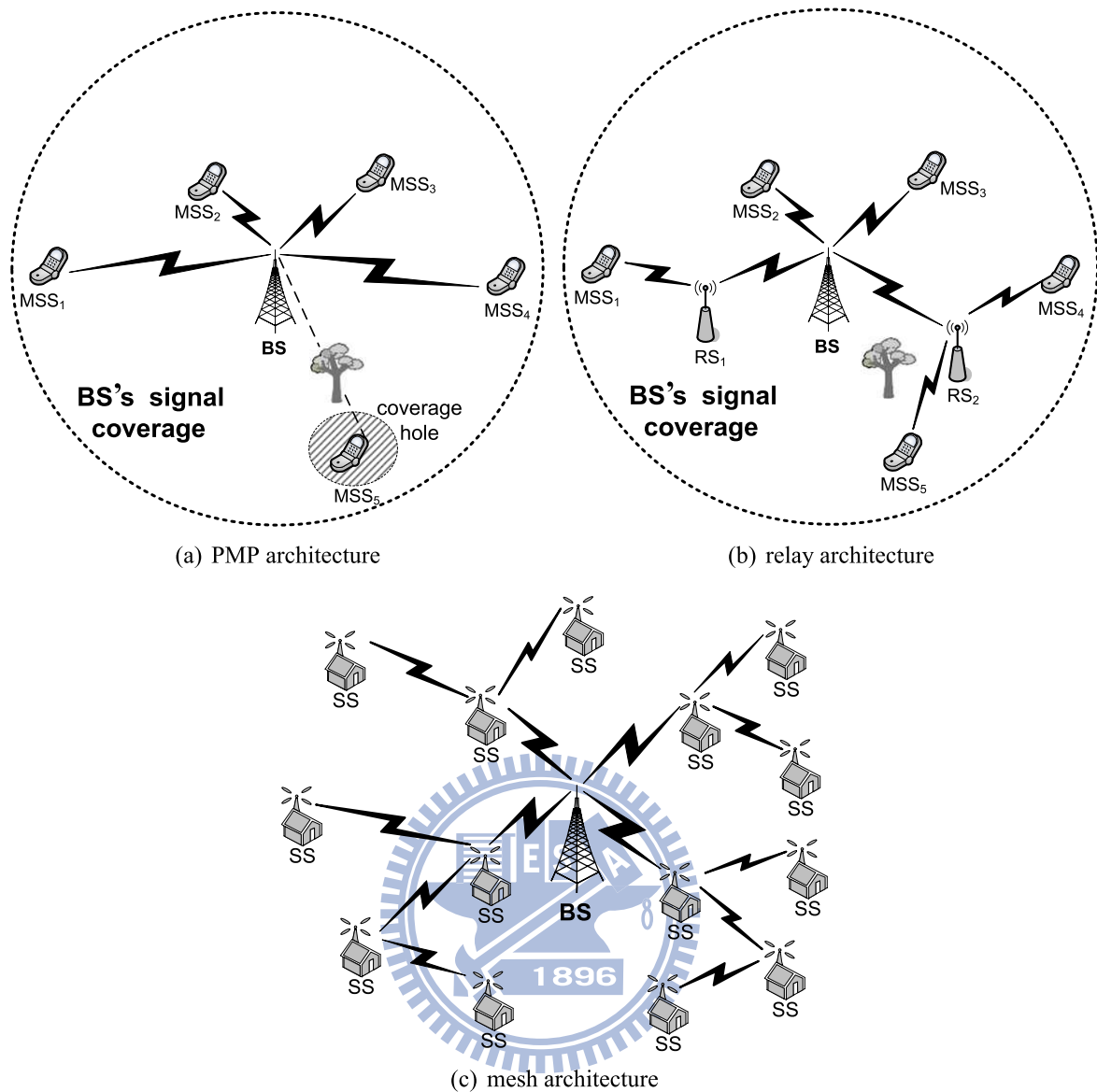


Figure 2.1: The three network architectures supported by WiMAX.

under the PMP architecture, the IEEE 802.16j standard [30] suggests deploying some RSs to help relay data between the BS and MSSs, as shown in Fig. 2.1(b). Each RS can be viewed as an ‘extended’ BS to enhance the received signal power at MSSs (such as MSS₁ and MSS₄) and eliminate the shadowing effect (such as MSS₅). The standard defines two types of RSs. When MSSs are not aware of the existence of RSs, these RSs are called *transparent*. Otherwise, they are *non-transparent*. Transparent RSs are used to increase network performance while non-transparent RSs are used to expand the BS’s signal coverage. Transparent RSs are not responsible for arranging the radio resource to MSSs (such a job is handled by the BS), so they are easier to implement than the non-transparent RSs. Thus, this dissertation aims at relay

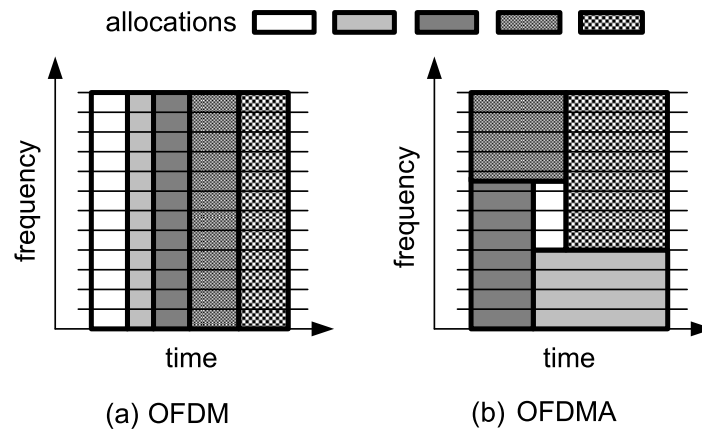


Figure 2.2: Two accessing techniques adopted in the WiMAX physical layer, where the radio resource is distributed among five allocations.

networks with transparent RSs. In a relay network, each MSS can choose to directly communicate with the BS or ask an RS to relay its data in a two-hop manner. However, any two MSSs or any two RSs cannot directly communicate with each other. In this way, the network will form a two-level tree rooted at the BS. Note that with RSs, concurrent RS-MSS communications may be realized due to spatial reuse.

Mesh architecture: Unlike the above two architectures, a mesh network consists of one BS and multiple *static* SSs (for example, these SSs can be set on the top of buildings to provide wireless access of the whole buildings). Specified by the IEEE 802.16d standard, all SSs will be organized in an ad hoc manner to cover a huge area. Two SSs can communicate with each other if they are within each other's transmission range. Each SS can act as either an end point or a router to relay data for its neighbors. Since the BS is responsible for managing the radio resource, all SSs have to send their requests containing traffic demands to the BS. Then, the BS will use the topology information along with SSs' requests to construct a routing tree for SSs to transmit/receive their data, as shown in Fig. 2.1(c). It can be observed that more concurrent communications could coexist since some SSs are deployed far away from each other.

2.2 Accessing Techniques in The Physical Layer

The WiMAX physical layer supports two types of accessing techniques, OFDM and OFDMA, as shown in Fig. 2.2.

OFDM technique: The mesh architecture adopts OFDM as the accessing technique in the physical layer. OFDM supports *non-line of sight (NLOS)* communications and multicarrier

transmissions, where each SS is given the complete control of all subcarriers. The BS adopts the concept of *time division multiple access (TDMA)* to share the radio resource among all SSs. In other words, for multiple SSs that are within each other's transmission range, only one SS is allowed to access the channel at any time. Therefore, the BS only needs to determine which time slot should be allocated to which SS. Fig. 2.2(a) gives an example, where the radio resource is distributed among into five allocations. Each allocation can be viewed as a rectangle whose height covers all available frequency bands. Any two allocations do not overlap in the time domain.

OFDMA technique: The PMP and relay architectures adopt OFDMA as the accessing technique in the physical layer to support the mobility of MSSs. Unlike OFDM, different MSSs are allowed to transmit/receive data through different subcarriers at the same time to enhance the signal power of the MSS. Fig. 2.2(b) gives an example, where the five allocations together constitute the whole radio resource. Since the BS needs to determine which time slot and which subcarrier should be allocated to which MSS, an OFDMA BS will be more complex than an OFDM BS.

Note that a scheduler only determines the sizes of allocations but does not take care of how to arrange these allocations to fit into the two-dimensional time-frequency array (in Fig. 2.2). Such an issue has been addressed in the studies of [6, 50, 67].

2.3 Frame Structures

In WiMAX networks, the radio resource is divided into *frames*. According to different network architectures, various frame structures are also defined:

PMP architecture: Since the PMP architecture adopts the OFDMA accessing technique, the frame will be a two-dimensional array with time units in the time domain and subchannels in the frequency domain, as shown in Fig. 2.3(a). The basic unit of a frame is called a *subchannel-time slot* (or simply *slot*). Each frame is further divided into a *downlink subframe* and an *uplink subframe*. A downlink subframe is composed of the *preamble*, *control*, and *data* portions, while an uplink subframe only has the data portion. The preamble portion is used for time synchronization. The control portion contains the *frame control header (FCH)*, *downlink map (DL_MAP)*, and *uplink map (UL_MAP)* fields. The DL_MAP and UL_MAP fields are used to indicate the downlink and uplink resource allocation in the current frame, respectively. In the data portion, each allocation is a subarray of slots, called a *burst*. From Fig. 2.3(a), each burst

Table 2.1: The six MCSs supported by WiMAX.

level	MCS	data carried by each slot	minimum SINR
1	QPSK 1/2	48 bits	6 dBm
2	QPSK 3/4	72 bits	8.5 dBm
3	16QAM 1/2	96 bits	11.5 dBm
4	16QAM 3/4	144 bits	15 dBm
5	64QAM 2/3	192 bits	19 dBm
6	64QAM 3/4	216 bits	21 dBm

in the downlink subframe is shaped by a rectangle whose width may be multiple subchannels. On the other hand, the bursts in the uplink subframe should be arranged in a row-wise manner, where each burst has a width of only one subchannel. In practice, each MSS can be allocated with more than one burst. However, any two bursts cannot overlap with each other.

Each downlink/uplink burst is with a *modulation and coding scheme (MCS)* and requires one *information element (IE)* recorded in the DL_MAP/UL_MAP field to indicate its size and location in the downlink/uplink subframe. Table 4.1 lists the six MCSs support by WiMAX. Note that each burst can only carry the data of exact one MSS. Therefore, the number of bursts (and thus IEs) will increase when the BS admits more MSSs to access the radio resource. Each IE requires 60 bits encoded by QPSK1/2 (that is, the lowest MCS level). From Table 4.1, each slot can carry data of 48 bits, so an IE will occupy $\frac{5}{4}$ slots. Because IEs and bursts share the same space in the downlink subframe, too many IEs may degrade the network performance.

Relay architecture: Since both the PMP and relay architectures adopt the OFDMA accessing technique, their frame structures will share some common features. For example, the frame is also modeled by a two-dimensional array over both the time and frequency domains. The bursts allocated in the downlink subframe are shaped by rectangles with different widths while the bursts in the uplink subframe are arranged in a row-wise manner. In addition, each downlink/uplink burst spends one IE in the DL_MAP/UL_MAP field to record its corresponding allocation information.

However, because of the existence of RSs, there are two types of frames, namely *BS frames* and *RS frames*. Generally speaking, a BS frame has a ‘complementary’ RS frame, as shown in Fig. 2.3(b). For a BS frame, its downlink subframe has a *BS-MSS/RS region* to allocate downlink bursts for the BS to transmit data to MSSs or RSs; its uplink subframe has an *MSS-BS region* and an *RS-BS region* to allocate uplink bursts for MSSs and RSs to submit their data to the BS, respectively. On the other hand, for an RS frame, its downlink subframe has an *RS-MSS*

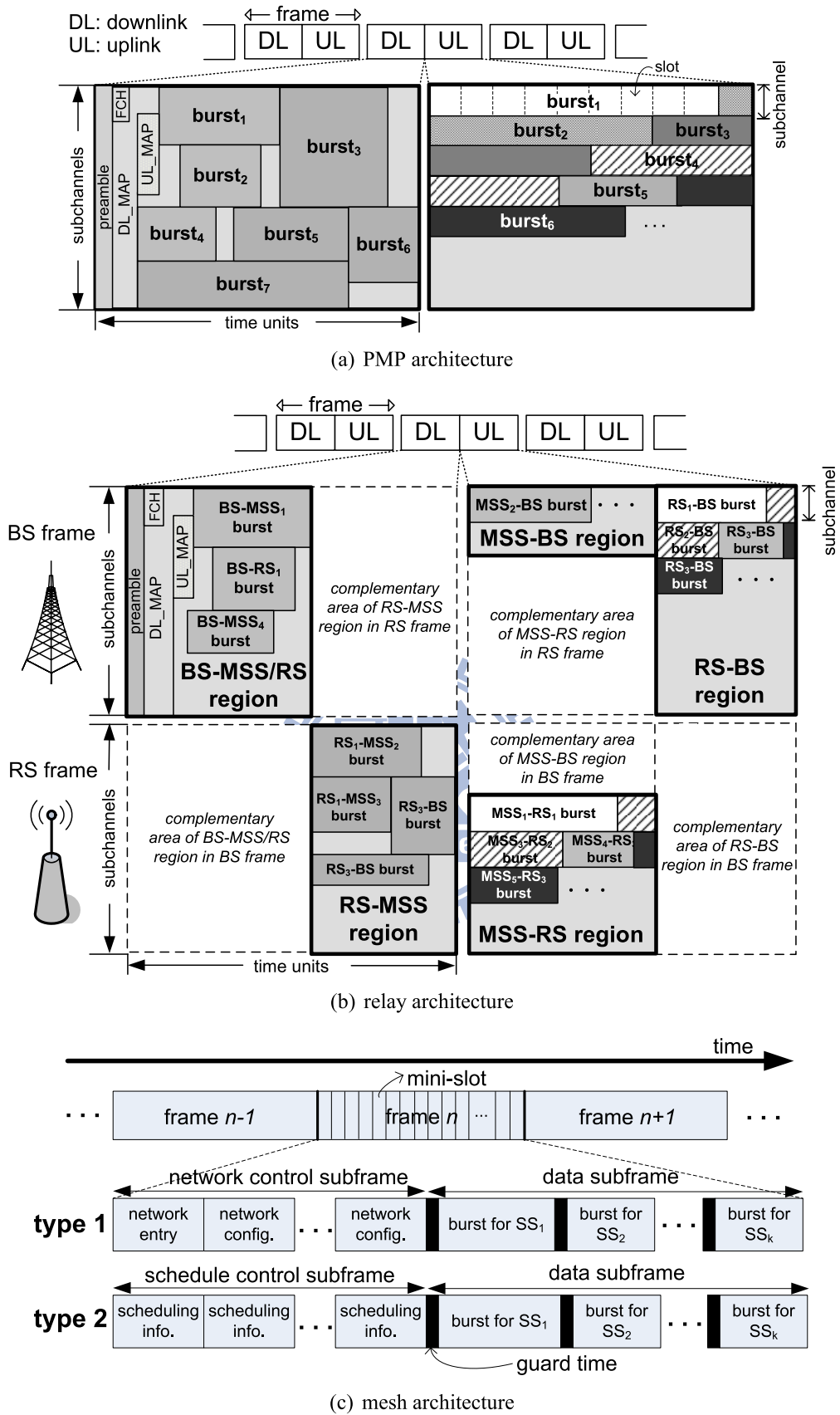


Figure 2.3: The frame structures under different network architectures.

region to allocate downlink bursts for the RS to relay data to MSSs; its uplink subframe has an *MSS-RS region* to allocate uplink bursts for MSSs to submit their data to the BS through the RS. Each RS is considered as ‘bufferless’ in the sense that the data received by the RS from the BS/MSS must be delivered to the MS/BS during the same frame. Taking the uplink subframe in Fig. 2.3(b) as an example, since an MSS_1 - RS_1 burst is allocated in the MSS-RS region, there must be an RS_1 -BS burst allocated in the RS-BS region.

Because the BS is the only receiver, any two bursts in the BS-MSS/RS, MSS-BS, and RS-BS regions cannot overlap with each other. However, by exploiting spatial reuse, concurrent MSS-RS or RS-MSS communications may be allowed. Therefore, some bursts could be overlapped with each other in the RS-MSS and MSS-RS regions to improve network efficiency.

Mesh architecture: Taking OFDM as the accessing technique in the physical layer, the frame under the mesh architecture is modeled by an one-dimensional array over the time domain. The basic unit of each frame is called a *mini-slot*. Two types of frames are defined, as shown in Fig. 2.3(c). A type-1 frame consists of a *network control subframe* and a *data subframe*, where the former carries some network formation information such as how to construct the routing tree while the latter carries the bursts of SSs. The length of the network control subframe is fixed. For each burst, it requires a *guard time* in front of it to conduct time synchronization and avoid propagation delay interfering the following transmission. Such a guard time is usually viewed as transmission overhead because it does not carry the SS’s data. Note that the burst of each SS may mix its downlink and uplink data. On the other hand, a type-2 frame has a fixed-length *schedule control subframe* used to specify the resource allocation in the following data subframe. Each *scheduling information* field contains the burst accessing information such as which mini-slots in the corresponding burst are used for uplink or downlink communication.

Type-1 frames are used for network configurations and type-2 frames are used for normal transmission. It can be observed that the transmission overhead caused by guard times will degrade network performance and thus how to alleviate these overhead is a critical issue.

2.4 QoS Service Classes

To satisfy the different requirements of various data traffics, WiMAX defines five types of QoS service classes:

Unsolicited grant service (UGS): The UGS class provides fixed periodic bandwidth allocation for *constant bit rate (CBR)* traffics such as E1/T1 circuit emulation. Each MSS or SS

only needs to negotiate with the BS about the QoS parameters such as maximum sustained rate, maximum latency, and tolerated jitter at the first time when the connection is established. Then, no further negotiation is required. The UGS class can guarantee the maximum latency for those delay-critical real-time services. However, the radio resource may be wasted if the granted traffics do not fully utilize the allocated bandwidth.

Real-time polling service (rtPS): The rtPS class supports *variable bit rate (VBR)* traffics such as compressed videos. Unlike UGS, the BS has to periodically poll each MSS or SS for its QoS parameters such as maximum sustained rate, maximum latency, tolerated jitter, and minimum reserved rate. The benefit is that the BS can adjust bandwidth allocation according to the real demands of traffics. However, periodical polling may also spend the radio resource.

Extended real-time polling service (ertPS): The ertPS class is specially designed for *voice over IP (VoIP)* with silence suppression, where no traffic is sent during silent periods. Both ertPS and UGS share the same QoS parameters. The BS will allocate the bandwidth with the maximum sustained rate when the VoIP traffic is active and no bandwidth when it becomes silent. In this way, the BS only has to poll MSSs or SSs during the silent period to determine whether their VoIP traffics become active again.

Non-real-time polling service (nrtPS): The nrtPS class considers those non-real-time traffics with minimum reserved rates. The *file transfer protocol (FTP)* is one representative example. The BS will preserve bandwidth according to the minimum reserved rate to avoid starving the non-real-time traffic.

Best effort service (BE): All other traffics belong to this service class. The BS will distribute the remaining bandwidth (after allocating to the traffics of all other four service classes) to the traffics of the BE class, so there is no guarantee of throughput or delay.

Chapter 3

A Cross-Layer Resource Allocation in WiMAX PMP Networks

3.1 Motivations

The WiMAX PMP network [29] is developed for wide-range broadband wireless access. The BS manages network resources for MSSs' data traffics, which are classified into *real-time traffics* (e.g., unsolicited grant service (UGS), real-time polling service (rtPS), and extended rtPS (ertPS)) and *non-real-time traffics* (e.g., non-real-time polling service (nrtPS) and best effort (BE)). Due to the subchannel-time frame structure, each allocated burst is needs to be specified by a *IE* in the DL-MAP field. Since these IEs occupy frame space and do not carry MSSs' data, they are considered as *control overheads*. Explicitly, how to efficiently reduce IE overheads will significantly affect network performance since it determines frame utilization.

To manage resources to all data traffics, the standard defines a *scheduler* in the MAC layer and a *burst allocator* in the physical layer. However, their designs are left as open issues to implementers. Thus, the *co-designing* both the scheduler and the burst allocator is needed to improve network performance, which covers overhead reduction, real-time and non-real-time traffic scheduling, and burs allocation. However, it is not easy to co-design both the scheduler and the burst allocator. In the following, we list the design issues of the scheduler that should be took into consideration:

- The scheduler should improve network throughput while maintain long-term fairness. Since the BS may send data to MSSs using different transmission rates (due to network situations), the scheduler will prefer those MSSs using higher transmission rates but should avoid starving those MSSs using lower transmission rates.
- The scheduler should satisfy the delay constraints of real-time traffics to avoid high packet

dropping ratios. However, it should also meet the requirements of non-real-time traffics.

- To well utilize the limited frame space, the scheduler has to reduce IE overheads when assigning resources to MSSs' data traffics. This requires the knowledge of available frame space and burst arrangement design from the burst allocator.

On the other hand, the design of the burst allocator should address the three issues:

- The burst allocator should arrange IEs and downlink bursts for the MSSs' resource requests from the scheduler in the OFDMA channel-time structure to well utilize the frame space and reduce the control overhead. Under the PUSC model, since all subchannels are equally adequate for all MSSs, the problem of arranging IEs and downlink bursts will become a 2D mapping problem, which is NP-complete [7]. To simplify the burst arrangement problem, an advance planning for the MSSs' resource requests in the scheduler is needed. This requires a co-designing for the scheduler and the burst allocator.
- To satisfy traffic requirements such as real-time delay constraints, the burst allocator has to arrange bursts based on the traffic scheduling knowledge from the scheduler. For example, those bursts for urgent real-time traffics should be allocated first to avoid packet dropping.
- Simplicity is a critical concern because a frame is typically 5 ms [36], which means that the burst allocation scheme needs to be executed every 5 ms.

3.2 Related Work

Most of prior studies on resource allocation in 802.16 OFDMA networks solely implement either the scheduler or the burst allocator. For the implementation of the scheduler, the study of [46] proposes a scheduling scheme according to MSSs' signal-to-noise ratios to achieve rate maximization. The work of [59] proposes a utility function to evaluate the tradeoff between network throughput and long-term fairness. In the work of [53], an opportunistic scheduler is proposed by adopting the instantaneous channel quality of each MSS to maintain fairness. However, these studies do not consider the delay requirements of real-time traffics. The work of [4] tries to minimize the blocking probability of MSSs' traffic requests and thus the packet dropping ratios of real-time traffics may be reduced. Nevertheless, all of the above studies [46, 59, 53, 4] do not address the issue of overhead reduction. The work of [38] tries to reduce

IE overhead from the perspective of the scheduler, where the number of MSSs to be served in each subframe is reduced to avoid generating too many IEs. However, without the help of the burst allocator, not only the efficiency of overhead reduction becomes insignificant but also some important data (*e.g.*, urgent real-time traffics) may not be allocated with bursts because of out of frame space. In this case, some MSSs may encounter serious packet dropping.

On the other hand, several studies consider implementing the burst allocator. The work of [63] proposes a new control message for periodic resource assignment to reduce duplicate signaling. Reference [37] suggests piggybacking IEs on data packets to increase the utilization of downlink subframes. However, both studies [63, 37] involve in modifying the standard. The work of [7] proposes two heuristics for burst allocation: The first heuristic scans the free space in a downlink subframe row by row to try to fully utilize the space, but it may generate a large number of IEs. The second heuristic pre-segments a subframe into several rectangles, and a request will choose a rectangle larger than it for allocation; however, this scheme requires prior knowledge of the request distribution. The work of [68] allocates bursts for large requests first. Nevertheless, larger requests may not be necessarily more important or urgent. Several studies consider allocating bursts in a column-by-column manner. In the work of [51], bursts with the same modulation and coding scheme are combined into a large one. However, this scheme is not compliant to the standard because a burst may contain requests from multiple MSSs. The study of [52] pads zero bits in each column's end, which may cause low subframe utilization. The work of [17] adopts a backward, column-wise allocation scheme, where the bursts are allocated from the right-down side to the left-up side of the subframe. However, this scheme requires $3n$ bursts for n MSSs in the worst case. As can be seen, existing research efforts may pad too many useless bits, generate too many IEs, or leave unused slot holes.

Few studies implement both the scheduler and the burst allocator, but they do not consider reducing IE overhead. The studies of [39, 20] try to arrange resources to MSSs to maximize their data rates and maintain fairness. However, they do not consider the delay requirements of real-time traffics. The studies of [76, 75] develop an one-tier priority-based scheduler to allocate resources to each MSS to exactly satisfy its demand. Thus, the delay requirement of real-time traffics could be guaranteed but network throughput may be degraded. Nevertheless, all of the studies [39, 20, 76, 75] neglect the issue of overhead reduction, which may lead to low subframe utilization and low network throughput. We will show by simulations in Section 5.4 that, without reducing IE overhead, the QoS (quality of service) requirements of MSSs' traffics

Table 3.1: Comparison of prior work and our cross-layer framework

features	network throughput	long-term fairness	rate satisfaction [†]	real-time traffic	subframe utilization	burst allocation complexity
references [46, 59, 53]	✓	✓	✓			N/A
references [4]	✓	✓	✓	✓		N/A
reference [38]	✓	✓	✓		✓	N/A
references [7, 51, 17]					✓	$O(n), O(n), O(n^2)$
references [68, 52]				✓	✓	$O(n)$
reference [39, 20]	✓	✓	✓			$O(n^2), O(n)$
references [76, 75]	✓		✓	✓		$O(n)$
our framework	✓	✓	✓	✓	✓	$O(n)$

[†] n is the number of MSSs.

[‡] The rate satisfaction is to evaluate the degree of starvation of non-real-time traffics.

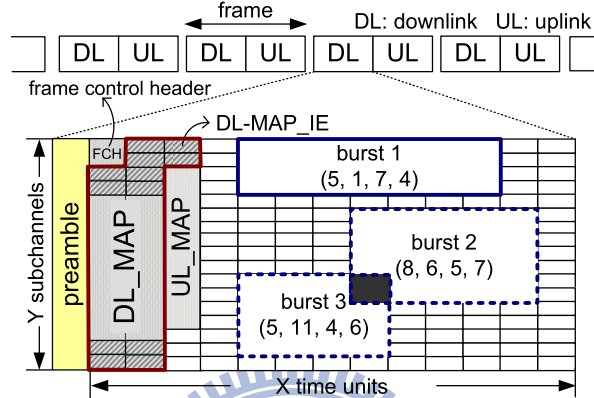


Figure 3.1: The structure of an IEEE 802.16 OFDMA downlink subframe under the TDD mode.

may not be satisfied, especially when the network becomes saturated.

Table 3.1 compares the features of prior studies and our cross-layer framework. It can be shown that our cross-layer framework covers all of the features. In addition, our cross-layer framework has the least computation complexity in burst allocation. Thus, it can be implemented on most of WiMAX low-cost chips.

3.3 Problem Definition

We consider the downlink communication in an 802.16 OFDMA network using the TDD mode. The mandatory PUSC model is adopted so that there is no issue of subchannel diversity (because MSSs will report only their average channel qualities to the BS. The BS supports multiple MSSs in a point-to-multipoint manner, where each MSS has its admitted real-time and non-real-time traffic rates. The BS has to arrange the radio resource to the MSSs according to their traffic demands.

The radio resource is divided into frames (referring to Fig. 3.1). A downlink subframe is composed of X time units (in the time domain) and Y subchannels (in the frequency domain).

The downlink allocation unit is a burst in the $X \times Y$ array. Each burst is denoted by (x, y, w, h) , where x is the starting time unit, y is the starting subchannel, w is the burst's width, and h is the burst's height. An MSS can own more than one burst in a subframe. However, no two bursts can overlap with each other. Fig. 3.1 gives some examples. Bursts 1 and 2 can coexist, but bursts 2 and 3 cannot coexist. Each burst requires one IE in DL-MAP to describe its size and location in the subframe. Note that, from the scheduler's perspective, the number of bursts (and thus IEs) will increase when more MSSs are scheduled. On the other hand, from the burst allocator's perspective, more IEs are required when an MSS's data are distributed over multiple bursts. An IE requires 60 bits encoded by the QPSK1/2 modulation and coding scheme [36]. Since each slot can carry 48 bits by QPSK1/2, an IE occupies $\frac{5}{4}$ slots, which has significant impact on the available space to allocate bursts in a downlink subframe.

The resource allocation problem is formulated as follows: There are n MSSs in the network, where each MSS M_i , $i = 1..n$, is admitted with an average real-time data rate of R_i^{rt} (in bits/frame) and a minimal non-real-time data rate of R_i^{nrt} (in bits/frame). Let C_i be the current transmission rate¹ (in bits/slot) for the BS to send data to M_i , which may change over frames. The objective is to design a cross-layer framework containing both the scheduler and the burst allocator to arrange bursts to MSSs, such that we can reduce IE overhead, improve network throughput, achieve long-term fairness, alleviate real-time traffic delays, and maximally utilize downlink subframes. In addition, the design of the cross-layer framework should not be too complicated so that it can execute within a frame duration (*i.e.*, 5 ms) and implemented in most low-cost WiMAX chips. Note that the *fairness index (FI)* in [14] is adopted to evaluate the long-term fairness of a scheme as follows:

$$FI = \frac{(\sum_{i=1}^n SD_i)^2}{n \sum_{i=1}^n (SD_i)^2},$$

where SD_i is the *share degree* of M_i which is calculated by

$$SD_i = \frac{\sum_{j=0}^{T-1} (\tilde{A}_i^{rt}(f_c - j) + \tilde{A}_i^{nrt}(f_c - j))}{T \times (R_i^{rt} + R_i^{nrt})}, \quad (3.1)$$

where $\tilde{A}_i^{rt}(x)$ and $\tilde{A}_i^{nrt}(x)$ are the amounts of real-time and non-real-time traffics allocated to M_i in the x th frame, respectively, f_c is the current frame index, and T is the window size (in frames) over which we measure fairness. We denote $U_d(x)$ the *utilization* of the x th downlink

¹The estimation of the transmission rate highly depends on the path loss, fading, and propagation model. Here, we assume that the BS can accurately estimate the transmission rate for each MSS and will discuss how to conduct the estimation in Section 5.4.

