# QoS-aware Resource Allocation in Wireless Communication Systems

by

Chi En Huang

B.A.Sc., The University of British Columbia, 1997

M.Eng., Cornell University, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

**Doctor of Philosophy**

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

The University of British Columbia
(Vancouver)

September 2012

# Abstract

With the rapid growth in demand for wireless communications, service providers are expected to provide always-on, seamless and ubiquitous wireless data services to a large number of users with different applications and different Quality of Service (QoS) requirements. The multimedia traffic is envisioned to be a concurrent mix of real-time traffic and non-real-time traffic. However, radio spectrum is a scarce resource in wireless communications. In order to adapt to the changing wireless channel conditions and meet the diverse QoS requirements, efficient and flexible packet scheduling algorithms play an increasingly important role in radio resource management (RRM).

Much of the published work in RRM has focused on exploiting multi-user and multi-channel diversities. In this thesis, we adopt an adaptive cross layer approach to exploit multi-application diversity in single-carrier communication systems and additionally, multi-bit diversity in multi-carrier communication systems. Efficient and practical resource allocation (RA) algorithms with finer scheduling granularity and increased flexibility are developed to meet QoS requirements. Specifically, for single-carrier communication systems, we develop RA algorithms with flow and user multiplexing while jointly considering physical-layer time-varying channel conditions as well as application-layer QoS requirements. For multi-carrier communication systems, we propose a bitQoS-aware RA framework to adaptively match the QoS requirements of the user application bits to the characteristics of the narrowband channels.

The performance gains achievable from the proposed bitQoS-aware RA framework are demonstrated with suboptimal algorithms using water-filling and bit-loading approaches. Efficient algorithms to obtain optimal and near-optimal solutions to the joint subcarrier, power and bit allocation problem with continuous and discrete rate adaptation, respectively, are developed. The increased control signaling that may be incurred, as well as the computational complexity as a result of the finer scheduling granularity, are also taken into consideration to establish the viability of the proposed RA framework and algorithms for deployment in practical networks. The results show that the proposed framework and algorithms can achieve a higher system throughput with substantial performance gains in the considered QoS metrics compared to RA algorithms that do not take QoS requirements into account or do not consider multi-application diversity and/or multi-bit diversity.

# Preface

Each of Chapters 2 to 8 is based on manuscripts that have been accepted, submitted or to be submitted for publication in international peer-reviewed journals and conferences. The manuscripts are all co-authored by myself as the first author and my supervisor, Dr. Cyril Leung. In all these works, I played the primary role in designing and performing the research, doing data analysis and preparing manuscripts under the supervision of Dr. Cyril Leung.

List of publications resulting from this PhD work are:

- C. E. Huang and C. Leung, "Multi-flow merging gain in scheduling for flow-based wireless networks," in *Proc. IEEE PACRIM*, Aug. 2007, pp. 553–556.

- C. E. Huang and C. Leung, "Adaptive cross layer scheduling with flow multiplexing," in *Proc. IEEE WCNC*, Mar. 2008, pp. 1871–1876.

- C. E. Huang and C. Leung, "Downlink mixed-traffic scheduling with packet division multiplexing," in *Proc. ACM PM2HW2N*, Oct. 2008, pp. 165–172.

- C. E. Huang and C. Leung, "QoS-aware bit scheduling in multi-user OFDM systems," in *Proc. IEEE WCNC*, Mar. 2011, pp. 215–220.

- C. E. Huang and C. Leung, "BitQoS-aware resource allocation for multi-user mixed-traffic OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2067-2082, Jun. 2012.

- C. E. Huang and C. Leung, "Scheduling signaling overhead in bitQoS-aware multi-flow OFDM systems," submitted.

- C. E. Huang and C. Leung, “On the optimality of bitQoS-aware resource allocation in OFDMA systems,” submitted.

- C. E. Huang and C. Leung, “Determination of scheduling block size in bitQoS-aware OFDMA systems,” in preparation.

# Table of Contents

# List of Tables

# List of Figures

# List of Symbols

| | |
|---|---|
| $I$ | number of users |
| $i$ | user index, $i \in \mathcal{I} = \{1, \ldots, I\}$ |
| $J_i$ | number of flows for user $i$ |
| $j$ | flow index, $j \in \mathcal{J}_i = \{1, \ldots, J_i\}$ |
| $J_{sys}$ | number of flows in the system, defined as $J_{sys} \triangleq \sum_{i=1}^{I} J_i$ |
| $N$ | number of subcarriers |
| $n$ | subcarrier index, $n \in \mathcal{N} = \{1, \ldots, N\}$ |
| $B_i^j(k)$ | data buffer queue length for user $i$, flow $j$ at time $k$ |
| $z$ | bit index, $z \in \{1, \ldots, B_i^j(k)\}$ |
| $K$ | simulation length |
| $k$ | time index, $k \in \{1, \ldots, K\}$ |
| $\alpha_{i,n}$ | channel gain of subcarrier $n$ for user $i$ |
| $P_{total}$ | total BS transmit power |
| $T_s$ | OFDM symbol duration |
| $c_{i,n}$ | number of bits that can be carried on subcarrier $n$ for user $i$ |
| $\sigma_0^2$ | noise power |
| $\zeta$ | signal-to-noise ratio gap parameter |
| $\lambda_j$ | average traffic arrival rate of flow $j$ |
| $\psi_i^{j,z}$ | bitQoS value of bit $z$ for user $i$, flow $j$ |
| $\boldsymbol{\theta}_i^{j,z}$ | tuple of QoS parameters associated with bit $z$ for user $i$, flow $j$ |
| $a_{i,n}$ | subcarrier assignment optimization variable for user $i$, subcarrier $n$ |
| $p_{i,n}$ | transmit power allocation optimization variable for user $i$, subcarrier $n$ |
| $b_{i,n}^{j,z}$ | bit assignment optimization variable for bit $z$ of user $i$, flow $j$ on subcarrier $n$ |

| | |
|---|---|
| $\pi_j$ | application flow priority of flow $j$ |
| $w_i^{j,z}(k)$ | waiting time of bit $z$ for user $i$, flow $j$ at time $k$ |
| $T_j$ | scheduling delay threshold of flow $j$ |
| $\eta_j$ | comfort latency threshold of flow $j$ |
| $\xi_j$ | delay sensitivity of flow $j$ |
| $c_j$, $d_j$ | coefficients for the bitQoS function of flow $j$ |
| $\boldsymbol{U}^{NFM}(k)$ | subcarrier-to-flow vector at time $k$ |
| $\boldsymbol{U}^{FM}(k)$ | subcarrier-to-user vector at time $k$ |
| $\boldsymbol{V}_n^{FM}(k)$ | bit-to-flow vector for subcarrier $n$ at time $k$ |
| $M_n(k)$ | number of bits carried by subcarrier $n$ at time $k$ |
| $C$ | analytical system throughput |
| $R$ | total number of allocated bits |
| $B$ | maximum data buffer queue length, defined as $B = \max_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}_i} B_i^j$ |
| $L$ | number of iterations performed by a bisection algorithm |
| $\kappa$ | number of iterations performed by MDU |
| $EPSize_i(k)$ | physical layer encoder packet size for user $i$ at time $k$ |
| $\mu$ | effective service rate |
| $d_i^{j,z}(k)$ | size of the packet at position $z$ in the data buffer for user $i$, flow $j$ at time $k$ |
| $\overline{v_i}(k)$ | running average throughput of user $i$ over the last $N_{window}$ time slots at time $k$ |
| $CSI_i(k)$ | channel state information for user $i$ at time $k$ |
| $u_i^{j,z}(k)$ | packet urgency value of user $i$, flow $j$, packet $z$ at time $k$ |
| $\rho$ | system loading factor |
| $G_\chi^Y$ | performance gain of scheduling policy $Y$ with QoS measure $\chi$ |
| $a_{i,n}^j$ | subcarrier assignment optimization variable for user $i$, flow $j$, subcarrier $n$ in the bitQoS-aware resource allocation framework with no flow merging |
| $p_{i,n}^j$ | transmit power allocation optimization variable for user $i$, flow $j$, subcarrier $n$ in the bitQoS-aware resource allocation framework with no flow merging |
| $c_{i,n}^j$ | number of bits that can be carried on subcarrier $n$ for user $i$, flow $j$ in the bitQoS-aware resource allocation framework with no flow merging |

| | |
|---|---|
| $W$ | system bandwidth |
| $\Delta f$ | subcarrier spacing |
| $\mathcal{J}_{sys}$ | set of indices to all $flow(i,j), \forall i \in \mathcal{I}, j \in \mathcal{J}_i$ |
| $J_{sys}$ | number of flows in the system, defined as $J_{sys} = \sum_{i \in \mathcal{I}} J_i$ |
| $\boldsymbol{U}^{FMGS}(k)$ | subcarrier-to-user vector at time $k$ for scheduling policies with FMGS |
| $\boldsymbol{V}_n^{FMGS}(k)$ | bit-to-flow vector for subcarrier $n$ at time $k$ for scheduling policies with FMGS |
| $H^{NFM}$, $H^{FM}$, $H^{FMGS}$ | entropy of scheduling signaling information for NFM, FM and FMGS, respectively |
| $\Upsilon_{\boldsymbol{w}}^{RLE}$, $\Upsilon_{\boldsymbol{w}}^{LZW}$ | number of bits required to represent a data block $\boldsymbol{w}$ using RLE and LZW, respectively |

Note: In this thesis, in order to distinguish a random variable from a sample value, the former is denoted by an uppercase letter, whereas the latter is denoted by a lowercase letter.

# List of Acronyms

**3G** Third Generation

**3GPP2** 3rd Generation Partnership Project 2

**ACLS-FM** Adaptive Cross Layer Scheduling with Flow Multiplexing

**ACLS-FUM** Adaptive Cross Layer Scheduling with Flow and User Multiplexing

**AMC** Adaptive Modulation and Coding

**BE** Best Effort

**BER** Bit Error Rate

**BS** Base Station

**CDF** Cumulative Distribution Function

**CDMA** Code Division Multiple Access

**CSI** Channel State Information

**CSDPS** Channel State Dependent Packet Scheduling

**DPA** Default Packet Application

**DRC** Date Rate Control

**EF** Expedited Forwarding

**FDD** Frequency Division Duplex

**FER** Frame Error Rate

**FIFO** First In, First Out

**HARQ** Hybrid Automatic Repeat-reQuest

**HOL** Head-of-Line

**IP** Internet Protocol

**KKT** Karush-Kuhn-Tucker

**LHS** Left-Hand-Side

**LTE** Long Term Evolution

**LZW** Lempel-Ziv-Welch

**M-LWDF** Modified Largest Weighted Delay First

**MAC** Medium Access Control

**MDU** Max-Delay-Utility

**MFM** Multi-Flow Merging

**MFPA** Multi-Flow Packet Application

**MILP** Mixed-Integer Linear Programming

**MINLP** Mixed-Integer Non-Linear Programming

**MS** Mobile Station

**MSO** Markov Service Option

**MUP** Multi-User Packet

**OFDM** Orthogonal Frequency Division Multiplexing

**OFDMA** Orthogonal Frequency Division Multiple Access

**OSI** Open Systems Interconnection

**PDM** Packet Division Multiplexing

**PDU** Protocol Data Unit

**PF** Proportional Fair

**QoS** Quality of Service

**RA** Resource Allocation

**RLE** Run-Length Encoding

**RRM** Radio Resource Management

**SNR** Signal-to-Noise Ratio

**TCP** Transmission Control Protocol

**TDD** Time Division Duplex

**UMTS** Universal Mobile Telecommunications System

**VoIP** Voice over Internet Protocol

# Acknowledgments

I would like to take this opportunity to express my utmost gratitude and sincere appreciation to my supervisor, Dr. Cyril Leung, whose continued guidance, persistent encouragement and deep insight in the research area have helped me immeasurably throughout the course of my thesis research. This thesis would never have been written without his assistance. I would also like to thank my supervisory committee members for their time and effort.

I am deeply indebted to my family for their constant support and immense encouragement over the years.

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant OGP0001731 and by the UBC PMC-Sierra Professorship in Networking and Communications.

*To my parents and JHYC*

# Chapter 1

# Introduction

## 1.1 Motivation

Radio spectrum is a scarce and expensive resource in wireless communications. This has led to extensive research in Radio Resource Management (RRM) with the objective of improving the achievable system capacity. While a large number of mature Resource Allocation (RA) algorithms for wireline networks have been studied [1, 2], they are not directly applicable to wireless networks due to distinct characteristics of the wireless channel such as user mobility, time-varying link capacity, high error rates, scarce bandwidth and power constraint of the Mobile Station (MS). With the tremendous growth in the wireless communications industry, wireless networks are expected to provide always-on, seamless and ubiquitous wireless data services to a large number of users with different applications and different Quality of Service (QoS) requirements. The multimedia traffic is envisioned to be mostly Internet Protocol (IP) based and to be a mix of real-time traffic such as voice, videoconferencing and gaming, and non-real-time traffic such as web browsing, file transfers and messaging [3]. The expected increase in peak rate and throughput requirements will be achieved using a combination of wider channel bandwidths and increased spectral efficiency. QoS requirements will include minimum acceptable throughput, maximum latency and maximum delay jitter, maximum packet loss and packet error rates and *a priori* determined priority classes of users and applications. In order to adapt to the time-varying wireless channel conditions and meet the diverse QoS requirements for a large number of users, wireless networks will need efficient and flexible packet scheduling algorithms.

## 1.2 Related Work

In this section, we review techniques for wireless resource allocation including scheduling algorithms that exploit multi-user and multi-channel diversities, cross layer resource allocation and resource allocation in Orthogonal Frequency Division Multiplexing (OFDM) networks.

### 1.2.1 Resource Allocation in Wireless Communication Systems

A common objective in RRM is to improve system capacity while meeting the diverse QoS requirements. While it is desirable that an optimal scheduling algorithm shall attempt to achieve key objectives that include efficient link utilization, fairness, throughput guarantees, low algorithm complexity, scalability and system stability [4, 5], some of these objectives are conflicting in nature. Hence, appropriate trade-offs need to be made to satisfy specific system service requirements.

In [6], a comprehensive survey of wireless scheduling algorithms to support the provision of QoS requirements for various types of broadband multimedia wireless networks are classified and examined. A Channel State Dependent Packet Scheduling (CSDPS) algorithm is proposed in [7] where the authors show that by deferring transmission of packets on a wireless link that is experiencing bursty errors to reduce retransmissions and exploit channel diversity gains, significant improvement in channel utilization can be achieved. However, the proposed CSDPS algorithm does not guarantee fairness to users and does not provide any bounds on packet delay. In [8], a Proportional Fair (PF) algorithm is proposed which exploits multi-user diversity to maximize system throughput on the forward link of a Code Division Multiple Access (CDMA) network by scheduling data transmission based on the relative channel quality of the competing users, while at the same time maintaining fairness across the entire competing user population. A user $i$ which has not transmitted for a long time due to a relatively low carrier-to-interference ratio gets its priority $Q_i(k)$ raised where

$$Q_i(k) = \frac{CSI_i(k)}{\bar{v}_i(k)}. \tag{1.1}$$

In (1.1), $CSI_i(k)$ is the channel state information of user $i$ at time $k$, and $\bar{v}_i(k)$ is the average throughput of user $i$ over a time window up to time $k$. The forward link throughput performance of a cdma2000 1xEV-DO system employing the PF algorithm is presented in [9].

While the specifications for Third Generation (3G) networks do not specify the details of the scheduler, some form of PF scheduler is typically used. Recent work in RRM has focused on supporting QoS of multimedia traffic. In [10], the delays are explicitly controlled by inclusion of the queue lengths in the scheduling algorithm. Token based rate control mechanisms are studied in [11] to provide minimum throughput guarantees. In [12], the PF algorithm is modified to take into account delay requirements of real-time data and it is shown that with the simple modifications, the scheduler can provide effective and fair service to both real-time and non-real-time data. Formulation of QoS requirements as stochastic constraints are expressed in [13] where a general structure for opportunistic scheduling policies that exploit channel and buffer content variations is presented. Since issues of efficient and fair resource allocation have been well studied in economics, utility-based resource allocation and scheduling are studied in [5, 14] by quantifying resource use (bandwidth, power, etc.) or performance criteria (data rate, delay, etc.) into corresponding price values and optimizing the established utility pricing system. In [15], an access scheme for multiplexing (from one session at each transmission) multimedia traffic over the air that can achieve absolute QoS guarantees in terms of Average Packet queuing Delay (APD), Packet Loss Rate (PLR), Packet Delay Variation (i.e. jitter) (PDV) and Packet Transfer Delay (PTD) for different service classes is proposed. Per-session guaranteed QoS for multimedia traffic is introduced in [10, 16] for scheduling of uplink and downlink flows.

### 1.2.2 Cross Layer Resource Allocation

Cross layer design is an interdisciplinary research area which involves signal processing, adaptive coding and modulation, channel modeling, traffic modeling, queuing theory, and network protocol design and optimization techniques [17]. As a wide variety of cross layer related designs have been studied in literature, we focus our literature survey on the general application of cross layer optimization in RRM. Fig. 1.1 shows a typical cross layer design model which attempts to optimize functionality across blurred delineation of layers.

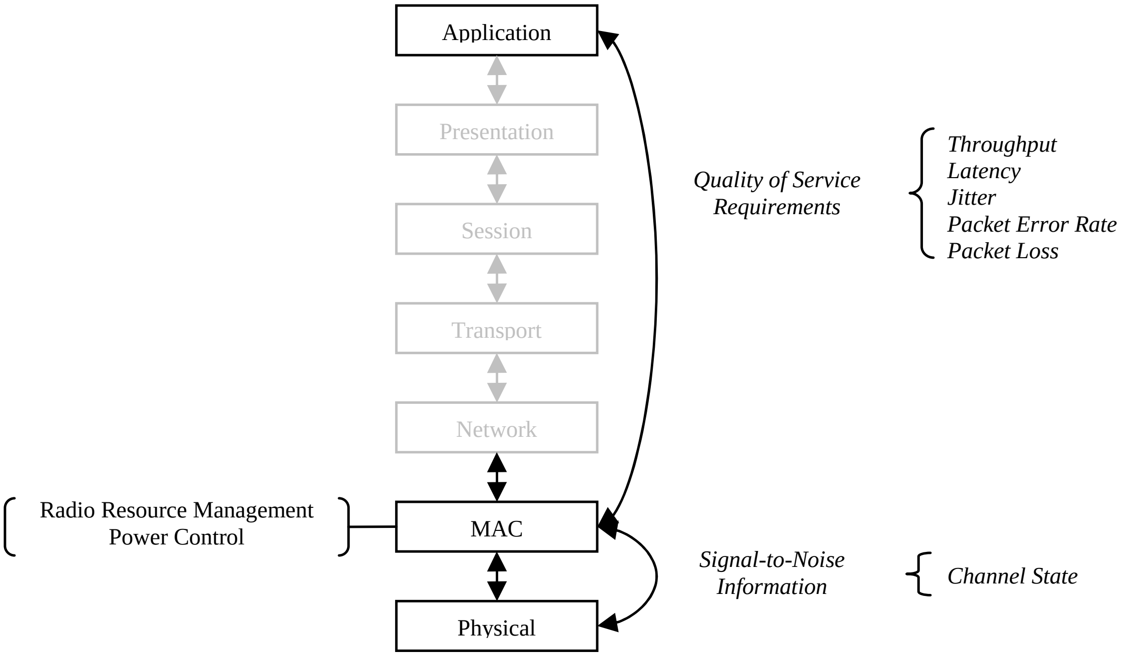


**Figure 1.1:** Cross Layer Design Model

An important aspect of wireless communications is its dynamic behavior. While the conventional layered Open Systems Interconnection (OSI) model [18] has served communication system designers well in the past by exploiting the advantage of modularity in system design, the structure is inflexible, requiring the various layers to communicate in a strictly defined manner. In most cases, layers are designed to operate in worst-case scenarios rather than adapting to conditions as they change, leading to inefficient use of both spectrum and energy. Evolving wireless networks are seriously challenging this design architecture, mandating the need for the various OSI layers to adapt to the channel variations and QoS requirements [19] and to be considered together [20–22] in order to provide more efficient methods of allocating network resources over the wireless network. In [3], an overview of the cross layer design paradigm shift is provided as wireless communication networks evolve from a circuit-switched to a packet-switched infrastructure. In [22], a general survey of the recent myriad of cross layer design proposals is presented along with a suggested definition and taxonomy for classifying cross layer designs. Open challenges to cross layer optimization are listed in [20, 22] to establish a platform upon which new research can be built.

An overview of cross layer design approaches for resource allocation is provided in [23] which proposes a cross layer design approach that exploits physical and application layer information to transmit real-time video over time-varying CDMA channels. Simulation results are presented to show the effectiveness of the proposed approach. In [24], information obtained from the fast power control algorithm is used to define a low complexity prioritization function to exploit short-term channel variations and to schedule transmissions for a Universal Mobile Telecommunications System (UMTS) downlink channel. Simulations results show improved system performance in terms of capacity and delay. [25] introduces an adaptive cross layer packet scheduler which minimizes a prescribed cost function given the current channel qualities and delay states of the packets in the queue. It is shown that the cross layer scheduling algorithm outperforms both the weighted fair queuing (WFQ) and earliest deadline first (EDF) schedulers with respect to both packet delay and user throughput.

While most recent papers tout the advantages of a cross layer approach to resource allocation for next generation wireless networks, a cautionary perspective is raised in [26], which points out a trade-off between performance gains and upkeep difficulty of system architecture violations introduced by cross layer designs.

### 1.2.3 Resource Allocation in OFDM Networks

OFDM is a promising technique for communication systems due to its high spectral efficiency and flexibility in dynamically allocating resources to multiple users. While spectral efficiency has improved significantly with the deployment of beyond 3G OFDM-based cellular air interfaces [27, 28], unallocated radio spectrum is scarce in most populated regions. The problem of dynamic bit-loading, transmission power allocation and subcarrier assignment for multi-user OFDM systems has attracted a great deal of interest. It is shown in [29] that the system efficiency can be significantly improved by allocating the power and subcarriers based on knowledge of the users' channel qualities. In [30], it is shown that the downlink system throughput is maximized when each subcarrier is assigned to the user with the best channel gain on that subcarrier and power is then allocated to the subcarriers using the water-filling algorithm. However, fairness among users is not considered in [29, 30] and it is possible that when the path loss differences are large among users, the users experiencing poor channel gains for an extended period of time may be starved. In [31], the optimal subcarrier assignment is formulated as a max-min convex optimization problem to maximize the worst user's capacity. However, since the max-min approach deals with the worst-case scenario in which the smallest user capacity is maximized, thereby ensuring that all users achieve similar data rates, it penalizes users with better channels and reduces system efficiency. In [32], a set of proportional rate constraints is introduced into the throughput maximization problem to allow each user to achieve a required data rate. The above-mentioned works exploit multi-user and multi-channel diversities to maximize system throughput and/or minimize total transmit power. However, they do not consider application QoS requirements which allow users to subscribe to the different levels of service available in contemporary wireless networks [27, 28, 33].

Radio RA algorithms that take QoS information of different traffic classes from the application layer and channel information from the physical layer into consideration to exploit multi-flow (concurrent applications with different QoS requirements) diversity have been studied for mixed-traffic networks [34–41]. The Modified Largest Weighted Delay First (M-LWDF) [35, 42] is a throughput-optimal algorithm that exploits multi-user diversity across time by buffering bursty traffic and improves throughput performance by trading delay for throughput. It provides QoS for data users by ensuring a minimum throughput guarantee and maintaining delays smaller than a predetermined threshold with a given probability.

RA algorithms based on M-LWDF with buffer and channel information have been studied in [36, 37, 39, 40]. In [36, 37], the authors consider a mixed-traffic environment and propose a utility-based cross layer RA framework in which utility functions are used to represent application QoS requirements. Based on this framework, a Max-Delay-Utility scheduling policy, hereafter referred to as MDU, is proposed in [38]. MDU aims to maximize the aggregate utility with respect to the user average waiting time while taking into account channel conditions and data queue information. An urgency and efficiency based packet scheduling algorithm is proposed in [39] to support both real-time and non-real-time traffic. The aim is to maximize the throughput of non-real-time traffic while satisfying the QoS requirements of real-time traffic by serving non-real-time traffic until the real-time packets approach their deadlines. In [40], the different traffic classes are handled separately by considering Head-of-Line (HOL) packet waiting time for real-time traffic and the queue length for non-real-time traffic. In [41], the authors present a joint bit rate, subcarrier and power allocation problem which take into consideration limits on the subcarrier transmit power in addition to an overall system power constraint.

## 1.3 Objectives and Contributions

Wireless communications, in particular CDMA and Orthogonal Frequency Division Multiple Access (OFDMA) cellular networks, has emerged as one of the largest sectors of the telecommunications industry and one of the most promising growth areas into the next decade. To meet the challenges of deploying an efficient wireless multimedia network, it is useful to consider network functions (i.e., the various OSI layers) together when designing the network to take into account QoS requirements at the Medium Access Control (MAC) layer where the scheduling and RA algorithms reside. As the scarce radio spectrum is shared by a large number of users, in this thesis, the research objective is to design and analyze efficient and practical adaptive cross layer (physical, MAC and application layers) RA algorithms for single-carrier CDMA communication systems and multi-carrier OFDMA communication systems that jointly consider the physical layer time-varying channel conditions as well as application layer QoS requirements so as to more efficiently utilize the radio spectrum.

In addition to exploiting multi-user and multi-channel diversities as in existing studies, we increase the flexibility and granularity of the RA algorithms by exploiting multi-application and multi-bit diversities to take advantage of the mechanisms and optimization features introduced in the air interfaces [27, 43]. In particular, for CDMA communication systems, we develop RA algorithms with flow and user multiplexing to take advantage of the flow-oriented QoS approach and Packet Division Multiplexing (PDM) to provide a unified approach to intra-user (between flows of a user) and inter-user (between users) QoS and to permit the Base Station (BS) to serve multiple users in the same physical layer encoder packet, respectively.

As cellular networks adopt OFDM as a modulation scheme due to its high spectral efficiency and flexibility in dynamically allocating resources to multiple users, for OFDMA communication systems, since data is loaded onto subcarriers in units of bits, we consider QoS at the bit-level rather than at the flow-level as in existing studies and define a bitQoS function which maps the QoS parameters of an application bit into a numerical value. We establish a bitQoS-aware RA framework which adaptively matches the QoS requirements of the user application bits to the characteristics of the OFDM subcarriers in a mixed-traffic environment. The proposed bitQoS-aware RA framework is formulated as an optimization problem with the objective of finding the joint subcarrier, power and bit assignment to maximize the total bitQoS-weighted throughput, subject to the total power constraint. However, as the formulated optimization problem is a Mixed-Integer Non-Linear Programming (MINLP) problem whose solution is computationally complex given the large number of subcarriers and users in a practical system, we demonstrate the performance gains achievable from the proposed framework with suboptimal algorithms using water-filling and bit-loading approaches. We then formulate the bitQoS RA framework as a convex optimization problem and use the Karush-Kuhn-Tucker (KKT) conditions to develop efficient algorithms to obtain optimal and near-optimal solutions to the joint subcarrier, power and bit allocation problem with continuous and discrete rate adaptation, respectively. To assess the viability of the bitQoS-aware RA framework, we formulate a model to determine and analyze the scheduling signaling overhead, including the scheduling signaling information entropy, and consider different schemes to compress the associated control signaling. The computational complexities of the proposed RA algorithms are also assessed for deployment consideration in practical networks.

## 1.4 Thesis Overview

The thesis is organized as follows: RA algorithms for single-carrier CDMA communication systems are studied in Chapters 2 and 3 and RA algorithms for multi-carrier OFDMA communication systems are studied in Chapters 4-8. The structure of the thesis is illustrated in Fig. 1.2.

In Chapter 2, we exploit multi-application diversity in flow-based single-carrier CDMA communication systems and quantify the performance gains obtainable with Multi-Flow Merging (MFM) in terms of user throughput, user latency and user packet drop probability. In addition, we incorporate the concept of MFM and propose an adaptive cross layer (physical, MAC and application layers) scheduling policy which further takes into account the time-varying channel conditions from the physical layer and includes QoS requirements from the application layer.

In Chapter 3, we extend the scheduling policy proposed in Chapter 2 to take into account PDM introduced in cdma2000 1xEV-DO Revision A. PDM permits the BS to service multiple users in the same physical layer encoder packet in a single time slot with the use of Multi-User Packet (MUP) transmission. We consider a mix of real-time voice services and non-real-time data applications and study the improvements in packing efficiency and latency performances. The QoS performance gains with flow and user multiplexing are quantified in terms of user throughput, user latency, user packet drop probability and user jitter in a mixed-traffic environment.

In Chapter 4, we propose a bitQoS-aware RA framework which exploits multi-bit diversity in addition to multi-application diversity to increase the flexibility and granularity of the RA algorithms in multi-carrier OFDMA communication systems. The proposed bitQoS-

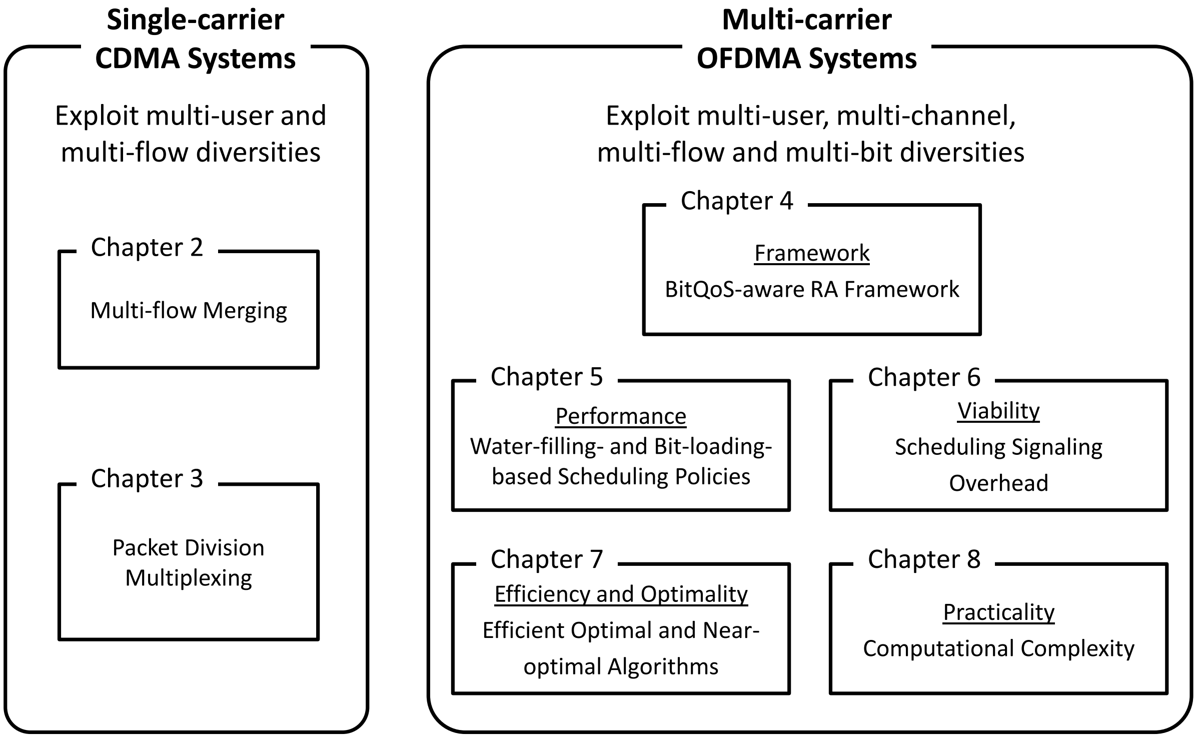

Single-carrier
CDMA Systems
Exploit multi-user and multi-flow diversities
Chapter 2
Multi-flow Merging
Chapter 3
Packet Division Multiplexing
Multi-carrier
OFDMA Systems
Exploit multi-user, multi-channel, multi-flow and multi-bit diversities
Chapter 4
Framework
BitQoS-aware RA Framework
Chapter 5
Performance
Water-filling- and Bit-loading-based Scheduling Policies
Chapter 6
Viability
Scheduling Signaling Overhead
Chapter 7
Efficiency and Optimality
Efficient Optimal and Near-optimal Algorithms
Chapter 8
Practicality
Computational Complexity

**Figure 1.2:** Structure of the Thesis

aware RA framework is formulated as two optimization problems, with no flow merging and with flow merging, with the objective of finding the joint subcarrier, power and bit assignment to maximize the total bitQoS-weighted throughput subject to the total power constraint. The system model which includes the network model and traffic classes are described and the performance evaluation methodology along with the comparative schemes used to assess the performance of the proposed bitQoS-aware RA framework are presented.

In Chapter 5, we evaluate the performance of the bitQoS-aware RA framework and propose two iterative subcarrier-power-bit allocation algorithms, one based on the water-filling approach and the other on the bit-loading approach, to quantify the achievable performance gains. In addition, the potential performance gains by allowing bits from different application flows of a user to be merged into a single OFDM subcarrier is examined. The performance gains obtainable are quantified in terms of system throughput, user throughput, user latency, user jitter and user packet drop probability for systems under different loads.

In Chapter 6, we establish the viability of the bitQoS-aware RA framework by taking into account the scheduling signaling overhead associated with the increased scheduling granularity of the proposed bitQoS-aware RA framework. This is critical since valuable resources that could otherwise be used to transmit application bits need to be reserved for control signaling. We formulate a scheduling signaling overhead model to analyze the scheduling signaling information required and consider different schemes to compress the scheduling signaling information. To assess the tradeoff between the scheduling gain and the increased scheduling signaling overhead of the proposed bitQoS-aware RA framework, the effective throughput gains (with the scheduling signaling overhead taken into account) are quantified.

In Chapter 7, with the performance gains and viability of the proposed bitQoS-aware RA framework established in Chapters 5 and 6, we use the KKT conditions to establish necessary and sufficient optimality conditions and develop efficient algorithms to obtain optimal and near-optimal solutions to the joint subcarrier, power and bit allocation problem with continuous and discrete rate adaptation, respectively. The performance of the proposed KKT-based algorithms is evaluated in terms of their closeness to optimality and computation time. In addition, the sensitivities of the objective value and computation time to tuning parameters in the KKT-based algorithms are also discussed.

In Chapter 8, we assess the computational complexity of the scheduling policies proposed for the bitQoS-aware RA framework and evaluate their practicality for real-time resource allocation in Long Term Evolution (LTE), an OFDM-based air interface.

In Chapter 9, the main contributions of the thesis and suggestions for future research are presented.

# Chapter 2

# Flow Multiplexing in Single-carrier CDMA Systems [1]

## 2.1 Introduction

While much of the existing work in RRM has focused on exploiting multi-user (channel) diversity and more recently exploiting multi-application (flow) diversity, we present in this chapter the performance gains of MFM in scheduling and propose an adaptive cross layer scheduling policy that is realizable in a framework such as that provided in cdma2000 1xEV-DO Revision A [43], which takes into account the time-varying Channel State Information (CSI) from the physical layer and includes QoS requirements from the application layer. We refer to this as the Adaptive Cross Layer Scheduling with Flow Multiplexing (ACLS-FM) scheduling policy.

This chapter is organized as follows: in Section 2.2, we briefly describe the enhancements to the cdma2000 1xEV-DO Revision A air interface and the included Multi-Flow Packet

[1] The material in this chapter is based on the following:
C. E. Huang and C. Leung, "Multi-flow merging gain in scheduling for flow-based wireless networks," in *Proc. IEEE PACRIM*, Aug. 2007, pp. 553–556. © 2007 IEEE.
http://dx.doi.org/10.1109/PACRIM.2007.4313296
C. E. Huang and C. Leung, "Adaptive cross layer scheduling with flow multiplexing," in *Proc. IEEE WCNC*, Mar. 2008, pp. 1871–1876. © 2008 IEEE.
http://dx.doi.org/10.1109/WCNC.2008.333

Application (MFPA). Section 2.3 presents the system model that includes the network model, traffic classes and data buffer parameters. The concept of MFM is illustrated in Section 2.4 along with an example scheduling policy. The ACLS-FM scheduling policy is described in Section 2.5. Simulation results are presented in Section 2.6 and the main findings are summarized in Section 2.7.

## 2.2 Background on cdma2000 1xEV-DO

The cdma2000 1xEV-DO (1x Evolution-Data Optimized) air interface is an evolution of the cdma2000 family of 3G mobile telecommunications air interface, standardized by the 3rd Generation Partnership Project 2 (3GPP2), that utilizes CDMA to provide high-speed packet data services to wireless users. However, unlike the other variants of CDMA based systems such as IS-95 [44], where the forward link transmit power is shared among all active mobiles within a sector to maintain simultaneous, continuous voice channels, cdma2000 1xEV-DO systems time-division multiplex (TDM) the forward link data transmission and transmit at full power to produce the highest possible energy per bit to noise ratio ($E_b/N_0$) to each active mobile. This allows the base station to transmit user data at the highest data rate supported by the time-varying wireless channel that the MS determines from the pilot channel carrier-to-interference ratio [45]. The reverse link remains similar to the IS-95 and utilizes code division multiplexing. 1xEV-DO Release 0 provides a peak physical layer data rate of 2.4 Mbps in the forward link and 153.6 kbps in the reverse link [46]. Forward link data is transmitted in successive $26\frac{2}{3}$ ms frames, which are divided into sixteen $1\frac{2}{3}$ ms slots in which packets of data are transmitted. The transmission duration of a single packet may vary from 1 to 16 slots.

The successor to 1xEV-DO Release 0 is 1xEV-DO Revision A [43] which includes enhancements that provide significant gains in spectral efficiency and substantial QoS support for inter-user (between users) and intra-user (between flows of a user) QoS in both the forward and reverse links. In addition to a rich variety of link adaptation techniques, such as power control, data rate control and Adaptive Modulation and Coding (AMC), 1xEV-DO Revision

A makes use of higher order modulation and Hybrid Automatic Repeat-reQuest (HARQ) to achieve higher peak data rates of 3 Mbps in the forward link and 1.8 Mbps in the reverse link [43, 47]. HARQ reduces the effects of power control imperfections due to variations in channel state and multiple-access interference to achieve higher reverse link spectral efficiency via early termination of physical packet transmissions, leading to improved throughput and reduced packet delay. Shorter packets, along with finer rate quantization, multi-user packet (MUP) transmission, and uninterrupted data transfer during forward link cell switching contribute to lower latency. In the reverse traffic channel MAC, key QoS-sensitive support includes efficient support for latency-sensitive and delay-tolerant applications, resource allocation among flows associated within a MS and MAC layer ARQ [43, 48]. A flow is an octet stream that can be used to carry packets between the MS and the BS. While some of these features increase system throughput and spectral efficiency, others improve the operator's ability to guarantee acceptable latency performance for delay sensitive applications such as interactive voice and video, and still others provide a mechanism for application coexistence. In particular, we highlight the following features of 1xEV-DO Revision A that are considered in our research.

1xEV-DO Release 0 systems support per flow QoS on the forward link and per MS QoS on the reverse link through the Default Packet Application (DPA). The DPA consists of a link layer protocol that provides octet retransmissions and duplicate detection, a location update protocol that provides mobility between data service networks and a flow control protocol that provides flow control of data traffic [46]. There is no differentiation of packets from different applications with different QoS requirements.

In 1xEV-DO Revision A, a flow-oriented QoS approach [43, 49] is adopted and provides a unified approach to inter-user and intra-user QoS. MFPA is included and provides multiple octet streams that can be used to carry octets between the mobile station and base station. MFPA, along with the reverse link multi-flow MAC with per-flow QoS support, provides the framework for the exploitation of MFM gain in both the forward and reverse links. Packets from latency-sensitive flows that arrive later at the base station following a large packet from a delay-tolerant flow can be transmitted first instead of being transmitted in the order of arrival, hence reducing latency and jitter for multimedia traffic.

## 2.3 System Model

The network model, traffic classes and data buffer parameters used are described in this section.