Table 3.2: Summary of notations

notation	definition
n	the number of admitted MSSs in the network
X	the number of units in time domain of a downlink subframe
Y	the number of subchannels in frequency domain of a downlink subframe
FS	the free space (in slots) in a downlink subframe
T	the window size (in frames)
Δ_{bkt}	the bucket size (in slots)
C_i	the current transmission rate (in bits/slot) for the BS to send data to MSS M_i
C_i^{avg}	the average transmission rate (in bits/slot) for the BS to send data to M_i in recent T frames
R_i^{rt}/R_i^{nrt}	the admitted data rate (in bits/frame) of M_i 's real-time/non-real-time traffics
B_i^{rt}/B_i^{nrt}	the amount of real-time/non-real-time queued data (in bits) of M_i
Q_i^{rt}/Q_i^{nrt}	M_i 's real-time/non-real-time resource assignments (in bits) generated by the scheduler
A_i^{rt}/A_i^{nrt}	the amount of real-time/non-real-time data (in bits) allocated to M_i by the burst allocator
I_i^{rt}/I_i^{nrt}	M_i 's importance factors to allocate real-time/non-real-time traffics
S_i^{nrt}	the non-real-time rate satisfaction ratio of M_i in recent T frames
θ_{IE}	the size of an IE (in slots)
B	the number of buckets in a downlink subframe

subframe, which is defined by the ratio of the number of slots used to transmit data to $X \times Y$. Thus, the average downlink utilization over T frames is $\frac{\sum_{j=0}^{T-1} U_d(f_c-j)}{T}$. Table 3.2 summarizes the notations used in this chapter.

3.4 The Proposed Cross-Layer Framework

Fig. 3.2 shows the system architecture of our cross-layer framework, which is composed of two components: the *two-tier, priority-based scheduler* and the *bucket-based burst allocator*. The transmission rate C_i for each MSS M_i (label 1 in Fig. 3.2) is periodically reported to the scheduler and the burst allocator. Each M_i 's admitted rates R_i^{rt} and R_i^{nrt} (label 2) are sent to the scheduler when M_i first associates with the BS or when R_i^{rt} and R_i^{nrt} change. The scheduler also monitors the current amounts of queued real-time and non-real-time data B_i^{rt} and B_i^{nrt} (label 3). The burst allocator informs the scheduler of the bucket size Δ_{bkt} and the available *free space* FS in the current downlink subframe (label 4) to help the scheduler distribute resources among MSSs' traffics, where

$$FS = X \times Y - (\text{FCH size}) - (\text{UL-MAP size}) - (\text{size of DL-MAP control fields}), \quad (3.2)$$

where FCH is the frame control header. The UL-MAP size can be known in advance since the uplink subframe is allocated before the downlink subframe. The DL-MAP control fields contain all parts of DL-MAP except IEs, which are yet to be decided by the burst allocator. The

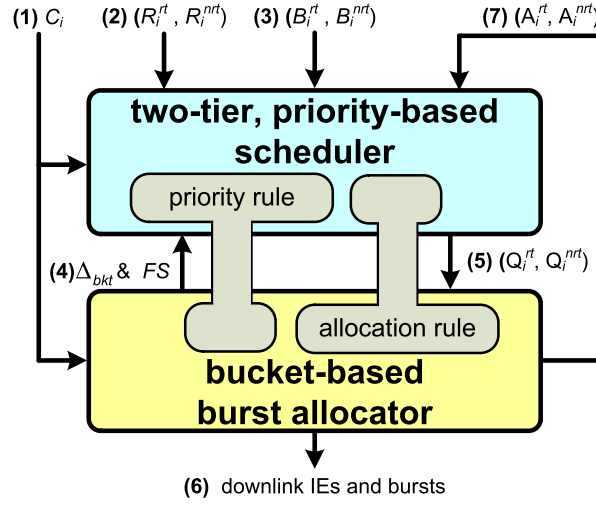


Figure 3.2: The system architecture of the proposed cross-layer framework, where $i = 1..n$.

scheduler's mission is to generate each M_i 's real-time and non-real-time resource assignments Q_i^{rt} and Q_i^{nrt} (label 5) to the burst allocator. Based on Q_i^{rt} and Q_i^{nrt} , the burst allocator arranges IEs and bursts to each M_i (label 6). The actual real-time and non-real-time traffics allocated to M_i are written as A_i^{rt} and A_i^{nrt} (label 7) and are fed to the scheduler for future scheduling.

In our cross-layer framework, the *priority rule* defined in the scheduler helps the burst allocator to determine how to arrange bursts for MSSs' traffics. On the other hand, the *allocation rule* defined in the burst allocator also helps the scheduler to determine how to assign resources to MSSs' traffics. Both the priority and allocation rules are like tenons in the cross-layer framework, which make the scheduler and the burst allocator tightly cooperate with each other.

Due to the NP-complete nature of the burst allocation problem and the hardware constraints of low-cost WiMAX chips, it is inefficient and yet infeasible to derive an optimal solution to arrange IEs and bursts in a short frame duration. Therefore, to keep our burst allocator simple and efficient, we adopt a *bucket* concept as follows: The available free space FS in the current subframe is sliced horizontally into a number of buckets, each of size Δ_{bkt} (see Fig. 3.4 for an example). The size Δ_{bkt} actually serves as the allocation unit in our scheme. As to be seen, the scheduler always keeps $(Q_i^{rt} + Q_i^{nrt})$ as a multiple of Δ_{bkt} for each M_i . In this way, the burst allocator can easily arrange bursts in a 'bucket-by-bucket' manner, well utilize frame resource, and generate quite few bursts and thus IEs (which will be proved having an upper bound later in Section 3.4.2). In addition, the long-term fairness is achieved because the actual allocation (A_i^{rt}, A_i^{nrt}) by the burst allocator is likely to be quite close to the assignment (Q_i^{rt}, Q_i^{nrt}) by the scheduler, for each $i = 1..n$.

3.4.1 Two-Tier, Priority-Based Scheduler

In each frame, the scheduler will generate resource assignments (Q_i^{rt}, Q_i^{nrt}) , $i = 1..n$, to the burst allocator. To generate these assignments, the scheduler adopts a two-tier priority rule. In the first tier, traffics are differentiated by their types and given priority levels according to the following order:

- P1.** Urgent real-time traffics whose packets will pass their deadlines at the end of this frame.
- P2.** Real-time traffics ranked top γ ratio ($0 < \gamma < 1$) sorted by their importance.
- P3.** Non-real-time traffics sorted by their importance.

Then, in the second tier, traffics with the same type are assigned with different priorities by their *importance*, which is calculated by their 1) current transmission rates, 2) average transmission rates, 3) admitted data rates, and 4) queue lengths. In particular, for priority level **P2**, we rank the *importance* of M_i 's real-time traffic by

$$I_i^{rt} = C_i \times \frac{C_i}{C_i^{avg}} \times \frac{B_i^{rt}}{R_i^{rt}}. \quad (3.3)$$

Here, the importance I_i^{rt} involves three factors multiplied together:

1. A higher transmission rate C_i gives M_i a higher rating to improve network throughput.
2. A higher ratio $\frac{C_i}{C_i^{avg}}$ gives M_i a higher rating to prevent starvation for MSSs with low average rates, where C_i^{avg} is the average transmission rate for the BS to send data to M_i in the most recent T frames. Specifically, supposing that an MSS encounters a bad channel condition for a long period (*i.e.*, a lower C_i^{avg} value), we still prefer this MSS if it can now enjoy a higher transmission rate (*i.e.*, $C_i > C_i^{avg}$). In addition, a higher $\frac{C_i}{C_i^{avg}}$ value means that the MSSs is currently in a better condition so that we give it a higher priority to improve the potential throughput.
3. A higher ratio $\frac{B_i^{rt}}{R_i^{rt}}$ gives M_i a higher rating to favor MSSs with more backlogs.

Similarly, for priority level **P3**, we rank the importance of M_i 's non-real-time traffic by

$$I_i^{nrt} = C_i \times \frac{C_i}{C_i^{avg}} \times \frac{1}{S_i^{nrt}}, \quad (3.4)$$

where S_i^{nrt} is the *non-real-time rate satisfaction ratio* of M_i in the most recent T frames, which is calculated by

$$S_i^{nrt} = \frac{\sum_{j=0}^{T-1} A_i^{nrt}(f_c - j)}{T \times R_i^{nrt}}. \quad (3.5)$$

A small S_i^{nrt} means that M_i 's non-real-time traffic may be starved. Thus, a smaller S_i^{nrt} gives M_i a higher rating.

The above two-tier priority rule not only prevents urgent real-time traffics from incurring packet dropping (through the first tier) but also maintains long-term fairness (through the second tier). The network throughput is also improved by giving a higher priority to those MSSs using higher transmission rates (in the second tier). In addition, by giving a γ ratio of non-urgent real-time traffics with level-2 priority, not only the amount of urgent real-time traffics in the next frame can be reduced, but also non-real-time traffics can have opportunity to send their data.

Below, we present the detailed operations of our scheduler. Let e_i be a binary flag to indicate whether an IE has been allocated for M_i , $i = 1..n$. Initially, we set all $e_i = 0$, $i = 1..n$. Besides, the free space FS is deducted by $(\frac{\gamma}{\Delta_{bkt}} - 1) \times \theta_{IE}$ to preserve the space for potential IEs caused by the burst allocator (this will be discussed in the next section), where $\theta_{IE} = \frac{5}{4}$ is the size of an IE.

1. Let U_i^{rt} be the data amount of M_i 's urgent real-time traffic in the current frame. For all M_i with $U_i^{rt} > 0$, we sort them according to their C_i values in a descending order. Then, we schedule the free space FS for each of them as follows, until $FS \leq 0$:
 - (a) Reserve an IE for M_i by setting $FS = FS - \theta_{IE}$. Then, set $e_i = 1$.
 - (b) If $FS > 0$, assign resource $Q_i^{rt} = \min \{FS \times C_i, U_i^{rt}\}$ to M_i and set $FS = FS - \lceil \frac{Q_i^{rt}}{C_i} \rceil$. Then, deduct Q_i^{rt} from B_i^{rt} .
2. After step 1, if $FS > 0$, we sort all M_i that have real-time traffics according to their I_i^{rt} values (by Eq. (3.3)). Then, we schedule the resource for each of them as follows, until either all MSSs in the top γ ratio are examined or $FS \leq 0$:
 - (a) If $e_i = 0$, reserve an IE for M_i by setting $FS = FS - \theta_{IE}$ and $e_i = 1$.
 - (b) If $FS > 0$, assign more resource $\delta = \min \{FS \times C_i, B_i^{rt}\}$ to M_i . Then, set $Q_i^{rt} = Q_i^{rt} + \delta$ and $FS = FS - \lceil \frac{\delta}{C_i} \rceil$. Deduct δ from B_i^{rt} .

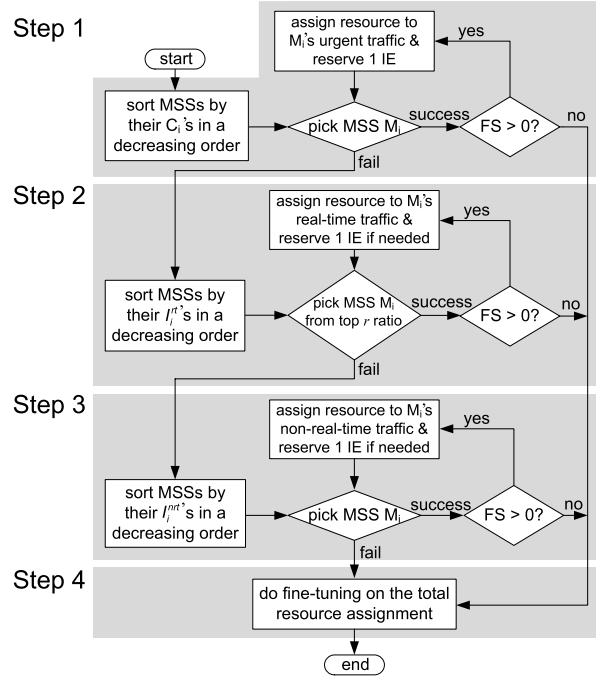


Figure 3.3: The flowchart of the two-tier, priority-based scheduler.

3. After step 2, if $FS > 0$, we sort all M_i according to their I_i^{nrt} values (by Eq. (3.4)). Then, we schedule the resource for each of them as follows, until either all MSSs are examined or $FS \leq 0$:

(a) If $e_i = 0$, reserve an IE for M_i by setting $FS = FS - \theta_{IE}$ and $e_i = 1$.

(b) If $FS > 0$, assign more resource $\delta = \min\{FS \times C_i, B_i^{nrt}\}$ to M_i . Then, set $Q_i^{nrt} = \delta$ and $FS = FS - \lceil \frac{\delta}{C_i} \rceil$. Deduct δ from B_i^{nrt} .

4. Since the bucket size Δ_{bkt} is the allocation unit in our burst allocator, in this step, we will do a fine-tuning on Q_i^{rt} and Q_i^{nrt} such that $(Q_i^{rt} + Q_i^{nrt})$ is aligned to a multiple of Δ_{bkt} for each M_i . To do so, we will gradually *remove* some slots from Q_i^{nrt} and then Q_i^{rt} , until $(\frac{Q_i^{rt} + Q_i^{nrt}}{C_i} \bmod \Delta_{bkt}) = 0$. One exception is when most of data in Q_i^{rt} are urgent, which makes removing any resource from M_i impossible. In this case, we will *add* more slots to M_i until $(\frac{Q_i^{rt} + Q_i^{nrt}}{C_i} \bmod \Delta_{bkt}) = 0$. The above adjustment (*i.e.*, removal and addition) may make the total resource assignment below or beyond the available resource FS . If so, we will further remove some slots from the MSSs with less importance or add some slots to the MSSs with more importance, until the total resource assignment is equal to the initial free space given by the burst allocator.

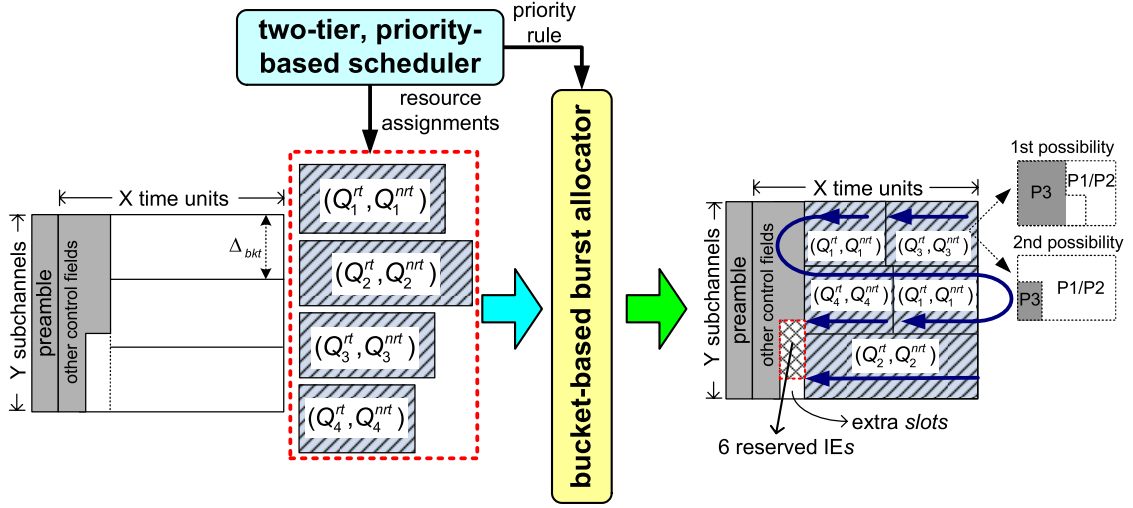


Figure 3.4: An example of the bucket-based burst allocation with three buckets and four resource assignments.

Fig. 3.3 illustrates the flowchart of the scheduler. To summarize, our scheduler generates the resource assignment according to three priorities: (**P1**) urgent traffics, (**P2**) real-time traffics, and (**P3**) non-real-time traffics. Step 1 first schedules those MSSs with urgent traffics to alleviate their real-time traffic delays. Step 2 schedules those top γ ratio of MSSs to reduce the number of MSSs that may have urgent traffics in the following frames. This step also helps reduce the IE overhead of future frames caused by urgent traffics, which is neglected by prior studies. Step 3 schedules those MSSs with lower non-real-time satisfaction ratios to prevent them from starvation. Finally, step 4 reshapes all assignments such that each $(Q_i^{rt} + Q_i^{nrt})$ is divisible by Δ_{bkt} . This step will help the burst allocator to fully utilize a downlink subframe.

We then analyze the time complexity of our scheduler. In step 1, sorting MSSs by their C_i values takes $O(n \lg n)$ time and scheduling the resources for the MSSs with urgent traffics takes $O(n)$ time. In step 2, sorting MSSs by their I_i^{rt} values requires $O(n \lg n)$ time and scheduling the resources for the top γ ratio of MSSs requires at most $O(\gamma n)$ time. In step 3, sorting MSSs by their I_i^{nrt} values costs $O(n \lg n)$ time and scheduling the resources for the MSSs with non-real-time traffics takes $O(n)$ time. In step 4, reshaping all requests spends at most $O(n)$ time. Thus, the total time complexity is $O(n \lg n + n + n \lg n + \gamma n + n \lg n + n + n) = O(n \lg n)$.

3.4.2 Bucket-Based Burst Allocator

Ideally, the free space FS in Eq. (3.2) should accommodate each resource assignment (Q_i^{rt}, Q_i^{nrt}) calculated by the scheduler and its corresponding IE(s). However, since the burst allocation

problem is NP-complete, our bucket-based heuristic will try to squeeze as more MSSs' assignments into FS as possible and allocate one burst per assignment with a very high possibility. If more than one burst is required, more IEs are needed, in which case some assignments originally arranged by the scheduler may be trimmed down or even kicked out by the burst allocator. Given the free space FS by Eq. (3.2), bucket size Δ_{bkt} , and assignments (Q_i^{rt}, Q_i^{nrt}) 's from the scheduler, our bucket-based heuristic works as follows:

1. Slice FS horizontally² into $\frac{Y}{\Delta_{bkt}}$ buckets, each of a height Δ_{bkt} , where Y is divisible by Δ_{bkt} . Fig. 3.4 shows an example by slicing FS into three buckets.
2. Let k be the number of resource assignments given by the scheduler. We reserve $\lceil (k + \frac{Y}{\Delta_{bkt}} - 1) \times \theta_{IE} \rceil$ slots for IEs at the left side of the subframe. In fact, the scheduler has also reserved the space for these IEs, and its purpose will become clear later on. Fig. 3.4 gives an example. Since there are four assignments, $4 + 3 - 1$ IEs are reserved.
3. We then assign bursts to satisfy these resource assignments according to their priorities originally defined in the scheduler. Since each assignment (Q_i^{rt}, Q_i^{nrt}) may have data mixed in categories of **P1**, **P2**, and **P3**, we redefine its priority as follows:
 - (a) An assignment with data in **P1** has a higher priority than an assignment without data in **P1**.
 - (b) Without the existence of data in **P1**, an assignment with data in **P2** has a higher priority than an assignment without data in **P2**.

Then, bursts are allocated in a bucket-by-bucket manner. Specifically, when an assignment (Q_i^{rt}, Q_i^{nrt}) is examined, it will be placed starting from the previous stop point and fill up the bucket from right to left, until either (Q_i^{rt}, Q_i^{nrt}) is satisfied or the left end of the bucket is encountered. In the later case, we will move to the right end of the next bucket and repeat the above allocation process again. In addition, this 'cross-bucket' behavior will require one extra IE for the request. The above operation is repeated until either all assignments are examined or all buckets are exhausted. Fig. 3.4 gives an example, where the four assignments are prioritized by $(Q_3^{rt}, Q_3^{nrt}) > (Q_1^{rt}, Q_1^{nrt}) > (Q_4^{rt}, Q_4^{nrt}) > (Q_2^{rt}, Q_2^{nrt})$. Assignment (Q_1^{rt}, Q_1^{nrt}) requires two IEs since it involves in one cross-bucket behavior.

²We can also slice FS vertically, but the effect will be the same.

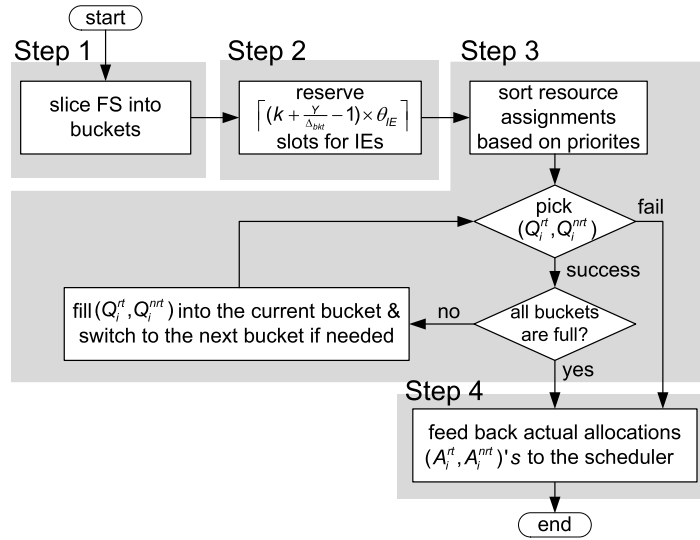


Figure 3.5: The flowchart of the bucket-based burst allocator.

4. According to the allocation in step 3, we place each resource assignment (Q_i^{rt}, Q_i^{nr}) into its burst(s). Besides, the amount of actual allocation is written into each (A_i^{rt}, A_i^{nr}) and fed back to the scheduler for future scheduling.

Fig. 3.5 illustrates the flowchart of the burst allocator. We make some remarks below. First, because there are $\frac{Y}{\Delta_{bkt}}$ buckets, there are at most $\left(\frac{Y}{\Delta_{bkt}} - 1\right)$ *cross-bucket* burst assignments and thus at most $\left(\frac{Y}{\Delta_{bkt}} - 1\right)$ extra IEs are needed. To accommodate this need, some assignments may be trimmed down slightly. This is why (Q_i^{rt}, Q_i^{nr}) and (A_i^{rt}, A_i^{nr}) are not necessarily the same. However, the difference should be very small. Second, the bucket which is located at the boundary of reserved IEs and data (e.g., the third bucket in Fig. 3.4) may have some extra slots (e.g., the lower-left corner of the third bucket). These extra slots are ignored in the above process for ease of presentation, but they can be used to allocate bursts to further improve space efficiency. Third, since each cross-bucket behavior will require one extra IE and there are $\frac{Y}{\Delta_{bkt}}$ buckets, the number of IEs required is bounded, as proved in Theorem 1.

Theorem 1. *In the bucket-based burst allocator, the $\left(k + \frac{Y}{\Delta_{bkt}} - 1\right)$ IEs reserved in step 2 are sufficient for the burst allocation in step 3.*

Proof. Given $\frac{Y}{\Delta_{bkt}}$ buckets $\hat{b}_1, \hat{b}_2, \dots$, and $\hat{b}_{\frac{Y}{\Delta_{bkt}}}$, we can concatenate them into one virtual bucket \hat{b} with $\left(\frac{Y}{\Delta_{bkt}} - 1\right)$ joints. We then allocate one virtual burst for each request from the scheduler in \hat{b} , so we have at most k virtual bursts. Then, we replace each virtual burst by one real burst. However, we require one extra real burst whenever the replaced virtual burst crosses

one joint. The worst case occurs when each of $\left(\frac{Y}{\Delta_{bkt}} - 1\right)$ joints is crossed by one virtual burst. In this case, we require $\left(k + \frac{Y}{\Delta_{bkt}} - 1\right)$ real bursts to replace all virtual bursts. Since each real burst requires one IE, we have to reserve at most $\left(k + \frac{Y}{\Delta_{bkt}} - 1\right)$ IEs. \square

In comparison, a naive burst allocation will require the worst case of $3k$ IEs if the allocation goes in a row-major or column-major way [17] (because each request may require up to 3 IEs). In our scheme, the bucket size Δ_{bkt} can be dynamically adjusted to reflect the ‘grain size’ of our allocation. A larger grain size may cause fewer IEs, but sacrifice fairness; a smaller grain size may cause more IEs, but improve fairness. We will discuss the effect of Δ_{bkt} in Section 3.6.6.

We then analyze the time complexity of our burst allocator. Since we allocate bursts in a zigzag manner, the time complexity is proportional to the number of bursts. By Theorem 1, we have at most $\left(k + \frac{Y}{\Delta_{bkt}} - 1\right)$ bursts. Since we have $k \leq n$ and $\frac{Y}{\Delta_{bkt}}$ is usually smaller than n , the time complexity is $O\left(k + \frac{Y}{\Delta_{bkt}} - 1\right) = O(n)$.

To conclude, the proposed scheduler and burst allocator are dependent with each other by the following two designs: First, the scheduler reserves the extra IE space caused by the bucket partition and arranges resources to MSSs’ traffics so that the resource assignments can align to buckets. Thus, we can enhance the possibility that the burst allocator fully satisfies the resource assignments from the scheduler. Second, the burst allocator follows the priority rule in the scheduler to arrange bursts. Thus, even if the frame space is not enough to satisfy all traffics, urgent real-time traffics can be still arranged with bursts to catch their approaching deadlines.

3.5 Analysis of Network Throughput Loss by The Bucket-Based Scheme

Given an ideal scheduler, we analyze the loss of network throughput caused by our bucket-based burst allocator. To simplify the analysis, we assume that the network has only traffics of priority levels **P1** and **P3**, and each MSS has infinite data in **P3**. (Traffics of **P2** will eventually become urgent traffics of **P1**.) Then, we calculate the difference between the expected throughput by our burst allocator and the maximum throughput by an *ideal* one. In the ideal burst allocator, the number of IEs is equal to the number of resource assignments from the scheduler. Also, the frame resource is always allocated to urgent traffics (**P1**) first and then to non-real-time traffics (**P3**) with the highest transmission rate. It follows that two factors may degrade network throughput by our burst allocator: 1) extra IEs incurred by step 3 in Section 3.4.2 and 2) the

data padding of low-rate non-real-time traffics at the boundary between the data in **P1** and **P3**. Specifically, each burst must begin with the data in **P1** followed by the data in **P3**. Furthermore, if the data in **P3** covers more than one column, it must be sent at the highest transmission rate. If the data in **P3** covers less than a column, it may be sent at a non-highest transmission rate. In the right-hand side of Fig. 3.4, it shows these two possibilities, where **P2** is empty. Note that in the first possibility, all data in **P3** must be transmitted at the highest rate; otherwise, the shaded area will be allocated to the data in **P3** of other MSSs using the highest rate.

Following the above formulation, our objective is to find the throughput loss \mathcal{L} by our burst allocator compared with the ideal one:

$$\mathcal{L} = E[\tilde{O}] \times c_{\text{high}} + E[\tilde{S}], \quad (3.6)$$

where \tilde{O} is the random variable representing the number of extra IEs caused by buckets and \tilde{S} is the random variable representing the throughput degradation (in bits) caused by the low-rate padding in the shaded area of the second possibility in the right-hand side of Fig. 3.4. To simplify the analysis, we assume that there are only two transmission rates c_{high} and c_{low} , where $c_{\text{high}} > c_{\text{low}}$. The probability that an MSS is in either rate is equal.

3.5.1 Calculation of $E[\tilde{O}]$

We first give an example to show how our analysis works. Suppose that we have three MSSs and three buckets. Each bucket has two *arrangement units*, each having Δ_{bkt} slots. Thus, there are totally six arrangement units, denoted by $O_1, O_2, O_3, O_4, O_5,$ and O_6 . Resources allocated to the three MSSs can be represented by two separators '|'. For example, we list three possible allocations: 1) $O_1O_2|O_3O_4|O_5O_6$, 2) $O_1O_2O_3O_4||O_5O_6$, and 3) $O_1|O_2O_3O_4O_5|O_6$. In arrangement 1, we need no extra IE. In arrangement 2, MSS 2 receives no resource, but MSS 1 needs one extra IE. In arrangement 3, MSS 2 requires two IEs.

We will use arrangement units and separators to conduct the analysis. Suppose that we have n MSSs, $\frac{Y}{\Delta_{bkt}} (= B)$ number of buckets, and $X \times B (= \alpha)$ arrangement units (*i.e.*, each bucket has X arrangement units). This can be represented by arbitrarily placing $(n - 1)$ separators along a sequence of α arrangement units. Bucket boundaries appear after each i th arrangement unit such that i is a multiple of X . Note that only $(B - 1)$ bucket boundaries can cause extra IEs as mentioned in Section 3.4. Whenever no separator appears at a bucket boundary, one extra IE is needed. There are totally $\frac{(\alpha + (n - 1))!}{\alpha!(n - 1)!}$ ways to place these separators. Let $\tilde{\mathcal{E}}$ be the random

variable representing the number of bucket boundaries, where each of them is inserted by at least one separator. The probability of ($\tilde{\mathcal{E}} = e$) is calculated by

$$Prob[\tilde{\mathcal{E}} = e] = \frac{C_e^{B-1} \times \frac{(\alpha - (B-1-e) + (n-1-e))!}{(\alpha - (B-1-e))!(n-1-e)!}}{\frac{(\alpha + (n-1))!}{\alpha!(n-1)!}}. \quad (3.7)$$

Note that the term C_e^{B-1} is the combinations to choose e boundaries from the $(B - 1)$ bucket boundaries. Each of these e boundaries is inserted by at least one separator. The remaining $(B - 1 - e)$ bucket boundaries must not be inserted by any separator. To understand the second term in the numerator of Eq. (3.7), we can denote by x_0 the number of separators before the first arrangement unit and by x_i the number of separators after the i th arrangement unit, $i = 1.. \alpha$. Explicitly, we have

$$x_0 + x_1 + \dots + x_\alpha = n - 1, \quad \forall x_i \in \{0, 1, 2, \dots\}.$$

However, when $\tilde{\mathcal{E}} = e$, $(B - 1 - e)$ of these x_i 's must be 0. Also, e of these x_i 's must be larger than or equal to 1. Then, this problem is equivalent to finding the number of combinations of

$$y_0 + y_1 + \dots + y_j + \dots + y_{\alpha - (B-1-e)} = n - 1 - e, \quad \forall y_j \in \{0, 1, 2, \dots\}.$$

It follows that there are $\frac{(\alpha - (B-1-e) + (n-1-e))!}{(\alpha - (B-1-e))!(n-1-e)!}$ combinations. Therefore, $E[\tilde{\mathcal{O}}]$ can be obtained by

$$\begin{aligned} E[\tilde{\mathcal{O}}] &= \sum_{e=0}^{B-1} (\text{number of extra IEs when } \tilde{\mathcal{E}} = e) \times Prob[\tilde{\mathcal{E}} = e] \\ &= \sum_{e=0}^{B-1} (B - 1 - e) \times \frac{C_e^{B-1} \times \frac{(\alpha - (B-1-e) + (n-1-e))!}{(\alpha - (B-1-e))!(n-1-e)!}}{\frac{(\alpha + (n-1))!}{\alpha!(n-1)!}}. \end{aligned} \quad (3.8)$$

3.5.2 Calculation of $E[\tilde{S}]$

Recall that $E[\tilde{S}]$ is the expected throughput degradation caused by the transmission of a burst at a low rate and the burst contains some data padding of non-real-time traffics. To calculate $E[\tilde{S}]$, let us define \tilde{N}_L as the random variable of the number of MSSs using the low transmission rate c_{low} . Since there is no throughput degradation by MSSs using the high transmission rate c_{high} , the overall expected throughput degradation is

$$E[\tilde{S}] = \sum_{m=1}^n E[\tilde{S} | \tilde{N}_L = m] \times Prob[\tilde{N}_L = m]. \quad (3.9)$$

Let \tilde{U}_i be the random variable representing the data amount of M_i 's urgent traffic, $i = 1..n$. Here, we assume that \tilde{U}_i is uniformly distributed among $[1, \mathcal{R}]$, where $\mathcal{R} \in \mathbb{N}$. Let \tilde{X}_j^L be the random variable representing the amount of throughput degradation (in bits) due to the data padding of M_j 's non-real-time traffic when using c_{low} . Since the throughput degradation caused by MSSs using c_{high} is zero, we have

$$E[\tilde{S} | \tilde{N}_L = m] = E \left[\sum_{j=1}^m \tilde{X}_j^L \right]. \quad (3.10)$$

Explicitly, \tilde{X}_i^L and \tilde{X}_j^L are independent of each other for any $i \neq j$, so we have

$$E \left[\sum_{j=1}^m \tilde{X}_j^L \right] = \sum_{j=1}^m E \left[\tilde{X}_j^L \right]. \quad (3.11)$$

Now, let us define I_i^U as an indicator to represent whether or not M_i has urgent traffic such that $I_i^U = 1$ if M_i has urgent traffic; otherwise, $I_i^U = 0$. Since the bursts of low-rate MSSs without urgent traffics will not contain the data padding of non-real-time traffics, no throughput degradation will be caused by them. So, we can derive

$$E[\tilde{X}_j^L] = E[\tilde{X}_j^L | I_j^U = 1] \times Prob[I_j^U = 1] = \left(\sum_{u=1}^{\mathcal{R}} \frac{f(\tilde{U}_j = u)}{\mathcal{R}} \right) \times Prob[I_j^U = 1], \quad (3.12)$$

where

$$f(\tilde{U}_j = u) = \left(\left\lceil \frac{u}{\Delta_{bkt} \times c_{\text{low}}} \right\rceil - \frac{u}{\Delta_{bkt} \times c_{\text{low}}} \right) \times \Delta_{bkt} \times (c_{\text{high}} - c_{\text{low}})$$

is a function to represent the throughput degradation caused by a low-rate MSS with non-real-time data padding when $\tilde{U}_j = u$.

By combining Eqs. (3.9), (3.10), (3.11), and (3.12), we can derive that

$$E[\tilde{S}] = \sum_{m=1}^n \left(\sum_{j=1}^m \left(\sum_{u=1}^{\mathcal{R}} \frac{f(\tilde{U}_j = u)}{\mathcal{R}} \right) \times Prob[I_j^U = 1] \right) \times Prob[\tilde{N}_L = m]. \quad (3.13)$$

Finally, the throughput loss by our burst allocator can be calculated by combining Eqs. (3.8) and (3.13) into Eq. (3.6).

3.6 Performance Evaluation

To verify the effectiveness of our cross-layer framework, we develop a simulator in C++ based on the architecture in [41], as shown in Fig. 3.6. The simulator contains three layers: The *traffic*

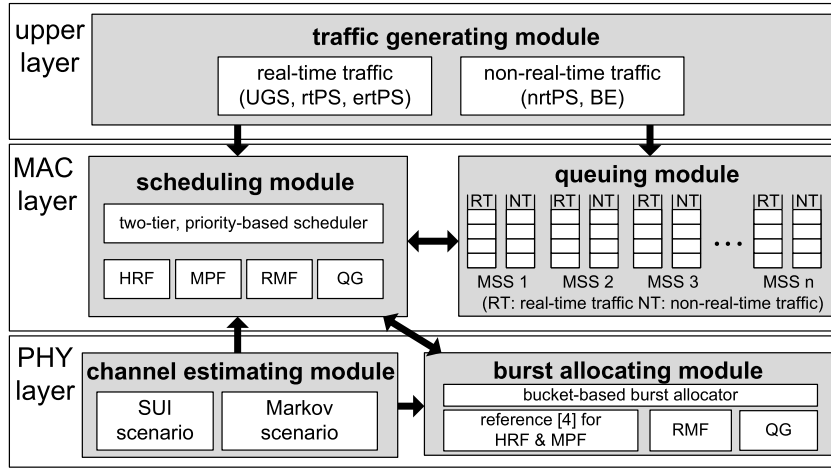


Figure 3.6: The architecture of our C++ simulator.

generating module in the upper layer creates the MSSs' demands according to their real-time and non-real-time traffic requirements. In the MAC layer, the *queuing module* maintains the data queues for each MSS and the *scheduling module* conducts the actions of the scheduler. In the PHY (physical) layer, the *channel estimating module* simulates the channel conditions and estimates the transmission rate of each MSS and the *burst allocating module* conducts the actions of the burst allocator. The arrows in Fig. 3.6 show the interaction between all the modules in our simulator. In particular, the traffic generating module will generate traffics and feed them to the scheduling module for allocating resources and the queuing module for simulating the queue of each traffic. The channel estimating module will send the transmission rates of MSSs to both the scheduling and burst allocating modules for their references. In addition, the scheduling module and the burst allocating module will interact with each other, especially for our scheme.

The simulator adopts a FFT (fast Fourier transform) size of 1024 and the zone category as PUSC with reuse 1. The frame duration is 5 ms. In this way, we have $X = 12$ and $Y = 30$. Six *modulation and coding schemes (MCSs)* are adopted, denoted by a set $MCS = \{QPSK1/2, QPSK3/4, 16QAM1/2, 16QAM3/4, 64QAM2/3, 64QAM3/4\}$. For the traffic generating module, the types of real-time traffics include UGS, rtPS, and ertPS; the types of non-real-time traffics include nrtPS and BE. Each MSS has an admitted real-time data rate R_i^{rt} of $0 \sim 200$ bits and an admitted non-real-time data rate R_i^{nrt} of $0 \sim 500$ bits per frame. In each frame, each MSS generates $0 \sim 2R_i^{rt}$ amount of real-time data and $R_i^{nrt} \sim 4R_i^{nrt}$ amount of non-real-time data.

Table 3.3: The amounts of data carried by each slot and the minimum required SNR thresholds of different MCSs

index	MCSs	data carried by each slot	minimum required SNR
1	QPSK 1/2	48 bits	6 dBm
2	QPSK 3/4	72 bits	8.5 dBm
3	16QAM 1/2	96 bits	11.5 dBm
4	16QAM 3/4	144 bits	15 dBm
5	64QAM 2/3	192 bits	19 dBm
6	64QAM 3/4	216 bits	21 dBm

Table 3.4: The simulation parameters used in the SUI scenario

parameter	value
P_{BS}	1000 milliwatts
subchannel bandwidth (BW)	10 MHz
path loss model	SUI
antenna height	BS: 30 meters; MSS: 2 meters
thermal noise	-100 dBm

For the channel estimating module, we develop two scenarios to estimate the transmission rate of each MSS. The first scenario, called the *SUI (Stanford university interim) scenario*, is based on the SUI path loss model recommended by the 802.16 task group [1]. In particular, each MSS will roam inside the BS's signal coverage (which is the largest area that the BS can communicate with each MSS using the lowest QPSK1/2 MCS) and move following the random waypoint model with the maximal speed of 20 meters per second [8]. The transmission rate of each MSS M_i is determined by its received SNR (signal-to-noise ratio):

$$SNR(BS, M_i) = 10 \cdot \log_{10} \left(\frac{\tilde{P}(BS, M_i)}{BW \cdot N_o} \right),$$

where BW is the effective channel bandwidth (in Hz), N_o is the thermal noise level, and $\tilde{P}(BS, M_i)$ is the received signal power at M_i , which is defined by

$$\tilde{P}(BS, M_i) = \frac{G_{BS} \cdot G_{M_i} \cdot P_{BS}}{L(BS, M_i)},$$

where P_{BS} is the transmission power of the BS; G_{BS} and G_{M_i} are the antenna gains at the BS and M_i , respectively, and $L(BS, M_i)$ is the path loss from the BS to M_i . Given M_i 's SNR, the BS can determine M_i 's MCS based on Table 4.1. Specifically, the BS will choose the highest MCS whose minimum required SNR is smaller than $SNR(BS, M_i)$. Table 3.4 lists the parameters used in the SUI scenario.

The second scenario, called the Markov scenario, adopts a six-state Markov chain [66] to simulate the channel condition of each MSS, as shown in Fig. 3.7. Specifically, let $MCS[i]$

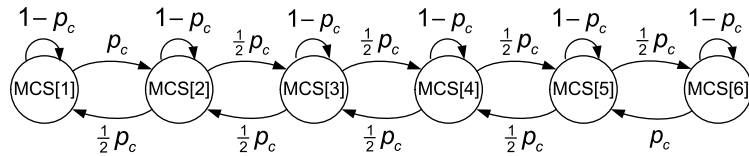


Figure 3.7: A six-state Markov chain to model the channel condition.

be the i th MCS, $i = 1..6$. Suppose that an MSS uses $MCS[i]$ to fit its channel condition at the current frame. The probabilities that the MSS switches to $MCS[i - 1]$ and $MCS[i + 1]$ in the next frame are both $\frac{1}{2}p_c$, and the probability that it remains unchanged is $1 - p_c$. For the boundary cases of $i = 1$ and 6 , the probabilities of switching to $MCS[2]$ and $MCS[5]$, respectively, are both p_c . Unless otherwise stated, we set $p_c = 0.5$ and the initial i value of each MSS is randomly selected from 2 to 5.

We compare our cross-layer framework against the *high rate first (HRF)* scheme [76], the *modified proportional fair (MPF)* scheme [38], the *rate maximization with fairness consideration (RMF)* scheme [20], and the *QoS guarantee (QG)* scheme [75]. HRF always first selects the MSS with the highest transmission rate C_i to serve. MPF assigns priorities to MSSs, where an MSS with a higher C_i value and a lower amount of received data is given a higher priority. RMF first allocates resources to those unsatisfied MSSs according to their minimum requirements, where MSSs are sorted by their transmission rates. If there remains resources, they are allocated to the MSSs with higher transmission rates. Similarly, QG first satisfies the minimum requirements of each MSS's traffics, which are divided into real-time and non-real-time ones. Then, the remaining resources are allocated to those MSSs with higher transmission rates. Since both HRF and MPF implement only the scheduler, we adopt the scheme in [7] as their burst allocators. In our framework, we use $B = 5$ buckets and set $\gamma = 0.3$ in **P2** unless otherwise stated. In Section 3.6.6, we will discuss the effects of these two parameters on the system performance. The duration of each experiment is at least 2000 frames.

3.6.1 Network Throughput

We first compare the network throughput under different number of MSSs (*i.e.*, n), where the network throughput is defined by the amount of MSSs' data (in bits) transmitted by the BS during 2000 frames. We observe the case when the network becomes saturated, where there are $60 \sim 90$ MSSs to be served. Fig. 3.8 shows the simulation results under both the SUI and the Markov scenarios, where the trends are similar. Explicitly, when the number of MSSs

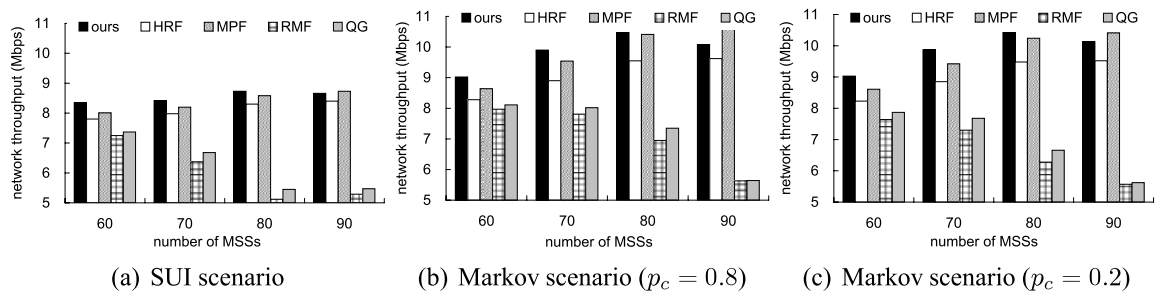


Figure 3.8: Comparison on network throughput.

grows, the throughput increases but will eventually steady when there are too many MSSs (*i.e.*, $n \geq 80$). The throughput under the SUI scenario is lower than that under the Markov scenario because some MSSs may move around the boundary of the BS's coverage, leading a lower SNR and thus a lower MCS. Under the Markov scenario, a higher p_c means that each MSS may change its MCS more frequently and vice versa.

Generally speaking, both RMF and QG ignore the effect of IE overheads on network performance so that their throughput will be degraded. Although HRF serves those MSSs with higher transmission rates first, its throughput is not the highest. The reason is that HRF not only ignores the importance of IE overheads but also neglects the effect of $\frac{C_i}{C_i^{avg}}$ factor on potential throughput when scheduling traffics. The throughput of MPF is higher than that of RMF, QG, and HRF due to two reasons: First, MPF prefers those MSSs using higher transmission rates, which is similar to HRF. However, HRF incurs higher IE overheads because of the scheduling methodology (which will be verified in Section 3.6.2). Second, both RMF and QG try to schedule every traffic in each frame, which generates too many IEs (in fact, we can postpone scheduling some traffics to reduce IE overheads while still guarantee long-term fairness; this will be verified in Sections VI-B and VI-C). On the other hand, MPF enjoys higher throughput because it takes care of IE overheads from the viewpoint of the scheduler. In particular, our cross-layer framework has the highest throughput in most cases because of the following reasons: First, our scheduler assigns a higher priority to those MSSs with higher C_i and $\frac{C_i}{C_i^{avg}}$ values, and thus makes MSSs receive their data in higher transmission rates. Second, both our scheduler and burst allocator can effectively decrease the number of IEs and acquire more sub-frame space for data transmission. Note that when $n = 90$, our cross-layer framework will try to satisfy a large number of urgent traffics to avoid their packets being dropped. In this case, its throughput is slightly lower than that of MPF but our cross-layer framework can significantly reduce the real-time packet dropping ratio, as will be shown in Section 3.6.4.

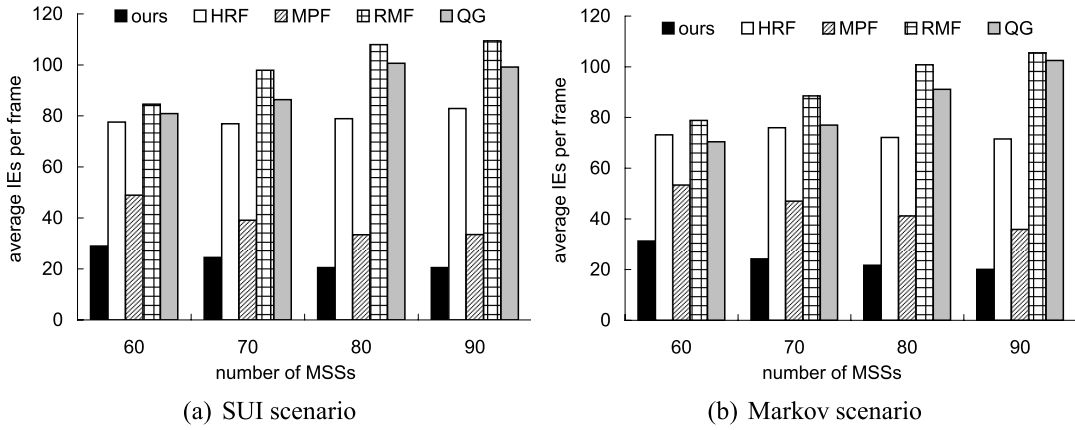


Figure 3.9: Comparison on IE overheads.

3.6.2 IE Overheads and Subframe Utilization

Fig. 3.9 shows the average number of IEs in each downlink subframe. As discussed earlier, HRF, RMF, and QG do not consider IE overheads so that they will generate a large number of IEs. The situation becomes worse when the number of MSSs grows, since each MSS needs to be allocated with at least one burst (and thus one IE). By considering IE overheads in the scheduler, MPF can reduce the average number of IEs per frame. It can be observed that when the number of MSSs grows, the number of IEs in MPF reduces. The reason is that MPF allocates more resources to MSSs in a frame to reduce the total number of scheduled MSSs, thus reducing the number of allocated bursts (and IEs). From Fig. 3.9, our cross-layer framework generates the smallest number of IEs per frame, because not only both the proposed scheduler and burst allocator do consider IE overheads, but also the framework can adjust the number of non-urgent real-time traffics to be served to avoid generating too many bursts.

IE overheads have strong impact on the utilization of downlink subframes, as reflected in Fig. 3.10. Since HRF, RMF, and QG generate a large number of IEs, their subframe utilization will be lower than MPF and our cross-layer framework. It can be observed that the number of buckets B significantly affects the subframe utilization of our cross-layer framework. In particular, a too large B (e.g., 30) will reduce the amount of data carried in each bucket and thus generate many small bursts. On the other hand, a too small B (e.g., 1) may degrade the functionality of buckets and thus some resource assignments may not fully utilize the bursts allocated to them. From Fig. 3.10, we suggest setting $B = 5$ to get the best utilization and the analysis result in Section 3.6.7 will also validate this point.

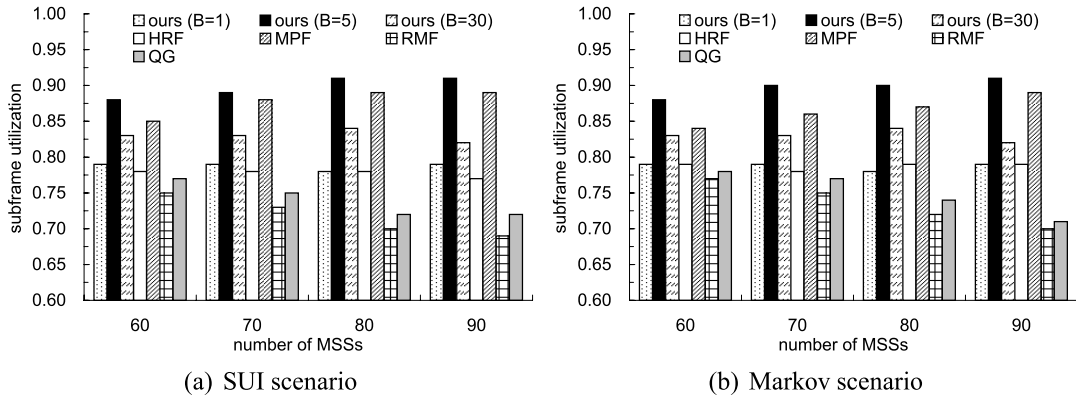


Figure 3.10: Comparison on subframe utilization.

3.6.3 Long-Term Fairness

Next, we verify whether each scheme can guarantee long-term fairness under a highly congested network, where there are 140 ~ 200 MSSs. Fig. 3.11 shows the fairness indices of all schemes. Recall that the network becomes saturated when there are 80 MSSs. Thus, it is impossible to get a fairness index of one because the network resource is not enough to satisfy the requirements of all traffics. From Fig. 3.11, HRF incurs the lowest index because it always serves those MSSs using higher transmission rates. By considering the amount of allocated data of each MSS, MPF can have a higher index than HRF. QG and RMF try to satisfy the minimum requirement of each traffic in every frame and thus leading higher indices. Since RMF allocates the resources to MSSs sorted by their transmission rates, its index will be lower than QG.

Our cross-layer framework has the highest fairness index (more than 0.85) due to two reasons: First, our priority-based scheduler only schedules γ ratio of non-urgent real-time traffics to avoid starving non-real-time traffics. Second, our cross-layer framework tries to reduce the IE overheads and acquire more frame space to allocate bursts for MSSs' traffics. In this case, we have more resources to fairly distribute among MSSs. Thus, our cross-layer framework can maintain long-term fairness even in a highly congested network.

3.6.4 Packet Dropping Ratios of Real-Time Traffics

We then observe the packet dropping ratios of real-time traffics, where each MSS will generate $0 \sim 2R_i^{rt}$ amount of real-time data in each frame. When a real-time packet is not transmitted within 6 frames (*i.e.*, 30 ms) after being generated, it will be dropped. Fig. 3.12 shows the real-time packet dropping ratios of all schemes under 10 ~ 110 MSSs. Both HRF and MPF distribute resources to MSSs based on the transmission rates without considering the traffic types,

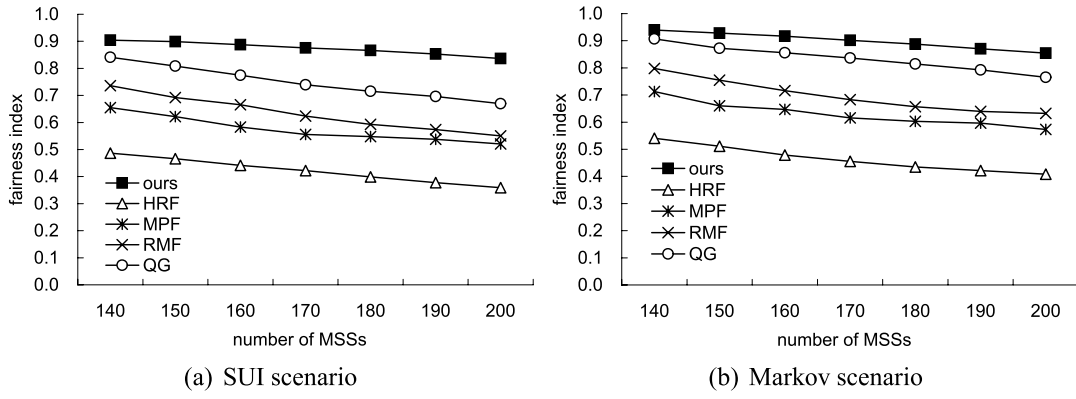


Figure 3.11: Comparison on long-term fairness.

so their ratios begin raising when $n \geq 50$. In this case, a large amount of non-real-time traffics will compete with real-time traffics for the limited resource. On the other hand, the ratios of RMF and QG begin raising when $n \geq 90$. Since both RMF and QG try to satisfy the minimum requirements of all traffics in each frame, they can avoid real-time packet dropping when the network is not saturated (*i.e.*, $n < 90$). Our cross-layer framework can have almost zero ratio due to three reasons: First, our priority-based scheduler assigns urgent real-time traffics with the highest priority. Also, it schedules a γ ratio of non-urgent real-time traffics to avoid generating too many urgent traffics in the following frames. Second, our bucket-based burst allocator arranges bursts based on the priorities from the scheduler, so the bursts of those urgent real-time traffics can be allocated first to avoid packet dropping. Third, both our scheduler and burst allocator try to reduce IE overheads and thus more urgent real-time traffics can be served in each frame.

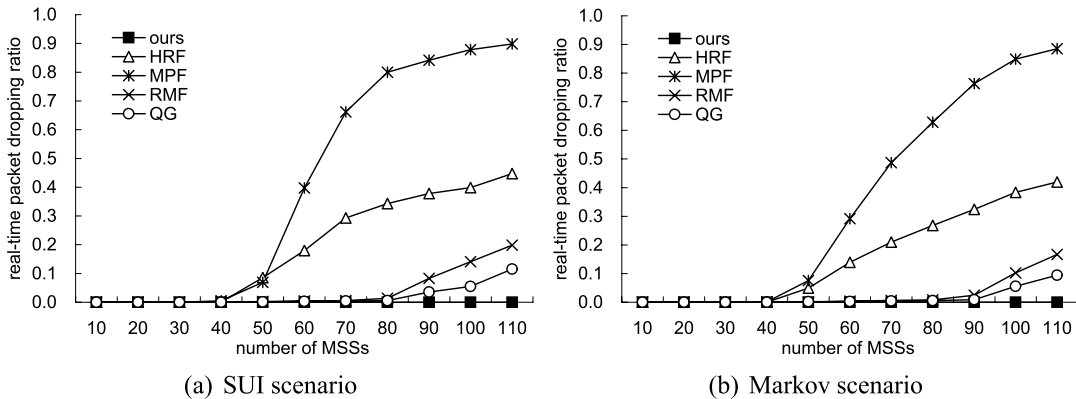


Figure 3.12: Comparison on real-time packet dropping ratios under different number of MSSs.

Fig. 3.13 shows the real-time packet dropping ratios of all schemes under different admitted non-real-time data rates, where the network is saturated. Since MPF proportionally distributes

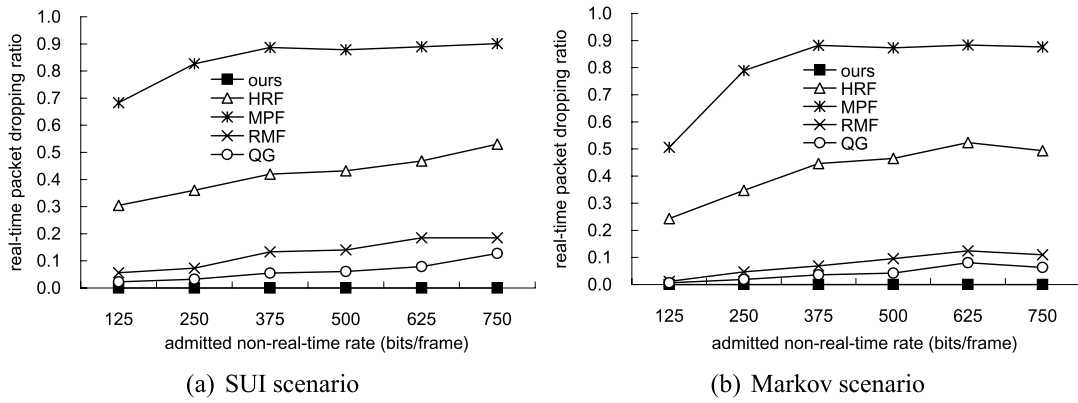


Figure 3.13: Comparison on real-time packet dropping ratios under different admitted non-real-time rates.

resources among MSSs, it incurs the highest real-time packet dropping ratio. On the other hand, since some MSSs with real-time traffics may have higher transmission rates, the ratio of HRF is lower than that of MPF. As discussed earlier, both RMF and QG try to satisfy the minimum requirement of each traffic, their ratios can become lower. Note that since QG differentiates real-time traffics from non-real-time ones, its ratio is lower than that of RMF. Our cross-layer framework always has the zero ratio because not only the bursts of urgent real-time traffics are allocated first but also our framework can acquire more frame space to serve urgent real-time traffics by reducing IE overheads.

Since the trends under both SUI and Markov scenarios are similar, we only show the results under the Markov scenario in the following experiments.

3.6.5 Satisfaction Ratios of Non-Real-Time Traffics

Next, we measure the satisfaction ratios of non-real-time traffics (by Eq. (3.5)) under a saturated network. Fig. 3.14 shows the satisfaction ratios of non-real-time traffics of the bottom 10% MSSs. When the non-real-time rate is larger than 125 bits/frame, the ratio of HRF is zero because these bottom 10% MSSs (whose transmission rates must be lower) are starved. The ratio of MPF starts diminishing when the non-real-time rate is larger than 250 bits/frame because MPF proportionally distributes resources among traffics. By satisfying the minimum requirement of each traffic, the ratios of RMF and QG are close to one. Our cross-layer framework can have a ratio of nearly one for the bottom 10% MSSs, which means that non-real-time traffics will not be starved even though our scheme prefers real-time traffics.

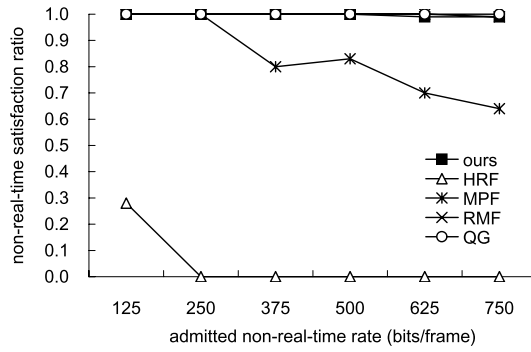


Figure 3.14: Comparison on non-real-time satisfaction ratios of the bottom 10% MSSs under the Markov scenario.