### 2.3.1 Network Model

We consider a 1xEV-DO Revision A-like packet cellular network random discrete-event model, as shown in Fig. 2.1, consisting of one BS servicing $I$ MSs. Let $\mathcal{I} = \{1, ..., I\}$ be the set of all users (MSs). Each MS $i$ can have up to $J_i$ data queues (flows) and let $\mathcal{J}_i = \{1, ..., J_i\}$ be the set of all flows. Forward link scheduling is centralized at the BS which communicates with all MSs. At each time slot $k$, where $k \in \mathbb{Z}_+ = \{1, 2, ..., K\}$, we assume that only one user is scheduled and that the scheduling decision time is negligible. Power control is not enabled in the forward link for 1xEV-DO systems and the BS transmits at full power to the MSs in all time slots. We assume that packets are received without errors, i.e. $0\%$ Frame Error Rate (FER), between the BS and MS. This simplifying assumption is made to illustrate the potential gains that ACLS-FM can provide. The BS is assumed to have knowledge of the channel state information, $CSI_i(k)$, for each MS $i$ at time $k$, queue status and QoS requirements for all the data queues. The service rate for a user during time slot $k$ is a function of the channel quality which is characterized by its received Signal-to-Noise Ratio (SNR) in time slot $k$. For simplicity, we assume that the physical layer encoder packet size, $EPSize_i(k)$, is chosen according to a uniform distribution from the set of eight discrete physical layer encoder packet sizes, where $EPSize \in \mathcal{E} = \{128, 256, 512, 1024, 2048, 3072, 4096, 5120\}$ bits. Let $\mu \triangleq EPSize/S \in \{4.8, ..., 3072\}$ kbps be the effective service rate, where $S \in \mathcal{S} = \{1, 2, 4, 8, 16\}$ time slot(s) of 1.667 ms duration.

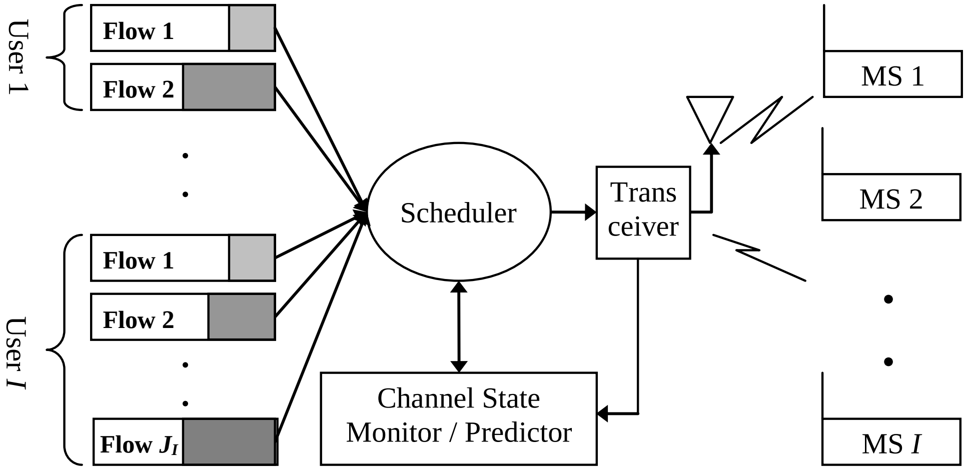


**Figure 2.1:** Forward Link Scheduler Model

### 2.3.2 Traffic Classes

Two traffic classes are considered: Best Effort (BE) traffic class representing Internet browsing-like applications and Expedited Forwarding (EF) traffic class representing Voice over Internet Protocol (VoIP)-like applications. We use the web browsing traffic arrival model in [50] to represent incoming BE traffic. The application layer Protocol Data Unit (PDU) is based on a truncated Pareto distribution with a mean of 25 kBytes and minimum and maximum sizes of 4.5 kBytes and 2 MBytes respectively. The application layer PDU interarrival time is geometrically distributed with a mean of 5 sec. For the EF traffic class, we use the VoIP traffic arrival model in [50]. In contrast to the web browsing model, source configuration and source files are used to generate VoIP traffic. The source file is generated based on the Markov Service Option (MSO) model IS-871 with alterations as detailed in [50]. The application layer PDU size and interarrival time have a mean of 152.4 bits and 0.04 sec, respectively. The average traffic arrival rates $\lambda_{BE}$ and $\lambda_{EF}$ are 40.0 kbps and 3.7 kbps per application respectively.

### 2.3.3 Data Buffer Parameters

The key data buffer parameters are as follows:

**Queue length:** $B_i^j(k) \in \mathbb{Z} = \{0, 1, 2, ...\}$ denotes the queue length, in packets, of the data buffer for user $i$, flow $j$ at time $k$. We assume that the size of data buffer itself is infinite (i.e. no packet blocking). Packets arriving in the data queues are chronologically ordered and serviced in a First In, First Out (FIFO) fashion. Packets in the data buffer are indexed by $z$, $z \in \{1, ..., B_i^j(k)\}$.

**Packet size:** $d_i^{j,z}(k)$ denotes the size, in bits, of the packet at position $z$ in the data buffer for user $i$, flow $j$ at time $k$.

**Waiting time:** $w_i^{j,z}(k) \in (0, \infty)$ denotes the amount of time, in seconds, that the packet at position $z$ in user $i$, flow $j$ buffer has waited. Each packet is time stamped upon arrival in the data buffer, and the waiting time is found by simply subtracting the arrival time from the current time $k$. Packets are dropped if $w_i^{j,z}(k)$ exceeds the flow scheduling delay thresholds $T_j \in \mathbb{R}_+$.

**Flow Priority:** $\pi_j(k) \in \mathbb{R}_+$ denotes the intra-user QoS requirement of flow $j$ at time $k$. $\pi_j(k)$ is a function of $k$ to allow for time-varying intra-user priority changes. However, $\pi_j(k)$ is not a function of user $i$ as it is assumed that flows of the same applications have the same QoS requirement.

## 2.4 Multi-flow Merging

An illustration of MFM for a user $i$ having up to $J_i$ data buffers (flows) is shown in Fig. 2.2. Each application layer PDU is segmented into a number of packets. Packets from different flows can be multiplexed into the same physical layer encoder packet of size $EPSize_i(k)$ for transmission at time $k$.

### 2.4.1 Multi-Flow Merging Scheduling Policy

To explore the benefits of MFM, we extend the existing PF scheduling policy to allow transmission of packets from multiple data queues using a single physical layer encoder packet and refer to this as the MFM scheduling policy. The MFM scheduling policy consists of two

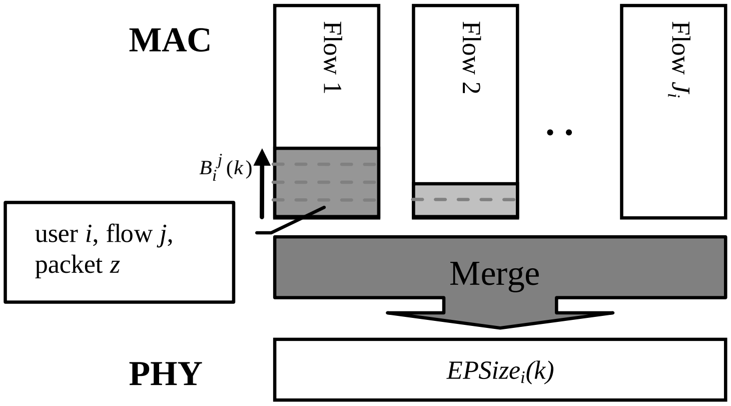


**Figure 2.2:** Illustration of MFM

steps: in Step 1, similar to PF, a user is selected based on the ratio of its $CSI_i(k)$ and corresponding running average throughput $\overline{v_i}(k)$ over the last $N_{window}$ time slots at time $k$; in Step 2, packets from the multiple data queues of the selected user are merged into the physical layer encoder packet. More specifically,

Step 1: Let $Q_i(k)$ denote the priority of user $i$ at scheduling period $k$:

$$Q_i(k) = \begin{cases} \dfrac{CSI_i(k)}{\overline{v_i}(k)} & \text{if} \displaystyle\sum_{j=1}^{J_i} B_i^j(k) > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{2.1}$$

A user with no data to send is assigned a priority of 0 and is ignored in the selection process. The user to be scheduled at time $k$ is determined as:

$$i^*(k) = \arg\max_{i \in \mathcal{I}} Q_i(k). \tag{2.2}$$

Step 2: Packets are selected, one at a time in an iterative fashion, from the data queues of user $i^*$ and added to the physical layer encoder packet until either the physical layer encoder packet of size $EPSize_{i^*}(k)$ is filled or there are no more packets in the data queues. The probability $p_j(\tau)$ that a packet is selected from flow $j$ at iteration $\tau$ is set to:

$$p_j(\tau) = \frac{B_{i^*}^j(k)}{\sum\limits_{j=1}^{J_i} B_{i^*}^j(k)}, \quad \forall j \in \mathcal{J}_i. \tag{2.3}$$

Thus the probability of merging a packet from flow $j$ is given by the ratio of its queue length to the sum of all data queue lengths for user $i^*$. While other simple schemes such as a deterministic longest-queue-first (LQF) scheme to more complex merging schemes are possible, this simple probabilistic scheme was chosen as an example to highlight the realizable gains from multi-flow merging whilst taking into account possible data queue starvation due to the different average data arrival rates in a mixed traffic (BE + EF) environment.

## 2.5 Adaptive Cross Layer Scheduling with Flow Multiplexing Scheduling Policy

The proposed Adaptive Cross Layer Scheduling with Flow Multiplexing (ACLS-FM) scheduling policy consists of a packet urgency function (to meet latency requirements), a packet priority function (for intra-user QoS adjustments), a flowing merging policy (to determine which flows and how many bits from each flow to service) and a user selection policy (to fairly schedule users).

### 2.5.1 Packet Urgency Function

The packet urgency function allows a packet from a latency-sensitive application flow to have its service priority raised when its waiting time exceeds a predetermined threshold. Let $u_i^{j,z}(k) \in \mathbb{R}_+$ denote the packet urgency (PU) value of user $i$, flow $j$, packet $z$ at time $k$. The PU value is given by the following packet urgency function

$$u_i^{j,z}(k) = c_j \xi_j^{(w_i^{j,z}(k) - \eta_j)}, \tag{2.4}$$

where $\xi_j \in \mathbb{R}_+$ is the urgency base and $c_j \in \mathbb{R}_+$ is the scaling factor for flow $j$. The parameter $\eta_j \in \mathbb{R}_+$ is the comfort latency threshold and is generally set to a value that is less than the flow scheduling delay threshold $T_j$. An illustration of the PU functions for BE and EF traffic are shown in Fig. 2.3. In the region where $w_i^{j,z}(k) \leq \eta_{EF}$, the BE traffic has a higher PU value than the EF traffic; this is meant to reduce BE traffic backlog, if necessary.

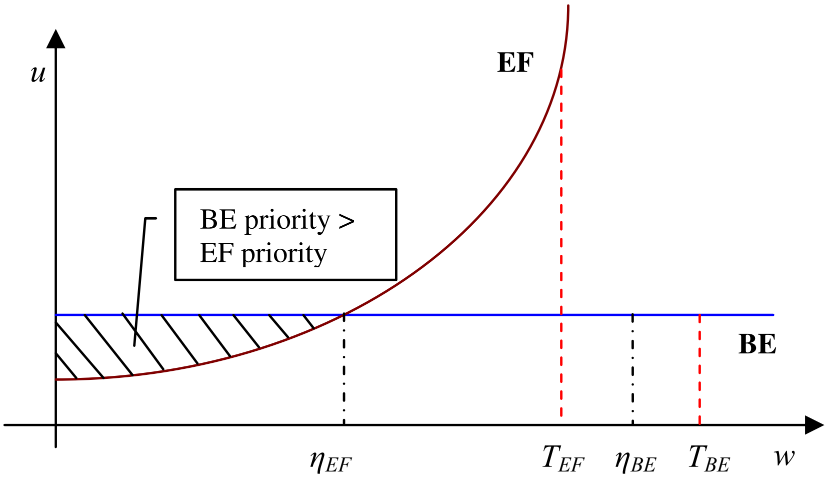


**Figure 2.3:** BE and EF Packet Urgency Functions

### 2.5.2 Packet Priority Function

Let $\psi_i^{j,z}(k) \in [0, 1]$ denote the packet priority (PP) value of user $i$, flow $j$, packet $z$ at time $k$. The PP value is calculated using the following packet priority function

$$\psi_i^{j,z}(k) = \pi_j(k)^{o_\pi} d_i^{j,z}(k)^{o_d} u_i^{j,z}(k)^{o_u}, \tag{2.5}$$

where $o_\pi$, $o_d$, and $o_u \in \mathbb{R}_+$ are non-negative weighting constants. Each component of $\psi_i^{j,z}(k)$ is normalized to its maximum value: $\pi_j(k)$ is normalized to $\max(\pi_j(k)) \; \forall \, j$, $d_i^{j,z}(k)$ is normalized to $\max(d_i^{j,z}(k)) \; \forall \, i, j, z$ and $u_i^{j,z}(k)$ is normalized to $\max(c_j \xi_j^{(T_j - \eta_j)}) \; \forall \, j$.

### 2.5.3 Flow Merging Policy

The objective of the flow merging policy is to merge packets from the different flow data buffers of a given user $i$ into a physical layer encoder single user packet at time $k$ such that the sum PP value of the selected packets is maximized subject to the $EPSize_i(k)$ and FIFO packet service constraints. The flow merging policy is formulated as follows:

$$\begin{aligned} \text{OP2.1:} \quad \max_{a_i^{j,z}(k)} \quad & \sum_{j=1}^{J_i} \sum_{z=1}^{B_i^j(k)} \psi_i^{j,z}(k) a_i^{j,z}(k) \\ \text{s. t.} \quad & \sum_{j=1}^{J_i} \sum_{z=1}^{B_i^j(k)} d_i^{j,z}(k) a_i^{j,z}(k) \le EPSize_i(k), \\ & a_i^{j,z}(k) \in \{0,1\}, \qquad \forall\, j \in \mathcal{J}_i, \\ & \qquad\qquad\qquad\qquad \forall\, z \in \{1,\dots,B_i^j(k)\}, \\ & a_i^{j,z}(k) \le a_i^{j,z'}(k), \qquad \forall\, z' \le z, \end{aligned} \tag{2.6}$$

where the binary variable $a_i^{j,z}(k) = 1$ if user $i$, flow $j$, packet $z$ is selected, and $a_i^{j,z}(k) = 0$ otherwise. The constraint $a_i^{j,z}(k) \le a_i^{j,z'}(k)$, $\forall\, z' \le z$ ensures that the packets in any data buffer are serviced in a FIFO fashion. The optimal solution is denoted by $\boldsymbol{A}^*$, where $\boldsymbol{A}^*$ is a binary matrix consisting of elements $a_i^{j,z}(k)$ that maximizes the PP sum of the objective function in OP2.1.

To obtain the optimal solution $\boldsymbol{A}^*$, we first determine the set of unique feasible solutions, denoted by $\mathcal{Y}$, where each element $\boldsymbol{y}$ is a vector consisting of $J_i$ elements. The $j^{th}$ element in $\boldsymbol{y}$ represents the number of data packets selected from flow $j$ that satisfies the constraints of the optimization problem formulated in OP2.1. The optimal solution $\boldsymbol{A}^*$ is then mapped by $\boldsymbol{y}^* \in \mathcal{Y}$ which maximizes the objective function. In the event of a tie, $\boldsymbol{y}^*$ is then selected randomly with equal probabilities. The set $\mathcal{Y}$ can be iteratively determined using $J_i$-nested loops. The loop counter for each nested loop $j$ is $[0,\dots,B_i^j(k)]$ and represents the number of data packets selected from flow $j$. A loop terminates when the total size of the selected data packets exceeds $EPSize_i(k)$.

Let $U_i(k)$ denote the maximal sum packet priority (MSPP) value for user $i$ at time $k$ attained by $\boldsymbol{A}^*$, i.e.

$$U_i(k) = \sum_{j=1}^{J_i} \sum_{z=1}^{B_i^j(k)} \psi_i^{j,z}(k) a_i^{j,z}(k), \qquad a_i^{j,z}(k) \in \boldsymbol{A}^*. \tag{2.7}$$

### 2.5.4 User Selection Policy

Let $Q_i(k)$ denote the priority of user $i$ at scheduling period $k$:

$$Q_i(k) = \kappa_i \frac{CSI_i(k)^{\alpha} U_i(k)^{\beta}}{\bar{v}_i(k)^{\varepsilon}}, \tag{2.8}$$

where $CSI_i(k)$ is the channel state information, $U_i(k)$ is the MSPP value and $\bar{v}_i(k)$ denotes the running average throughput over the last $N_{window}$ time slots for user $i$ at time $k$. The parameter $\kappa_i \in \mathbb{R}_+$ can be used to establish relative user priorities and $\alpha$, $\beta$, $\varepsilon \in \mathbb{R}_+$ are non-negative weighting constants. The user $i^*$ to be scheduled at time $k$ is determined as:

$$i^*(k) = \arg\max_{i \in \mathcal{I}} Q_i(k). \tag{2.9}$$

## 2.6 Simulation Results

The MFM and ACLS-FM scheduling policies described in Sections 2.4.1 and 2.5 were simulated in Matlab using the system model described in Section 2.3. Simulation results were obtained for BE only traffic, EF only traffic and mixed traffic (BE + EF) scenarios with system loading factor, $\rho \triangleq \overline{\lambda}/\overline{\mu}$, values of 0.10, 0.50 and 0.90, where $\overline{\lambda} = \sum_{i=1}^{I} \sum_{j=1}^{J_i} \lambda_i^j$ and $\overline{\mu} = \frac{1}{\|\mathcal{S}\| \|\mathcal{E}\|} \sum_{S \in \mathcal{S}} \sum_{EPSize \in \mathcal{E}} \frac{EPSize}{S}$. $\|\cdot\|$ is the cardinality of a set. For the mixed traffic scenario, an equal number of BE and EF traffic flows were simulated. To achieve the desired system loading factor, $I$ and $J_i$ were varied. The simulation parameter values are listed in Table 2.1. The scheduling delay thresholds were set at 3.0 sec for BE traffic class (to avoid Transmission Control Protocol (TCP) retransmissions) [51] and 0.070 sec for EF traffic class (to achieve a “Users Satisfied” mouth-to-ear delay rating) [52].

**Table 2.1:** Simulation Parameter Values

| **Parameter** | **Value** | **Parameter** | **Value** |
|---|---|---|---|
| $N_{window}$ | 100 slots | $S$ | 1 slot |
| $d_i^{j,z}(k)$ | 128 bits $\forall\, i \in \mathcal{I}$<br>$\forall\, j \in \mathcal{J}_i,\, z = 1, ..., B_i^j(k)$ | $\xi_{BE}$ | 1.0 |
| $\pi_j(k)$ | $1\ \forall\, j \in \mathcal{J}_i, k = 1, ..., K$ | $\xi_{EF}$ | 1.5 |
| $\lambda_j$ | $1\ \forall\, j \in \mathcal{J}_i$ | $T_{BE}$ | 3.000 sec |
| $o_\pi, o_d, o_u$ | 1 | $T_{EF}$ | 0.070 sec |
| $\kappa_i$ | $1\ \forall\, i \in \mathcal{I}$ | $\eta_{BE}$ | 1.500 sec |
| $\alpha, \beta, \varepsilon$ | 1 | $\eta_{EF}$ | 0.035 sec |

### 2.6.1 Comparative Scheduling Policies

The performance of the MFM and ACLS-FM scheduling policies are compared with those of four other scheduling policies: Modified Greedy (MG), Modified Round Robin (MRR), MFM and Modified Proportional Fair (MPF). The MG, MRR and MPF scheduling policies are described below. The term, $EPSize_i(k)$, denotes the physical layer encoder packet size for user $i$ at time $k$.

#### MG Scheduling Policy

The Classical Greedy (CG) scheduling policy [53] $i^*(k) = \arg\max_{i \in \mathcal{I}} CSI_i(k)$ is strictly opportunistic and simply selects the user $i^*$ with the best channel condition. While CG provides a throughput upper-bound, it does not specify how the flows of the selected user are to be scheduled. In MG, each traffic flow is regarded as a separate user. At each scheduling period $k$, MG services the flow with the best channel condition and longest data queue. Specifically,

Step 1: Let $Q_{MG}{}_i^j(k)$ denote the priority of user $i$, flow $j$ at scheduling period $k$:

$$Q_{MG}{}_i^j(k) = \min\{EPSize_i(k), \sum_{z=1}^{B_i^j(k)} d_i^{j,z}(k)\}, \qquad \forall\, i \in \mathcal{I}, j \in \mathcal{J}_i. \tag{2.10}$$

The user and flow to be scheduled at time $k$ is determined as:

$$(i^*(k), j^*(k)) = \arg\max_{\substack{i \in \mathcal{I} \\ j \in \mathcal{J}_i}} Q_{MG}{}_i^j(k). \tag{2.11}$$

Step 2: Packets are selected, one at a time in an iterative fashion, from the data queue of user $i^*$, flow $j^*$ and added to the physical layer encoder packet until either the physical layer encoder packet $EPSize_{i^*}(k)$ is filled or that there are no more packets in the data queue $j^*$.