3.6.6 Effects of System Parameters

We then observe the effects of system parameters in our cross-layer framework on network throughput, subframe utilization, and IE overheads under a saturated network (*i.e.*, the number of MSS is 90). Fig. 3.15 shows the impact of the number of buckets (*i.e.*, B) on network throughput, utilization, and overhead ratios when $Y = 32$. Here, the *overhead ratio* is defined as the ratio of the number of slots used for MAP information (*e.g.*, DL-MAP, UL-MAP, and IEs) to the total number of slots in a downlink subframe. Generally speaking, the utilization decreases when the overhead ratio increases, since they are complementary to each other. From Fig. 3.15, the utilization first increases and then decreases when B grows. The former increment is due to that some resource assignments do not fully utilize their allocated bursts. On the other hand, the later decrement is because the burst allocator generates too many bursts to satisfy the thinner buckets. The overhead ratio increases when B increases, because more IEs are generated. In addition, when $B \leq 4$, the throughput increases when B grows, because more buckets may serve more requests. On the other hand, when $B \geq 8$, such a trend reverses because more IEs are generated, causing lower utilization. From Fig. 3.15, we suggest setting $B = 4 \sim 8$ since this range of B value improves both throughput and utilization while reduces IE overheads.

Fig. 3.16 shows the effects of γ and B on real-time packet dropping ratios and network throughput in our cross-layer framework. Explicitly, the real-time packet dropping ratio decreases when γ grows, because more real-time traffics can be served. However, when γ increases, the throughput may decrease because the scheduler has to select more non-urgent real-time traffics to serve. In this case, some real-time traffics with lower transmission rates may be served, which degrades the throughput. As mentioned earlier, a large B may generate more

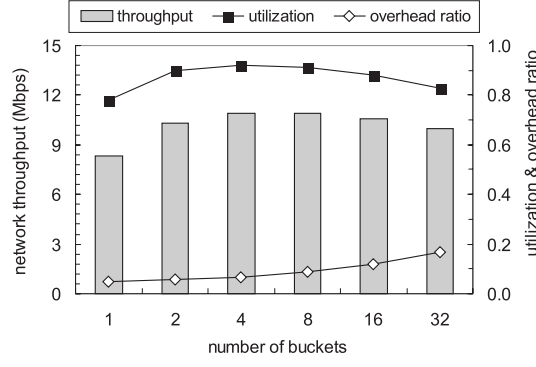


Figure 3.15: Effect of the number of buckets (B) on network throughput, subframe utilization, and IE overheads under the Markov scenario.

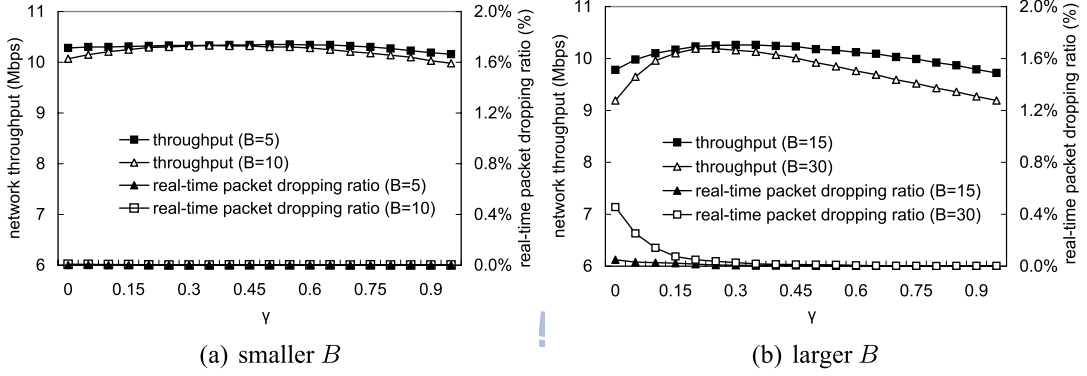


Figure 3.16: Effect of γ on network throughput and real-time packet dropping ratios under the Markov scenario.

IEs and thus reduce the utilization. Thus, the throughput in the case of larger B (e.g., $B = 15$ and 30) starts dropping earlier than that in the case of smaller B (e.g., $B = 5$ and 10). From Fig. 3.16, we suggest setting $\gamma = 0.15 \sim 0.45$ since this range of γ value not only improves network throughput but also reduces real-time packet dropping ratios, under different values of B .

3.6.7 Verification of Throughput Analysis

Finally, we verify our analytic results in the part, where two transmission rates, $c_{\text{low}} = 48$ bits/slot and $c_{\text{high}} = 96$ bits/slot, are adopted. The probabilities that an MSS can use c_{low} and c_{high} are both 0.5. Then, the probability that m MSSs can use c_{low} is

$$\begin{aligned} \text{Prob}[\tilde{N}_L = m] &= C_m^n \times (\text{Prob}[\text{transmission rate} = c_{\text{low}}])^m \times (\text{Prob}[\text{transmission rate} = c_{\text{high}}])^{n-m} \\ &= \frac{n!}{m!(n-m)!} \times (0.5)^m \times (0.5)^{n-m} = \frac{(0.5)^n \times n!}{m!(n-m)!}. \end{aligned}$$

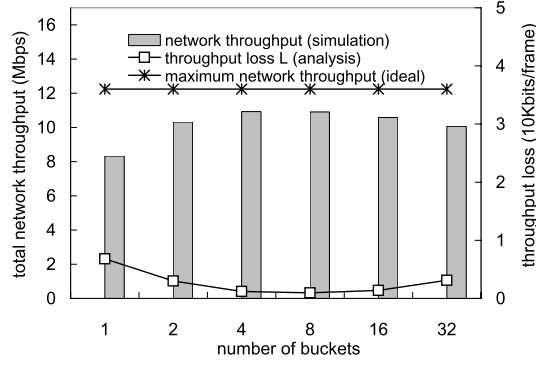


Figure 3.17: Effect of the number of buckets (B) on the throughput loss \mathcal{L} (by analysis) and the total network throughput (by simulation) under the Markov scenario.

In addition, the probability that an MSS M_i has urgent data is

$$Prob[I_i^U = 1] = (1 - \gamma)^{T_D - 1},$$

where T_D is the deadline of real-time data that will be dropped (in frames). Note that since the scheduler will serve all queued real-time data from the top γn MSSs in each frame, after $(T_D - 1)$ frames, the probability of an MSS with urgent data is no more than $(1 - \gamma)^{T_D - 1}$. In our simulation, we set $\gamma = 0.3$, $T_D = 6$ frames, and $\mathcal{R} = 200$ bits/frame.

Fig. 3.17 show the analysis and simulation results. When $B < 4$, the throughput loss \mathcal{L} decreases but the network throughput increases as B increases. On the other hand, when $B > 8$, \mathcal{L} increases but the network throughput decreases as B increases. This result indicates that the minimum value of \mathcal{L} by analysis appears at the range of $B = [4, 8]$ while the maximum network throughput by simulation appears at the same range of $B = [4, 8]$. Thus, our analysis and simulation results are consistent. From Fig. 3.17, we suggest setting $B = 4 \sim 8$ to maximize the network throughput and minimize \mathcal{L} , which matches the results in Fig. 3.15. Therefore, our analysis can validate the simulation results and provide guidelines for the setting of the burst allocator.

Chapter 4

A Power and Bandwidth Allocation in WiMAX Relay Networks

4.1 Motivations

Recently, to overcome the coverage hole, shadow, and NLOS (non-line-of-sight) limitations, the 802.16j extension [30] is proposed to add *relay stations (RSs)*. It has been proved in [9, 42, 57] that MSSs can enjoy higher throughput and/or lower energy consumption with the help of RSs. The standard defines two types of RSs. An RS is called *transparent* if MSSs are not aware of its existence. Otherwise, it is *non-transparent*. Transparent RSs are considered easier to implement than non-transparent ones since they do not need to manage the resources of networks [26].

In the literature, several studies [18, 24, 64, 65] evaluate the network capacity of an IEEE 802.16j network. References [44, 45, 49, 13] address the placement of RSs to improve the network performance. References [21, 40, 22] discuss the selection of RSs to enhance the network capacity. For transparent-relay networks, [60] shows how to leverage channel diversity and concurrent transmissions to increase network throughput. Reference [70] suggests reusing frequency and placing RSs in an irregular manner to improve network throughput. In [48], a Markov decision process is used for admission control and a chance-constrained assignment scheme is proposed to minimize the number of RSs required and to maximize their rates. An isolation band around each RS cluster is adopted in [69] to allow more frequency reuse between RSs and the BS. References [23, 25] adopt a minimal coloring approach to maximize down-link capacity while reducing the differences among MSSs' rates. The above studies all aim at improving network capacity but do not consider the energy conservation of MSSs. A solution of multiple-choice knapsack problem is exploited in [73] to reduce the energy consumption of MSSs, but it considers the PMP mode and does not exploit RSs to help save MSSs' energy.

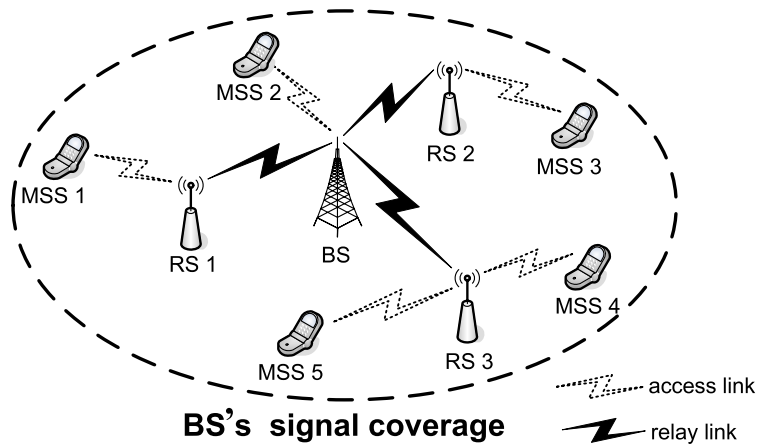


Figure 4.1: The uplink communication of an 802.16j transparent-relay network.

As can be seen, existing works have not well addressed the energy conservation issue in IEEE 802.16j networks. This work tries to minimize MSSs' energy consumption subject to satisfying their traffic demands in each frame by selecting proper paths, rates (in terms of *modulation and coding schemes*, or MCSs in short), and spatial reuse. We show this problem to be NP-complete and propose two energy-efficient heuristics. Below, we first introduce our network architecture and energy model. Then, we define our resource allocation problem and prove its NP-completeness property.

4.2 Preliminaries

4.2.1 Network Model

In an 802.16j transparent-relay network, there is one BS supporting multiple MSSs, as shown in Fig. 4.1. The coverage range of the BS is defined as the reachable area when the lowest MCS (such as QPSK1/2) and the largest power are used. Inside the coverage range, RSs are deployed to help relay data between MSSs and the BS. An MSS can send its data to the BS either directly or indirectly through an RS. However, there are no communication links between two RSs and two MSSs. Therefore, the network is a two-level tree with the BS as the root and MSSs as the leaves. The standard defines two types of links for uplink communication. A link is called an *access link* if it connects to an MSS at one end; otherwise, it is called a *relay link*. Fig. 4.1 shows some examples.

The network resource is divided into *frames*, where a frame is a two-dimensional (subchannel \times time slot) array. Each frame is further divided into a *downlink subframe* and an *uplink*

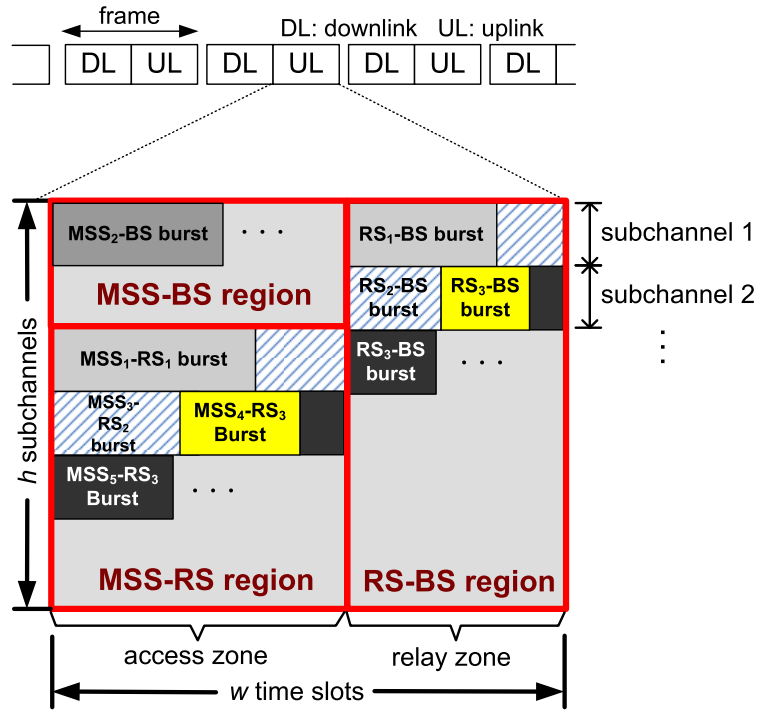


Figure 4.2: The structure of the uplink subframe.

subframe. We show the uplink subframe in Fig. 4.2. It is divided into an *access zone* and a *relay zone*, which are designed for access links and relay links, respectively. The access zone is further divided into an MSS-BS region and an MSS-RS region. For convenience, the relay zone is also called the RS-BS region. Note that these regions have no overlap with each other. However, their sizes can be changed frame by frame.

In this work, we adopt the *PUSC (partial usage of subchannel) mode*, which is very suitable for mobile applications [56]. Under the PUSC mode, *bursts* are the basic resource allocation units, where a burst is a sequence of slots arranged in a row-wise manner, as shown in Fig. 4.2. Note that a burst may cross multiple subchannels. The transmission power and rates of MSSs and RSs are adjustable. However, the transmission rate of an MSS within one burst should be fixed. The BS is responsible for allocating bursts for MSSs and RSs. In MSS-BS and RS-BS regions, since the BS is the only receiver, no two bursts can overlap. In the MSS-RS region, however, spatial reuse is allowed.

4.2.2 Energy Cost Model

Table 4.1 shows the available MCSs in IEEE 802.16j and their rates and required SINRs, denoted by $rate(\cdot)$ and $\delta(\cdot)$, respectively. Let d_i be the number of bits to be transmitted by MSS _{i}

Table 4.1: MCSs supported by IEEE 802.16j.

MCS	scheme	$rate(MCS_k)$	$\delta(MCS_k)$
MCS_1	QPSK 1/2	48 bits/slot	6 dBm
MCS_2	QPSK 3/4	72 bits/slot	8.5 dBm
MCS_3	16QAM 1/2	96 bits/slot	11.5 dBm
MCS_4	16QAM 3/4	144 bits/slot	15 dBm
MCS_5	64QAM 2/3	192 bits/slot	19 dBm
MCS_6	64QAM 3/4	216 bits/slot	21 dBm

in a frame. If MSS_i adopts MCS_k , then it requires $T_i = \left\lceil \frac{d_i}{rate(MCS_k)} \right\rceil$ slots to transmit its data. So, the energy cost of MSS_i is $E_i = T_i \times P_i$, where P_i is the required transmission power (in mW). Suppose that there are n MSS s to be served. The total energy cost is

$$E_{\text{total}} = \sum_{i=1}^n E_i.$$

The transmission power P_i is modeled as follows. Consider any receiver j (which can be any RS or the BS). With power P_i , the received signal power at receiver j is

$$\tilde{P}(i, j) = \frac{G_i \cdot G_j \cdot P_i}{L(i, j)}, \quad (4.1)$$

where G_i and G_j are the antenna gains at MSS_i and receiver j , respectively, and $L(i, j)$ is the path loss from MSS_i to receiver j . Here, we adopt the standard *two-ray ground model* [47] to calculate $L(i, j)$, which is recommended by the 802.16j task group. So, the SINR (in dBm) perceived by receiver j is

$$SINR(i, j) = 10 \cdot \log_{10} \left(\frac{\tilde{P}(i, j)}{B \cdot N_o + I(i, j)} \right), \quad (4.2)$$

where B is the effective channel bandwidth (in Hz), N_o is the thermal noise level, and $I(i, j)$ is the interference caused by other transmitters, which is evaluated by

$$I(i, j) = \sum_{l \neq i} \tilde{P}(l, j).$$

MSS_i 's data can be correctly decoded by receiver j if

$$SINR(i, j) \geq \delta(MCS_k). \quad (4.3)$$

By integrating Eqs. (4.1) and (4.2) into Eq. (4.3), the minimum power required for MSS_i to reach receiver j using MCS_k is

$$P_i \geq \frac{10^{\frac{\delta(MCS_k)}{10}} (B \cdot N_o + I(i, j)) \cdot L(i, j)}{G_i \cdot G_j}. \quad (4.4)$$

4.2.3 Problem Definition

We are given an 802.16j network containing one BS, m RSs, and n MSSs. Each MSS $_i$, $i = 1..n$, has a maximum transmission power of P_i^{MAX} (mW per subchannel) and has an uplink traffic demand of d_i bits per frame granted by the traffic management of the BS¹. We assume that MSSs may move around within the BS's signal coverage, but the relative distances among BS, RSs, and MSSs can be estimated², from which we can construct the network topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the node set and \mathcal{E} is the communication link set. A *path* on \mathcal{G} can be either a direct link from an MSS to the BS or a link from an MSS to an RS and then to the BS. An uplink frame has h subchannels and w time slots. Bursts in the MSS-RS region can overlap with each other so as to exploit spatial reuse. But, bursts in the MSS-BS and RS-BS regions cannot overlap. If there is a burst allocated in the MSS-RS region, a “matching” burst must be allocated in the RS-BS region to relay the former data. For example, in Fig. 4.2, since an MSS₁-RS₁ burst is allocated in the MSS-RS region, there must be a corresponding RS₁-BS burst allocated in the RS-BS region. However, the sizes of these two bursts may not be the same because they may use different MCSs.

Let \mathcal{R} be the set of all possible paths on \mathcal{G} . The *energy-conserved resource allocation (ERA)* problem asks how to find a set of transmission paths $\mathcal{R}_p \subseteq \mathcal{R}$ and the corresponding MCSs, bursts, and transmission power for MSSs under an $h \times w$ frame space constraint such that the total energy cost E_{total} is minimized. Specifically, we denote by $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i)$ the *transmission schedule* of MSS $_i$ in a frame, where $J(i) = 0..m$ and $K(i) = 1..6$. For ease of presentation, we use RS₀ as a special case to represent the BS. So, when $J(i) = 0$, it means that MSS $_i$ transmits to the BS directly using MCS $_{K(i)}$ with power P_i ; otherwise, it means that MSS $_i$ transmits to RS $_{J(i)}$ using MCS $_{K(i)}$ with power P_i and then RS $_{J(i)}$ relays the data to the BS using the best possible MCS. In either case, P_i has to be bounded between the minimum required power and P_i^{MAX} , i.e.,

$$\frac{10^{\frac{\delta(MCS_{K(i)})}{10}}(B \cdot N_o + I(i, J(i))) \cdot L(i, J(i))}{G_i \cdot G_{J(i)}} \leq P_i \leq P_i^{MAX}. \quad (4.5)$$

In addition, we use $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_G\}$ to denote the set of transmission groups in a frame. Each $\tau_\ell \in \mathcal{T}$ is a transmission group consisting of either one MSS-BS transmission schedule

¹Here, we assume that the BS has a traffic scheduler and admission controller to manage MSSs' demands according to their QoS requirements.

²The relative distances between MSS and RSs/BS can be estimated periodically by RSs and the BS through some existing techniques such as evaluating the *received signal strength (RSS)* [11, 10].

or multiple MSS-RS transmission schedules. When there are multiple schedules in τ_ℓ , it means that the MSSs therein can concurrently transmit to their corresponding RSs with overlapping (however, the corresponding RS-BS transmissions cannot overlap with each other). Let \mathcal{B}_ℓ be the binary indicator such that $\mathcal{B}_\ell = 1$ if τ_ℓ contains a single MSS-BS transmission and $\mathcal{B}_\ell = 0$ otherwise. Assume that s_a is a transmission schedule in group τ_ℓ , i.e., $s_a \in \tau_\ell$. Then, the total number of slots required by the transmission group τ_ℓ is expressed by

$$S_{tot}(\tau_\ell) = \begin{cases} \left\lceil \frac{d_a}{rate(MCS_{K(a)})} \right\rceil, & \text{if } \mathcal{B}_\ell = 1 \\ S_g(\tau_\ell) + \sum_{\forall s_a \in \tau_\ell} \left\lceil \frac{d_a}{rate(MCS_{\hat{K}(a)})} \right\rceil, & \text{if } \mathcal{B}_\ell = 0 \end{cases}.$$

In the case of $\mathcal{B}_\ell = 1$, it is the number of required slots in the MSS-BS region. In the case of $\mathcal{B}_\ell = 0$, it is the required slots in the MSS-RS region plus those in the RS-BS region. Here, $MCS_{\hat{K}(a)}$ is the best feasible MCS level for $RS_{J(a)}$ to relay MSS_a 's data to the BS. $S_g(\tau_\ell)$ is the required slots of τ_ℓ in the MSS-RS region. Because of concurrent transmissions, we can conduct $S_g(\tau_\ell)$ as follows:

$$S_g(\tau_\ell) = \max_{\forall s_a \in \tau_\ell} \left\{ \left\lceil \frac{d_a}{rate(MCS_{K(a)})} \right\rceil \right\}.$$

Because the total required slots of all transmission schedules cannot exceed the frame space, we have

$$\sum_{\tau_\ell \in \mathcal{T}} S_{tot}(\tau_\ell) \leq h \times w. \quad (4.6)$$

The goal of the ERA problem is to minimize the total energy consumption of all MSSs:

$$\min_{s_i, i=1..n} E_{total} = \sum_{i=1..n} T_i \cdot P_i = \sum_{i=1..n} \left\lceil \frac{d_i}{rate(MCS_{K(i)})} \right\rceil \cdot P_i, \quad (4.7)$$

by calculating the transmission schedule s_i for each MSS_i and group τ_ℓ that s_i belongs to, under the power constraint in Eq. (4.5) and the frame space constraint in Eq. (4.6).

Theorem 2. *The ERA problem is NP-complete.*

Proof. To simplify the proof, we consider the case of no spatial reuse in the MSS-RS region and each MSS has only one fixed transmission power. So, the MCS and burst(s) of each path are unique. Thus, the energy cost of an MSS on each path is uniquely determined. Then, we formulate the resource allocation problem as a decision problem: *Energy-conserved resource allocation decision (ERAD)* problem: Given the network topology \mathcal{G} and the demand of each

MSS, we ask whether or not there exists a path set \mathcal{R}_p on \mathcal{G} such that all MSSs can conserve the total amount of energy \mathcal{Q} to satisfy their demands. Then, we show ERAD problem is NP-complete.

We first show that the ERAD problem belongs to NP. Given a problem instance and a solution containing the path set, it can be verified whether or not the solution is valid in polynomial time. Thus, this part is proved.

We then reduce the *multiple-choice knapsack (MCK) problem* [35], which is known to be NP-complete, to the ERAD problem. Consider that there are n disjointed classes of objects, where each class i contains N_i objects. In each class i , every object $x_{i,j}$ has a profit $q_{i,j}$ and a weight $u_{i,j}$. Besides, there is a knapsack with capacity of \mathcal{U} . The MCK problem asks whether or not we can select exact one object from each class such that the total object weight is no larger than \mathcal{U} and the total object profit is \mathcal{Q} .

We then construct an instance of the ERAD problem as follows. Let n be the number of MSSs. Each MSS $_i$ has N_i paths to the BS. When MSS $_i$ selects a path $x_{i,j}$, it will conserve energy of $q_{i,j}$ ³ and the system should allocate burst(s) of a total size of $u_{i,j}$ to transmit MSS $_i$'s data to the BS. The total frame space is $w \cdot h = \mathcal{U}$. Our goal is to let all MSSs conserve energy of \mathcal{Q} and satisfy their demands. We show that the MCK problem has a solution if and only if the ERAD problem has a solution.

Suppose that we have a solution to the ERAD problem, which is a path set \mathcal{R}_p with MSSs' conserved energy and burst allocations. Each MSS can choose exact one path and we need to assign paths to all MSSs to satisfy their demands. The total size of bursts cannot exceed the frame space \mathcal{U} and the conserved energy of all MSSs is \mathcal{Q} . By viewing the paths of an MSS as a class of objects and the frame as the knapsack, the paths in \mathcal{R}_p all constitute a solution to the MCK problem. This proves the *if* part.

Conversely, let $\{x_{1,\alpha_1}, x_{2,\alpha_2}, \dots, x_{n,\alpha_n}\}$ be a solution to the MCK problem. Then, for each MSS $_i$, $i = 1..n$, we select a path such that MSS $_i$ conserves energy of q_{i,α_i} and the size of allocated burst(s) to transmit MSS $_i$'s data to the BS is u_{i,α_i} . In this way, the conserved energy of all MSSs will be \mathcal{Q} and the overall burst size is no larger than \mathcal{U} . This constitutes a solution to the ERAD problem, thus proving the *only if* part. \square

³Note that the conserved energy of an MSS's path is compared to the same MSS's path with the most energy cost.

4.3 Two Heuristics to the ERA Problem

Since the ERA problem is NP-complete, finding an optimal solution is impractical due to the time complexity. Thus, we propose two energy-efficient heuristics, the DFA and EFA schemes. Below, we first give the rationale of our heuristics and then depict the DFA and EFA schemes.

4.3.1 The Rationale of Our Designs

We first observe what the key factors are and how they affect the goal (energy consumption) and the constraint (resource usage) of the ERA problem. Explicitly, we reveal that the transmission rate, the number of concurrent transmissions, and the distance to the receiver (either an RS or the BS) have a great impact on these two terms. To show how these three factors affect the energy consumption and resource usage of each MSS, we conduct an experiment as shown in Fig. 4.3. Consider a network consisting of one BS, four RSs, and four MSSs. Each MSS selects a distinct RS to relay its data and the network allows four concurrent transmissions. Assume that the distance between each MSS and its RS is the same and each MSS has an identical uplink demand. Fig. 4.4 shows the results on normalized energy consumption and resource usage of an MSS. In Fig. 4.4(a), the transmission rate of an MSS is normalized by the highest MCS. We can observe that when a lower MCS is used, the MSS will need more resources (i.e., frame space) but can reduce its consumed energy. The benefit ratio of the conserved energy to the increased resource usage is more significant when the MSS degrades its MCS from a higher level (such as 5 or 6) to a next lower one (such as 4). In this case, the MSS can greatly reduce its energy consumption by increasing only a small amount of resource usage. On the other hand, from Fig. 4.4(b), it can be observed that more concurrent transmissions can decrease resource usage linearly but increase the energy consumption drastically. Although concurrent transmissions can help resource reuse but it harms MSSs in terms of the energy consumption. Finally, in Fig. 4.4(c), it can be observed that the resource usage is not affected by the distance to the receiver when the MCS is fixed, but it can save the consumed energy greatly when the MSS chooses a closer RS to relay its data.

From the experiments in Fig. 4.4, we can obtain two important observations:

- To reduce the energy consumption of an MSS, we have to decrease its MCS level (and thus the transmission rate), the number of concurrent transmissions, and the distance to the receiver. However, doing these will also increase the resource usage of the MSS. This

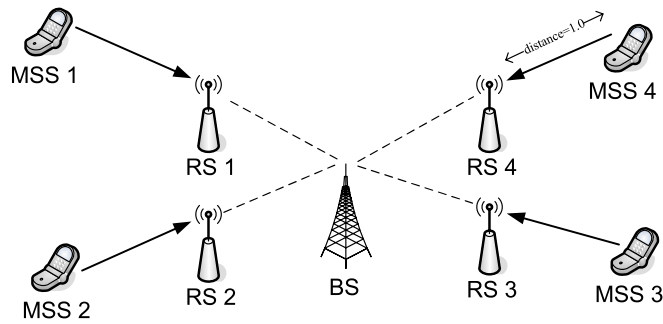
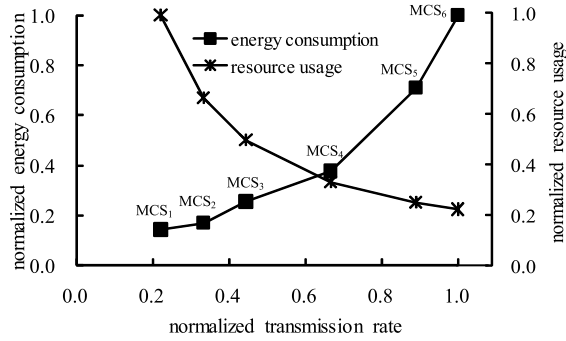


Figure 4.3: The example of a transparent-relay network with one BS, four RSs, and four MSSs.

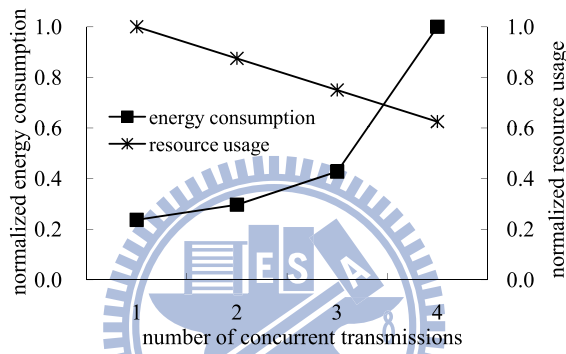
means that the energy conservation is inversely proportional to the used resource. Thus, we should keep in mind that the overall resource usage of all MSSs cannot exceed the frame space constraint when reducing energy.

- The amount of MSS's energy reduction is "jointly" decided by its MCS, the number of concurrent transmissions, and the distance to the receiver. In order to minimize the MSS's energy consumption, it is insufficient to decrease the three factors individually. Since the experiments show that the benefit ratio of energy decrement to resource increment for each factor is greatly different. An MSS may save more energy by considering more than one factor simultaneously. For example, an MSS may not be able to relay its data to an RS closer to it because that RS is used by another MSS. When considering both the factors of concurrent transmissions and the distance to the receiver, the MSS can change to another transmission group and choose such RS to further save energy (even if it may increase the number of concurrent transmissions in that group). This adjusting may be more efficient than that of considering only one factor (such as the MCS). Therefore, we need to consider the possible combination of three factors when trying to reduce MSSs' energy consumption.