### MRR Scheduling Policy

The MRR scheduling policy assigns equal service to each traffic flow and in order regardless of queue length and channel condition. The MRR scheduling policy (where each of the $I$ users has $J_i$ flows) is specified as follows:

Step 1: The user and flow to be scheduled at time $k$ is determined as:

$$\begin{aligned} i^*(k) &= \left\lceil \frac{(k-1) \text{ mod } IJ_i + 1}{J_i} \right\rceil, \\ j^*(k) &= (k-1) \text{ mod } J_i + 1. \end{aligned} \tag{2.12}$$

Step 2: Same as Step 2 of the MG scheduling policy.

### MPF Scheduling Policy

The PF scheduling policy [9] exploits multi-user diversity to maximize system throughput by scheduling data transmission based on the relative channel quality of the competing users, while at the same time maintaining fairness across users. Note that in the classic PF scheduling policy, there is no provision for choosing which flow to schedule from among the flows of a given user. Thus for purposes of comparison, each traffic flow is regarded as a separate user and we refer to this as the MPF scheduling policy. The MPF scheduling policy is defined as:

Step 1: Let ${Q_{MPF}}_i^j(k)$ denote the priority of user $i$, flow $j$ at scheduling period $k$:

$$ {Q_{MPF}}_i^j(k) = \begin{cases} \dfrac{CSI_i(k)}{\overline{v}_i^j(k)} & \text{if } B_i^j(k) > 0 \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}_i, \tag{2.13} $$

where $\overline{v}_i^j(k)$ denotes the running average throughput over the last $N_{window}$ time slots for user $i$, flow $j$ at time $k$. A flow with no data to send is assigned a priority of 0 and is ignored in the selection process. The user and flow to be scheduled at time $k$ is determined as:

$$ (i^*(k), j^*(k)) = \arg\max_{\substack{i \in \mathcal{I} \\ j \in \mathcal{J}_i}} {Q_{MPF}}_i^j(k). \tag{2.14} $$

Step 2: Same as Step 2 of the MG scheduling policy.

### 2.6.2 Performance Measure

To evaluate the system performance, we define the scheduling policy performance gain, $G_\chi^Y$, as

$$ G_\chi^Y = C \, \frac{\overline{\chi_Y} - \overline{\chi}}{\overline{\chi}} \times 100\%, \tag{2.15} $$

where $Y$ is either the MFM or ACLS-FM scheduling policy and $\chi$ is the QoS measure of interest: throughput (*TP*), latency (*LT*) and packet drop probability (*PDP*). The term $C$ in (2.15) takes value $+1$ for *TP* and $-1$ for *LT* and *PDP*. The terms $\overline{\chi_Y}$ and $\overline{\chi}$ are the average QoS values for scheduling policy $Y$ and MPF respectively. The MPF scheduling policy is used for evaluating the system performance as it (or its variants) is the most commonly used scheme in wireless networks.

### 2.6.3 MFM Results

Some simulation results are shown in Table 2.2 for $\rho = 0.90$. The $G_{TP}^{MFM}$, $G_{LT}^{MFM}$ and $G_{PDP}^{MFM}$ columns show the throughput, latency and packet drop probability gains of MFM compared to MPF. Cumulative distribution function (CDF) plots for user throughput, user latency and

**Table 2.2:** Simulation Results of MFM with $\rho = 0.90$

| $I$ | $J_i$ | $G_{TP}^{MFM}$ | $G_{LT}^{MFM}$ | $G_{PDP}^{MFM}$ |
|---|---|---|---|---|
| 20 | 2 BE | 0.19 % | 0.00 % | 0.00 % |
| 110 | 2 EF | 52.14 % | 0.00 % | 13.15 % |
| 22 | 10 EF | 246.83 % | 23.53 % | 60.98 % |
| 30 | 1 BE, 1 EF | 22.87 % | 17.60 % | 13.64 % |
| 6 | 5 BE, 5 EF | 36.83 % | 67.97 % | 20.80 % |

user packet drop probability obtained from a simulation of 22 users, each with 10 EF traffic flows are shown in Fig. 2.4, 2.5 and 2.6 respectively. The improvements in $G_{TP}^{MFM}$ and $G_{LT}^{MFM}$ relative to MPF come from the reduction in wastage in the physical layer encoder packet when the flow queue sizes (in bits) are typically quite small compared to the physical layer encoder packet size. The results show that $G_{TP}^{MFM}$ increases as the system loading factor increases for both the EF only traffic and mixed traffic scenarios but is negligible for the BE only traffic scenario due to minimal unfilled space left in the physical layer encoder packet for merging. For the same system loading factor, $G_{TP}^{MFM}$ increases as the number of traffic flows is increased with a corresponding decrease in the number of users due to multi-application diversity. Further decrease in latency is realized due to the possible multiplexing of packets from different data queues in a scheduling period. Application layer PDU from EF flows that arrive later at the access network following a large application layer PDU from a BE flow can be transmitted first instead of being transmitted in the order of arrival, hence reducing latency. $G_{LT}^{MFM}$ exhibits the highest gain in a mixed traffic scenario. As expected, the results show that an increase in $G_{LT}^{MFM}$ results in a corresponding increase in $G_{PDP}^{MFM}$.

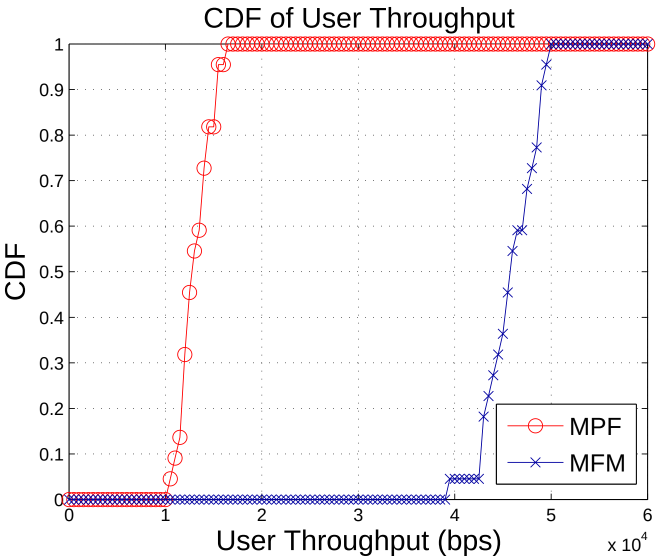


**Figure 2.4:** CDF of User Throughput, $\rho = 0.90$, $I = 22$, 10 EF Flows for each User

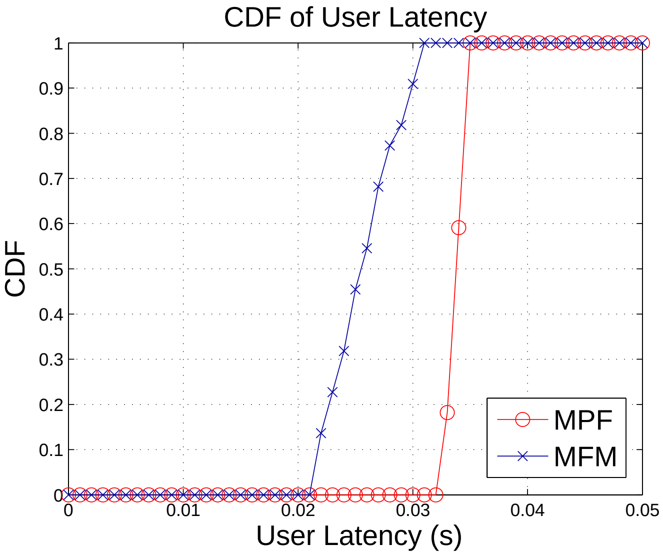


**Figure 2.5:** CDF of User Latency, $\rho = 0.90$, $I = 22$, 10 EF Flows for each User

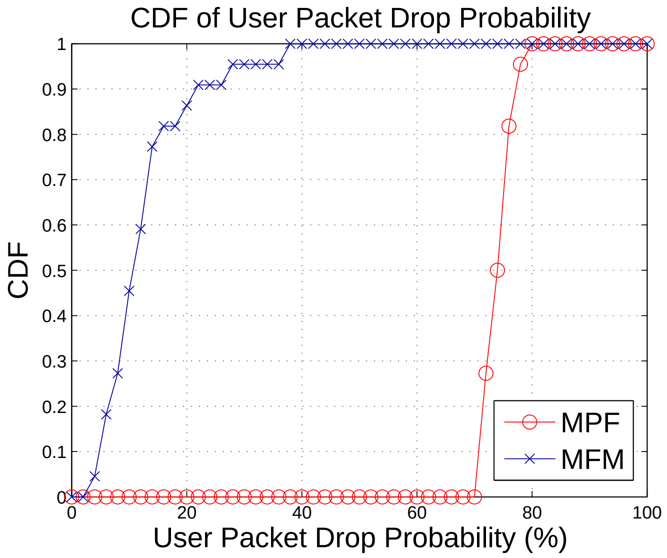


**Figure 2.6:** CDF of User Packet Drop Probability, $\rho = 0.90$, $I = 22$, 10 EF Flows for each User

### 2.6.4 ACLS-FM Results

Some simulation results are shown in Table 2.3 for $\rho$ = 0.10, 0.50 and 0.90. The $G_{TP}^{ACLS-FM}$, $G_{LT}^{ACLS-FM}$ and $G_{PDP}^{ACLS-FM}$ columns show the throughput, latency and packet drop probability gains of ACLS-FM compared to MPF. CDF plots for user throughput, user latency and user packet drop probability for a system with $\rho = 0.90$ for EF only and mixed traffic (BE + EF) are shown in Fig. 2.7 and Fig. 2.8 respectively.

The simulation results confirm that ACLS-FM generally performs better than the other four scheduling policies defined in Section 2.6.1 in terms of user throughput, user latency and user packet drop probability. We note in Fig. 2.8a that while MG has a higher system aggregate throughput, ACLS-FM can have a higher user throughput compared to MG. In addition to exploiting the benefits of MFM, ACLS-FM achieves additional performance gains from the PU function defined in Section 2.5.1, which allows a packet from a EF (latency-sensitive) flow to have its urgency increased as its waiting time, $w_i^{EF,\,z}$, exceeds a predetermined threshold, $\eta_{EF}$, to meet its latency requirements. For the period $w_i^{BE,\,z} < \eta_{EF}$, packets from the BE (delay-tolerant) flows are given a higher urgency to reduce the buffer backlog as a mechanism to achieve a higher system throughput. The PP function defined in Section 2.5.2 allows for further intra-user adjustments through the coupling of flow priority, $\pi_j$, packet size, $d_i^{j,z}$, and packet urgency, $u_i^{j,z}$. Fairness among users is taken into account in the user selection policy defined in Section 2.5.4.

For BE only traffic, as shown in Table 2.3a, ACLS-FM provides little performance gains over MPF regardless of the system loading factor. This is due to the fact that BE only traffic is high data rate and bursty in nature, leaving minimal unfilled space in the physical layer encoder packet for exploiting MFM.

However, for EF only traffic, as shown in Fig. 2.7, ACLS-FM provides significant per-

**Table 2.3:** Results of ACLS-FM for (a) BE only (b) EF only (c) BE + EF

**(a)** BE only

| $\rho$ | $I$ | $J_i$ | $G_{TP}^{ACLS-FM}$ | $G_{LT}^{ACLS-FM}$ | $G_{PDP}^{ACLS-FM}$ |
|---|---|---|---|---|---|
| 0.10 | 3 | 2 BE | 0.00 % | 0.52 % | 0.00 % |
| 0.50 | 10 | 2 BE | 0.79 % | 0.00 % | 0.44 % |
| 0.90 | 20 | 2 BE | 0.43 % | 2.59 % | 0.00 % |

**(b)** EF only

| $\rho$ | $I$ | $J_i$ | $G_{TP}^{ACLS-FM}$ | $G_{LT}^{ACLS-FM}$ | $G_{PDP}^{ACLS-FM}$ |
|---|---|---|---|---|---|
| 0.10 | 20 | 2 EF | 7.53 % | 52.63 % | 6.93 % |
| 0.50 | 60 | 2 EF | 91.37 % | 3.13 % | 36.48 % |
| 0.90 | 110 | 2 EF | 87.06 % | 2.94 % | 21.88 % |
| 0.90 | 22 | 10 EF | 304.02 % | 50.00 % | 74.85 % |
| 0.90 | 14 | 16 EF | 300.78 % | 67.65 % | 74.69 % |

**(c)** BE + EF

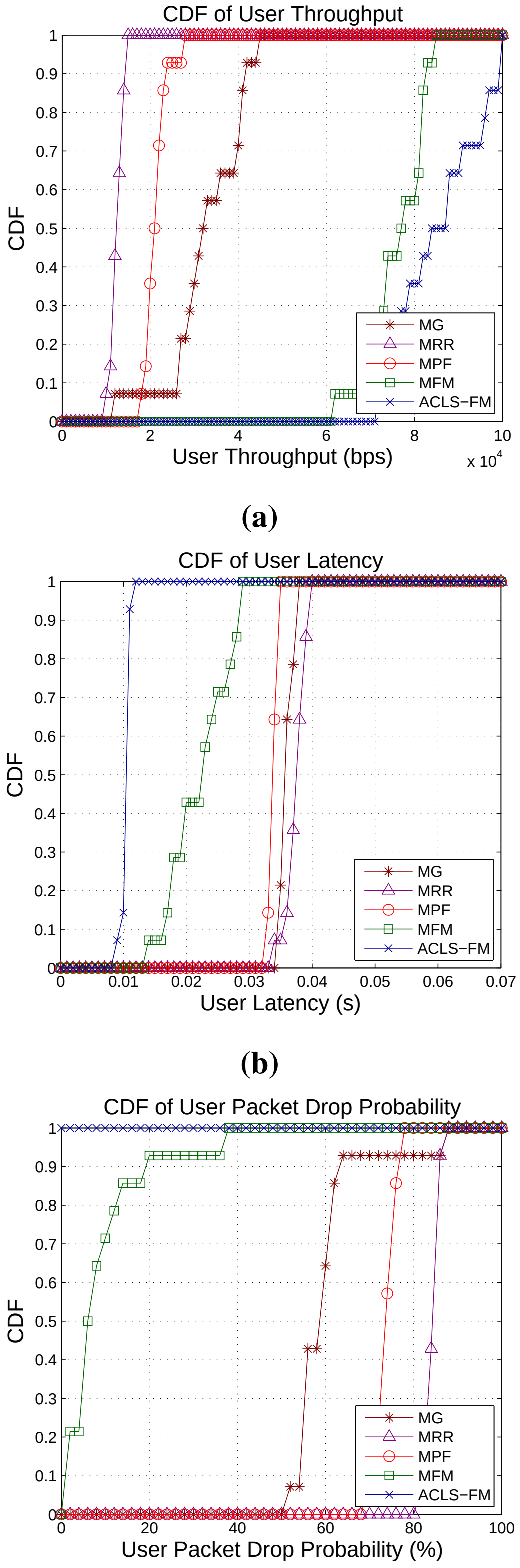
| $\rho$ | $I$ | $J_i$ | $G_{TP}^{ACLS-FM}$ | $G_{LT}^{ACLS-FM}$ | $G_{PDP}^{ACLS-FM}$ |
|---|---|---|---|---|---|
| 0.10 | 5 | 1 BE, 1 EF | 0.00 % | 10.92 % | 0.00 % |
| 0.50 | 20 | 1 BE, 1 EF | 31.54 % | 28.79 % | 16.18 % |
| 0.90 | 4 | 8 BE, 8 EF | 114.15 % | 45.80 % | 35.95 % |

formance gains, especially as $\rho$ increases. As with the MFM scheduling policy, the gain of ACLS-FM is achieved through a reduction of wastage in the physical layer encoder packet. In addition, with the inclusion of the packet urgency function $u_i^{j,z}(k)$ in the ACLS-FM scheduling policy, $G_{PDP}^{ACLS-FM}$ is achieved as the number of packets dropped due to the violation of the scheduling delay threshold is substantially reduced, which in turn leads to additional $G_{TP}^{ACLS-FM}$. Further $G_{LT}^{ACLS-FM}$ is realized due to the consideration of the MSPP $U_i(k)$ of a user in the user selection policy in (2.9) which selects a user with more urgent packets. Comparing the cases of $I = 110$, $J_i = 2$ and $I = 22$, $J_i = 10$ for $\rho = 0.90$ shown in Table 2.3b, we see that as the number of flows per user increases, a corresponding increase in performance gains is obtained due to the exploitation of multi-application diversity. For $\rho = 0.90$ and 16 EF flows per user shown in Fig. 2.7c, ACLS-FM achieves a near-0% *PDP* in comparison to an average of 45% *PDP* for the other 4 scheduling policies at the 95th percentile.

For the mixed traffic (BE+EF) scenario shown in Fig. 2.8, ACLS-FM has the second highest throughput performance. MG provides the best throughput performance at the expense of starving EF traffic as it also has the highest EF *PDP* as shown in Fig. 2.8c. On the other hand, ACLS-FM achieves a near-0% *PDP* for EF traffic and the second lowest *PDP* for BE traffic. As shown in Fig. 2.8b, ACLS-FM has the lowest latency for BE traffic, and while MFM has a lower latency for EF traffic than ACLS-FM, that is achieved at the expense of a 50% EF *PDP* at the 95th percentile (shown in Fig. 2.8c). In a mixed traffic scenario, MFM has a higher EF than BE *PDP* shown in Fig. 2.8c as the flow merging policy for MFM determines the probability of merging a packet from a flow by the ratio of its queue length to the sum of all queue lengths. It is worth noting that while MPF has the second lowest EF *PDP*, it also has the second highest BE *PDP* as it trades-off BE packets to achieve its intended throughput fairness objective. On the other hand, MG trades-off EF packets (highest EF *PDP*) for BE packets to achieve a high throughput.

## 2.7 Conclusion

The performance gains of a scheduling policy which exploits MFM in terms of user throughput, user latency and user packet drop probability were quantified. The substantial gains of MFM results from wastage reduction in the physical layer encoder packet and multiplexing of ackets with different latency tolerances in a scheduling period. Only queue length information is needed to implement the MFM scheduling policy. With the promising gains and simplicity in implementation of MFM, we propose an ACLS-FM scheduling policy that integrates MFM and jointly considers physical-layer time-varying channel conditions as well as application-layer QoS requirements. In addition to exploiting the benefits of MFM, ACLS-FM realizes additional performance gains through the use of a cross layer design, utilizing a packet urgency function, packet priority function, flow merging policy and user selection policy. The simulation results confirm that ACLS-FM achieves substantial performance gains in the considered QoS performance measures (user throughput, user latency and user packet drop probability) when compared to other commonly used scheduling policies.