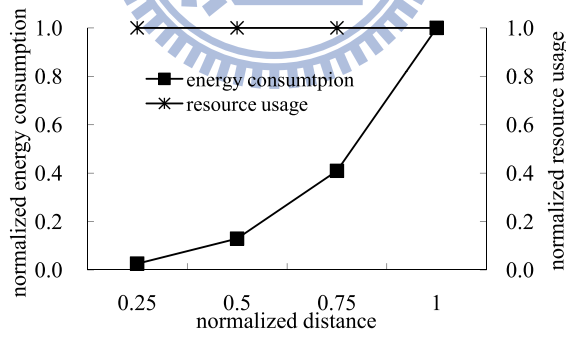
Based on the two observations and the three key factors, our DFA and EFA heuristics adopt a gradient-like search method to find the suboptimal solutions, as shown in Fig. 4.5. For ease of presentation, we say that a solution is demand-satisfied if it can satisfy all MSSs' demands. Besides, a solution is *feasible* if it is not only demand-satisfied but also the overall frame usage does not exceed the frame space. Given the solution set, DFA first selects a feasible solution which consumes as less frame space as possible to be its starting point. Then, it adopts a forward search to approximate the optimal solution. In each step of search, it tries to adjust the transmission schedule of one MSS by evaluating the combinations of the three factors mentioned



(a)



(b)



(c)

Figure 4.4: The effects of rate, concurrent transmission, and receiver distance on energy consumption and resource usage.

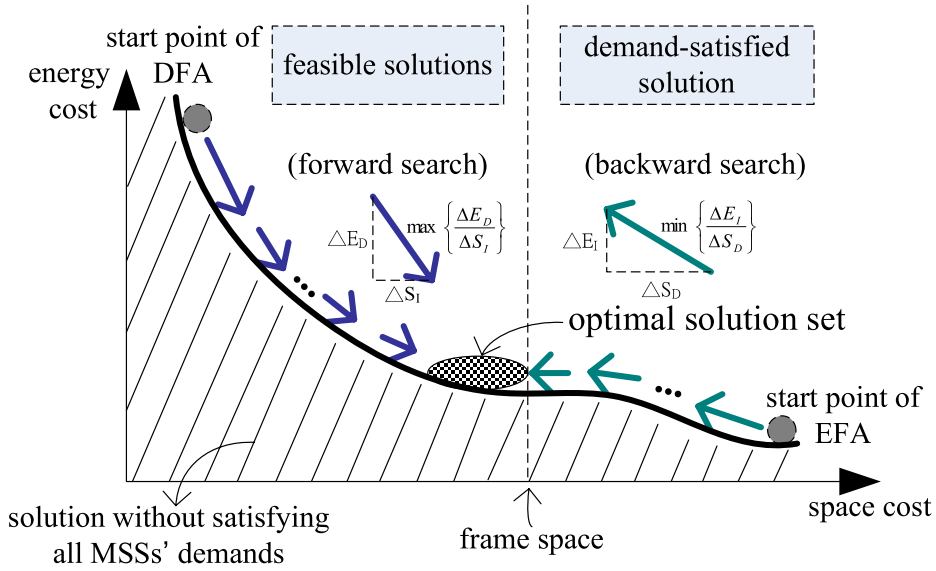


Figure 4.5: The concepts of forward and backward searches by the gradient-like method.

above such that the new solution is also feasible and the gradient of $\Delta E_D / \Delta S_I$ is maximum, where ΔE_D is the decrement of energy and ΔS_I is the increment of space usage after adjustment. The forward search is repeated until $\Delta E_D / \Delta S_I$ approximates to zero (that is, we cannot further reduce the energy consumption since $\Delta E_D \approx 0$). On the other hand, EFA first selects a demand-satisfied solution that allows MSSs to consume as less energy as possible to be its starting point. Then, it adopts a backward search to approximate the optimal solution. In each step of search, it tries to adjust the transmission schedule of one MSS such that the new schedule is also demand-satisfied while $\Delta E_I / \Delta S_D$ is minimum, where ΔE_I is the increment of energy and ΔS_D is the decrement of space usage after adjustment. The backward search is repeated until the solution becomes feasible.

Fig. 4.6 shows the flow charts of the two heuristics. In DFA, the first “Demand-First Path Assignment” phase tries to satisfy MSSs’ demands by selecting the best MCSs and paths and exploiting spatial reuse such that the use of frame space is minimized. However, the above process assumes that each MSS transmits at its largest power. So, the second “MCS, Path, and Transmission Group Adjustment” phase tries to reduce MSSs’ energy consumption by lowering their transmission rates and adjusting their paths and transmission groups. Each step of reduction is based on the gradient concept. Finally, the third “Burst Allocation and Region Assignment” phase determines the sizes of the MSS-BS, MSS-RS, and RS-BS regions and allocates uplink bursts for MSSs and RSs. On the other hand, EFA first relaxes the frame space constraint to find the initial solution with the minimum total energy consumption in its first

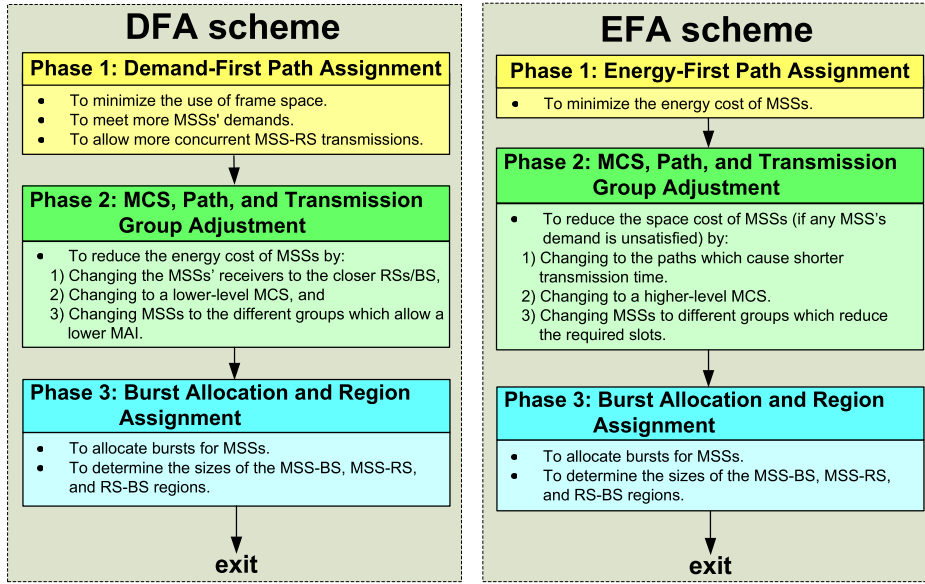


Figure 4.6: Flowcharts of our proposed heuristics.

“Energy-First Path Assignment” phase. In this phase, MSSs choose the closest RSs and the lowest MCSs without spatial reuse. The second “MCS, Path, and Transmission Group Adjustment” phase works based on the gradient concept to approach the optimum by raising MSSs’ energy consumption until packing all demands into the frame, i.e., reducing the required space by using more power. The third is the “Burst Allocation and Region Assignment” phase. Since the two schemes start from different initial solutions and apply different strategies, they have different limitations and thus lead to different performances. This will be clear later on.

4.3.2 Demand-First Allocation (DFA) Scheme

4.3.2.1 Phase 1 — Burst and Path Assignment

Assuming that the energy consumption of MSSs is not a concern, phase 1 has the following objectives: i) to minimize the use of frame space, ii) to meet more MSSs’ demands, and iii) to allow more concurrent MSS-RS transmissions. This phase helps choose each MSS_{*i*}’s initial path, transmission group, and MCS using the maximum power.

To exploit spatial reuse in the MSS-RS region, we model the *maximum allowable interference (MAI)* $\widehat{\mathcal{I}}_{(i,J(i))}^{K(i)}$ at relay RS_{*J(i)*} if MSS_{*i*} chooses RS_{*J(i)*} as its relay using MCS_{*K(i)*} with power P_i^{MAX} , $i = 1..n$, $J(i) = 0..m$, and $K(i) = 1..6$. Recall the $I(i, J(i))$ in Eq. (4.4), which stands for the current perceived interference for the transmission from MSS_{*i*} to RS_{*J(i)*}. With the relative distance between MSSs and BS/RSs, we can derive the path loss $L(i, J(i))$ of each MSS-BS/RS pair. From Eq. (4.4), each $\widehat{\mathcal{I}}_{(i,J(i))}^{K(i)}$ of an MSS_{*i*} transmitting to RS_{*J(i)*} using

$MCS_{K(i)}$ with P_i^{MAX} is

$$\widehat{\mathcal{I}}_{(i,J(i))}^{K(i)} = \frac{G_i \cdot G_{J(i)} \cdot P_i^{MAX}}{10^{\frac{\delta(MCS_{K(i)})}{10}} \cdot L(i, J(i))} - B \cdot N_0. \quad (4.8)$$

We should keep $\widehat{\mathcal{I}}_{(i,J(i))}^{K(i)} \geq I(i, J(i))$. Note that using a lower-level MCS can tolerate a higher interference, so $\widehat{\mathcal{I}}_{(i,J(i))}^{K(i)} < \widehat{\mathcal{I}}_{(i,J(i))}^{K(i)-1}$. Also note that for the BS, $\widehat{\mathcal{I}}_{(i,0)}^{K(i)} = 0$, since no concurrent transmission to the BS is allowed. For simplicity, we will pre-calculate all values of $\widehat{\mathcal{I}}_{(i,J(i))}^{K(i)}$ and maintain an MAI table using $(MSS_i, RS_{J(i)}, MCS_{K(i)})$ as the index.

Given the network topology \mathcal{G} , the path set \mathcal{R} , and MSS_i 's demand $d_i, i = 1..n$, phase 1 starts from a set \mathcal{T} of n empty transmission groups and greedily adds more transmission schedules to \mathcal{T} , until all frame space is exhausted or all MSSs are satisfied. Each transmission schedule has the format $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i)$, which means that MSS_i is scheduled to send its data to $RS_{J(i)}$ using $MCS_{K(i)}$ at power P_i . In case that $J(i) \neq 0$, it is implied that $RS_{J(i)}$ will relay MSS_i 's data to the BS using the best possible MCS level. Note that in this phase, P_i is always equal to P_i^{MAX} .

- Step 1) Set all MSSs as *unsatisfied*. Set the initial value of \mathcal{T} to be $\{\phi, \phi, \dots, \phi\}$ (i.e., with n empty sets) and set $F = h \times w$ as the initial amount of free slots.
- Step 2) Consider each unsatisfied MSS_i . If we adding the path from MSS_i to $RS_{J(i)}$ using $MCS_{K(i)}$ to the transmission group $\tau_\ell \in \mathcal{T}$ at power P_i^{MAX} (that is, adding $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i^{MAX})$ to group τ_ℓ), the extra number of slots required will be

$$S_{ex}(s_i, \tau_\ell) = \begin{cases} \left\lceil \frac{d_i}{rate(MCS_{K(i)})} \right\rceil, & \text{if } J(i) = 0, \tau_\ell = \phi \\ \infty, & \text{if } J(i) = 0, \tau_\ell \neq \phi \\ \max \left\{ \left\lceil \frac{d_i}{rate(MCS_{K(i)})} \right\rceil - S_g(\tau_\ell), 0 \right\} & \text{if } inf(s_i, \tau_\ell) = TRUE, J(i) \neq 0 \\ \left\lceil \frac{d_i}{rate(MCS_{\widehat{K}(i)})} \right\rceil, & \text{if } inf(s_i, \tau_\ell) = FALSE, J(i) \neq 0, \end{cases} \quad (4.9)$$

where $S_g(\tau_\ell)$ is the number of slots required by τ_ℓ in the MSS-RS region, $inf(s_i, \tau_\ell)$ is a function to determine if adding $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i^{MAX})$ to τ_ℓ is interference-free, and $MCS_{\widehat{K}(i)}$ is the best feasible MCS from $RS_{J(i)}$ to the BS. In the first case of $J(i) = 0$, it is the cost to the MSS-BS region. In the second case, it means adding an MSS-BS transmission to a non-empty group is infeasible. In the third case, it is the extra cost to the MSS-RS region plus that to the RS-BS region. In the fourth case, it means adding this

path to τ_ℓ is infeasible. Function $inf(s_i, \tau_\ell)$ returns *TRUE* (i.e., interference-free) if and only if the following three conditions are *all* satisfied:

1. $RS_{J(i)}$ does not appear in τ_ℓ . That is, for each $s_a = (RS_{J(a)}, MCS_{K(a)}, P_a^{MAX}) \in \tau_\ell$, $J(a) \neq J(i)$.
2. $RS_{J(i)}$ can receive correctly considering all interferences. That is,

$$\sum_{\forall s_a=(RS_{J(a)}, MCS_{K(a)}, P_a^{MAX}) \in \tau_\ell} \tilde{P}(a, J(i)) \leq \hat{\mathcal{I}}_{(i, J(i))}^{K(i)}.$$

3. After adding the interference caused by MSS_i with $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i^{MAX})$, $RS_{J(a)}$ can still receive correctly. That is, for each $s_a = (RS_{J(a)}, MCS_{K(a)}, P_a^{MAX}) \in \tau_\ell$, $I(a, J(a)) + \tilde{P}(i, J(a)) \leq \hat{\mathcal{I}}_{(a, J(a))}^{K(a)}$.

After step 2), we have the extra cost to schedule each unsatisfied MSS_i for all combinations of $RS_{J(i)}$, $MCS_{K(i)}$, and τ_ℓ .

Step 3) From the extra costs of all unsatisfied $MSSs$, pick the one causing the least cost of $S_{ex}(s_i, \tau_\ell)$. If $S_{ex}(s_i, \tau_\ell) \leq F$, add $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i^{MAX})$ to τ_ℓ directly; otherwise, adjust the demand d_i of MSS_i proportionally to fit into F and add $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i^{MAX})$ to τ_ℓ . Then, update F by deducting the allocated resource and set MSS_i as *satisfied*. Also, update $I(a, J(a))$ for each satisfied MSS_a in τ_ℓ . Finally, update $S_{ex}(\cdot)$ s of all unsatisfied $MSSs$ ' schedules for τ_ℓ . Note that after step 3), one MSS will be satisfied.

Step 4) If there still is space in an uplink subframe and there still exists any unsatisfied MSS , go back to step 3); otherwise, go to the next phase.

4.3.2.2 Phase 2 — MCS, Path, and Group Adjustment

Phase 1 aims at reducing the use of frame space, but the maximum powers have been used by all $MSSs$. This phase tries to make some adjustments and lower their energy costs by taking advantage of the extra free frame space F . We try three possibilities to reduce an MSS 's energy:

i) Change its receiver to a closer RS/BS. ii) Change to a lower-level MCS. iii) Change to a different transmission group with a different MCS and receiver. In particular, for possibility ii),

recall that the energy cost of MSS_i can be written as $E_i = T_i \times P_i = \left\lceil \frac{d_i}{rate(MCS_{K(i)})} \right\rceil \times P_i$. By

Table 4.2: Energy costs per bit for different MCSs.

level k	energy cost (mW)
1	0.082β
2	0.098β
3	0.147β
4	0.219β
5	0.413β
6	0.582β

ignoring the ceiling function and assuming a fixed interference level of $B \cdot N_o + I(i, J(i))$, the energy cost per bit to reach the SINR in Table 4.1 can be written as

$$E_i = \frac{1}{\text{rate}(MCS_{K(i)})} \times (10^{\frac{\delta(MCS_{K(i)})}{10}} \beta), \quad (4.10)$$

where $\beta = \frac{(B \cdot N_o + I(i, J(i))) \cdot L(i, J(i))}{G_i \cdot G_{J(i)}} > 0$. In Table 4.2, we do see that the energy cost per bit decreases as the MCS level decreases.

Given the current set \mathcal{T} and the remaining free resource F from phase 1, phase 2 works as follows:

Step 1) For each $\tau_\ell \in \mathcal{T}$, consider each transmission schedule $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i) \in \tau_\ell$.

There are three possibilities for MSS_i to reduce its energy: a) Change its MCS and power. b) Change its relay and power. c) Change its group, relay, MCS, and power. For each s_i , we may find multiple combinations of $s'_i = (RS_{J'(i)}, MCS_{K'(i)}, P'_i)$ and $\tau_{\ell'}$ such that s'_i is the new transmission schedule for MSS_i and $\tau_{\ell'}$ is the transmission group to accommodate s'_i (which may or may not be equal to τ_ℓ).

To find all feasible s'_i and $\tau_{\ell'}$, let us consider the above three cases. In case a), since $RS_{J(i)}$ is unchanged, we can simply try different $MCS_{K'(i)}$ and then use Eq. (4.4) based on the existing interference $I(i, J(i))$ perceived by $RS_{J(i)}$ to compute the best power P'_i . With this new power P'_i , we also need to check if this would exceed the tolerable interference of any other RS in τ_ℓ . If so, this transmission schedule is not feasible. In case b), since τ_ℓ is unchanged, we try other unused RSs in τ_ℓ and follow the procedure in case a) to find appropriate MCSs and power. Similarly, we need to check if this would excess interference to existing RSs. In case c), we will try to delete s_i from τ_ℓ and add MSS_i 's demand to other $\tau_{\ell'}$. For each $\tau_{\ell'}$, the same procedure in case b) can be used to identify all possible s'_i .

Note that after step 1), we have all new feasible s'_i and $\tau_{\ell'}$ for MSS_i .

Step 2) For each $(s'_i, \tau_{\ell'})$ pair, we calculate the saving of energy and the cost of extra slots for MSS_i to make this change. The saving of energy is written as

$$\Delta_E((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'})) = \left(\left\lceil \frac{d_i}{\text{rate}(MCS_{K(i)})} \right\rceil \times P_i \right) - \left(\left\lceil \frac{d_i}{\text{rate}(MCS_{K'(i)})} \right\rceil \times P'_i \right). \quad (4.11)$$

Then, the cost of extra slots is derived as

$$\Delta_S((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'})) = \begin{cases} (S_{tot}(\tau_{\ell} - \{s_i\}) + S_{tot}(\tau_{\ell'} \cup \{s'_i\})) \\ - (S_{tot}(\tau_{\ell}) + S_{tot}(\tau_{\ell'})), & \text{if } \tau_{\ell} \neq \tau_{\ell'} \\ S_{tot}(\tau_{\ell'} - \{s_i\} \cup \{s'_i\}) - S_{tot}(\tau_{\ell}), & \text{if } \tau_{\ell} = \tau_{\ell'} \end{cases} \quad (4.12)$$

Note that $\Delta_S((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))$ should not exceed the available resource F and the saving $\Delta_E((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))$ should be positive. Otherwise, this pair $(s'_i, \tau_{\ell'})$ is infeasible and should not be considered.

Step 3) From all feasible pairs $(s'_i, \tau_{\ell'})$, we use the *energy-per-extra-slot ratio*

$$\frac{\Delta_E((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))}{\Delta_S((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))}$$

as the metric (this is recognized as the “gradient” in our scheme). The $(s'_i, \tau_{\ell'})$ pair with the largest ratio is selected (this represents the “steepest gradient” in the energy cost). Then, we remove s_i from τ_{ℓ} , add s'_i to $\tau_{\ell'}$, deduct $\Delta_S((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))$ from F , and update all interference levels of all RSs in τ_{ℓ} and $\tau_{\ell'}$. Then, we calculate $\Delta_E(\cdot)$ and $\Delta_S(\cdot)$ for each schedule s_a in τ_{ℓ} and each schedule s_b in $\tau_{\ell'}$. If any change in τ_{ℓ} and $\tau_{\ell'}$ is done, go to step 3); otherwise, go to the next phase.

We make some remarks below. First, updating an MSS’s power level is possible even if no extra slots are needed. The reason is that when an MSS lowers its power, other RSs may experience lower interference levels, making it possible for other MSSs to meet the required SINRs using lower power. From our experience, such a positive cycle would repeatedly benefit lots of MSSs. Second, the above process will eventually terminate. To speed up our algorithm, we can set a threshold ∂ on Δ_E or on the number of iterations.

4.3.2.3 Phase 3 — Burst Allocation and Region Assignment

After phase 2, all MSSs’ paths, MCSs, power, and transmission groups are determined. This phase will allocate bursts for MSSs and determine the sizes of the MSS-BS, MSS-RS, and RS-BS regions accordingly.

Given the current set \mathcal{T} from Phase 2, Phase 3 works as follows.

Step 1) Let $R_{MSS-BS}(\mathcal{T})$, $R_{MSS-RS}(\mathcal{T})$, and $R_{RS-BS}(\mathcal{T})$ be the sizes of the MSS-BS, MSS-RS, and RS-BS regions, respectively. Calculate them as follows:

$$\begin{aligned}
R_{MSS-BS}(\mathcal{T}) &= \sum_{\forall \tau_\ell \in \mathcal{T}} \sum_{\forall s_i = (RS_{J(i)}, MCS_{K(i)}, P_i) \in \tau_\ell : J(i) = 0} \left\lceil \frac{d_i}{rate(MCS_{K(i)})} \right\rceil \\
R_{MSS-RS}(\mathcal{T}) &= \sum_{\forall \tau_\ell \in \mathcal{T} : s_i = (RS_{J(i)}, MCS_{K(i)}, P_i) \in \tau_\ell, J(i) \neq 0} S_g(\tau_\ell) \\
R_{RS-BS}(\mathcal{T}) &= \sum_{\forall \tau_\ell \in \mathcal{T}} \sum_{\forall s_i = (RS_{J(i)}, MCS_{K(i)}, P_i) \in \tau_\ell, J(i) \neq 0} \left\lceil \frac{d_i}{rate(MCS_{\hat{K}(i)})} \right\rceil. \quad (4.13)
\end{aligned}$$

Step 2) According to each schedule $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i)$ in $\tau_\ell \in \mathcal{T}$, allocate MSS_i the corresponding burst(s) to the MSS-BS, MSS-RS, and RS-BS regions accordingly.

To summarize, the DFA scheme finds its best solution by first calculating a temporal solution that can consume the minimum frame space and then iteratively refines the solution to reduce MSSs' energy consumption. The above refinement is repeated until either the frame space is exhausted or the total energy consumption is minimized. However, deriving the minimal space solution (in phase 1) takes a lot of time. In addition, phase 2 might face convergence problem because the value of energy-per-extra-slot ratio is usually difficult to converge since each MCS, path, and group adjustment in phase 2 may incur a chain reaction such that a large number of iterations will be required to reach its best solution. Therefore, we apply a threshold to limit the number of iterations in phase 2 to guarantee the convergence of DFA. In the next section, we will discuss how to address the convergence issue.

4.3.3 Energy-First Allocation (EFA) Scheme

To solve the problem in DFA, EFA makes the following improvements:

1. EFA first relaxes the constraint of frame space so that it can easily find a temporal solution which consumes the least energy as the starting point. This significantly reduces the computational complexity.
2. Unlike DFA that reduces the energy consumption (which is continuous) in phase 2, EFA tries to reduce the frame usage in a discrete manner (because the basic unit of the frame space is a slot). This not only alleviates the computation cost but also guarantees the convergence of EFA.

3. EFA adopts simultaneous equations to calculate the minimum transmission power of MSSs in each transmission group. This can help to further reduce the energy consumption of MSSs.

The EFA scheme starts with a trivial set \mathcal{T} of transmission groups where each group contains only one MSS with the closest RS/BS using the lowest MCS. It is thus a solution with the least energy cost. However, the total number of slots required may exceed the frame space. We then adjust these schedules by changing their power, MCSs, paths, and transmission groups based on gradient-like search, until they fit into one frame space.

Phase 1: For each MSS_{*i*}, we create a transmission schedule $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i)$ such that RS_{*J(i)*} is the closest to MSS_{*i*}, $MCS_{K(i)} = MCS_6$ (the lowest one), and P_i is the lowest power required to communicate with RS_{*J(i)*}. Then, we let each s_i be in one transmission group by setting $\tau_i = \{s_i\}, i = 1..n$. Let L be the total required slots of \mathcal{T} . Initially, $L = R_{MSS-BS}(\mathcal{T}) + R_{MSS-RS}(\mathcal{T}) + R_{RS-BS}(\mathcal{T})$. Finally, check whether $L \leq w \times h$. If yes, go to phase 3. Otherwise, go to phase 2 to reduce the space cost for possibilities.

Phase 2: For each $\tau_\ell \in \mathcal{T}$, consider the transmission schedule $s_i = (RS_{J(i)}, MCS_{K(i)}, P_i) \in \tau_\ell$. There are three possibilities for MSS_{*i*} to reduce the space cost. a) Within the same group τ_ℓ , MSS_{*i*} can still transmit to RS_{*J(i)*} but using a higher MCS. b) Within the same group τ_ℓ , MSS_{*i*} can still use MCS_{*K(i)*} but changing its relay. (Note that the best feasible MCS for each RS to the BS may be different so that the space cost will be also different). c) MSS_{*i*} switches to another group and then selects proper MCS and relay. For each possibility, we use s'_i as the new schedule for MSS_{*i*} and $\tau_{\ell'}$ as the new group accommodating s'_i .

Step 1) To find all feasible s'_i and $\tau_{\ell'}$, we consider the above possibilities a), b), and c). Unlike DFA, EFA tries to further reduce energy by optimizing the transmission power of multiple MSSs in the transmission group $\tau_{\ell'}$ when s'_i joins it. Therefore, we propose using simultaneous equations to derive the minimum required power of all MSSs in group $\tau_{\ell'}$. Suppose that if adding s'_i to $\tau_{\ell'}$, we have a set of schedules $\{s_a = (RS_{J(a)}, MCS_{K(a)}, P_a)\}$ in $\tau_{\ell'}$, $|\tau_{\ell'}| = z$, where $J(a)$ and $K(a)$ are the indexes of the RS and MCS used by MSS_{*a*}. Let P_a be the power of MSS_{*a*}, $0 \leq P_a \leq P_a^{MAX}$. It follows that the SINR perceived by RS_{*J(a)*} should be over $\delta(MCS_{K(a)})$, i.e.,

$$SINR(a, J(a)) = 10 \cdot \log_{10} \left(\frac{\tilde{P}(a, J(a))}{B \cdot N_o + I(a, J(a))} \right) \geq \delta(MCS_{K(a)}). \quad (4.14)$$

To minimize the power, we make the equal mark (i.e., “=”) hold. Thus, we have

$$\frac{\frac{G_a \cdot G_{J(a)} \cdot P_a}{L(a, J(a))}}{B \cdot N_o + \sum_{s_{a'} \in \tau_{\ell'}, a' \neq a} \frac{G_{a'} \cdot G_{J(a')} \cdot P_{a'}}{L(a', J(a'))}} = 10^{\frac{\delta(MCS_{K(a)})}{10}}. \quad (4.15)$$

Since the right-hand side is a constant, Eq. (4.15) can be converted into a simultaneous equations for each $p_a, s_a \in \tau_{\ell'}$. Repeating this for each $MSS_{s_a}, s_a \in \tau_{\ell'}$, we obtain z equalities. Then, by solving these equalities, we can find the best power P_a for each MSS_{s_a} in $\tau_{\ell'}$ in polynomial time and check whether they are feasible for concurrent transmissions by $P_a \leq P_a^{MAX}$.

After step 1), we have all new feasible s'_i and $\tau_{\ell'}$ for MSS_i .

Step 2) For each $(s'_i, \tau_{\ell'})$ pair, we calculate the cost of extra consumed energy and saving of slots for MSS_i to make this change. Given any transmission group τ , let $E_g(\tau)$ be the summation of energy consumed by all transmission schedule $s_a = (RS_{J(a)}, MCS_{K(a)}, P_a)$ in τ , which can be defined as

$$E_g(\tau) = \sum_{s_a = (RS_{J(a)}, MCS_{K(a)}, P_a) \in \tau} \left[\frac{d_a}{rate(MCS_{K(a)})} \right] \times P_a, \quad (4.16)$$

Then, the cost of extra consumed energy is derived as

$$\Delta_{\Psi}((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'})) = \begin{cases} (E_g(\tau_{\ell} - \{s_i\}) + E_g(\tau_{\ell'} \cup \{s'_i\})) \\ - (E_g(\tau_{\ell}) + E_g(\tau_{\ell'})), & \text{if } \tau_{\ell} \neq \tau_{\ell'} \\ E_g(\tau_{\ell'} - \{s_i\} \cup \{s'_i\}) - E_g(\tau_{\ell}), & \text{if } \tau_{\ell} = \tau_{\ell'}. \end{cases} \quad (4.17)$$

The saving of slots is written as

$$\Delta_{\Omega}((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'})) = \begin{cases} (S_{tot}(\tau_{\ell}) + S_{tot}(\tau_{\ell'})) \\ - (S_{tot}(\tau_{\ell} - \{s_i\}) + S_{tot}(\tau_{\ell'} \cup \{s'_i\})), & \text{if } \tau_{\ell} \neq \tau_{\ell'} \\ S_{tot}(\tau_{\ell}) - S_{tot}(\tau_{\ell'} - \{s_i\} \cup \{s'_i\}), & \text{if } \tau_{\ell} = \tau_{\ell'}. \end{cases} \quad (4.18)$$

Note that $\Delta_{\Omega}((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))$ should be positive. Otherwise, this pair $(s'_i, \tau_{\ell'})$ provides no benefit and should not be considered.

Step 3) From all feasible pairs $(s'_i, \tau_{\ell'})$, we use the *slot-per-extra-energy ratio*

$$\frac{\Delta_{\Omega}((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))}{\Delta_{\Psi}((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))}$$

as the metric (this is recognized the as “gradient” in our scheme). The $(s'_i, \tau_{\ell'})$ pair with the largest ratio is selected (which represents the “steepest gradient” in space cost). Then, we remove s_i from τ_{ℓ} , add s'_i to $\tau_{\ell'}$, deduct $\Delta_{\Omega}((s_i, \tau_{\ell}), (s'_i, \tau_{\ell'}))$ from L . Then, we recalculate $\Delta_{\Psi}(\cdot)$ and $\Delta_{\Omega}(\cdot)$ for each schedule s_a in τ_{ℓ} and each schedule s_b in $\tau_{\ell'}$ accordingly.

Go back to step 3) if $L > w \times h$ and there is any change in τ_ℓ or τ_ℓ' ; otherwise, go to the next phase.

Note that it is possible that more than two schedules have the largest burst size in a group. By changing one of them, no space is saved. In this case, we can try to raise their MCSs by one level simultaneously to further reduce the space cost.

Phase 3: If the total number of required slots still exceeds the frame space, i.e., $L > w \times h$, we can shrink the sizes of some MSSs' bursts until the overall allocation can fit the frame space. Then, we adopt the phase 3 in DFA to allocate bursts and determine the sizes of the MSS-BS, MSS-RS, and RS-BS regions accordingly.

4.3.4 Analysis of Time Complexity

For DFA, phase 1 initially costs $O(n(m+1)6) = O(nm)$ to model the MAI for all MSSs transmitting to all possible RSs/BS with six MCSs, where $(m+1)$ means that there are m relay paths and one direct path to the BS. In step 1, it costs $O(n)$. In step 2, for each schedule, it costs $O(m)$ because it has at most m schedules in a transmission group to be verified whether adding the new schedule is interference-free. Then, since we may have at most $O(n(m+1)6)$ possible schedules and n possible transmission groups for all MSSs, the time complexity of step 2 is $O(m) \cdot O(n(m+1)6) \cdot n = O(n^2m^2)$. In step 3, it costs $O(n^2m^2)$ because it has at most $(n(m+1)6) \cdot n$ schedules to be picked and at most n schedules to be updated for their costs to that group (each costs $O(m^2)$). In step 4, it may go back to step 3 at most n times since there are n MSSs. Therefore, the time complexity of phase 1 in the DFA scheme costs $O(n) + O(n^2m^2) + n \cdot O(n^2m) = O(n^3m^2)$. For phase 2, step 1 and 2 cost $O(n^2m^2)$ because there are at most $(n(m+1)6)$ possible new schedules and n possible groups to be tried. Then, each schedule needs to verify whether they are interference-free (which costs $O(m)$). Thus, it can calculate the extra cost and conserved energy accordingly. In step 3, it costs $O(n^2m^2)$ because it has at most $(n(m+1)6) \cdot n$ schedules to be chosen and at most n schedules to be updated for their costs to those groups (each costs $O(m^2)$). Besides, it will go back to step 3 at most $\frac{P_i^{MAX} \cdot (w \cdot h)}{\partial}$ times, where $P_i^{MAX} \cdot (w \cdot h)$ is the maximum energy cost of an MSS and ∂ is a threshold on Δ_E . Since we have n MSSs, phase 2 costs $n \cdot \frac{P_i^{MAX} \cdot (w \cdot h)}{\partial} \cdot O(n^2m^2) = O(n^3m^2)$ if n and m are sufficiently large. For phase 3, it costs $O(n)$ to calculate the region sizes and to allocate at most $2n$ bursts if all MSSs use relays to transmit data. Therefore, the DFA scheme

Table 4.3: The parameters in our simulator.

parameter	value
channel bandwidth	10 MHz
FFT size	1024
zone category	PUSC with reuse 1
slot-time	200.94 μ s
uplink frame duration	2.5 ms
uplink subframe space	12 \times 30
MCS	Table 4.1
traffic	UGS, rtPS, nrtPS, and BE
demand d_i	Table 4.4
path loss model	two-ray ground
thermal noise	-100 dBm
P_i^{MAX}	1000 mW (milliwatt)
threshold ϑ	50

costs $O(n^3m^2) + O(n^3m^2) + O(n) = (n^3m^2)$.

For EFA, in phase 1, it costs $O(nm)$ to choose the closest RS and lowest MCS for each MSS. For phase 2, step 1 costs $O(n)$. In step 2, each schedule costs $O(m^3)$ to solve m simultaneous equations by the *Gaussian Elimination* because there are at most m transmission schedules in one transmission group. Since we may have at most $O(n(m+1)6)$ possible schedules and n possible transmission groups, the time complexity of step 2 is $O(n^2m^4)$. In step 3, it costs $O(n^2m^4)$ because it has no more than $(n(m+1)6) \cdot n$ schedules to be chosen and then takes $O(nm^4)$ to update. Besides, it will go back to step 1 at most $L - (w \cdot h)$ times, where L is the total number of slots required by the schedules in phase 1, which is proportional to the number of demands (i.e., n). Phase 3 costs $O(n)$. Therefore, EFA scheme costs $O(nm) + \{O(n) + O(n^2m^4) + [L - (w \cdot h)] \cdot O(n^2m^4)\} + O(n) = O(n^3m^4)$.

4.4 Performance Evaluation

In this section, we develop a simulator in Java to verify the effectiveness of our heuristics. The system parameters of our simulator are listed in Table 4.3. We consider four types of traffic: UGS, rtPS, nrtPS, and BE. Table 4.4 lists the parameters used to model these traffic. The network contains one BS and several RSs and MSSs. RSs are uniformly deployed inside the 2/3 coverage range of the BS to get the best performance gain [23, 25] and the number of RSs is ranged from 0 to 32. MSSs are randomly deployed inside the BS's coverage and the number of MSSs is ranged from 10 to 80. Each MSS may move inside the BS's coverage following the

Table 4.4: The traffic model used in our simulator.

Traffic class	traffic type	bandwidth (bytes/frame)		
		minimum	maximum	average
UGS	CBR	40 ~ 150	50 ~ 150	50 ~ 150
rtPS	VBR	50 ~ 100	100 ~ 150	75 ~ 125
	gaming	1.2	3	1.2
	VoIP	1.4	1.4	1.4
nrtPS	VBR	50 ~ 100	100 ~ 125	75 ~ 125
	FTP	4	10	7
	real trace	40	110	85
BE	VBR	0	0 ~ 150	0 ~ 75
	HTTP	0	7	3.6

random waypoint model with the maximal speed of 20 meters per second [8].

We compare our proposed DFA and EFA schemes against the *minimal-coloring (MC) scheme* [23, 25] and the *modified solution of MCK problem (sMCKP)* [73]. The MC scheme considers spatial reuse while the sMCKP scheme addresses the energy consumption of MSSs. Specifically, the MC scheme first selects a path with the minimum transmission time (by using the highest MCS level) for each MSS. Then, this scheme assigns one color for those MSS-RS communications that can coexist and tries to use the minimum number of colors. In this way, the spatial reuse can be realized. On the other hand, the sMCKP scheme calculates a benefit value of each MSS, which is defined by the ratio of the amount of energy reduction to the increase of burst size when the MSS changes from its current MCS level to another level. Then, sMCKP iteratively selects one MSS with the maximum benefit value and changes its MCS accordingly, until the maximum benefit is zero. However, sMCKP does not exploit RSs to help relay MSSs' data.

For the MC scheme and our heuristics, we use the terms “-SR” and “-NSR” to indicate whether or not they adopt spatial reuse. In our heuristics, we can set the MAI values as zeros for the DFA scheme and keep the schedules in original groups for the EFA scheme to realize *no* spatial reuse.

In addition, to further investigate the performance of our proposed schemes. We define two ideal performance boundaries in terms of *energy consumption lower bound (ELB)* and *demand satisfaction ratio upper bound (DUB)*. ELB assigns each MSS a schedule in a group containing only itself and chooses a closest RS/BS as its receiver using the lowest MCS without

consideration of frame space limitation. ELB is expressed as follows.

$$\sum_{i=1..n} \left\lceil \frac{d_i}{\text{rate}(MCS_6)} \right\rceil \times \frac{10^{\frac{\delta(MCS_6)}{10}} (B \cdot N_o + 0) \cdot L(i, j^*)}{G_i \cdot G_{j^*}}, \quad (4.19)$$

where $j^* = \arg \min_{J(i)=1..m} \{L(i, J(i))\}$. The right part of Eq. (4.19) is the transmission power derived from Eq. (4.4) by adopting the equal sign. On the other hand, DUB schedules each MSS to transmit to the BS if its required slots is less than that the MSS's RS required to transmit to the BS. In addition, we assume that each transmission group can accommodate the number of MSS-RS transmissions up to the number of RSs in the network, i.e, DUB considers the interference perceived at any RS as zero no matter there are concurrent MSS-RS transmissions or not (thus so called *ideal*). Hence, DUB can be expressed as $\min\{\frac{F}{L}, 1\}$, where L is the total required slots, defined by

$$L = \sum_{i \in \mathcal{I}} \left\lceil \frac{d_i}{\text{rate}(MCS_{K^B(i)})} \right\rceil + \sum_{i \notin \mathcal{I}} \frac{\left\lceil \frac{d_i}{\text{rate}(MCS_{K^R(i)})} \right\rceil}{m} + \sum_{i \notin \mathcal{I}} \left\lceil \frac{d_i}{\text{rate}(MCS_{\hat{K}(i)})} \right\rceil. \quad (4.20)$$

\mathcal{I} is a set of MSSs with the BS as its receiver, i.e.,

$$\mathcal{I} = \left\{ i \mid \left\lceil \frac{d_i}{\text{rate}(MCS_{K^B(i)})} \right\rceil < 0 + \left\lceil \frac{d_i}{\text{rate}(MCS_{\hat{K}(i)})} \right\rceil, i = 1..n \right\},$$

and $MCS_{K^B(i)}$ and $MCS_{K^R(i)}$ are the highest feasible MCSs of MSS_{*i*} transmitting to the BS and the RS, respectively. The first part of Eq. (4.20) is the number of required slots in MSS-BS region. The second part and the third part of Eq. (4.20) are the costs in MSS-RS and RS-BS regions, respectively. Now, let's explain why DUB takes the MSS-RS cost as the second part of Eq. (4.20). As we know, the number of required slots of an MSS-RS transmissions is determined by the largest burst size in all MSS-RS transmissions of the corresponding transmission group. Assume we have G non-empty transmission groups in the MSS-RS region, $\tau_\ell, \ell = 1..G$. Let $V_\ell, \ell = 1..G$, be the largest burst size in the ℓ th transmission group. It is known that the following equation is established,

$$\sum_{i \notin \mathcal{I}} \left\lceil \frac{d_i}{\text{rate}(MCS_{K^R(i)})} \right\rceil = \sum_{\ell=1..G} \sum_{s_i \in \tau_\ell} \left\lceil \frac{d_i}{\text{rate}(MCS_{K^R(i)})} \right\rceil \leq \sum_{\ell=1..G} |\tau_\ell| \cdot V_\ell.$$

From above equation, we can derive that

$$\sum_{\ell=1..G} V_\ell \geq \sum_{i \notin \mathcal{I}} \frac{\left\lceil \frac{d_i}{\text{rate}(MCS_{K^R(i)})} \right\rceil}{|\tau_\ell|} \geq \sum_{i \notin \mathcal{I}} \frac{\left\lceil \frac{d_i}{\text{rate}(MCS_{K^R(i)})} \right\rceil}{m}.$$

Hence, the second part of Eq. (4.20) is a lower bound of the number of required slots in the MSS-RS region.

4.4.1 Energy Consumption

We first evaluate the total energy consumption of MSSs per frame under different numbers of MSSs, as shown in Fig. 4.7. The number of RSs is 8 and the network is under the non-saturated condition. Note that the y-axis is drawn with exponential scales. Clearly, the energy consumption of MSSs under all schemes increases when the number of MSSs increases. The sMCKP scheme makes MSSs consume the most energy because it does not exploit RSs to reduce the transmission power of MSSs. For the case without spatial reuse, the proposed DFA-NSR and EFA-NSR schemes can save energy up to 72% and 80% of MSSs' energy, respectively, compared with the MC-NSR scheme. The reason is that the proposed schemes can determine better MCSs and closer RSs for MSSs to conserve energy. On the other hand, by allowing spatial reuse, the proposed DFA-SR and EFA-SR schemes can reduce unnecessary energy consumption of MSSs compared to the ones without spatial reuse. Although the MC-SR scheme adopts spatial reuse to allow concurrent transmissions, it does not change MSSs' paths or lower MCSs for energy conservation when the free resource remains. Thus, it outperforms the case without spatial reuse. In addition, we can observe that EFA-SR scheme saves more energy than the DFA scheme. This is because EFA scheme exploits the optimal power, deriving by the simultaneous equations, when conducting spatial reuse. Fig. 4.7 shows that the proposed DFA-SR and EFA-SR schemes can save up to 86% and 92% of MSSs' energy, respectively, compared with the MC-SR scheme. It is important to note that the performance of our EFA-SR scheme approximates to the energy consumption lower bound. Specifically, when the number of MSSs is 10, 20, 30, 40, and 50, the performance errors between the DFA-SR/EFA-SR schemes and the energy consumption lower bound are 0%/0%, 0%/0%, 22%/0.2%, 95%/11.0%, and 589%/39.0%, respectively.

We then measure the total energy consumption of MSSs under different numbers of RSs, as shown in Fig. 4.8. Note that the y-axis is drawn with exponential scales. Since the sMCKP scheme does not exploit RSs, its energy consumption is always the same. On the other hand, the energy consumption of the MC scheme and our heuristics decreases when the number of RSs increases because each MSS has more RSs to select to save its energy. Similarly, for the case without spatial reuse, the DFA-NSR and EFA-NSR schemes can save energy up to 77% and 85% of MSSs' energy, respectively, compared with the MC-NSR scheme. Furthermore, by allowing spatial reuse, the proposed DFA-SR and EFA-SR schemes outperform other schemes. From Fig. 4.8, the proposed DFA-SR and EFA-SR schemes can save up to 90% and 98% of

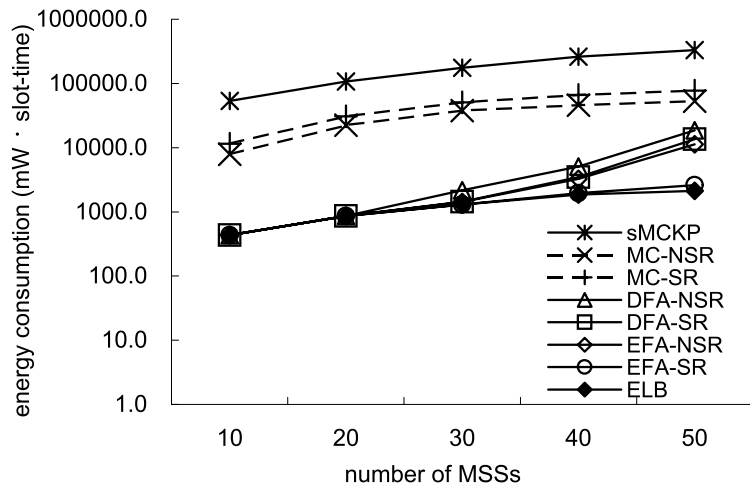


Figure 4.7: The energy consumption of MSSs under different numbers of MSSs, where there are 8 RSs.

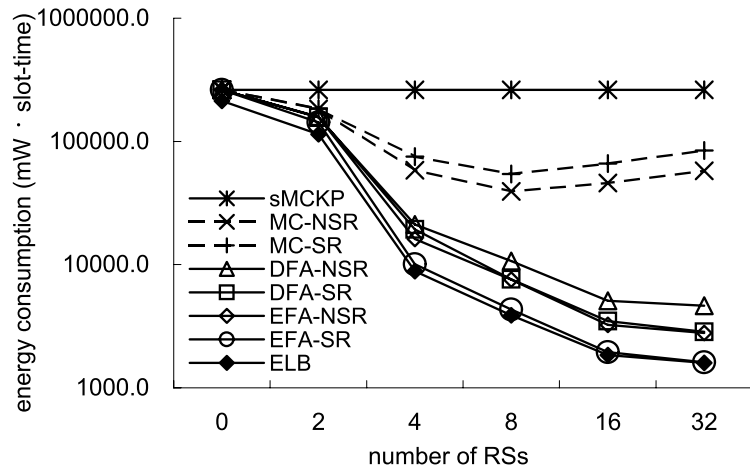


Figure 4.8: The energy consumption of MSSs under different numbers of RSs, where there are 50 MSSs.

MSSs' energy, respectively, compared with the MC-SR scheme. It is important to note that the performance of our EFA-SR scheme approximates to the energy consumption lower bound. Specifically, when the number of RSs is 0, 2, 4, 8, 16, and 32, the performance errors between the DFA-SR/EFA-SR schemes and the energy consumption lower bound are 21%/21%, 61%/56%, 490%/56%, 589%/39%, 539%/22%, and 485%/16%, respectively.

4.4.2 Satisfaction Ratio

Next, we investigate the *satisfaction ratio* of MSSs, which is defined by the ratio of the amount of *satisfied* demands to the total amount of demands per frame. When the satisfaction ratio is 1, it means that the scheme can satisfy all MSSs' demands. Fig. 4.9 shows the satisfaction ratios

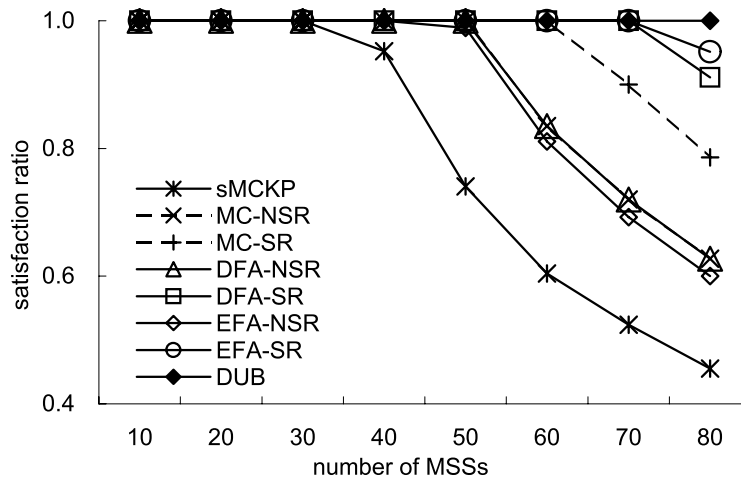


Figure 4.9: The satisfaction ratio of MSSs under different numbers of MSSs, where there are 32 RSs.

of all schemes under different numbers of MSSs, where the number of RSs is 32. When there are less than 30 MSSs, all schemes have a satisfaction ratio of 1 because the network is not saturated. The sMCKP scheme has the lowest satisfaction ratio when the number of MSSs is more than 30, because this scheme does not exploit RSs to improve network capacity. Without spatial reuse, the satisfaction ratios of the MC-NSR scheme and the proposed heuristics, DFA-NSR and EFA-NSR schemes, are similar. However, by exploiting spatial reuse, the proposed schemes always have higher satisfaction ratios than other schemes. The EFA-SR scheme performs the best because it can compactly overlap bursts to satisfy more MSSs' demands. It is important to note that the performance of our EFA-SR scheme approximates to the demand satisfaction ratio upper bound. Specifically, when the number of MSSs is 10, 20, 30, 40, and 50, the performance errors between the DFA-SR/EFA-SR schemes and the demand satisfaction ratio upper bound are 0%/0%, 0%/0%, 0%/0%, 0%/0%, and 6%/5%, respectively.

Fig. 4.10 shows the satisfaction ratios of all schemes under different numbers of RSs, where the number of MSSs is 70. Again, the satisfaction ratio of the sMCKP scheme is not affected by the number of RSs because it does not consider the existence of RSs. Without spatial reuse, our heuristics, the DFA-NSR and EFA-NSR schemes, perform similarly to the MC-NSR scheme. With spatial reuse, when the number of RSs is more than 8, increasing the number of RSs will decrease the satisfaction ratio of the MC-SR scheme. The reason is that the MC-SR scheme makes all MSSs transmit at their highest MCS levels. In this case, more interference may arise when there are more RSs. On the other hand, the proposed DFA-SR and EFA-SR schemes can better utilize RSs than the MC-SR scheme such that the satisfaction ratio increases when

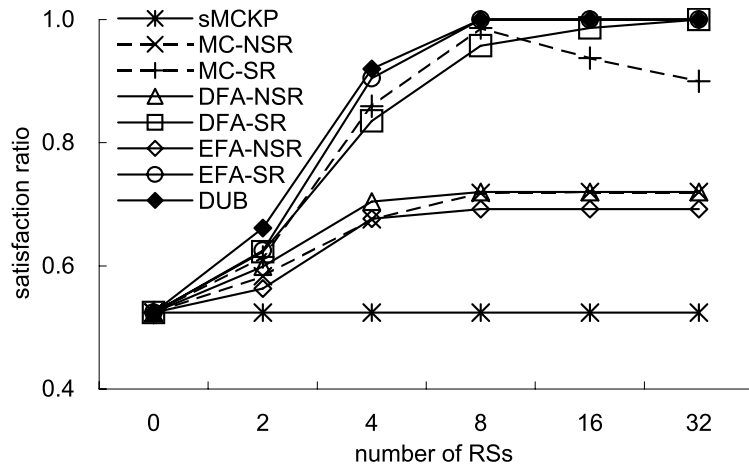
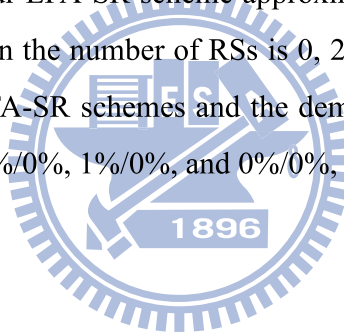


Figure 4.10: The satisfaction ratio of MSSs under different numbers of RSs, where there are 70 MSSs.

the number of RSs increases. Especially when the number of RSs is more than 4, the EFA-SR scheme always has a satisfaction ratio of 1. This is because it uses RSs to fully exploit spatial reuse and compactly overlap bursts to satisfy the demands of MSSs. It is important to note that the performance of our EFA-SR scheme approximates the demand satisfaction ratio upper bound. Specifically, when the number of RSs is 0, 2, 4, 8, 16, and 32, the performance errors between the DFA-SR/EFA-SR schemes and the demand satisfaction ratio upper bound are 0%/0%, 6%/5%, 9%/2%, 4%/0%, 1%/0%, and 0%/0%, respectively.



Chapter 5

A Regular Mini-Slot Allocation in WiMAX Mesh Networks

5.1 Motivations

The WiMAX mesh network (or shorten as ‘WMN’) has a BS connecting to multiple SSs via multi-hop transmissions. The transmission follows a *time division multiple access (TDMA)* mechanism over the underlying OFDM physical layer. The wireless resource on each link in WMNs is a sequence of mini-slots. However, before actually transmitting on a mini-slot, a sender must wait for a fixed number of mini-slots, called *transmission overhead*, to avoid collisions [15, 3, 72]. This is a guard time to synchronize and tolerate the air-propagation delay of the transmission occurring on the mini-slot right before the aforementioned overhead mini-slots. Once starting its actual transmission, a node may send on several consecutive mini-slots. References [72, 61] observe that if the (actual) transmission is too short, most of the time will be occupied by the transmission overhead. On the other hand, if the transmission is too long, it may hurt fairness and pipeline efficiency (i.e., there could be less concurrent transmissions) in the pipelines. So, a good scheduling should balance between the ratio of transmission overhead and the pipeline efficiency by adjusting the sizes of (actual) transmissions. In this work, we propose to use three metrics to evaluate a scheduling scheme (i) the total latency (i.e., the time to deliver all data to BS), (ii) the scheduling complexity, and (iii) the signaling overhead (i.e., the cost to notify all SSs their schedules).

In the literature, several works [71, 62, 12, 58, 19, 27, 32, 74] have studied the scheduling problem in WMNs. Reference [71] proposes an interference-aware, cross-layer scheduling scheme to increase the utilization of a WiMAX mesh network. Reference [62] suggests using concurrent transmissions to improve overall end-to-end throughput of a WMN. Reference [12]

shows how to increase spatial reuse to improve system throughput and provide fair access for subscriber stations. Reference [58] presents a flow-control scheduling to provide quality of service guarantee for flows. Reference [19] shows how to maximize spatial reuse to achieve better overall network throughput and higher spectral efficiency. Reference [27] proposes four criteria to select conflict-free links to reduce the scheduling length. These criteria include random selection, min-interference selection, nearest-to-BS selection, and farthest-to-BS selection. Reference [32] considers that each transmission can transmit one piece of data and tries to maximize spatial reuse to minimize the total transmission time. However, all above schemes do not consider the cost of *transmission overhead* (to be defined later on). For example, the results in [27, 32] are not optimal because they do not take this into account. Considering transmission overheads, [74] proposes to always find the maximal number of concurrent transmission links in each round. This problem has been shown to be NP-hard [31]. Although it performs close to optimum, its computational complexity is too high to be used by the BS. Also, the signaling overhead incurred by [74] is quite high because the scheduling patterns for SSs are not regular.

In this work, we focus on the most regular topologies, such as chain, grid, and triangle networks, which have been proved to have many applications, such as the WMNs deployed in rural areas in South Africa to provide Internet access [33], the VoIP testbed [5], and the mobility testbed developed in [34]. These topologies outperform random topologies in terms of their achievable network capacity, connectivity maintenance capability, and coverage-to-node ratios (about two times that of random topologies) [55, 54]. Here, the chain topology is a special case of grid topologies, which is the most suitable for long-thin areas, such as railways and highways [32].

5.2 Problem Definition

We are given one BS and n SSs, $SS_i, i = 1..n$. These BS and SSs are deployed in a chain, grid or triangle topologies, as shown in Fig. 5.1. The BS can be placed at any location in the topology. All nodes share the same communication channel. The amount of data that a node can transmit per mini-slot is d bytes. Since the topology is regular, two nodes are allowed to transmit concurrently if they are at least H hops away from each other. We consider the uplink scheduling. So we abstract the uplink mini-slots of the system by concatenating them together into an infinite sequence and ignore the downlink mini-slots. Each SS_i has a traffic demand of p_i bytes. Our goal is to construct a scheduling tree \mathcal{T} such that each SS_i receives a collision-free

schedule T_i and the total time to deliver all SSs' data to the BS is as less as possible. In our work, we impose that the schedule T_i for each SS_i should be periodical as defined below.

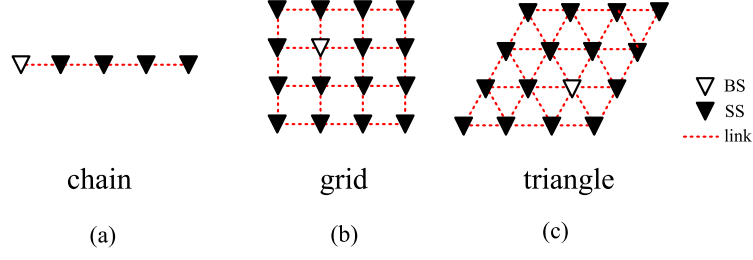


Figure 5.1: (a) A 5-node chain topology, (b) a 4×4 grid topology, and (c) a 4×4 triangle topology.

The transmission schedule is formulated as follows. For each SS_i and each mini-slot, we use a character in $\{0, 1, h\}$ to represent its state. A '0' means that the mini-slot is idle for SS_i . A '1' means that SS_i can transmit at most d bytes in this mini-slot. An 'h' means that SS_i is preparing to transmit (i.e., this mini-slot is considered a transmission overhead). To start an actual transmission, a SS must wait for α mini-slots of state 'h' so as to synchronize and tolerate the air-propagation time of the transmission occurring right before the overhead mini-slots, where α is a system-dependent constant. For example, when $\alpha = 2$, we can use a string '000hh111100' to indicate that a SS is idle in the first three mini-slots, waits for two overhead mini-slots, transmits for four mini-slots, and then stays idle in the last two mini-slots.

In this work, we enforce that all SSs' transmission schedules are periodical and regular. Specifically, all SSs' schedules have the same of period of ρ . Each SS's transmission schedule has the format of $(0^a h^\alpha 1^b 0^c)^*$, where $a \geq 0$, $b > 0$, and $c \geq 0$ are integers and $a + \alpha + b + c = \rho$. The 0s at the end of a schedule are necessary when we consider periodical schedule. Symbol '*' means a number of repetitions of the string in parentheses until all necessary data is delivered. Different SSs may have different patterns. For example, Fig. 5.2 shows a chain network with one BS and seven SSs. Only SS_7 has a traffic demand of $p_7 = 4$ bytes. Assuming $\alpha = 1$, $d = 1$, and $H = 3$, we show three schedules. In the first schedule, $b = 1$ mini-slot of data is transmitted in each cycle. The other parameters $a = 0/2/4$ and $c = 4/2/0$, respectively. So there are three types of schedule patterns: $(h10000)^*$, $(00h100)^*$, and $(0000h1)^*$. In the second schedule, $b = 2$ mini-slots in each cycle; $a = 0/3/6$ and $c = 6/3/0$, respectively. In the third schedule, $b = 4$; $a = 0/5/10/15/20/25/30$ and $c = 30/25/20/15/10/5/0$, respectively (however, only one cycle is needed). The second schedule is the most efficient. Our goal is to find the most efficient regular schedules.

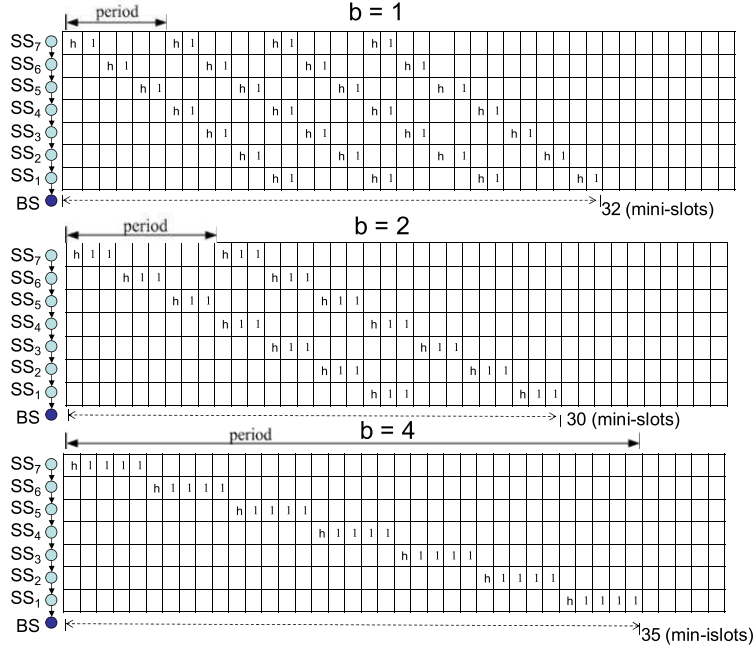


Figure 5.2: Transmission schedules for nodes in a chain network (idle state ‘0’ is omitted in the drawing).