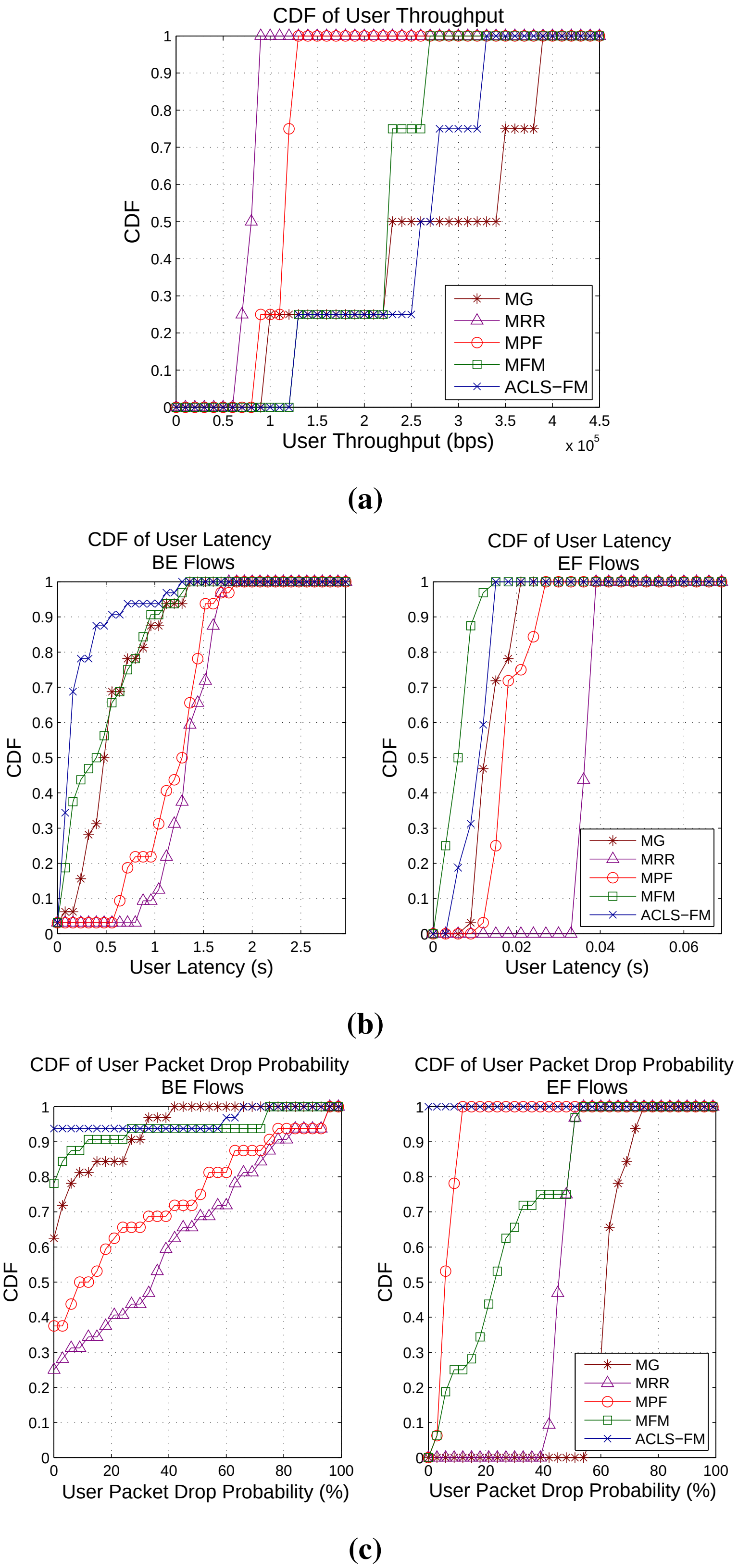


**(a)**

**(b)**

**(c)**

**Figure 2.7:** Performance for a System with $\rho = 0.90$, $I = 14$, 16 EF Flows for each User (a) CDF of User Throughput (b) CDF of User Latency (c) CDF of User Packet Drop Probability



**(a)**

**(b)**

**(c)**

**Figure 2.8:** Performance for a System with $\rho = 0.90$, $I = 4$, 8 BE Flows and 8 EF Flows for each User (a) CDF of User Throughput (b) CDF of User Latency (c) CDF of User Packet Drop Probability

# Chapter 3

# Packet Division Multiplexing in Single-carrier CDMA Systems [2]

## 3.1 Introduction

With the rapid introduction of multimedia services, wireless networks are expected to integrate a mix of real-time traffic and non-real-time traffic with different QoS requirements. This has driven the continued extensive research in RRM with the objective of improving achievable system capacity while at the same time meeting the diverse QoS requirements and adapting to the dynamically changing wireless conditions.

As part of the evolution of the cdma2000 family of 3G mobile telecommunications air interface, cdma2000 1xEV-DO Revision A [43] provides significant improvements at various protocol layers over cdma2000 1xEV-DO Release 0 [46]. These include higher peak data rates, HARQ transmission and enhancements that provide considerable gains in spectral efficiency and substantial QoS support to efficiently support both latency-sensitive and delay-tolerant applications. In addition, cdma2000 1xEV-DO Revision A also introduced PDM [43, 47, 48] in the forward link that permits the BS to service multiple users in the

[2] The material in this chapter is based on: C. E. Huang and C. Leung, "Downlink mixed-traffic scheduling with packet division multiplexing," in *Proc. ACM PM2HW2N*, Oct. 2008, pp. 165–172. © 2008 ACM. http://dx.doi.org/10.1145/1454630.1454655

same physical layer encoder packet in a single time slot with the use of *multi-user packet* (MUP) transmission. PDM not only improves the resource utilization (packing efficiency) by allowing delay-tolerant applications to fill up the physical layer encoder packet unused with higher priority, low rate latency-sensitive applications but also improves the transmission latency performance by overcoming the shortage of time slots and enables cdma2000 1xEV-DO Revision A to support a large number of low-rate latency-sensitive applications, leading to increased system throughput and spectral efficiency.

The feasibility of supporting a single traffic type with PDM in cdma2000 1xEV-DO Revision A is explored in [54]. Analytical models and simulation are developed to evaluate the expected capacity and delay performance of implementing VoIP traffic using cdma2000 1xEV-DO Revision A. The authors demonstrate in the study that MUP transmission plays a critical role in achieving the expected Erlang capacity for VoIP which is comparable to that of a circuit switched cdma2000 [55] system. MUP efficiency, in terms of the average number of VoIP packets contained in one physical layer encoder packet is also presented. In [56], the performance and capacity of VoIP traffic by itself and VoIP together with other traffic types are analyzed. It is shown that cdma2000 1xEV-DO Revision A can not only provide VoIP capacity that is comparable to IS-2000, but the simulation results also show that a significant amount of delay-tolerant traffic can be simultaneously supported along with VoIP.

In this chapter, we leverage upon the PDM and mixed traffic findings in [54] and [56] respectively and adopt the ACLS-FM scheduling policy approach introduced in Chapter 2. ACLS-FM integrates MFM (Section 2.4) and takes into account the time-varying channel conditions from the physical layer and QoS requirements from the application layer. Considerable performance gains achievable with ACLS-FM in terms of user throughput, user latency and user packet drop probability were quantified in Section 2.6. We extend ACLS-FM and propose an adaptive cross layer scheduling policy that incorporates PDM of the shared physical layer encoder packet. We refer to this scheme as the Adaptive Cross Layer Scheduling with Flow and User Multiplexing (ACLS-FUM) scheduling policy. We consider a mix of real-time voice services and non-real-time data applications and study the improvements in

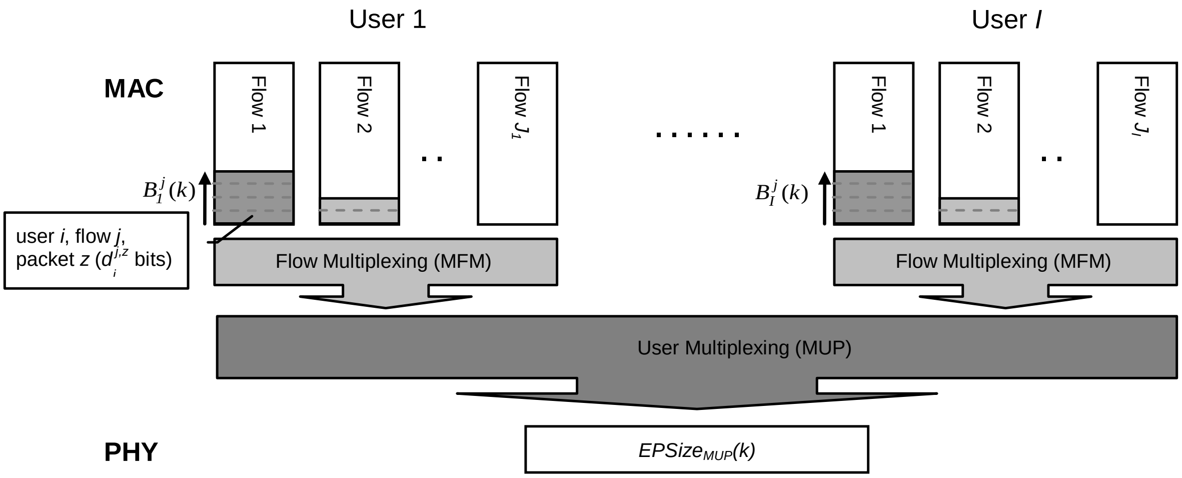


**Figure 3.1:** Illustration of MUP with MFM

packing efficiency and latency performance. The resulting performance gains that are realizable in a framework such as that provided in cdma2000 1xEV-DO Revision A are quantified. An illustration of MUP with MFM is shown in Fig. 3.1.

This chapter is organized as follows: in Section 3.2, we present the system model that includes the network model, traffic classes and data buffer parameters. The ACLS-FUM scheduling policy is described in Section 3.3. Simulation results are presented in Section 3.4 and the main findings are summarized in Section 3.5.

## 3.2 System Model

The network model used in this chapter is described in this section. The traffic classes and data buffer parameters used are described in Sections 2.3.2 and 2.3.3, respectively.

We consider a 1xEV-DO Revision A-like packet cellular network random discrete-event model, as shown in Fig. 2.1, consisting of one BS servicing $I$ mobile stations (MSs). Let $\mathcal{I} = \{1, ..., I\}$ be the set of all users (MSs). Each MS $i$ can have up to $J_i$ data queues (flows) and let $\mathcal{J}_i = \{1, ..., J_i\}$ be the set of all flows for MS $i$. Forward link scheduling is centralized at the BS which communicates with all MSs. At each time slot $k$, where $k \in \mathbb{Z}_+ = \{1, 2, ..., K\}$, only one user is scheduled for *single-user packet* (SUP) transmission or up to eight users are scheduled for MUP transmission. We assume that the scheduling decision time is negligible.

Power control is not enabled in the forward link for 1xEV-DO systems and the BS transmits at full power to the MSs in all time slots. We assume that packets are received without errors, i.e. 0% FER, between the BS and MS. This simplifying assumption is made to illustrate the potential gains that ACLS-FUM can provide. It is expected that gains will also be achievable in a more realistic setting with non-zero FER values. The BS is assumed to have knowledge of the channel state information, $CSI_i(k)$, for each MS $i$ at time $k$, queue status and QoS requirements for all the data queues. The maximum service rate for a user during time slot $k$ is a function of its channel quality which is characterized by its received SNR in time slot $k$. Multiple application layer protocol data units (PDUs) from the same user can be transmitted in the same physical layer encoder packet in the same time slot using SUP transmission. Furthermore, application layer PDUs destined for different users are either scheduled and transmitted in different time slots using SUP transmission or multiplexed into the same physical layer encoder packet and transmitted in the same time slot using MUP transmission.

For simplicity, we assume that the physical layer encoder packet size, $EPSize_i(k)$, is chosen according to a uniform distribution from the set of eight physical layer encoder packet sizes, where $EPSize \in \mathcal{E}_{\mathcal{SUP}} = \{128, 256, 512, 1024, 2048, 3072, 4096, 5120\}$ bits for SUP transmission. For MUP transmission, $EPSize \in \mathcal{E}_{\mathcal{MUP}} = \{1024, 2048, 3072, 4096, 5120\}$ bits and maps to the set of Date Rate Control (DRC) indices compatible with MUP transmission for data rates greater than 153.6 kbps [43]. Let $\mu \triangleq EPSize/S \in \{4.8, ..., 3072\}$ kbps be the effective service rate, where $S \in \mathcal{S} = \{1, 2, 4, 8, 16\}$ time slot(s), each of 1.667 ms duration. Each application layer PDU is segmented into a number of packets. Packets from up to $J_i$ = 16 different flows and $I$ = 8 different users can be multiplexed into the same physical layer encoder packet of size $EPSize_{MUP}(k)$ for transmission at time $k$.

## 3.3 Adaptive Cross Layer Scheduling with Flow and User Multiplexing Scheduling Policy

The proposed ACLS-FUM scheduling policy consists of a packet urgency function (to meet latency requirements), a packet priority function (for intra-user QoS adjustments), a transmission mode selection function (to determine SUP/ MUP transmission mode), SUP transmission mode, MUP transmission mode and a flow merging policy (to determine which flows and how many bits from each flow to service). The packet urgency function, packet priority function and flow merging policy are described in Chapter 2. A flow chart illustrating the ACLS-FUM scheduling policy is shown in Fig. 3.2.

### 3.3.1 Transmission Mode Selection Function

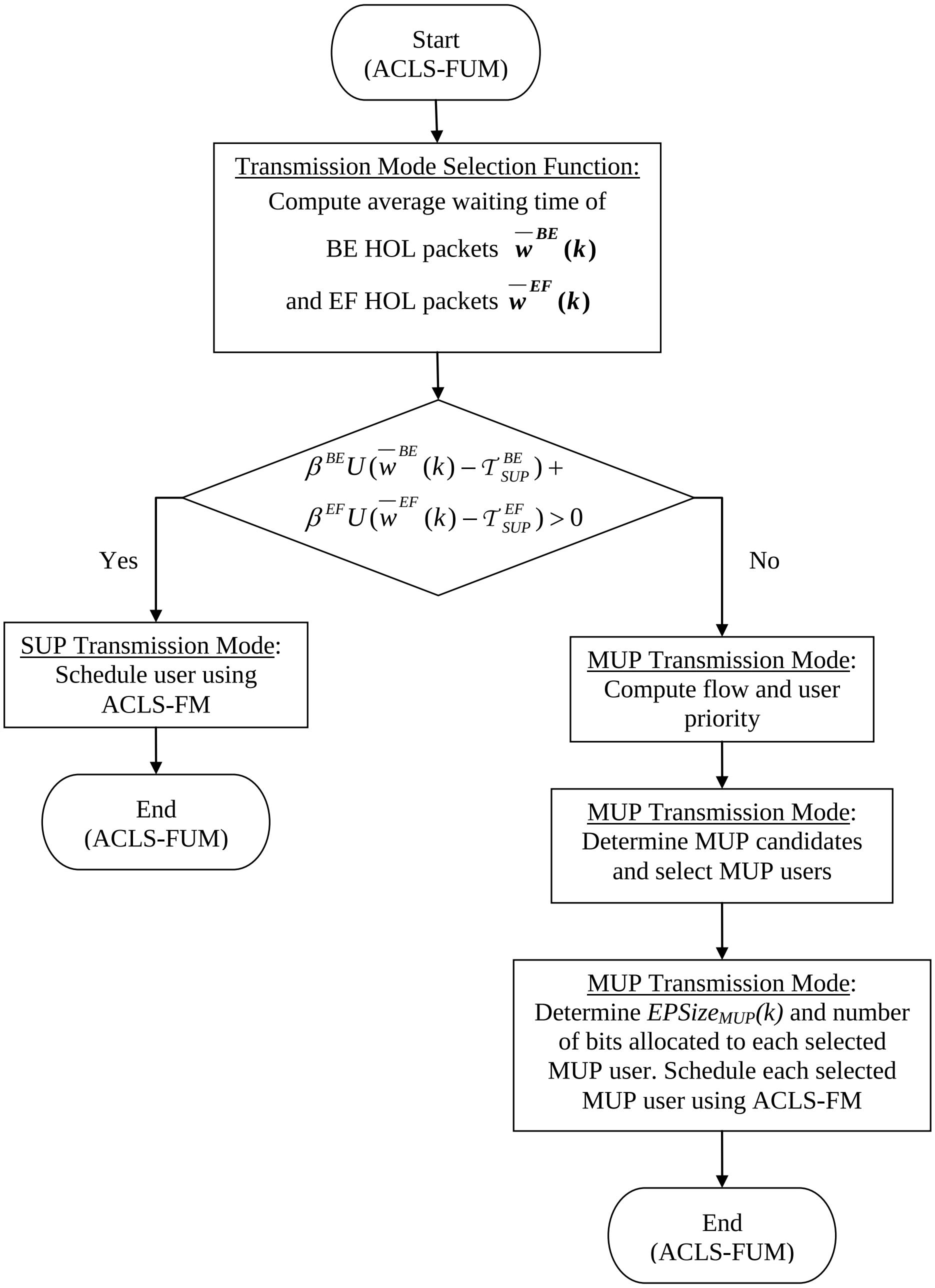
In order to give users that do not qualify for MUP transmission (cdma2000 1xEV-DO Revision A [43] precludes DRC index $< 3$ from MUP transmission) an opportunity to clear their backlog, the scheduling policy may transmit packets in SUP transmission mode if the average waiting time of the head-of-line (HOL) packets exceeds predefined thresholds. Specifically, we define the averaging waiting time of BE and EF HOL packets as follows

$$\overline{w}^{BE}(k) = \frac{1}{\displaystyle\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\mathbb{I}_{BE}(j)}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\mathbb{I}_{BE}(j)w_i^{j,HOL}(k) \tag{3.1}$$

$$\overline{w}^{EF}(k) = \frac{1}{\displaystyle\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\mathbb{I}_{EF}(j)}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\mathbb{I}_{EF}(j)w_i^{j,HOL}(k) \tag{3.2}$$

where $w_i^{j,HOL}(k)$ denotes the waiting time of the HOL packet for user $i$, flow $j$ at time $k$. The indicator function $\mathbb{I}_{BE}(j)$ is defined as

$$\mathbb{I}_{BE}(j) = \begin{cases} 1 & \text{if } j \text{ is an BE flow} \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$



Start
(ACLS-FUM)
Transmission Mode Selection Function:
Compute average waiting time of
BE HOL packets $\overline{\boldsymbol{w}}^{\boldsymbol{BE}}(\boldsymbol{k})$
and EF HOL packets $\overline{\boldsymbol{w}}^{\boldsymbol{EF}}(\boldsymbol{k})$
$\beta^{BE}U(\overline{w}^{BE}(k)-\mathcal{T}^{BE}_{SUP})+$
$\beta^{EF}U(\overline{w}^{EF}(k)-\mathcal{T}^{EF}_{SUP})>0$
Yes
No
SUP Transmission Mode:
Schedule user using
ACLS-FM
End
(ACLS-FUM)
MUP Transmission Mode:
Compute flow and user
priority
MUP Transmission Mode:
Determine MUP candidates
and select MUP users
MUP Transmission Mode:
Determine $EPSize_{MUP}(k)$ and number
of bits allocated to each selected
MUP user. Schedule each selected
MUP user using ACLS-FM
End
(ACLS-FUM)

**Figure 3.2:** ACLS-FUM Scheduling Policy Flow Chart

$\mathbb{I}_{EF}(j)$ is similarly defined. The terms $\sum_{j \in \mathcal{J}_i} \mathbb{I}_{BE}(j)$ and $\sum_{j \in \mathcal{J}_i} \mathbb{I}_{EF}(j)$ represent the number of BE and EF flows for user $i$ respectively. The ACLS-FUM scheduling policy will use SUP transmission mode at time $k$ if the following condition is true

$$\beta^{BE} U(\overline{w}^{BE}(k) - \mathcal{T}^{BE}_{SUP}) + \beta^{EF} U(\overline{w}^{EF}(k) - \mathcal{T}^{EF}_{SUP}) > 0, \tag{3.4}$$

where $U(x)$ denotes the unit step function, i.e. $U(x) = 1$ for $x > 0$ and $U(x) = 0$ otherwise. The parameters $\beta^{BE}$ and $\beta^{EF}$ take on values in $\{0, 1\}$ depending on whether BE and/or EF flows are included in the transmission mode selection function. $\mathcal{T}^{BE}_{SUP}$ and $\mathcal{T}^{EF}_{SUP}$ are the SUP thresholds for BE and EF flows respectively. They are generally set to values that are less than the scheduling delay thresholds $T_{BE}$ and $T_{EF}$. MUP transmission mode is used if (3.4) is false.

### 3.3.2 SUP Transmission Mode

In SUP transmission mode, only one user is serviced. Multiple packets from the same user are selected using ACLS-FM (see Section 2.5) and packed into the same physical layer encoder packet in the same time slot for SUP transmission.

### 3.3.3 MUP Transmission Mode

In MUP transmission mode, up to eight users are serviced. Multiple packets from different users are multiplexed into the same physical layer encoder packet in the same time slot for MUP transmission. The MUP transmission mode is performed as follows:

1) Flow and User Priority: Let $Q^j_i(k) \in [0, 1]$ denote the flow priority of user $i$, flow $j$ at time $k$, and be defined as

$$Q^j_i(k) = \frac{\alpha u^{j,HOL}_i(k) + (1 - \alpha) CSI_i(k)}{\overline{v}^j_i(k)}, \tag{3.5}$$

where $CSI_i(k)$ is the channel state information of user $i$ at time $k$, $u^{j,HOL}_i(k)$ is the PU value of the HOL packet and $\overline{v}^j_i(k)$ (for fairness consideration) denotes the running average throughput over the last $N_{window}$ time slots for user $i$, flow $j$ at time $k$. Each of these terms is normalized to its maximum value: $u_i^{j,HOL}(k)$ is normalized to $\max(c_j \xi_j^{(T_j-\eta_j)})$ $\forall j$, $CSI_i(k)$ which is mapped to $EPSize_i(k)$ is normalized to $\max\limits_{EPSize \in \mathcal{E}_{\mathcal{MUP}}} EPSize$ and $\overline{v}_i^j(k)$ is normalized to $\max\limits_{EPSize \in \mathcal{E}_{\mathcal{MUP}}} EPSize/S$. The parameter $\alpha \in [0,1]$ is a weighting constant that is used to adjust the relative weighting of HOL packet urgency and channel condition. The user priority of user $i$ at time $k$ is

$$Q_i(k) = \sum_{j=1}^{J_i} Q_i^j(k), \quad \forall\, i \in \mathcal{I}. \tag{3.6}$$

2) MUP User Selection: We define the set, $MUP_{cands}(k)$, of candidate users that qualify for MUP transmission at time $k$ as

$$MUP_{cands}(k) = \{\, i \in \mathcal{I} \mid EPSize_i(k) \geq 1024\} \tag{3.7}$$

and let $i^{max}(k)$ denote the MUP candidate which has the largest user priority:

$$i^{max}(k) = \arg \max_{i \in MUP_{cands}(k)} Q_i(k). \tag{3.8}$$

The set of MUP users, $MUP_{users}(k)$, to be scheduled at time $k$ is determined as:

$$MUP_{users}(k) = \{\, i \in MUP_{cands}(k) \mid CSI_i(k) \geq CSI_{i^{max}(k)}(k)\}. \tag{3.9}$$

In the case where $\|MUP_{users}(k)\| > 8$ ($\| \cdot \|$ denotes the cardinality of a set), the 8 users with the largest $Q_i(k)$ are selected for MUP transmission. In the event of any ties, the tied users are selected randomly with equal probabilities.

3) MUP User Bit Allocation: Let $EPSize_{MUP}(k)$ denote the physical layer encoder packet size used for MUP transmission and it is defined as

$$EPSize_{MUP}(k) = EPSize_{i^{max}(k)}(k). \tag{3.10}$$

The number of bits allocated to user $i$ for MUP transmission at time $k$ is denoted by $MUPSize_i(k)$. It is proportional to its user priority, $Q_i(k)$, which takes into account flow throughput fairness, HOL packet urgencies and user channel condition. $MUPSize_i(k)$ is determined by

$$MUPSize_i(k) = \frac{Q_i(k)}{\sum\limits_{i \in MUP_{users}(k)} Q_i(k)} EPSize_{MUP}(k), \qquad \forall\, i \in MUP_{users}(k). \tag{3.11}$$

Based on the number, $MUPSize_i(k)$, of bits allocated, packets for user $i \in MUP_{users}(k)$ are selected using ACLS-FM (see Section 2.5) and multiplexed into the same physical layer encoder packet in the same time slot for MUP transmission.

## 3.4 Simulation Results

The ACLS-FUM scheduling policy described in Section 3.3 was simulated in Matlab using the system model described in Section 3.2. Simulation results were obtained for BE only traffic, EF only traffic and mixed traffic (BE + EF) scenarios with system loading factor, $\rho \triangleq \overline{\lambda}/\overline{\mu}$, values of 0.10, 0.50 and 0.90, where $\overline{\lambda} = \sum\limits_{i=1}^{I} \sum\limits_{j=1}^{J_i} \lambda_i^j$ and $\overline{\mu} = \frac{1}{\|\mathcal{S}\| \|\mathcal{E}_{\mathcal{SUP}}\|} \sum\limits_{S \in \mathcal{S}} \sum\limits_{EPSize \in \mathcal{E}_{\mathcal{SUP}}} \frac{EPSize}{S}$. $\lambda_i^j$ is the average traffic arrival rate for user $i$, flow$j$. For the mixed traffic scenario, an equal number of BE and EF traffic flows were simulated. To achieve the desired system loading factor, $I$ and $J_i$ were varied. The simulation parameter values are listed in Table 3.1.

To evaluate the system performance, we define the ACLS-FUM scheduling policy performance gain, $G_{\chi}^{ACLS-FUM}$, as in (2.15). The performance metrics $G_{TP}^{ACLS-FUM}$, $G_{LT}^{ACLS-FUM}$, $G_{PDP}^{ACLS-FUM}$, $G_{JT,BE}^{ACLS-FUM}$ and $G_{JT,EF}^{ACLS-FUM}$ quantifies the throughput, latency, packet drop probability and jitter (BE and EF) gains of ACLS-FUM compared to MPF. CDF plots for user throughput, user latency, user packet drop probability and user jitter for a system with $\rho = 0.50$ for EF only and $\rho = 0.90$ for mixed traffic (BE + EF) are shown in Figs. 3.3 and 3.4 respectively. The simulation results confirm that ACLS-FUM performs better than the other

**Table 3.1:** Simulation Parameter Values

| **Param.** | **Value** | **Param.** | **Value** |
|---|---|---|---|
| $d_i^{j,z}(k)$ | 128 bits $\forall\, i \in \mathcal{I}$<br>$\forall\, j \in \mathcal{J}_i$, $z = 1, ..., B_i^j(k)$ | $\xi_{BE}$ | 1.0 |
| $N_{window}$ | 100 slots | $\xi_{EF}$ | 1.5 |
| $S$ | 1 slot | $T_{BE}$ | 3.000 sec |
| $c_j$ | 1 | $T_{EF}$ | 0.070 sec |
| $\mathcal{T}_{SUP}^{BE}$ | 1.500 sec | $\eta_{BE}$ | 1.500 sec |
| $\mathcal{T}_{SUP}^{EF}$ | 0.035 sec | $\eta_{EF}$ | 0.035 sec |
| $\beta^{BE}$ | 0 | $\alpha$ | 0.5 |
| $\beta^{EF}$ | 1 | | |

four scheduling policies defined in Section 2.6.1 in terms of user throughput, user latency, user packet drop probability and user jitter through the exploitation of both MFM and MUP.

The simulation scenario for $\rho = 0.50$ (110 users, each with 1 EF flow) is created to demonstrate the achievable performance gains solely from MUP transmission. From both Fig. 3.3 and the performance metrics where $G_{TP}^{ACLS-FUM} = 132.83\%$, $G_{LT}^{ACLS-FUM} = 77.42\%$, $G_{PDP}^{ACLS-FUM} = 99.99\%$ and $G_{JT,EF}^{ACLS-FUM} = 61.15\%$, it is clear that ACLS-FUM performs much better than any of the other four scheduling policies. ACLS-FUM performs better than MG in terms of user throughput as shown in Fig. 3.3a. The high $G_{TP}^{ACLS-FUM}$ is achieved due to the ability to multiplex packets for different users into the same physical layer encoder packet. ACLS-FUM also provides the best performance in terms of user jitter compared to the other four scheduling policies which have almost identical performance as shown in Fig. 3.3d. This improvement is achieved due to the ability to PDM the physical layer encoder packet using MUP transmission which provides an increase number of available time slots to support low-rate latency-sensitive EF traffic. High $G_{LT}^{ACLS-FUM}$ and $G_{PDP}^{ACLS-FUM}$ (near-0% PDP) are achieved from the reduction in wastage in the physical layer encoder packet and a corresponding queue length reduction.

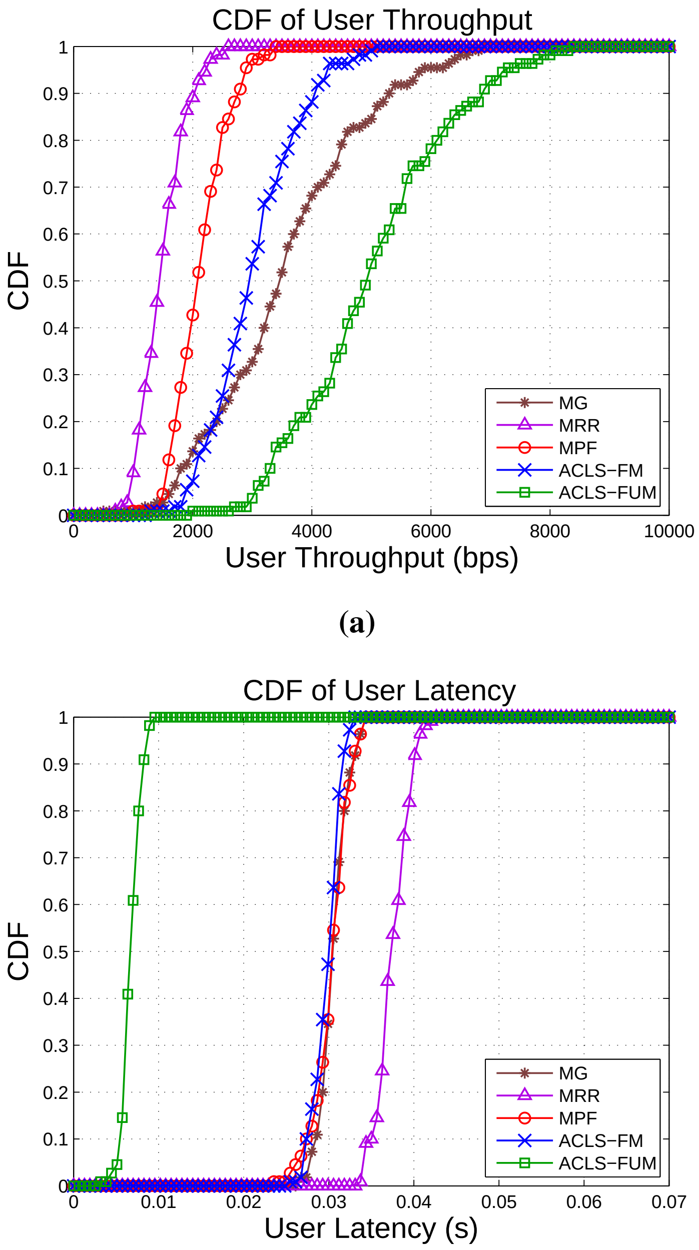
The simulation scenario for $\rho = 0.90$ (30 users, each with 1 BE and 1 EF flow) is created to demonstrate the achievable performance gains from MUP with MFM transmission in a mixed traffic (BE + EF) scenario. The results are presented in Fig. 3.4 with $G_{TP}^{ACLS-FUM} =$

66.58%, $G_{LT}^{ACLS-FUM} = 42.84\%$, $G_{PDP}^{ACLS-FUM} = 82.57\%$, $G_{JT,BE}^{ACLS-FUM} = 16.81\%$ and $G_{JT,EF}^{ACLS-FUM} = 39.06\%$. In this scenario, based on the transmission mode selection function defined in Section 3.3.1, 99.56% were MUP transmissions and the remaining 0.44% were SUP transmissions. As shown in Fig. 3.4, simulation results confirm that ACLS-FUM performs better than the other four scheduling polices defined in Section 2.6.1 in terms of user throughput, user latency, user packet drop probability and user jitter with the exception that MG has a slightly better BE throughput and BE packet drop probability. Fig. 3.4a shows that ACLS-FUM has the second highest throughput performance for BE (behind MG) and the highest throughput performance for EF. However, MG's BE throughput performance comes at a great sacrifice of EF traffic, which not only has the lowest EF throughput but also the highest EF PDP as shown in Fig. 3.4a and 3.4c respectively. In contrast, ACLS-FUM achieves a near-0% PDP for EF traffic and the second lowest PDP for BE traffic. ACLS-FUM also has the lowest latency for BE (up to the 80th percentile) and the lowest latency for EF as shown in Fig. 3.4b. While MG has a lower latency for the upper 20th percentile for BE, that is achieved at the expense of a 50% EF PDP at the 80th percentile as shown in Fig. 3.4c. ACLS-FUM achieves the lowest user jitter for both BE and EF traffic. It is worth highlighting that ACLS-FUM (MUP) outperforms ACLS-FM (SUP) in all four QoS metrics, primarily due to the increased packing efficiency of MUP transmission. A solution possible under SUP is also feasible under MUP. Therefore, optimizing over the set of possible MUP solutions will generally yield an improved optimal solution in any one of the four QoS metrics. An inductive proof of MUP throughput gain is presented in Appendix A.

From the simulation results, we note that the HOL average waiting time increases as the system loading $\rho$ increases. ACLS-FUM will more likely select the SUP transmission mode in an attempt to clear the users' backlog. However, this could degrade the system performance to that of ACLS-FM (see Chapter 2). As such, further considerations should be taken in account when defining the transmission mode selection function so as to 1) achieve a balanced tradeoff between backlog reduction and MUP benefit maximization and 2) attempt to determine (other than from the DRC index) whether a user's low DRC index request is due to bad channel conditions or due to a lack of transmit data in the queue. Scheduling priority should be given to the users that are in better channel conditions (provided that the user fairness constraint is met) so as to not compromise system capacity.

## 3.5 Conclusion

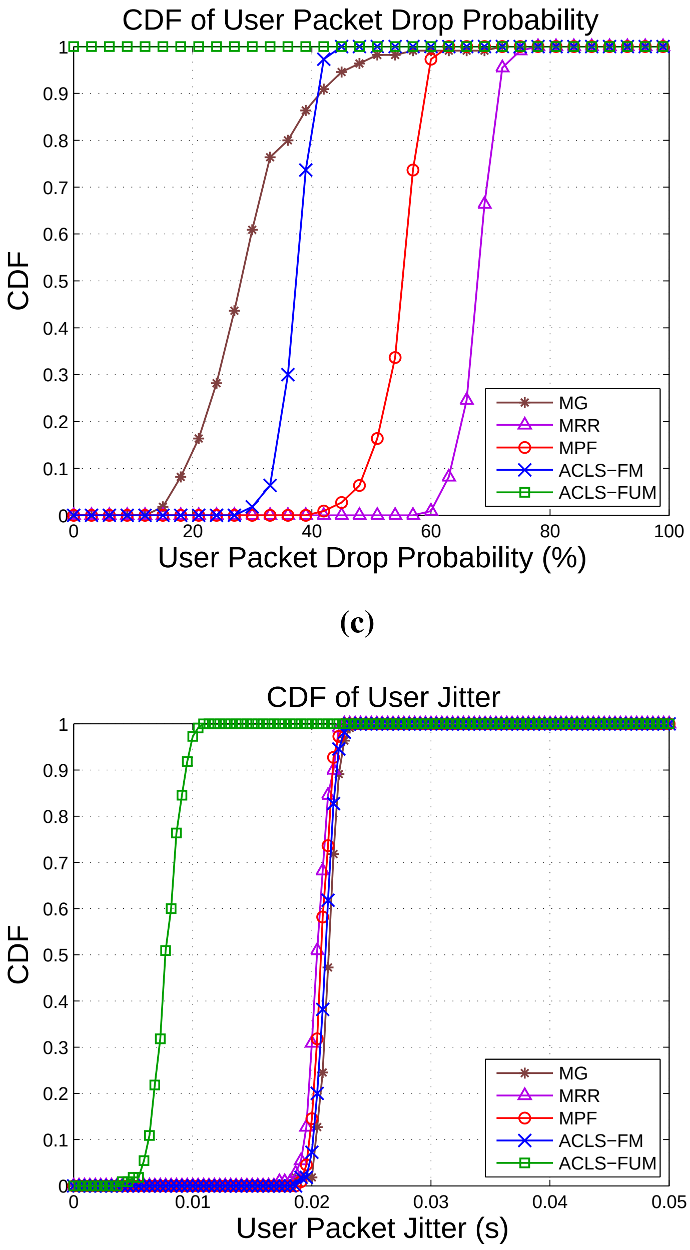
An ACLS-FUM scheduling policy that integrates both MFM and PDM while jointly considering physical-layer time-varying channel conditions as well as application-layer QoS requirements in a mixed traffic environment has been proposed and evaluated. In addition to exploiting the benefits of MFM and cross layer information, ACLS-FUM realizes additional performance gains by taking PDM of the shared physical layer encoder packet into account, further reducing wastage in the physical layer encoder packet. Simulation results show that ACLS-FUM can achieve substantial performance gains in user throughput, user latency, user packet drop probability and user jitter when compared to four other well-known scheduling policies.



**(a)**

**(b)**

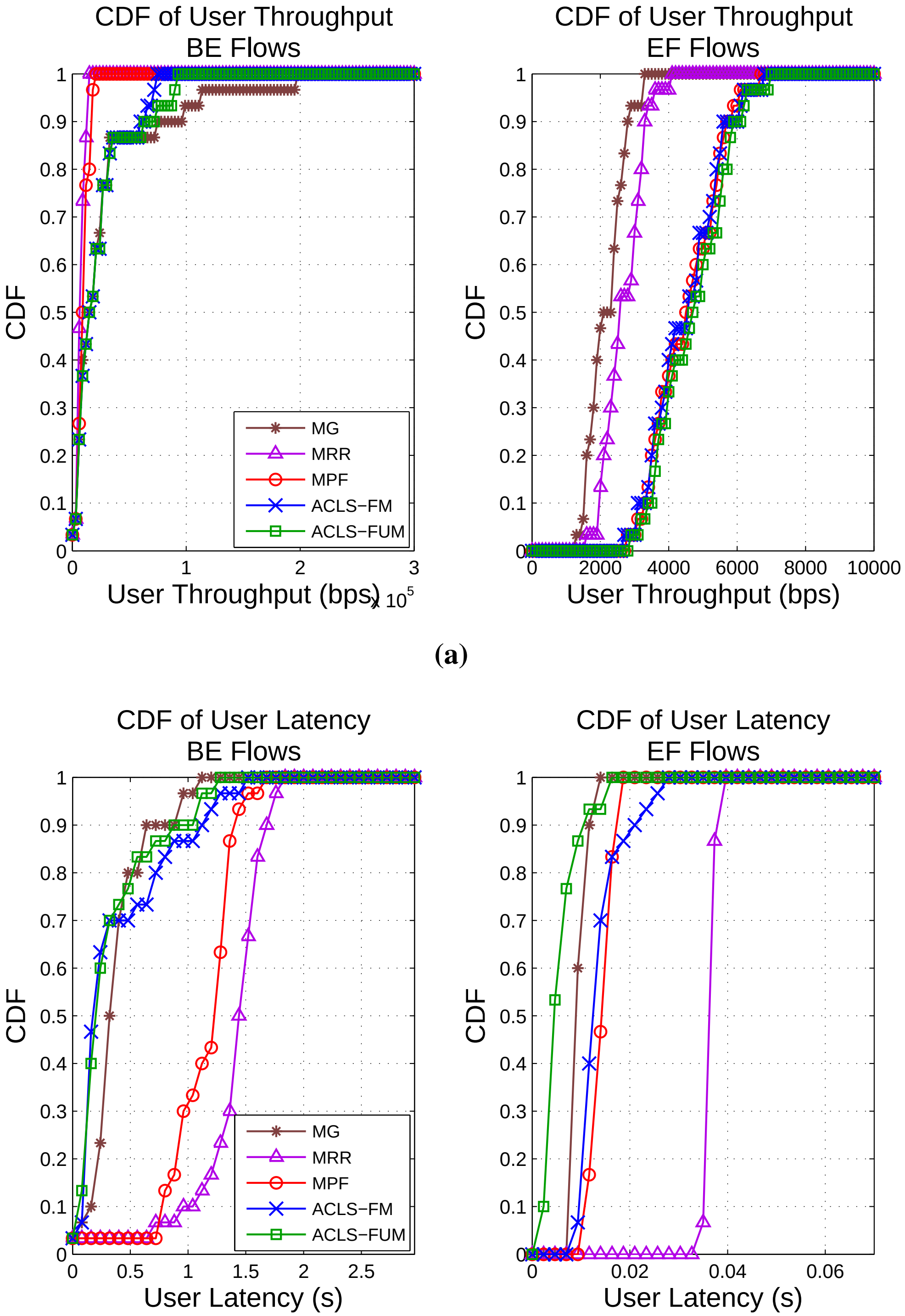
**Figure 3.3:** Performance for a System with $\rho = 0.50$, $I = 110$, 1 EF Flow for each User (a) CDF of User Throughput (b) CDF of User Latency