5.3 Scheduling Tree Construction Schemes

Next, we present our scheduling schemes for chain and grid topologies. We first consider the chain topology with different locations of source SSs on the chain. Then we use these results as basic components to solve the scheduling problem for grid and triangle topologies. Given a grid/triangle network, we first construct a fishbone-like tree. The fishbone-like tree is decomposed into individual chains, each of which can be scheduled using the previous chain solutions. Below, we present three solutions for a chain, from simpler to more complicated cases. Then, we combine these solutions for the grid/triangle topologies.

5.3.1 A Chain with A Single Request

Since there are only one source and one destination, we can model the chain, without loss of generality, as a path $SS_n \rightarrow SS_{n-1} \rightarrow \dots \rightarrow SS_1 \rightarrow BS$ such that only SS_n has a non-zero demand p_n . To increase parallelism, we partition these SSs into k concurrent transmission-able groups, where $k = H$ if $n \geq H$ and $k = n$ otherwise (recall that H is the least spatial-reuse distance). Specifically, we define group $G_j, j = 0..k - 1$, as follows:

$$G_j = \{SS_i \mid (n - i) \bmod k = j, i = 1..n\}. \quad (5.1)$$

Nodes in the same group have the same schedule. We simply denote by T_j the transmission

schedule of G_j . Since we are interested in having regular schedules, we enforce each G_j to have a schedule of the format $(0^{a_j}h^{\alpha}1^b0^{c_j})^*$, where a_j and c_j are group-specific constants and b is a fixed constant for all groups, such that the following conditions hold: (i) $a_0 = 0$, (ii) $a_j + \alpha + b + c_j = \rho$ is a constant and ρ is the period for all groups, and (iii) $a_j + \alpha + b = a_{j+1}$, $j = 0..k - 2$. Condition (iii) means that each G_{j+1} is obtained from G_j by shifting the latter to the right by $(\alpha + b)$ positions. Given any b , we can compute the total latency $L_1(n, p_n, b)$ to deliver SS_n 's data to the BS:

$$L_1(n, p_n, b) = \begin{cases} \lceil \frac{p_n}{b \cdot d} \rceil \cdot H \cdot (\alpha + b) + (n - H) \cdot (\alpha + b), & \text{if } n \geq H \\ \lceil \frac{p_n}{b \cdot d} \rceil \cdot n \cdot (\alpha + b), & \text{otherwise.} \end{cases} \quad (5.2)$$

When $n \geq H$, each cycle has a length of $H \cdot (\alpha + b)$ mini-slots. It takes $\lceil \frac{p_n}{b \cdot d} \rceil$ cycles for SS_n to transmit its last piece of data. At the end of the $\lceil \frac{p_n}{b \cdot d} \rceil$ th cycle, the last piece of SS_n 's data will arrive at node SS_{n-H} . Then it takes another $(n - H)$ hops, each requiring $(\alpha + b)$ mini-slots, to travel to the BS. This gives the upper term in Eq. (5.2). For the lower term, the derivation is similar.

Given fixed n, p_n , and α , we are interested in knowing the optimal value of b , denoted by \hat{b} , that gives the minimum latency L_1 . To do so, we need to confine that p_n is divisible by $b \cdot d$ in Eq. (5.2). To minimize Eq. (5.2), we can let $L_1 = 0$ and take the first-order derivative of b . This leads to

$$\hat{b} = \begin{cases} \sqrt{\frac{\alpha \cdot p_n \cdot H}{d(n-H)}}, & \text{if } n \geq H \\ \frac{p_n}{d}, & \text{otherwise.} \end{cases} \quad (5.3)$$

The value of \hat{b} in Eq. (5.3) is a real. The best value may appear in $\lceil \hat{b} \rceil$ or $\lfloor \hat{b} \rfloor$. Plugging this into Eq. (5.2), we can get the minimum \hat{L}_1 .

5.3.2 A Chain with Multiple Requests

Next, we consider a path $SS_n \rightarrow SS_{n-1} \rightarrow \dots \rightarrow SS_1 \rightarrow BS$ with multiple non-zero-load nodes. Without loss of generality, we assume SS_n 's load is non-zero. Similar to Section 3.1, we divide SSs into k groups $G_j, j = 1..k - 1$. Again, we enforce G_j 's schedule with the format $(0^{a_j}h^{\alpha}1^b0^{c_j})^*$, where a_j and c_j are group-specific constants and b is a fixed constant for all groups, such that the following conditions hold: (i) $a_0 = 0$, (ii) $a_j + \alpha + b + c_j = \rho$ is a constant and ρ is the period for all groups, and (iii) $a_j + \alpha + b = a_{j+1}$, $j = 0..k - 2$. Condition (iii) means that each G_{j+1} is obtained from G_j by shifting the latter to the right by $(\alpha + b)$ positions. To find an appropriate value of b , we imagine that all data are originated from SS_n by assuming

that all SSs have zero loads except that SS_n has a load $p'_n = \sum_{i=1}^n p_i$. Then we plug p'_n into p_n in Eq. (5.3) to find the best \hat{b} .

With this \hat{b} , we need to find the latency $L_2(n, p_1, p_2, \dots, p_n, \hat{b})$ to deliver all SSs' data on the original path. The transmission is similar to a pipeline delivery, but with some bubbles sometimes. To model the pipeline behavior, we do not take a 'micro-view' on the system. Instead, we take a 'macro-view' to partition the path into $n' = \lceil \frac{n}{k} \rceil$ trains, by traversing from the end (i.e., SS_n) toward the head (i.e., SS_1) of the path by grouping, every consecutive k SSs are as one train (when n is not divisible by k , the last few SSs are grouped into one train). We make two observations on these trains.

Observation 1. *In each cycle, a train can deliver up to $b \cdot d$ bytes of data to the next train, no matter where these data are located in which SSs of the train.*

However, a bubble appears when a train does not have sufficient data to be delivered to the next train. Below, we show when bubbles will not appear.

Observation 2. *Except the first $n' = \lceil \frac{n}{k} \rceil$ cycles, the BS will continuously receive $b \cdot d$ bytes of data in every cycle until no more data exists in the path.*

Proof. Consider the first cycle, the data delivered by n' th train is only its data. However, in the second cycle, the data delivered by the n' th train will be the n' th chain's data or both the n' th and $(n' - 1)$ th chains' data. So, in the n' th cycle, the data delivered by n' th train will be the n' th chain's data \sim 1st chain's data. If the n' th chain's has no sufficient data (i.e., $< b \cdot d$), the delivery will combine the previous train's data, i.e., $(n' - 1)$ th chain's data or even $(n' - 2)$ th chain's data, etc. So, if the n' th train deliver less than $b \cdot d$ (so called bubble) to BS after n' th cycle, it means that the amount of data on all trains must be less than $b \cdot d$ after n' th cycle. Or it must deliver $b \cdot d$ data without bubbles. It's proved. \square

Observation 2 implies that if we can derive the network state at the end of the n' th cycle, the latency can be easily derived. To derive the network state after each cycle, let $S_i = (w_1^{(i)}, \dots, w_{n'}^{(i)})$ be the network state at the end of the i th cycle, $i = 0..n'$, where $w_j^{(i)}$ is the total load remaining in the j th train at the end of the i th cycle. Initially, $w_j^{(0)}$ is the total loads of those SSs in the j th train. Then we enter a recursive process to find S_{i+1} from S_i , $i = 0..n' - 1$ as follows:

$$w_j^{(i+1)} = \begin{cases} \max\{w_j^{(i)} - bd, 0\} + \min\{w_{j-1}^{(i)}, bd\}, & j = 2..n' \\ \max\{w_j^{(i)} - bd, 0\}, & j = 1. \end{cases} \quad (5.4)$$

Eq. (5.4) is derived based on observation 1. In the upper equality, the first term is the remaining load of the j th train after subtracting delivered data and the second term is the amount of data received from the previous train. The lower equality is delivered similarly.

According to observation 2, after the n' th cycle, the BS will see no bubble until all data on the path is empty and it will take $\lceil \frac{\sum_{j=1}^{n'} w_j^{(n')}}{b \cdot d} \rceil$ more cycles to deliver all remaining data. This leads to

$$L_2(n, p_1, p_2, \dots, p_n, \hat{b}) = (\lceil \frac{\sum_{j=1}^{n'} w_j^{(n')}}{b \cdot d} \rceil + n') \cdot \rho, \quad (5.5)$$

where $\rho = k \cdot (\alpha + b)$ is the period of cycles. As has been clear from the context, previous \hat{b} in Eq. (5.5) is just an estimation. The optimal b may appear at a point to the left of \hat{b}^1 . One may repeatedly decrease \hat{b} to find a better value.

5.3.3 A Chain with Multiple Requests and BS in the Middle

Since the BS in the middle, we model the chain as a path $SS_\ell \rightarrow SS_{\ell-1} \rightarrow \dots \rightarrow SS_1 \rightarrow BS \leftarrow SS'_1 \leftarrow SS'_2 \leftarrow \dots \leftarrow SS'_r$. Without loss of generality, we assume $\ell \geq r$ and both SS_ℓ and SS'_r have non-zero loads. For simplicity, we call $SS_\ell \rightarrow SS_{\ell-1} \rightarrow \dots \rightarrow SS_1 \rightarrow BS$ the left chain C_L , and $BS \leftarrow SS'_1 \leftarrow SS'_2 \leftarrow \dots \leftarrow SS'_r$ the right chain C_R . The arrangement of concurrent transmission-able groups is more difficult because we intend to transmit sufficient data to the BS from both chains without congestion in each cycle. First, we need to identify a new value for k (the number of groups):

$$k = \begin{cases} H, & \text{if } 1 \leq H \leq 4 \\ 2H - 4, & \text{if } H \geq 5. \end{cases} \quad (5.6)$$

Eq. (5.6) means that when $1 \leq H \leq 4$, we can still manage to have the most compact number of concurrent transmission-able groups. However, when $H \geq 5$, the number of groups will exceed H , which is not most compact. Fortunately, in practice, $H \leq 4$ in most cases.

Now, for $j = 0..k - 1$, we define group G_j as follows:

$$G_j = \begin{aligned} & \{SS_i \mid (\ell - i) \bmod k = j, i = 1..\ell\} \\ & \cup \{SS'_i \mid (\ell - i + \Delta) \bmod k = j, i = 1..r\} \end{aligned} \quad (5.7)$$

In Eq. (5.7), nodes in C_L and C_R are grouped sequentially similarly to the earlier cases. However, for C_R , the grouping of nodes is shifted by a value of Δ to allow concurrent transmissions, where $\Delta = 1$ if $H = 2$ and $\Delta = H - 2$ if $H \geq 3$. Fig. 5.3 shows some examples with $\ell = 6$

¹The current \hat{b} is the upper bound of the optimal value because we previously imagine that all data are originated from SS_n .

and $r = 6$. In the example of $H = 2$, we shift the grouping of nodes in C_R by a value of $\Delta = 1$. In the examples of $H = 3$ and $H = 4$, we shift the grouping of nodes in C_R by a value of $\Delta = H - 2$. Such shifting avoids nodes nearby the BS from colliding with each other. When $H \geq 5$, the value of k is defined differently. However, shifting by $\Delta = H - 2$ still helps avoid collision. Note that when $H = 5$ and 6, there are 6 and 8 groups, respectively, where these numbers of groups are least for grouping on both chains to transmit without congestion. We will explain later on.

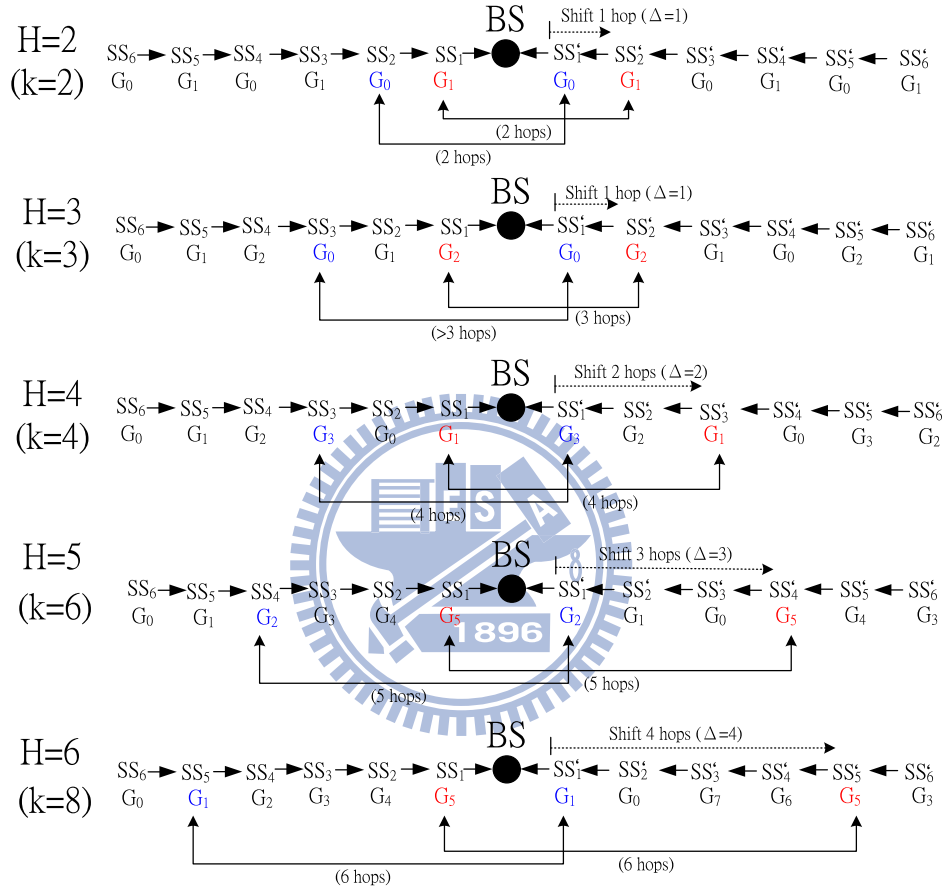


Figure 5.3: The arrangement of transmission-able groups for $H = 2, 3, 4, 5,$ and 6 when BS is in the middle.

When $H \leq 4$, the collision-free property of the above grouping can be proved by case-by-case validation. When $H \geq 5$, Fig. 5.4 gives a general proof. There are $k = 2H - 4$ groups. Consider nodes SS_i and SS'_i , $i = 1..2H - 4$ on C_L and C_R . Assume that the former i ($1 \leq i \leq 2H - 4$) SSs are grouped from G_0 and the remaining $2H - 4 - i$ SSs are grouped from G_i . We prove it in two aspects. First, since the indexes of groups on both two chains are increasing toward BS, if the pair of SS_x and SS_y , where they are in the same group but on different chains, have no interference to each other that will make all $SS_i, i \geq x$ and $SS'_i, i \geq y$

be interference free. By Eq. (5.7), all SSs on C_R are shifted by $H - 2$ of grouping sequence, the critical pairs (i.e., SS_1 and SS'_{H-1}) can be in same groups while keep a distance of H hops. That means those $SS_i, i \geq 1$ and $SS'_i, i \geq (H - 1)$ can be grouped successfully and will not cause any interference when those SSs are in their groups. On the other hand, as we know that SS'_1 is grouped by G_{i+H-3} . Since SS_{2H-4} is grouped in G_i , the smallest index of SS grouped by G_{i+H-3} will be SS_{H-1} , which has a distance of H hops to SS'_1 . By these two aspects, all $SS_i, i \geq (H - 1)$ and $SS'_i, i \geq 1$ will not cause any interference when those SSs are in their groups. Then, we can promise that each SS_i and each $SS'_i, i \geq 1$ can be grouped by Eq. (5.6) and Eq. (5.7) to transmit without any interference and congestion.

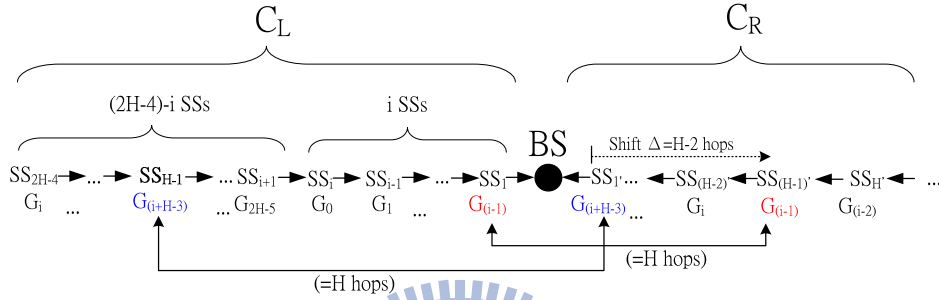


Figure 5.4: General collision-free proof for the cases of $H \geq 5$.

Theorem 3. By Eq. (5.6) and Eq. (5.7), SSs in the same group can transmit concurrently without collision for any value of H .

With theorem 1, we can allow C_L and C_R to transmit concurrently without collision. So the results in Section 3.2 can be applied here. (The only exception is that the first transmitting node in C_R , i.e., SS'_r , may not start from group G_0 . In this case, we can add some virtual nodes to C_R and make one start from G_0 .) For C_L , there will exist an optimal value of \hat{b}_ℓ such that $L_2(\ell, p_1, p_2, \dots, p_\ell, \hat{b}_\ell)$ is smallest. Similarly, for C_R , there will exist an optimal value of \hat{b}_r such that $L_2(r, p'_1, p'_2, \dots, p'_r, \hat{b}_r)$ is smallest. Plugging in any possible b , we can formulate the latency as follow:

$$L_3(\ell, p_1, p_2, \dots, p_\ell, r, p'_1, p'_2, \dots, p'_r, b) = \max \{L_2(\ell, p_1, p_2, \dots, p_\ell, b), L_2(r, p'_1, p'_2, \dots, p'_r, b)\} \quad (5.8)$$

The best value of b , called \hat{b} , may appear nearby or between \hat{b}_ℓ and \hat{b}_r .

5.3.4 A General Grid/Triangle Topology

Here we show how to extend our scheduling scheme to a grid/triangle topology. The scheduling is built on top of the previous chain scheduling results. First, we will construct a *fishbone-like*

tree from the grid/triangle network. The fishbone-like tree is further decomposed into horizontal and vertical chains. For example, Fig. 5.5(a) shows how such trees are formed. One of the chain passing the BS is called the *trunk chain*, and the others are called *branch chains*. Then, we schedule all branch chains to transmit their data to the trunk chain. Branch chains are divided into H groups and we schedule these groups to transmit sequentially. Finally, we schedule SSs in the trunk chain to transmit their data to the BS.

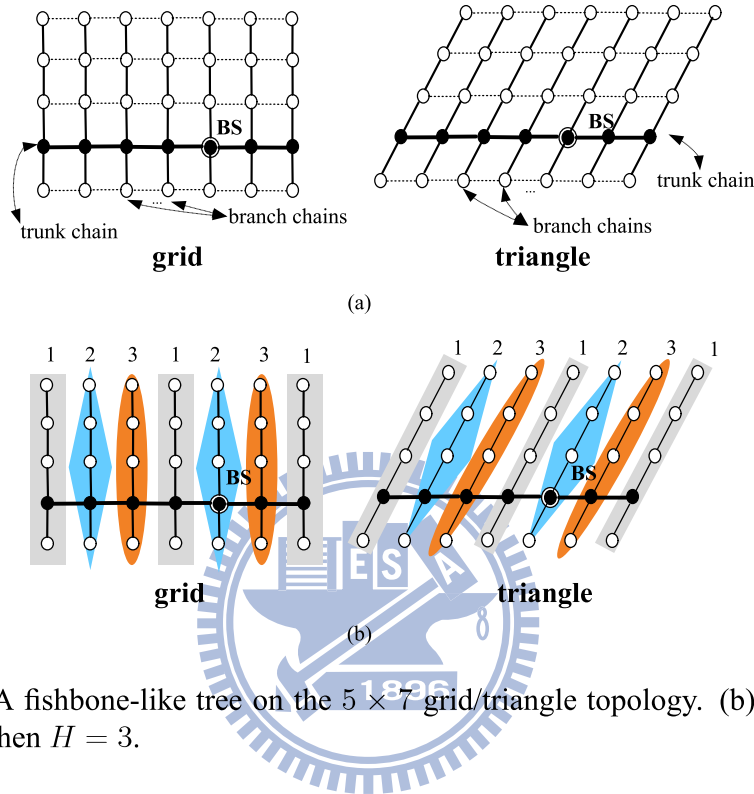


Figure 5.5: (a) A fishbone-like tree on the 5×7 grid/triangle topology. (b) The grouping of branch chains when $H = 3$.

Details of the scheme are as follows. We consider a $X \times Y$ grid/triangle topology. Without loss of generality, we assume $X \leq Y$ and we decompose the tree into Y vertical chains (branch chains) and one horizontal chain (trunk chain). Intuitively, the trunk chain is larger than the branch chains. There are two phases. In the first phase, branch chains are scheduled to transmit. These chains are divided into H groups. Since two parallel branch chains with a distance of H hops can transmit concurrently without interference, we assign a number between 1 to H to each branch chain in rotation from left to right. Chains marked by the same number are in the same group. Then we schedule each group of chains to transmit sequentially. For example, when $H = 3$, in Fig. 5.5(b), the seven branch chains are numbered by 1..3 in rotation. Then we let group 1 to transmit until all data are forwarded to the trunk chain, followed by group 2, and then group 3 in a similar way. Since chains in the same group can transmit individually without interference, we can apply the optimal \hat{b} for each chain as formulated above. The

latency of phase one is the sum of all groups' latencies. In the second phase, data are already all aggregated at the trunk chain. So, we can apply the easier result again to schedule nodes' transmissions on the trunk chain.

5.4 Performance Evaluation

In this section, we present our simulation results to verify the effectiveness of the proposed scheme. The simulator is written in JAVA language. Unless otherwise stated, the default parameters used in our simulation are $d = 1$ byte, $\alpha = 3$ mini-slots [15, 3], and $H = 3$ hops.

We compare our scheme against four schemes, named the basic IEEE 802.16d mesh operation [28], the **BGreedy** scheme [32], the **Max-transmission** scheme [74], and the **Priority-based** scheme [74]. The reason for selecting the **BGreedy** scheme for comparison is that it considers the pipeline efficiency, while that for selecting the **Max-transmission** scheme and the **Priority-based** scheme for comparison is that they consider the transmission overhead. The basic IEEE 802.16d mesh operation assigns the cumulated data plus a transmission overhead as the transmission for each SS without any spatial reuse. **BGreedy** scheme makes each transmission as short as possible to maximize pipeline efficiency. **Max-transmission** scheme always finds the maximal number of concurrent transmission links in the network round by round and assigns those links to transmit the minimal buffered data plus a transmission overhead. **Priority-based** scheme first finds all available links sets, which can transmit without interference, and chooses one with the maximal predefined priority. Then, it assigns those links to transmit the minimal buffered data plus a transmission overhead. Here, we adopt **LQR** priority, which performs the best performance in [74]. Such priority consults some network information, such as the hop counts, queue lengths, and transmission rates of the links. Then, except **BGreedy** scheme, we construct our fishbone-like tree for all other schemes because they do not discuss the routing tree construction in their works.

In the following results, we use the total latency to compare different schemes. We simulate three scenarios: a chain with a single request (SN1), a chain with multiple requests (SN2), and a grid with multiple requests (SN3). Unless otherwise stated, we use a 15-node chain and a 7×7 grid for the last two scenarios with BS in the middle. We remark that since the results of the triangle topology are almost the same as those in grid topology, we will only discuss the grid case.

5.4.1 Impact of Network Size

First, we investigate the effect of network size on the total latency (in mini-slots). Fig. 5.6(a) considers scenario SN1 with $p_n = 30$ bytes by varying n . Clearly, the total latencies of all schemes increase as n increases. **BGreedy**, **Max-transmission**, and **Priority-based** schemes perform the same because when the traffic demand is from only one SS, they schedule one transmission each time without any spatial reuse. Although **BGreedy** tries to maximize spatial reuse, its latency is still higher than ours because it disregards the transmission overhead. Ours has the best performance. This indicates the necessity of balancing between transmission overhead and pipeline efficiency. This effect is more significant when n is larger. Fig. 5.6 (b) and (c) shows our results for SN2 and SN3, respectively. Each SS has a randomly traffic demand from 0 to 20 bytes. Similar to SN1, the total latencies of all schemes increase as the network size increases. Although all schemes will exploit spatial reuse when there are multiple traffic demands, our scheme still outperforms all the other schemes. In addition, it is to be noted that the schedules generated by our scheme are regular and periodical, which is not so for other schemes.

5.4.2 Impact of Traffic Load

Next, we investigate the effect of average traffic load on total latency. Fig. 5.7(a) shows the results for SN1 when $n = 15$. Fig. 5.7(b) and (c) shows the results for SN2 and SN3, respectively, where each SS has a random traffic of 0 to 10, 0 to 20, and 0 to 40 bytes (thus the averages are 5, 10, and 20 bytes, respectively). The trends are similar to the previous cases; our scheme outperforms the other schemes significantly in SN1 and slightly in SN2 and SN3.

5.4.3 Impact of Transmission Overhead

Next, we investigate the impact of transmission overhead (α) on total latency. Fig. 5.8 shows our results. The simulation environment is similar to the previous cases except that we vary the value of α . Naturally, the total latencies of all schemes increase as α increases. In SN1, our significantly outperforms the other schemes in all values of α . In SN2 and SN3, the average traffic load of each station is 10 bytes. We see that a larger α will favor our scheme as compared to **Max-transmission** and **Priority-based** schemes. This is because our scheme enforces a regular schedule for each SS, thus losing some degree of pipeline efficiency. The impact is higher when $\alpha = 1$ and 2. When $\alpha \geq 3$, the transmission overhead is too high to be neglected. Thus,

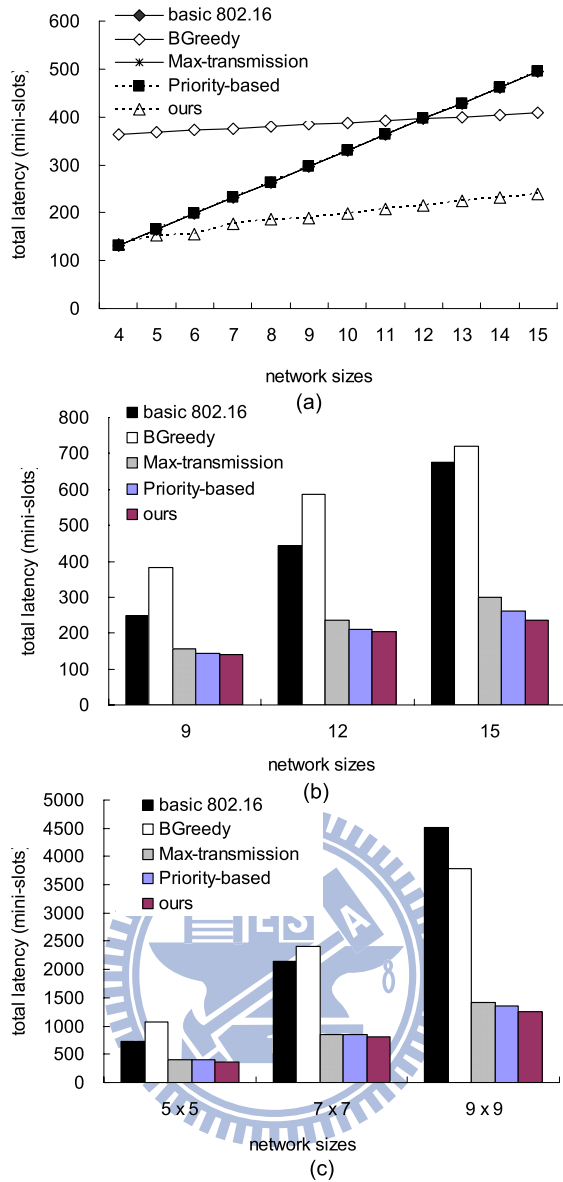


Figure 5.6: The impact of network size on total latency in scenarios SN1, SN2, and SN3.

balancing between transmission overhead and pipeline efficiency becomes quite important. In practice, α is greater than 2 [3].

5.4.4 Impact of BS Location

Next, we move the location of the BS around, as shown in Fig. 5.9. In SN2, we place the BS in the first, 4th, and 8th position of the chain ($n = 15$). In SN3, we place the BS at (1,1), (3,3), and (4,4) of the 7×7 grid network. In both cases, the total latencies of all schemes reduce as the BS is moved toward the center of the network because SSs are closer to the BS.

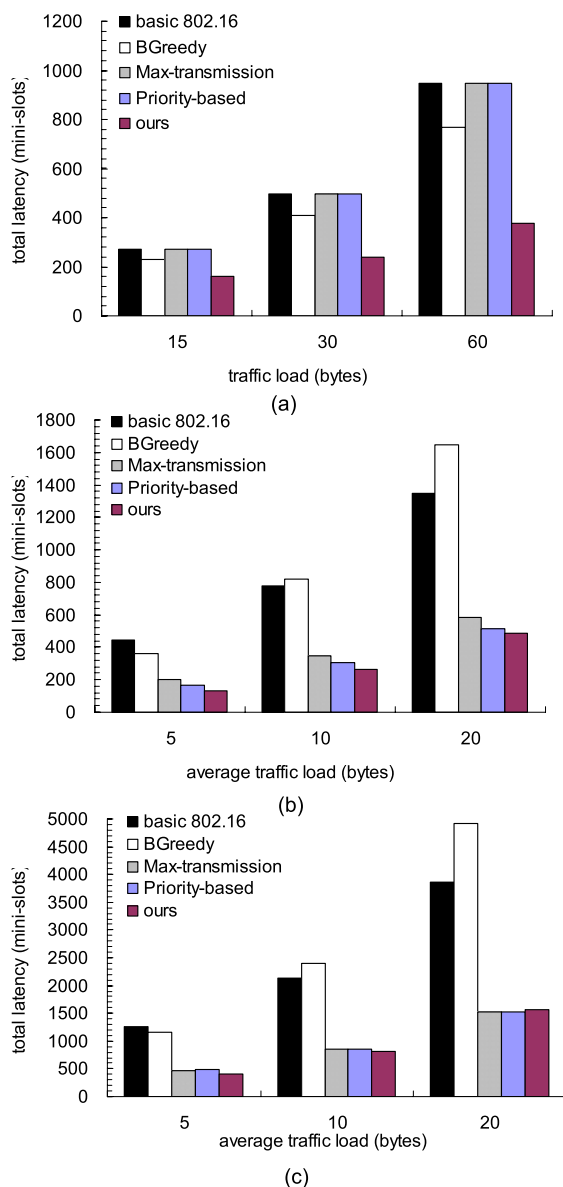


Figure 5.7: The impact of traffic load on total latency in scenarios SN1, SN2, and SN3.