**(c)**

**(d)**

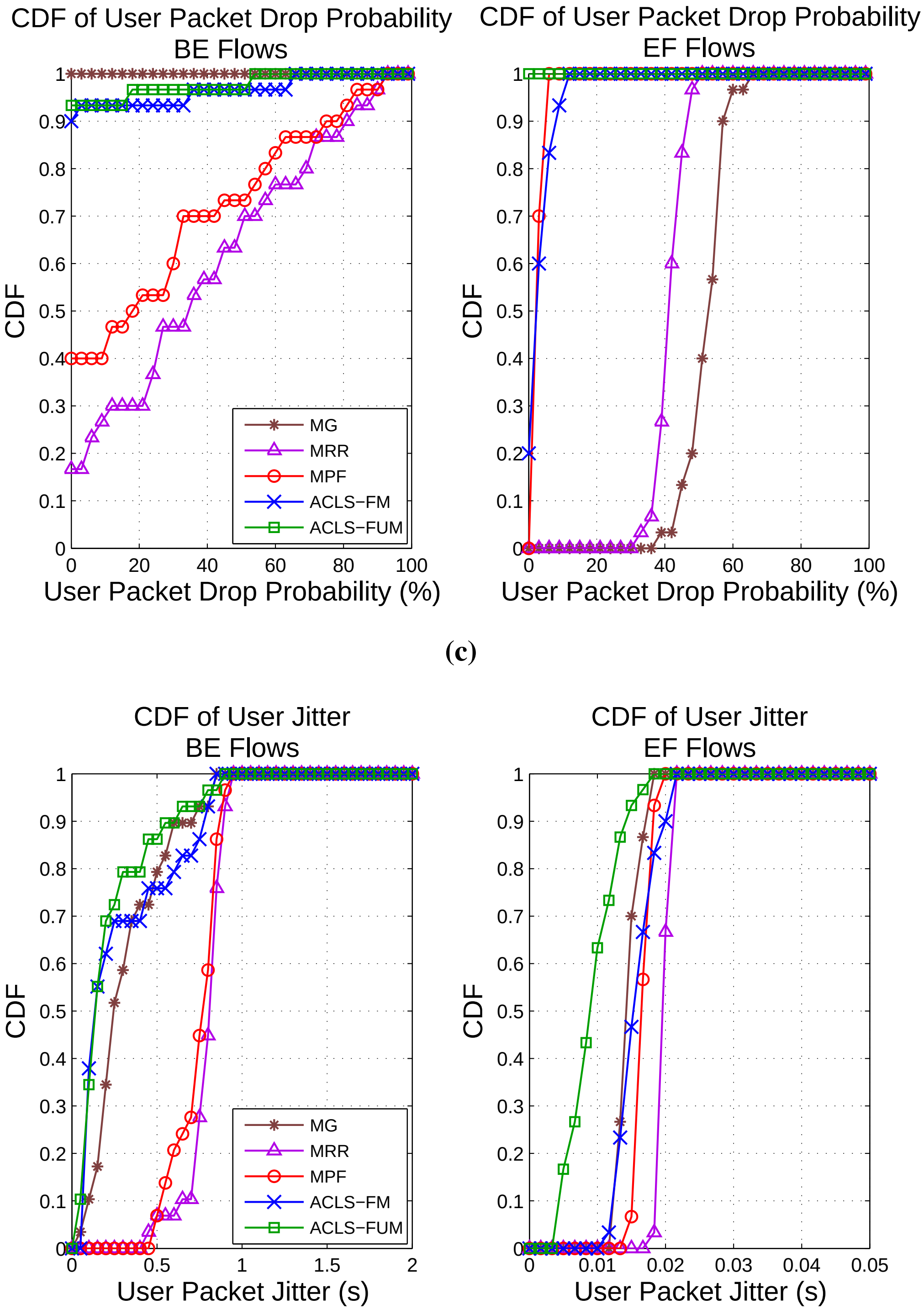
**Figure 3.3:** Performance for a System with $\rho = 0.50$, $I = 110$, 1 EF Flow for each User (Continued) (c) CDF of User Packet Drop Probability (d) CDF of User Jitter



**(a)**

**(b)**

**Figure 3.4:** Performance for a System with $\rho = 0.90$, $I = 30$, 1 BE and 1 EF Flow for each User (a) CDF of User Throughput (b) CDF of User Latency



**(c)**

**(d)**

**Figure 3.4:** Performance for a System with $\rho = 0.90$, $I = 30$, 1 BE and 1 EF Flow for each User (Continued) (c) CDF of User Packet Drop Probability (d) CDF of User Jitter

# Chapter 4

# BitQoS-aware Resource Allocation Framework for Multi-carrier OFDM Systems [3]

## 4.1 Introduction

Orthogonal Frequency Division Multiplexing (OFDM) [57–59] is a promising technique for communication systems due to its high spectral efficiency and is currently employed in many communication systems, e.g., LTE [27], Worldwide Interoperability for Microwave Access (WiMAX IEEE 802.16) [28] and Very High bit rate Digital Subscriber Line (VHDSL) [60]. In OFDM, the available transmission bandwidth is divided into mutually orthogonal narrow-band subcarriers and data is transmitted over these subcarriers. A higher spectral efficiency is possible as the orthogonality is achieved through proper selection of waveforms instead of reliance on guard bands as in conventional frequency division multiplexing (FDM). The system performance can be enhanced by adapting the modulation, coding and power to the channel quality of each subcarrier. In a multi-user system, as the channel quality on each

[3] The material in this chapter is based on: C. E. Huang and C. Leung, "BitQoS-aware resource allocation for multi-user mixed-traffic OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2067-2082, Jun. 2012. ⓒ 2012 IEEE.
http://dx.doi.org/10.1109/TVT.2012.2189030

subcarrier is likely to be independent among different users, OFDMA allows users to access subcarriers selectively, in time and frequency, to exploit multi-user and multi-channel diversities, providing increased scheduler flexibility and scalability to further improve system performance.

Dynamic bit-loading, transmission power allocation and subcarrier assignment schemes for multi-user OFDM systems have been devised to take advantage of the mechanisms and optimization features introduced in the air interfaces [27, 28, 33]. Much of the published work in OFDM RRM has focused on exploiting multi-user and multi-channel diversities [29–32] to maximize the system throughput subject to a total system transmit power constraint [30, 61, 62] or to minimize the total transmit power while satisfying a transmission rate for each user [63]. In addition, many of the RRM algorithms have previously focused on homogeneous traffic where the traffic type consists of only either real-time or non-real-time traffic traffic. More recently, multi-application (flow) diversity [34–37, 39–41] has been exploited to address concurrent heterogeneous application QoS requirements in mixed-traffic networks.

In this chapter, we propose to increase the flexibility and granularity of the resource allocation algorithms by considering QoS at the bit-level rather than at the flow-level as in previous works [36–41]. This is achieved by adaptively matching the QoS requirements of the user application bits to the characteristics of the OFDM subcarriers. As shown in Fig. 4.1, the bits from each application flow of a given user are mapped into OFDM subcarriers based on a bitQoS-aware scheduling policy to exploit both multi-application and multi-bit diversities. BitQoS represents a QoS prioritization mechanism which can take into consideration inter-user priorities, intra-user application QoS requirements and fairness in a multi-user mixed-traffic system. The selected application bits are then transmitted simultaneously on a set of OFDM subcarriers allocated to that user. The mapping between application bits and the OFDM subcarriers is signaled using the control channel accompanying the data channel. The receiver is then able to extract the application bits from the assigned OFDM subcarriers. While the proposed scheme requires additional scheduling signaling overhead and increased

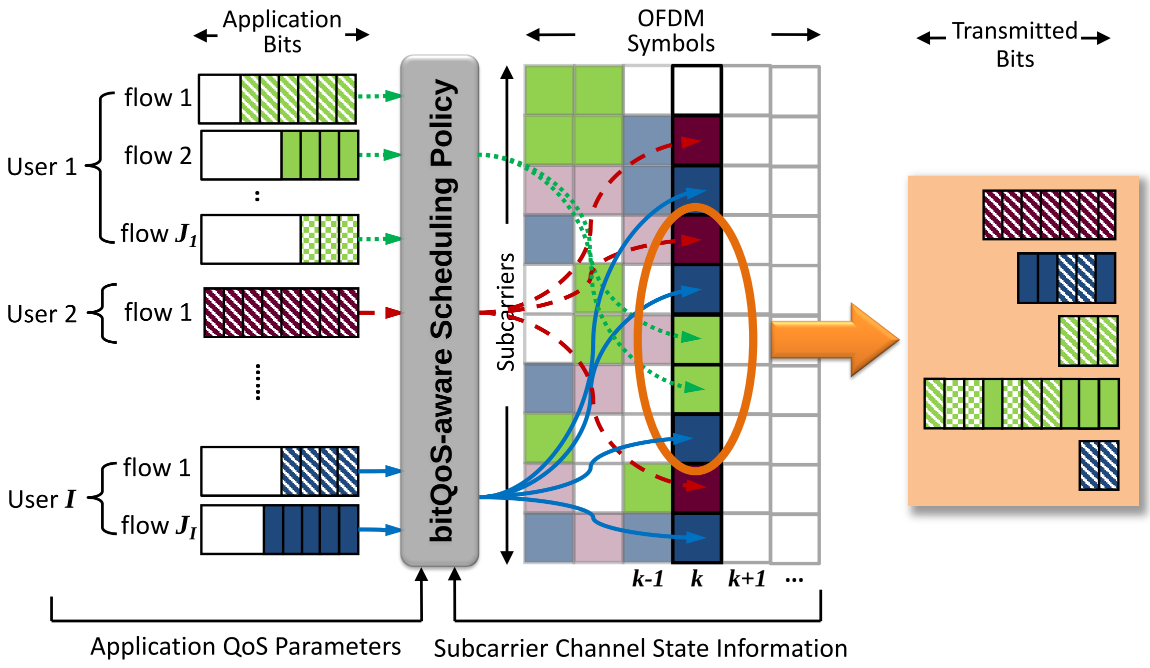


**Figure 4.1:** Mapping of application bits to OFDM subcarriers for the bitQoS-aware resource allocation framework. There are no restrictions as to whether each subcarrier can carry bits from more than one application flow of a user.

computational complexity, it provides the advantage of matching the QoS requirements of the application bits to the channel qualities of the OFDM subcarriers, and the critical ability to more closely meet the QoS requirements of multiple user application flows. This is not possible in flow-level scheduling since only flow-level QoS parameter values are considered. We formulate the proposed bitQoS-aware RA framework as two optimization problems: one with no flow merging and one with flow merging.

This chapter is organized as follows: in Section 4.2, we present the system model that includes the network model and traffic classes. The bitQoS-aware RA framework with no flow merging and with flow merging are described in Section 4.3. The performance measures, analytical system throughput and comparative schemes are presented in Section 4.4.

## 4.2 System Model

The network model of a multi-user OFDM system and the BE and EF traffic classes are described in this section.

### 4.2.1 Network Model

We consider forward link transmissions in a multi-user OFDM system consisting of one BS servicing $I$ users with $N$ subcarriers in a single cell. Let $\mathcal{I} = \{1, 2, \ldots, I\}$ denote the set of all users and $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of all subcarriers. User $i$ has $J_i$ application flows and let $\mathcal{J}_i = \{1, 2, \ldots, J_i\}$ denote the set of all application flows of user $i$. All application data packets to be transmitted to users are queued at the BS. We assume that the data buffer size at the BS is infinite (i.e., no packet blocking) and that the BS has knowledge of the data buffer parameters and QoS requirements for all the application flows. Bits in the data buffer are indexed by $z$, $z \in \{1, 2, \ldots, B_i^j(k)\}$, where $B_i^j(k)$ denotes the queue length, in bits, of the data buffer for user $i$, flow $j$ at time $k$, $k \in \{1, 2, \ldots, K\}$. Packets in the data queues are serviced in a FIFO fashion.

We assume that the BS has perfect knowledge of the channel gain, $\alpha_{i,n}$, of subcarrier $n$ for user $i$, $i \in \mathcal{I}$, $n \in \mathcal{N}$, from the feedback channel. In practice, for Time Division Duplex (TDD) systems, the BS is able to estimate the channel state information based on the received uplink transmission given the symmetry of the channel characteristics for the downlink and uplink, and for Frequency Division Duplex (FDD) systems, pilot symbols are inserted in the downlink transmission [64] for the MS to estimate the channel state information. For simplicity, we do not consider the path loss or the effects of shadowing from the BS to MSs and we assume that the subcarriers undergo independent and identically distributed (i.i.d.) Rayleigh fading to account for multipath fading. The fading rate is slow enough that $\alpha_{i,n}$ remains constant over an OFDM symbol duration, $T_s$, and the mean, $E\{|\alpha_{i,n}|^2\}$, of the channel power gain is assumed to be unity. Let $p_{i,n}$ denote the transmit power allocated to user $i$ on subcarrier $n$. The corresponding number of bits that can be carried per OFDM symbol [65] is

$$c_{i,n} = \log_2 \left( 1 + \frac{p_{i,n}|\alpha_{i,n}|^2}{\zeta \sigma_0^2} \right), \tag{4.1}$$

where $\sigma_0^2$ denotes the noise power and $\zeta$ is a SNR gap parameter. For practical signal constellations, $\zeta$ reflects the Bit Error Rate (BER) requirement [65]. The scheduling decision is performed on an OFDM symbol basis and the total BS transmit power is $P_{total}$. It is also assumed that the scheduling decision time is negligibly small compared to $T_s$ and that transmitted bits are received without errors.

### 4.2.2 Traffic Classes

Two traffic classes are considered: BE traffic representing Internet browsing-like applications and EF traffic representing VoIP-like applications. We use the web browsing traffic arrival model in [50] for incoming BE traffic. The application layer PDU size is based on a truncated Pareto distribution with a mean of 25 kBytes and minimum and maximum sizes of 4.5 kBytes and 2 MBytes respectively. The application layer PDU interarrival time is geometrically distributed with a mean of 5 sec and takes on values which are multiples of 1 sec. For the EF traffic class, the VoIP traffic arrival model in [50] is assumed. In contrast to the web browsing model, source configuration and source files are used to generate VoIP traffic. The source file is generated based on the MSO model IS-871 with alterations as detailed in [50]. The application layer PDU size and interarrival time have a mean of 152.4 bits and 0.04 sec, respectively. The average traffic arrival rates $\lambda_{BE}$ and $\lambda_{EF}$ are 40.0 kbps and 3.7 kbps per application respectively.

## 4.3 BitQoS-aware Resource Allocation Framework

In this section, we describe the bitQoS function and the bitQoS-aware resource allocation problem formulation with no flow merging and with flow merging.

### 4.3.1 BitQoS Function

Based on the application QoS requirements, data buffer parameters, inter- and intra-user priorities and fairness, the bitQoS function maps these QoS parameters of an application bit into a numerical value. The bitQoS function allows the scheduling priority of a bit to be raised when the QoS satisfaction level is low and vice versa. For example, for delay-sensitive traffic such as VoIP applications, the bitQoS function may be expressed as an exponentially increasing function of the bit waiting time, whereas for the BE traffic, the bitQoS function may be a constant. We define the bitQoS value of user $i$, flow $j$, bit $z$ as

$$\psi_i^{j,z} = f(\boldsymbol{\theta}_i^{j,z}), \tag{4.2}$$

where $f(\cdot)$ denotes the bitQoS function and $\boldsymbol{\theta}_i^{j,z}$ denotes the tuple of QoS parameters of interest associated with user $i$, flow $j$, bit $z$. For this work, we consider a bitQoS function which includes the following QoS parameters: application flow priority and bit waiting time,

$$\boldsymbol{\theta}_i^{j,z} = \{\pi_j, w_i^{j,z}(k)\}, \tag{4.3}$$

where $\pi_j \in \mathbb{R}_+$ is the application flow priority for flow $j$ and $w_i^{j,z}(k) \in [0, \infty)$ denotes the amount of time, in seconds, that the bit at position $z$ in user $i$, flow $j$ buffer has waited. The term, $\pi_j$, is included in (4.3) to account for the different traffic classes that may be present in a mixed-traffic system, and $w_i^{j,z}$ is included to account for the bit waiting time since latency is a key QoS requirement for delay-sensitive traffic. Each bit is time stamped upon arrival in the data buffer, and the waiting time is found by simply subtracting the arrival time from the current time $k$. Bits are dropped if $w_i^{j,z}(k)$ exceeds the application flow scheduling delay threshold $T_j \in \mathbb{R}_+$ as specified in Table 4.2. If any bit within an application data packet is dropped, then all the bits in that application data packet are dropped. We define the bitQoS function as

$$f(\boldsymbol{\theta}_i^{j,z}) = c_j \pi_j \xi_j^{d_j (w_i^{j,z}(k) - \eta_j)}. \tag{4.4}$$

The bitQoS function is expressed as an exponential of the bit waiting time $w_i^{j,z}(k)$, which allows bits from delay-sensitive application flows to have their service priority rapidly raised as the waiting time exceeds the comfort latency threshold $\eta_j \in \mathbb{R}_+$, where $\eta_j$ is set to a value smaller than $T_j$. In the region where $w_i^{j,z}(k) \leq \eta_{EF}$, the BE bits have a higher bitQoS value than the EF bits; this allows for a reduction in BE traffic backlog, if necessary. The base of the exponential function $\xi_j \in \mathbb{R}_+$ is set according to the delay sensitivity of the respective

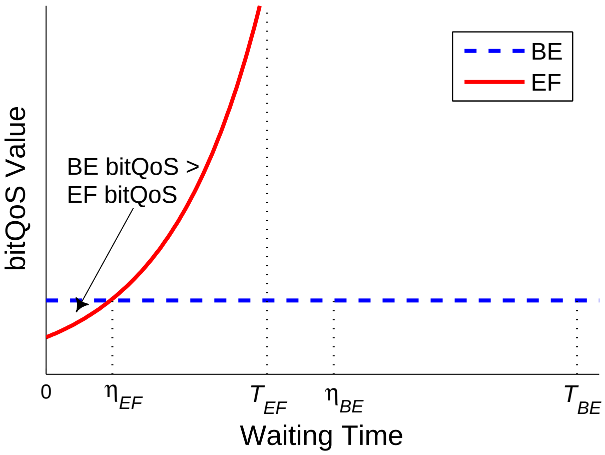


**Figure 4.2:** BE and EF BitQoS Functions

application flow. The coefficients, $c_j \in \mathbb{R}_+$ and $d_j \in \mathbb{R}_+$, are used to scale the exponential function if needed. Examples of the bitQoS functions for BE and EF are illustrated in Fig. 4.2.

### 4.3.2 BitQoS-aware Resource Allocation Framework with No Flow Merging

We formulate the proposed bitQoS-aware RA framework with no flow merging as an optimization problem with the objective of finding the joint subcarrier, power and bit assignment to maximize the total bitQoS-weighted throughput, subject to the total transmit power constraint. Let the optimization variable $a_{i,n}^{j}$ denote the subcarrier assignment variable which takes on the value 1 if subcarrier $n$ is allocated to user $i$, flow $j$ and 0 otherwise. Furthermore, let the optimization variable $b_{i,n}^{j,z}$ denote the bit assignment variable which takes on the value 1 if user $i$, flow $j$, bit $z$ is transmitted on subcarrier $n$ and 0 otherwise. Finally, the optimization variable $p_{i,n}^{j} \in [0, P_{total}]$ denotes the transmit power for user $i$, flow $j$ on subcarrier $n$. The optimization problem, OP4.1, is formulated as

$$\text{OP4.1:} \quad \max_{\substack{a_{i,n}^{j} \in \{0,1\} \\ p_{i,n}^{j} \in [0, P_{total}] \\ b_{i,n}^{j,z} \in \{0,1\}}} \quad \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{z=1}^{B_i^j} \sum_{n=1}^{N} f(\boldsymbol{\theta}_i^{j,z}) b_{i,n}^{j,z} \tag{4.5}$$

$$\text{subject to} \quad \sum_{i} \sum_{j} \sum_{n} p_{i,n}^{j} a_{i,n}^{j} \leq P_{total} \tag{4.6}$$

$$\sum_{z} b_{i,n}^{j,z} \leq c_{i,n}^{j} a_{i,n}^{j} \quad \forall i, j, n \tag{4.7}$$

$$\sum_{i} \sum_{j} a_{i,n}^{j} \leq 1 \quad \forall n \tag{4.8}$$

$$\sum_{n} b_{i,n}^{j,z} \leq 1 \quad \forall i, j, z. \tag{4.9}$$

Constraint (4.6) ensures that the sum of the transmit powers on all subcarriers does not exceed $P_{total}$. Constraint (4.7) ensures that the total number of bits that user $i$, flow $j$ can transmit on subcarrier $n$ does not exceed the throughput limit $c_{i,n}^{j}$ given in (4.1). Constraint (4.8) ensures that each subcarrier can only be assigned to at most a single application flow of a user so as to reduce the signaling overhead required for the application bits to OFDM subcarriers mapping. Constraint (4.9) ensures that each bit is only transmitted on one subcarrier.