5.4.5 Computational Complexity

Finally, we investigate the computational complexities of different schemes. We mainly compare our scheme against **Max-transmission** and **Priority-based** schemes. Note that it has been proved in [31] that the problem that **Max-transmission** and **Priority-based** schemes intend to solve is NP-hard. So we are interested in seeing the total CPU time incurred by these schemes as compared to ours. (The computation time is measured by the platform of IBM R61 with Intel Core 2 Duo T7300 2.0GHz and DDR2-800 SDRAM 2GB). From Fig. 5.10, we see that the computational complexities of all schemes increase as the network size increases. Since both **Max-transmission** and **Priority-based** schemes try to find out the maximal concurrent

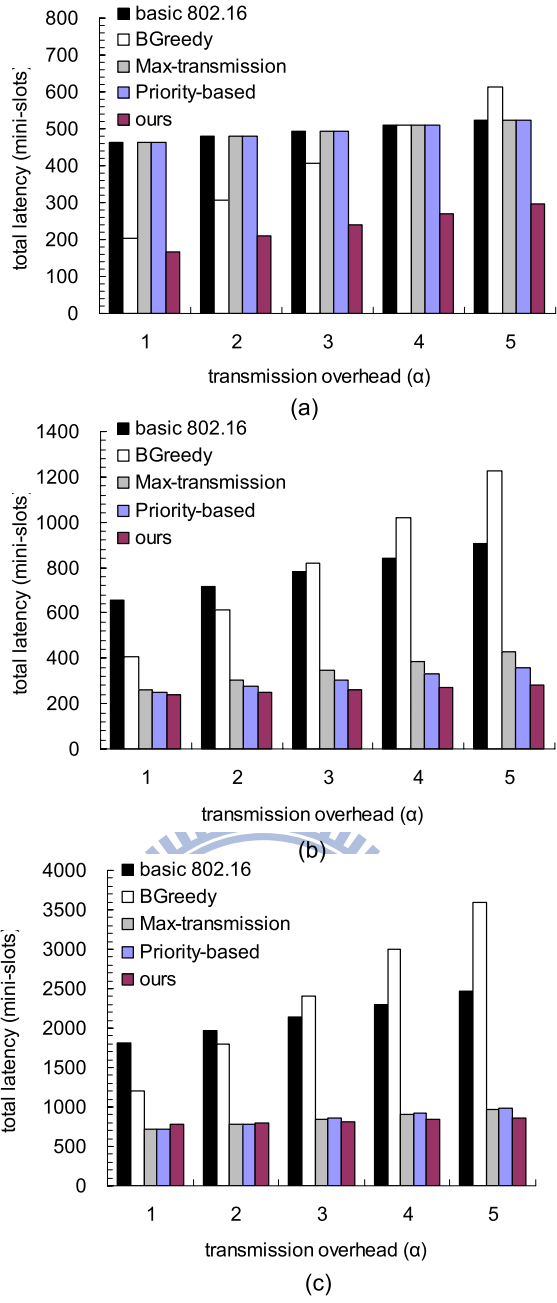


Figure 5.8: The impact of transmission overhead (α) on total latency in scenarios SN1, SN2, and SN3.

transmission set of Ss round by round until all data are delivered to BS, the processing time increases exponentially as n grows (note that the y-axis is drawn with exponential scales). For example, the computation costs of **Priority-based** are 4.03, 3.22, and 16.7 times of ours when $n = 4, 7,$ and $15,$ respectively, in SN2 and 96, 1039, and 6070 times of ours in the $5 \times 5,$ $7 \times 7,$ and 9×9 grid topologies, respectively. Because our scheme simplifies the grouping of Ss in both chain and grid topologies, and schedules transmissions in quite a regular way, it

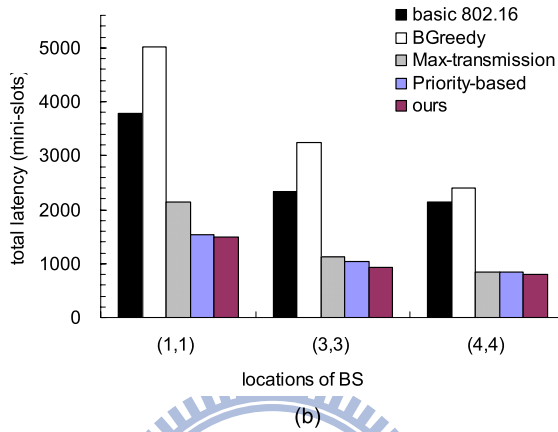
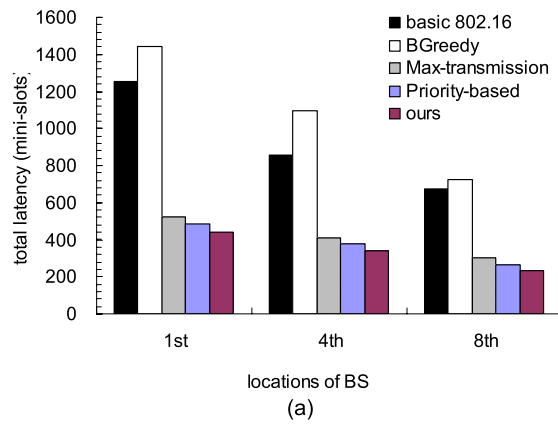


Figure 5.9: The impact of locations of BS on total latency in scenarios SN2 and SN3.

achieves a much lower cost. This further verifies that our scheme is more practical and is easier to implement, even when the network scales up.

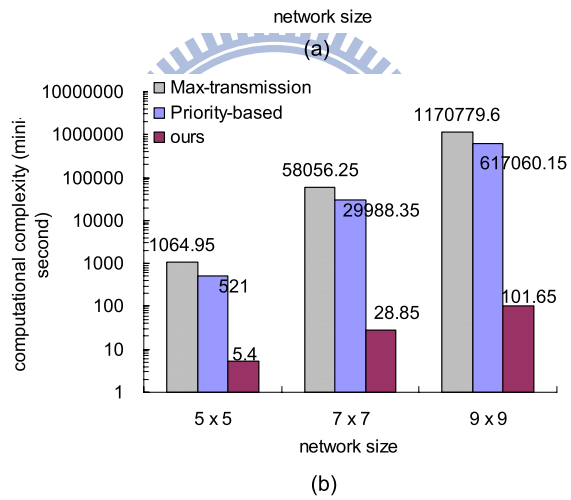
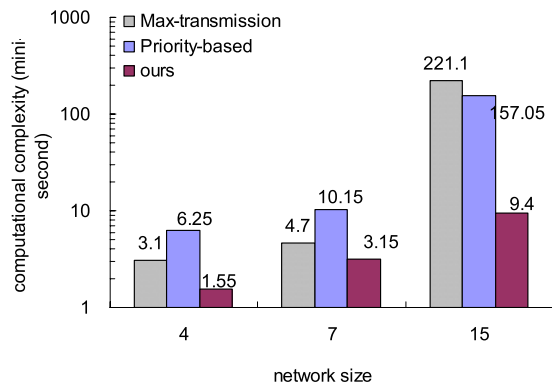


Figure 5.10: The impact of network size on computational complexity in scenarios SN2 and SN3.

Chapter 6

Conclusions and Future Directions

This dissertation contains three works for the resource and power allocation in WiMAX networks. The first work proposed a cross-layer approach considering traffic scheduling, burst allocation, and overhead reduction in WiMAX PMP networks. The second work proposed a power and bandwidth allocation considering MSSs' requirements and energy conservation in the WiMAX relay network. The third work proposed a regular mini-slot allocation considering the cost of transmission overhead and pipeline efficiency in WiMAX mesh networks.

In Chapter 3, we have proposed a cross-layer framework that covers the issues of overhead reduction, real-time and non-real-time traffic scheduling, and burst allocation in a WiMAX PMP network. Compared with existing solutions, our framework is more complete because it involves in co-designing of both the two-tier, priority-based scheduler and the bucket-based burst allocator. Our scheduler reduces potential IE overheads by adjusting the number of MSSs to be served. With a two-tier priority rule, it guarantees real-time traffic delays, ensures satisfaction ratios of non-real-time traffics, and maintains long-term fairness. On the other hand, our burst allocator incurs low complexity and guarantees a bounded number, $\left(k + \frac{Y}{\Delta_{bkt}} - 1\right)$, of IEs to accommodate data bursts. In addition, it follows the priority rule from the scheduler to avoid packet dropping of urgent real-time traffics. We have also analyzed the impact of the number of buckets on the throughput loss. Through both analyses and simulations, we show how to adjust the system parameters to reduce IE overheads, improve subframe utilization, and enhance network throughput. Besides, these results also verify that such a cross-layer framework significantly improves the resource allocation and utilization of downlink communications in WiMAX networks.

In Chapter 4, we have addressed the energy conservation issue in the uplink resource allocation problem of a WiMAX relay network. We show this problem to be NP-complete and

point out that using a higher MCS level and allowing more concurrent transmissions may harm an MSS in terms of its energy consumption. We have proposed two energy-efficient heuristics with different allocation strategies. The key idea is that we determine the better MCSs, paths, and transmission groups to adjust the use of the frame space and thus to satisfy more MSSs' demands while reduce their energy consumption. Simulation results have verified the effectiveness of our heuristics, where our heuristics can save more energy of MSSs while increasing their satisfaction ratios, as compared with existing schemes.

In Chapter 5, we have addressed the scheduling problem in chain- and grid-based WiMAX mesh networks. While most existing solutions try to address this NP-hard scheduling problem by searching for the sequence of concurrent transmission-able sets to maximize the spatial reuse factor, our approach tries to identify regular patterns that SSs can follow and repeatedly transmit easily. One special feature of our scheme is that it tries to balance transmission overhead and pipeline efficiency. In particular, our scheme tries to fill up the pipeline as full as possible to improve the pipeline efficiency. With these designs, our scheme does achieve better or equal total latency as compared to existing schemes, incurs much low computational cost as compared to existing schemes, and allows an easy implementation of the scheduler.

Based on the results presented above, several issues which can be discussed and improved are summarized as follows.

- To further improve the proposed schemes, such as the heuristics proposed for energy conservation in WiMAX relay networks, we may find their performance bounds, such as the energy consumption upper bound and the demand satisfaction ratio lower bound to improve the technique depth of the work.
- To make the addressed problems more general, we may take more features into account, such as the subchannel diversity of the OFDMA technique. Since the IEEE 802.16 supports the *adaptive modulation and coding (AMC)* mode to exploit the diverse channel qualities of subchannels, we may take this feature to accomplish the problems in both WiMAX PMP and relay networks.
- To make the proposed solutions be widely used, such as the regular mini-slot scheme proposed for WiMAX grid-based mesh networks, we may extend such solutions to a more general topology (e.g., the random topology).

Bibliography

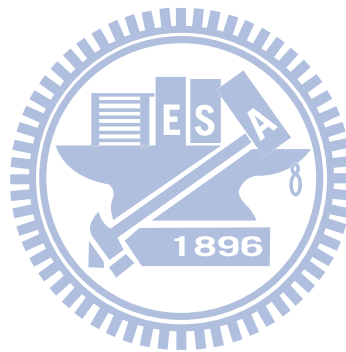
- [1] Channel models for fixed wireless applications.
- [2] Intel ships its next-generation WiMAX chip with support for mobile networks.
- [3] S. Ahson and M. Ilyas. *WiMAX: standards and security*. CRC Press, 2007.
- [4] N. A. Ali, M. Hayajneh, and H. Hassanein. Cross layer scheduling algorithm for IEEE 802.16 broadband wireless networks. *IEEE International Conference on Communications (ICC)*, pages 3858–3862, 2008.
- [5] N. Bayer, B. Xu, V. Rakocevic, and J. Habermann. Application-aware scheduling for VoIP in wireless mesh networks. *Computer Networks*, 54(2):257–277, 2010.
- [6] Y. Ben-Shimol, I. Kitroser, and Y. Dinitz. Two-dimensional mapping for wireless OFDMA systems. *IEEE Transactions on Broadcasting*, 52(3):388–396, 2006.
- [7] Y. Ben-Shimol, I. Kitroser, and Y. Dinitz. Two-dimensional mapping for wireless OFDMA systems. *IEEE Transactions on Broadcasting*, 52(3):388–396, 2006.
- [8] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, 2003.
- [9] J. Boyer, D. D. Falconer, and H. Yanikomeroglu. Multihop diversity in wireless relaying channels. *IEEE Transactions on Communications*, 52(10):1820–1830, 2004.
- [10] M. Bshara, U. Orguner, F. Gustafsson, and L. Van Biesen. Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on wimax networks. *IEEE Transactions on Vehicular Technology*, 59(1):283–294, 2010.
- [11] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000.
- [12] M. Cao, V. Raghunathan, and P. Kumar. A tractable algorithm for fair and efficient up-link scheduling of multi-hop WiMAX mesh networks. *IEEE Workshop on Wireless Mesh Networks*, pages 93–100, 2006.
- [13] C. Y. Chang, C. T. Chang, M. H. Li, and C. H. Chang. A novel relay placement mechanism for capacity enhancement in IEEE 802.16j WiMAX networks. *Proc. IEEE International Conference on Communications*, 2009.
- [14] D. M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Journal of Computer Networks and ISDN*, 17(1):1–14, 1989.
- [15] P. Djukic and S. Valaee. Delay aware link scheduling for multi-hop TDMA wireless networks. *IEEE/ACM Transactions on Networking (TON)*, 17(3):870–883, 2009.
- [16] C. Eklund, R. B. Marks, K. L. Stanwood, and S. Wang. IEEE standard 802.16: a technical overview of the wirelessMAN air interface for broadband wireless access. *IEEE Communications Magazine*, 40(6):97–107, 2002.

- [17] A. Erta, C. Cicconetti, and L. Lenzini. A downlink data region allocation algorithm for IEEE 802.16e OFDMA. *IEEE International Conference on Information, Communications & Signal Processing*, pages 1–5, 2007.
- [18] L. Erwu, W. Dongyao, L. Jimin, S. Gang, and J. Shan. Performance evaluation of bandwidth allocation in 802.16j mobile multi-hop relay networks. *Proc. IEEE Vehicular Technology Conference*, pages 939–943, 2007.
- [19] L. Fu, Z. Cao, and P. Fan. Spatial reuse in IEEE 802.16 based wireless mesh networks. *IEEE International Symposium on Communications and Information Technology*, pages 1358–1361, 2005.
- [20] L. Gao and S. Cui. Efficient subcarrier, power, and rate allocation with fairness consideration for OFDMA uplink. *IEEE Transactions on Wireless Communications*, 7(5):1507–1511, 2008.
- [21] Y. Ge, S. Wen, and Y. H. Ang. Analysis of optimal relay selection in IEEE 802.16 multihop relay networks. *Proc. IEEE Wireless Communications and Networking Conference*, 2009.
- [22] Y. Ge, S. Wen, Y. N. Ang, and Y. C. Liang. Optimal relay selection in IEEE 802.16j multihop relay vehicular networks. *IEEE Transactions on Vehicular Technology*, 2010.
- [23] V. Genc, S. Murphy, and J. Murphy. An interference-aware analytical model for performance analysis of transparent mode 802.16j systems. *IEEE Broadband Wireless Access Workshop (BWA) co-located with GLOBECOM*, pages 1–6, 2008.
- [24] V. Genc, S. Murphy, and J. Murphy. Performance analysis of transparent relays in 802.16j MMR networks. *Proc. International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 273–281, 2008.
- [25] V. Genc, S. Murphy, and J. Murphy. Analysis of transparent mode IEEE 802.16j system performance with varying numbers of relays and associated transmit power. *Proc. IEEE Wireless Communications and Networking Conference*, 2009.
- [26] V. Genc, S. Murphy, Y. Yang, and J. Murphy. IEEE 802.16j relay-based wireless access networks: an overview. *Proc. IEEE Wireless Communications*, 15(5):56–63, 2008.
- [27] B. Han, W. Jia, and L. Lin. Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks. *Computer Communications*, 30(4):782–792, 2007.
- [28] IEEE Standard 802.16-2004. IEEE standard for local and metropolitan area networks part 16: air interface for fixed broadband wireless access systems, 2004.
- [29] IEEE Standard 802.16e-2005. IEEE standard for local and metropolitan area networks part 16: air interface for fixed and mobile broadband wireless access systems amendment 2: physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1, 2006.
- [30] IEEE Standard 802.16j-2009. IEEE standard for local and metropolitan area networks part 16: air interface for broadband wireless access systems amendment 1: multiple relay specification, 2009.
- [31] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. *Wireless networks*, 11(4):471–487, 2005.
- [32] F. Jin, A. Arora, J. Hwan, and H. Choi. Routing and packet scheduling for throughput maximization in IEEE 802.16 mesh networks. In *Proceedings of IEEE Broadnets*, 2007.
- [33] D. Johnson. Evaluation of a single radio rural mesh network in South Africa. *International Conference on Information and Communication Technologies and Development*, pages 1–9, 2007.

- [34] D. Johnson and G. Hancke. Comparison of two routing metrics in OLSR on a grid based mesh network. *Ad Hoc Networks*, 7(2):374–387, 2009.
- [35] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer Verlag, 2004.
- [36] H. S. Kim and S. Yang. Tiny MAP: an efficient MAP in IEEE 802.16/WiMAX broadband wireless access systems. *Computer Communications*, 30(9):2122–2128, 2007.
- [37] J. Kim and D. H. Cho. Piggybacking scheme of MAP IE for minimizing MAC overhead in the IEEE 802.16e OFDMA systems. *IEEE Vehicular Technology Conference (VTC)*, pages 284–288, 2007.
- [38] J. Kim, E. Kim, and K. S. Kim. A new efficient BS scheduler and scheduling algorithm in WiBro systems. *IEEE International Conference on Advanced Communication Technology (ICACT)*, 3:1467–1470, 2006.
- [39] K. Kim, Y. Han, and S. L. Kim. Joint subcarrier and power allocation in uplink OFDMA systems. *IEEE Communications Letters*, 9(6):526–528, 2005.
- [40] W. H. Kuo. Recipient maximization routing scheme for multicast over IEEE 802.16j relay networks. *Proc. IEEE International Conference on Communications*, 2009.
- [41] T. Kwon, H. Lee, S. Choi, J. Kim, D. H. Cho, S. Cho, S. Yun, W. H. Park, and K. Kim. Design and implementation of simulator based on cross-layer protocol between MAC and PHY layers in WiBro compatible IEEE 802.16e OFDMA system. *IEEE Communications Magazine*, 43(12):136–146, 2005.
- [42] J. N. Laneman, D. N. C. Tse, and G. W. Wornell. Cooperative diversity in wireless networks: efficient protocols and outage behavior. *IEEE Transactions on Information Theory*, 50(12):3062–3080, 2004.
- [43] B. Li, Y. Qin, C. P. Low, and C. L. Gwee. A survey on mobile WiMAX. *IEEE Communications Magazine*, 45(12):70–75, 2007.
- [44] B. Lin, P. H. Ho, L. L. Xie, and X. Shen. Relay station placement in IEEE 802.16j dual-relay MMR networks. *Proc. IEEE International Conference on Communications*, pages 3437–3441, 2008.
- [45] H. C. Lu and W. Liao. Joint base station and relay station placement for IEEE 802.16j networks. *Proc. IEEE Global Telecommunications Conference*, 2009.
- [46] Y. Ma and D. Kim. Rate-maximization scheduling schemes for uplink OFDMA. *IEEE Transactions on Wireless Communications*, 8(6):3193–3205, 2009.
- [47] A. F. Molisch. *Wireless Communications*. Wiley, 2005.
- [48] D. Niyato, E. Hossain, D. Kim, and Z. Han. Relay-centric radio resource management and network planning in IEEE 802.16j mobile multihop relay networks. *IEEE Transactions on Wireless Communications*, 8(12):6115–6125, 2009.
- [49] D. Niyato, E. Hossain, D. I. Kim, and Z. Han. Joint optimization of placement and bandwidth reservation for relays in IEEE 802.16j mobile multihop networks. *Proc. IEEE International Conference on Communications*, 2009.
- [50] T. Ohseki, M. Morita, and T. Inoue. Burst construction and packet mapping scheme for OFDMA Downlinks in IEEE 802.16 systems. *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 4307–4311, 2007.
- [51] T. Ohseki, M. Morita, and T. Inoue. Burst construction and packet mapping scheme for OFDMA Downlinks in IEEE 802.16 systems. *IEEE Global Telecommunications Conference (GLOBECOM)*, pages 4307–4311, 2007.

- [52] X. Perez-Costa, P. Favaro, A. Zubow, D. Camps, and J. Arauz. On the challenges for the maximization of radio resources usage in WiMAX networks. *IEEE Consumer Communications and Networking Conference (CCNC)*, pages 890–896, 2008.
- [53] R. Pitic and A. Capone. An opportunistic scheduling scheme with minimum data-rate guarantees for OFDMA. *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1716–1721, 2008.
- [54] K. Ramachandran, I. Sheriff, E. Belding, and K. Almeroth. Routing stability in static wireless mesh networks. *Lecture Notes in Computer Science*, 44:73–82, 2007.
- [55] J. Robinson and E. Knightly. A performance study of deployment factors in wireless mesh networks. *International Conference on Computer Communications (INFOCOM)*, pages 2054–2062, 2007.
- [56] B. Rong, Y. Qian, and H. H. Chen. Adaptive power allocation and call admission control in multiservice WiMAX access networks. *IEEE Wireless Communications*, 14(1):14–19, 2007.
- [57] A. K. Sadek, W. Su, and K. J. R. Liu. Multinode cooperative communications in wireless networks. *IEEE Transactions on Signal Processing*, 55(1):341–355, 2007.
- [58] H. Shetiya and V. Sharma. Algorithms for routing and centralized scheduling in IEEE 802.16 mesh networks. *Wireless Communications and Networking Conference*, pages 147–152, 2006.
- [59] J. Shi and A. Hu. Maximum utility-based resource allocation algorithm in the IEEE 802.16 OFDMA System. *IEEE International Conference on Communications (ICC)*, pages 311–316, 2008.
- [60] K. Sundaresan and S. Rangarajan. Efficient algorithms for leveraging spatial reuse in OFDMA relay networks. *IEEE INFOCOM*, pages 1539–1547, 2009.
- [61] A. Taha and H. Hassanein. IEEE 802.16 mesh schedulers: issues and design challenges. *IEEE network*, 22(1):58–65, 2008.
- [62] J. Tao, F. Liu, Z. Zeng, and Z. Lin. Throughput enhancement in WiMax mesh networks using concurrent transmission. *International Conference on Wireless Communications, Networking and Mobile Computing*, pages 871–874, 2005.
- [63] Y. Tcha, M. S. Kim, and S. C. Lee. A compact MAP message to provide a virtual multi-frame structure for a periodic fixed bandwidth assignment scheme. *IEEE C802.16e-04/368r2*, 2004.
- [64] E. Visotsky, J. Bae, R. Peterson, R. Berry, and M. L. Honig. On the uplink capacity of an 802.16j system. *Proc. IEEE Wireless Communications and Networking Conference*, pages 2657–2662, 2008.
- [65] H. Wang, C. Xiong, and V. B. Iversen. Uplink capacity of multi-class IEEE 802.16j relay networks with adaptive modulation and coding. *Proc. IEEE International Conference on Communications*, 2009.
- [66] H. S. Wang and N. Moayeri. Finite-state Markov channel—a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, 1995.
- [67] T. Wang, H. Feng, and B. Hu. Two-dimensional resource allocation for OFDMA system. *IEEE International Conference on Communications Workshops*, pages 1–5, 2008.
- [68] T. Wang, H. Feng, and B. Hu. Two-dimensional resource allocation for OFDMA system. *IEEE International Conference on Communications Workshops*, pages 1–5, 2008.
- [69] W. Wang, C. Chen, Z. Guo, J. Cai, and X. S. Shen. Isolation band based frequency reuse scheme for IEEE 802.16j wireless relay networks. *Proc. ICST International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2008.

- [70] W. Wang, Z. Guo, J. Cai, X. S. Shen, and C. Chen. Multiple frequency reuse schemes in the two-hop IEEE 802.16j wireless relay networks with asymmetrical topology. *Computer Communications*, 32(11):1298–1305, 2009.
- [71] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. Haas. Interference-aware IEEE 802.16 WiMax mesh networks. *Vehicular Technology Conference*, pages 3102–3106, 2005.
- [72] Y. Xiao. *WiMAX/MobileFi: advanced research and technology*. Auerbach Publications, 2008.
- [73] J. P. Yoon, W. J. Kim, J. Y. Baek, and Y. J. Suh. Efficient uplink resource allocation for power saving in IEEE 802.16 OFDMA systems. *Proc. IEEE Vehicular Technology Conference*, pages 2167–2171, 2008.
- [74] J. Zhang, H. Hu, L. Rong, and H. Chen. Cross-layer scheduling algorithms for IEEE 802.16 based wireless mesh networks. *Wireless Personal Communications*, 51(3):375–378, 2009.
- [75] X. Zhu, J. Huo, X. Xu, C. Xu, and W. Ding. QoS-guaranteed scheduling and resource allocation algorithm for IEEE 802.16 OFDMA system. *IEEE International Conference on Communications (ICC)*, pages 3463–3468, 2008.
- [76] X. Zhu, J. Huo, S. Zhao, Z. Zeng, and W. Ding. An adaptive resource allocation scheme in OFDMA based multiservice WiMAX Systems. *IEEE International Conference on Advanced Communication Technology (ICACT)*, pages 593–597, 2008.



Curriculum Vitae

Jia-Ming Liang

Contact Information

Department of Computer Science
National Chiao Tung University
1001 University Road, Hsinchu, Taiwan 300
Email: jmliang@cs.nctu.edu.tw

Education

Ph.D.: Computer Science, National Chiao Tung University, Taiwan (2006.9 ~ 2011.5),
Advisor: Yu-Chee Tseng

M.S.: Department of Computer Science and Engineering, National Sun Yat-Sen
University, Taiwan (2004.9 ~ 2006.6), Advisor: Chun-Hung Richard Lin

B.S.: Department of Computer Science and Engineering, National Taiwan Ocean
University, Taiwan (2000.9 ~ 2004.6)

Research Interests

1. Broadband Wireless Communications.
2. Wireless Networks and Mobile Computing.
3. Link-layer Protocols.

Publication List

Journal Papers

1. J.-M. Liang, J.-J. Chen, Y.-C. Wang, and Y.-C. Tseng, “A cross-layer framework for overhead reduction, traffic scheduling, and burst allocation in IEEE 802.16 OFDMA networks”, *IEEE Transaction on Vehicular Technology*, Vol. 60, No. 4, pp. 1740–1755, 2011. (SCI, EI)
2. J.-M. Liang, Y.-C. Wang, J.-J. Chen, J.-H. Liu, and Y.-C. Tseng, “Energy-Efficient uplink resource allocation for IEEE 802.16j transparent-relay networks”, *Computer Networks*. (accepted, to appear) (SCIE, EI)
3. J.-M. Liang, H.-C. Wu, J.-J. Chen, and Y.-C. Tseng, “Mini-Slot scheduling for IEEE 802.16d chain and grid mesh networks”, *Computer Communications*, Vol. 33, No. 17, pp. 2048–2056, 2010. (SCIE, EI)

Conference Papers

1. J.-M. Liang, J.-J. Chen, Y.-C. Wang, Y.-C. Tseng, and B.-S. Lin, “Priority-Based scheduling algorithm for downlink traffics in IEEE 802.16 networks”, *IEEE Asia-Pacific Wireless Communications Symposium (APWCS)*, 2009.
2. J.-M. Liang, C.-W. Wang, L.-C. Wang, and Y.-C. Tseng, “The upper bound of capacity for a concurrent-transmission-based ad-Hoc network with single channel”, *IEEE Asia-Pacific Wireless Communications Symposium (APWCS)*, 2009.
3. J.-J. Chen, J.-M. Liang, and Y.-C. Tseng, “An energy efficient sleep scheduling considering QoS diversity for IEEE 802.16e wireless networks”, *IEEE International Communications Conference (ICC)*, 2010.
4. J.-M. Liang, J.-J. Chen, H.-C. Wu, and Y.-C. Tseng, “Simple and regular mini-slot scheduling for IEEE 802.16d grid-based mesh networks”, *IEEE Vehicular Technology Conference (VTC)*, 2010.
5. J.-M. Liang, Y.-C. Wang, J.-J. Chen, J.-H. Liu, and Y.-C. Tseng, “Efficient resource allocation for energy conservation in uplink transmissions of IEEE 802.16j transparent relay networks”, *IEEE/ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2010.

Book Chapters

1. J.-M. Liang, Y.-C. Wang, and Y.-C. Tseng, “Scheduling problems and solutions in WiMAX networks” (a book chapter in “Scheduling Problems and Solutions”), Nova Science Publishers, 2011. (ISBN: 978-1-61470-689-2)
2. J.-M. Liang, J.-J. Chen, Y.-C. Tseng, and B.-S. P. Lin, “Scheduling in WiMAX mesh networks” (a book chapter in “Horizons in Computer Science Research, Volume 5”), Nova Science Publishers, 2011. (ISBN: 978-1-61324-789-1)

Submitted Papers

1. J.-M. Liang, J.-J. Chen, C.-W. Liu, Y.-C. Tseng, and B.-S. P. Lin, “On tile-and-energy allocation in OFDMA broadband wireless networks”, submitted to *IEEE Communications Letters*.