### 4.3.3 BitQoS-aware Resource Allocation Framework with Flow Merging

In the previous section, each subcarrier assigned to a user is restricted to only carry bits from a single application flow of that user so as not to incur additional signaling overhead that may be required to indicate which application flow of the user each bit is from in the proposed bitQoS-aware RA framework. However, the bitQoS-aware RA framework with no flow merging increases the computational burden of the BS [34], as at each scheduling decision time $k$, the BS has to schedule $\sum_{i} J_i$ users instead of $I$ users. It may also result in some wastage in the event that there are not enough bits from an application flow to fill up subcarriers that have been assigned to it. Hence, we relax this constraint and allow application bits from different application flows of a user to be merged onto a single OFDM subcarrier to study the potential performance gains that can be further achieved with the bitQoS-aware RA framework with flow merging. The proposed bitQoS-aware RA framework with flow merging is formulated as an optimization problem with the objective of finding the joint subcarrier, power and bit assignment to maximize the total bitQoS-weighted throughput, subject to the total transmit power constraint. Let the optimization variable $a_{i,n}$ denote the subcarrier assignment variable which takes on the value 1 if subcarrier $n$ is allocated to user $i$ and 0 otherwise. Furthermore, let the optimization variable $b_{i,n}^{j,z}$ denote the bit assignment variable, which takes on the value 1 if user $i$, flow $j$, bit $z$ is transmitted on subcarrier $n$ and 0 otherwise. Finally, the optimization variable $p_{i,n} \in [0, P_{total}]$ denotes the transmit power for user $i$ on subcarrier $n$. The optimization problem, OP4.2, is formulated as

$$\text{OP4.2:} \quad \max_{\substack{a_{i,n}\in\{0,1\} \\ p_{i,n}\in[0,P_{total}] \\ b_{i,n}^{j,z}\in\{0,1\}}} \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{z=1}^{B_i^j}\sum_{n=1}^{N} f(\boldsymbol{\theta}_i^{j,z}) b_{i,n}^{j,z} \tag{4.10}$$

$$\text{subject to} \quad \sum_i \sum_n p_{i,n} a_{i,n} \leq P_{total} \tag{4.11}$$

$$\sum_j \sum_z b_{i,n}^{j,z} \leq c_{i,n} a_{i,n} \quad \forall i, n \tag{4.12}$$

$$\sum_i a_{i,n} \leq 1 \qquad \forall n \tag{4.13}$$

$$\sum_n b_{i,n}^{j,z} \leq 1 \qquad \forall i, j, z. \tag{4.14}$$

Constraint (4.11) ensures that the sum of the transmit powers on all subcarriers does not exceed $P_{total}$. Constraint (4.12) ensures that the total number of bits that user $i$ can transmit on subcarrier $n$ does not exceed the throughput limit $c_{i,n}$ given in (4.1). Constraint (4.13) ensures that each subcarrier can only be assigned to at most one user. Note that this constraint has been relaxed from the problem formulation in OP4.1 to allow the subcarrier to be assigned to more than one flow of that user. Constraint (4.14) ensures that each bit is only transmitted on one subcarrier.

**Table 4.1:** Simulation Parameter Values

| Parameter | Value |
|---|---|
| System bandwidth (kHz) | $W = 4.5$ |
| Number of subcarriers | $N = 18$ |
| OFDM symbol duration (sec) | $T_s = 0.004$ |
| Subcarrier spacing (Hz) | $\Delta f = 250$ |
| Application data packet size (bits) | 128 |
| Channel model | independent Rayleigh fading |
| Total transmit power (Watt) | $P_{total} = 1$ |
| SNR gap | $\zeta = 1$ |
| Noise power (Watt) | $\sigma_0^2 = 10^{-13}$ |
| MDU window length (OFDM symbols) | $W_{MDU} = 200$ |

## 4.4 Performance Evaluation

The bitQoS scheduling policies are simulated in Matlab using the system model described in Section 4.2. Each user is assumed to have 1 BE flow and 1 EF flow. The parameter values used in our simulation are listed in Tables 4.1 and 4.2.

### 4.4.1 Performance Measures

To evaluate the performance of the bitQoS-aware RA framework, we quantify the performance gains in terms of average system throughput, average user throughput, average user latency, average user jitter and average user packet drop probability. We define the user

**Table 4.2:** Traffic Parameter Values

| Parameter | BE Traffic | EF Traffic |
|---|---|---|
| Packet size | Truncated Pareto ($\alpha = 1.2$, $x_{min} = 4.5$ kBytes and $x_{max} = 2$ MBytes) | IS-871 with alterations as detailed in [50] |
| Packet interarrival time | Geometric distribution (mean = 5 sec) | IS-871 with alterations as detailed in [50] |
| Average traffic arrival rate (kbps) | $\lambda_{BE} = 40.0$ | $\lambda_{EF} = 3.70$ |
| Scheduling delay threshold (sec) | $T_{BE} = 3.000$ | $T_{EF} = 0.100$ |
| Comfort latency threshold (sec) | $\eta_{BE} = 0.100$ | $\eta_{EF} = 0.025$ |
| Flow priority | $\pi_{BE} = 1.00$ | $\pi_{EF} = 1.00$ |
| Flow scaling coefficients | $c_{BE} = 1.00$<br>$d_{BE} = 1000$ | $c_{EF} = 1.00$<br>$d_{EF} = 1000$ |
| Urgency base | $\xi_{BE} = 1.00$ | $\xi_{EF} = 1.05$ |

throughput, user latency, user jitter and user packet drop probability respectively as follows:

$$TP_{user}(i) = \frac{\sum_j \sum_k TP_i^j(k)}{K}, \tag{4.15}$$

$$LT_{user}(i) = \frac{\sum_{bit(i,j,z)\in\Phi_i^j} LT_i^{j,z}}{K \times TP_{user}(i)}, \tag{4.16}$$

$$JT_{user}(i) = \sqrt{\frac{\sum_{bit(i,j,z)\in\Phi_i^j} (LT_i^{j,z} - LT_{user}(i))^2}{K \times TP_{user}(i)}}, \tag{4.17}$$

$$PDP_{user}(i) = \frac{\sum_j \sum_k BD_i^j(k)}{K \times TP_{user}(i) + \sum_j \sum_k BD_i^j(k)}, \tag{4.18}$$

where $TP_i^j(k)$ denotes the number of bits that is contained in packets of user $i$, flow $j$ which are successfully received at time $k$, $LT_i^{j,z}$ denotes the amount of time $bit(i,j,z)$ has waited in the data buffer before being scheduled and $BD_i^j(k)$ denotes the number of bits that is contained in packets of user $i$, flow $j$ which are dropped at time $k$. The term, $\Phi_i^j$, denotes the set of all the scheduled bits of user $i$, flow $j$. Note that if any bit within an application data packet is dropped, then that application data packet is dropped. Furthermore, in the calculation of user latency and user jitter, only bits in the scheduled packets (i.e., packets that are not dropped due to exceeding $T_j$) are included. The average system throughput, $TP_{system}$, is defined as $TP_{system} = \sum\limits_{i} TP_{user}(i)$, and the average user throughput, average user latency, average user jitter and average user packet drop probability are obtained by averaging $TP_{user}(i)$, $LT_{user}(i)$, $JT_{user}(i)$ and $PDP_{user}(i)$ respectively, over the $I$ users.

### 4.4.2 Analytical System Throughput

For comparison, the analytical system throughput curve of a multi-user, multi-channel OFDM system with full buffer using water-filling is derived. The analytical system throughput is obtained by summing the capacities of the i.i.d. Rayleigh fading subcarriers subject to the total power constraint $P_{total}$ over the set $\mathcal{N}$ and distribution $p_{\Gamma_n}(\gamma_n)$ [59], where $\Gamma_n \triangleq |\alpha_{i^*(n),n}|^2 P_{total}/\zeta\sigma_0^2$ denotes the instantaneous SNR of subcarrier $n$ assuming the total BS transmit power, $P_{total}$, is allocated to that subcarrier and $i^*(n) = \arg\max\limits_{i\in\mathcal{I}}\{|\alpha_{i,n}|^2\}$:

$$C = \max_{p_{i^*(n),n}(\gamma_n):\sum\limits_{n}\int\limits_0^\infty p_{i^*(n),n}(\gamma_n)\cdot p_{\Gamma_n}(\gamma_n)d\gamma_n \leq P_{total}} \sum_{n}\int\limits_{0}^{\infty} \log_2\left(1+\frac{p_{i^*(n),n}(\gamma_n)\gamma_n}{P_{total}}\right) p_{\Gamma_n}(\gamma_n)d\gamma_n. \tag{4.19}$$

We use the results in [66] which derives the capacity of a single Rayleigh fading channel with multi-receiver antennas using selection combining under optimal simultaneous power and rate adaptation to obtain $p_{\Gamma_n}(\gamma_n) = \dfrac{I}{E\{\Gamma_n\}}\left[1-e^{\frac{-\gamma_n}{E\{\Gamma_n\}}}\right]^{I-1}\left[e^{-\frac{\gamma_n}{E\{\Gamma_n\}}}\right]$, where the term $E\{\Gamma_n\}$ denotes the expected value of $\Gamma_n$. The analytical system throughput can be expressed as

$$C = \log_2(e)\sum_{n=1}^{N}\sum_{z=1}^{I}(-1)^{z+1}\binom{I}{z}E_1\left(\frac{z\gamma_0}{E\{\Gamma_n\}}\right), \tag{4.20}$$

where $E_1(x) \triangleq \displaystyle\int_x^\infty \frac{e^{-u}}{u}du$ denotes the exponential integral of order 1. The optimal power allocation is a two-dimensional water-filling (over $\mathcal{N}$ and $p_{\Gamma_n}(\gamma_n)$) with a common cutoff

SNR value, $\gamma_0$, which is obtained by solving

$$\sum_{n=1}^{N}\sum_{z=1}^{I}(-1)^z\binom{I}{z}\left[\frac{z}{E\{\Gamma_n\}}E_1(\frac{z\gamma_0}{E\{\Gamma_n\}})-\frac{1}{\gamma_0}e^{-\frac{z\gamma_0}{E\{\Gamma_n\}}}\right]=1. \tag{4.21}$$

Since the Left-Hand-Side (LHS) of (4.21) is a monotonically decreasing function of $\gamma_0$, $\forall I \geq 1, N \geq 1, \gamma_0 > 0$ and $E\{\Gamma_n\} > 0$, the solution can be found numerically using a bisection algorithm (see proof in Appendix B).

### 4.4.3 Comparative Schemes

To provide a comparative performance assessment, we consider the following comparative scheduling policies: 1) Multi-user Water-filling (WF) scheduling policy, 2) Multi-user Water-filling with Full Buffer (WF-FB) scheduling policy, and 3) MDU scheduling policy [38]. Since there are no provisions in WF and WF-FB for choosing which application flow to schedule from among the flows of a given user, each application flow $j$ of user $i$ is regarded as a separate user with the same channel gain $\alpha_{i,n}$ in the simulation. When multiple users experiencing the same channel gain are considered for assignment to a subcarrier, one user is chosen completely at random. All scheduling policies, with the exception of WF-FB, adopt the traffic model described in Section 4.2.2.

#### Multi-user Water-filling

WF assigns each subcarrier to the user that has the best channel gain for that subcarrier, and the transmit power is distributed over the subcarriers using the water-filling algorithm [30]. The purpose of including this scheduling policy is to illustrate the performance of an algorithm that does not take QoS requirements into account but attempts to maximize the overall throughput of the system.

#### Multi-user Water-filling with Full Buffer

WF-FB is similar to WF described above, except that in WF-FB, a full buffer model is assumed for the incoming traffic, i.e., all data buffers are always full. While this model may not be realistic, it establishes an upper bound on the throughput achievable for a multi-user, multi-channel OFDM system with full buffer using water-filling.

### Max-Delay-Utility

MDU is a channel- and queue-aware, dynamic power-subcarrier assignment scheme which aims to maximize the aggregate utility with respect to the average waiting times [38]. The objective function of the optimization problem is

$$\max \sum_{i \in \mathcal{I}} \frac{|U_i'(\overline{w}_i(k))|}{\overline{r}_i(k)} r_i(k), \tag{4.22}$$

where $\overline{r}_i(k)$ is the long-term average throughput for user $i$ up to time $k$ which is obtained by averaging the instantaneous actual throughput of user $i$ over the last $W_{MDU}$ OFDM symbols, and $r_i(k)$ is the instantaneous achievable throughput for user $i$ at time $k$. The term $\overline{w}_i(k)$ denotes the average waiting time of user $i$ at time $k$ which is approximated by $\overline{w}_i(k) = \dfrac{Q_i(k)}{\overline{r}_i(k)}$, where $Q_i(k)$ is the queue length, in bits, of user $i$ at time $k$. The marginal utility function $U_i'(\cdot)$ is a non-decreasing function which is chosen based on the QoS requirements of the traffic classes. For our simulation, we adopt the marginal utility functions specified in [38]. The solution to the optimization problem in (4.22) is found by a combination of iterative subcarrier assignment, power allocation and the update of the marginal utility [38]. The purpose of including this scheduling policy is to illustrate the performance gains and tradeoffs of WFH-FM with respect to MDU, which only considers the flow-level QoS requirements.

# Chapter 5

# BitQoS-aware Resource Allocation Scheduling Policies [4]

## 5.1 Introduction

In Chapter 4, a novel bitQoS-aware RA framework is proposed to increase the flexibility and granularity of the resource allocation algorithms by considering QoS at the bit-level rather than only at the flow-level as in previous works [34–37, 39–41]. The proposed RA framework is formulated as MINLP optimization problems (OP4.1 and OP4.2), whose solutions are computationally complex given the large number of subcarriers and users in a practical system. To evaluate the performance of the bitQoS-aware RA framework, we propose lower complexity, iterative subcarrier-power-bit allocation algorithms, hereafter referred to as *Multi-user Water-filling with Heuristics* (WFH) and *Multi-user BitQoS-aware Bit-loading* (BABL) to quantify the achievable performance gains.

This chapter is organized as follows: in Section 5.2, the water-filling-based WFH schedul-

[4] The material in this chapter is based on the following:
C. E. Huang and C. Leung, "QoS-aware bit scheduling in multi-user OFDM systems," in *Proc. IEEE WCNC*, Mar. 2011, pp. 215–220. © 2011 IEEE.
http://dx.doi.org/10.1109/WCNC.2011.5779163
C. E. Huang and C. Leung, "BitQoS-aware resource allocation for multi-user mixed-traffic OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2067-2082, Jun. 2012. © 2012 IEEE.
http://dx.doi.org/10.1109/TVT.2012.2189030

ing policy is described and the bit-loading-based BABL is described in Section 5.3. Simulation results are presented in Section 5.4 including the performance of the bitQoS-aware RA framework with flow merging and with no flow merging. The main findings are summarized in Section 5.5.

## 5.2 Multi-user Water-filling with Heuristics

To evaluate the performance of the bitQoS-aware RA framework with flow merging and with no flow merging, we propose water-filling-based iterative subcarrier-power-bit allocation algorithms, hereafter referred to as *Multi-user Water-filling with Heuristics with Flow Merging* (WFH-FM) to solve the optimization problem, OP4.2, and *Multi-user Water-filling with Heuristics with No Flow Merging* (WFH-NFM) to solve the optimization problem, OP4.1. The goal of the WFH scheduling policies is to maximize the total bitQoS-weighted throughput. It uses the following two main steps: 1) multi-user water-filling for throughput maximization and 2) iterative subcarrier reassignment for bitQoS maximization.

### 5.2.1 WFH-FM Scheduling Policy

At each scheduling decision time $k$, we run the following resource allocation algorithm. To ease the notational burden, we omit the time index $k$ from the equations in this section. The bits from all flows of a user are combined into one queue, i.e., $J_i = 1$, and sorted in decreasing order based on their bitQoS values. A flow chart for the WFH-FM scheduling policy is shown in Fig. 5.1.

Step 1: Multi-user water-filling: To simplify the maximization of the total bitQoS-weighted throughput, we assume that the bitQoS values, $\psi_i^{j,z}$, of all the bits are equal, so that the objective function in OP4.2 can be rewritten as

$$\max_{\substack{a_{i,n}\in\{0,1\}\\ p_{i,n}\in[0,P_{total}]\\ b_{i,n}^{j,z}\in\{0,1\}}} \quad \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{z=1}^{B_i^j}\sum_{n=1}^{N} b_{i,n}^{j,z}. \tag{5.1}$$

This new optimization problem can be solved as a throughput maximization problem subject to a total power constraint [30, 67–69]. In [30], it is shown that when a full buffer model is assumed, the maximum throughput of a multi-user OFDM system can be achieved by assigning each subcarrier to the user with the best channel gain for that subcarrier and distributing the power over subcarriers using the water-filling algorithm. Let $i^*(n)$ denote the selected user that has the highest channel gain on subcarrier $n$, i.e., $i^*(n) = \arg\max_{i \in \mathcal{I}} |\alpha_{i,n}|$. In the event where multiple users experience the same channel gain $\max_{i \in \mathcal{I}} |\alpha_{i,n}|$, then $i^*(n)$ is chosen randomly from these users with equal probabilities. Thus, the current subcarrier assignment can be written as

$$\hat{a}_{i,n} = \begin{cases} 1, & \text{if } i = i^*(n) \\ 0, & \text{if } i \neq i^*(n) \end{cases} \qquad \forall n. \tag{5.2}$$

The term $\hat{a}_{i,n}$ is used to denote the current intermediate subcarrier assignment variable which may be different from the optimal subcarrier assignment variable, $a_{i,n}$, for OP4.2. Once the subcarrier assignment is determined, we can determine the amount of transmit power to be allocated to the subcarriers in order to maximize overall system throughput. This is achieved using the water-filling algorithm. The transmit power for user $i$ on subcarrier $n$ [30] is

$$\hat{p}_{i,n} = \begin{cases} \zeta\sigma_0^2[\frac{1}{\lambda_0} - \frac{1}{|\alpha_{i,n}|^2}]^+, & \text{if } i = i^*(n) \\ 0, & \text{if } i \neq i^*(n) \end{cases} \tag{5.3}$$

where $[x]^+ \triangleq \max\{x, 0\}$ and $\lambda_0$ is a threshold determined using the total power constraint (4.11). The bit assignment variable $\hat{b}_{i,n}^{j,z}$ is then obtained by assigning bits of user $i$ in a FIFO manner to the subcarriers in $\mathcal{V}_i$, one subcarrier at a time, where $\mathcal{V}_i = \{n \in \mathcal{N} | \hat{a}_{i,n} = 1\}$. This bit assignment is performed until either all the bits of user $i$ have been assigned or the throughput limits $c_{i,n}, \forall n \in \mathcal{V}_i$ have been reached.

We note that performing subcarrier assignments based only on channel gains may lead to situations where users in good channel conditions are assigned more subcarriers than needed, i.e., $\sum_n c_{i,n} > \sum_j B_i^j$. To address this issue, we perform an additional subcarrier reassignment step, called greedy water-filling, which aims to reduce the wastage of resources through reassignments of the users' excess subcarriers. We define $\mathcal{U} = \{i \in \mathcal{I} | \sum_n c_{i,n} > \sum_j B_i^j\}$ and $\mathcal{U}^c = \mathcal{I} - \mathcal{U}$ to denote the set of users that have excess subcarriers and the complement set of $\mathcal{U}$, respectively. Furthermore, we define $\Omega_{\mathcal{U}} = \{n \in \mathcal{N} | \hat{a}_{i,n} = 1, i \in \mathcal{U}\}$ to denote the set of subcarriers that are assigned to users in $\mathcal{U}$. The goal of the greedy water-filling is to iteratively reassign one subcarrier in $\Omega_{\mathcal{U}}$ at a time to a user in $\mathcal{U}^c$ such that the overall system throughput after reassignment is maximized. This can be done by computing the attainable throughput gain for every possible reassignment pair in the Cartesian product of $\mathcal{U}^c$ and $\Omega_{\mathcal{U}}$ and performing the subcarrier reassignment based on the pair yielding the highest throughput gain. Power allocation is updated after each reassignment using the water-filling algorithm. This procedure is repeated until the overall system throughput cannot be increased any further through the reassignments of the subcarriers in $\Omega_{\mathcal{U}}$. Based on the current assignment values $\hat{a}_{i,n}$, $\hat{p}_{i,n}$ and $\hat{b}_{i,n}^{j,z}$, the current intermediate objective value $\hat{\delta}_{obj}$ is given by

$$\hat{\delta}_{obj} = \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{z=1}^{B_i^j} \sum_{n=1}^{N} f(\boldsymbol{\theta}_i^{j,z}) \hat{b}_{i,n}^{j,z}. \tag{5.4}$$

Step 2: Iterative subcarrier reassignment: While the intermediate solution from Step 1 maximizes the overall system throughput, it may not be an optimal solution to OP4.2. In particular, if there exists any unassigned bit in the data buffer with a bitQoS value that is greater than those of any already assigned bits, then the intermediate solution may be improved upon by reassigning subcarriers to the users who have unassigned bits with larger bitQoS values. Let $\psi_{un}(i)$ denote the bitQoS value of the first unas-

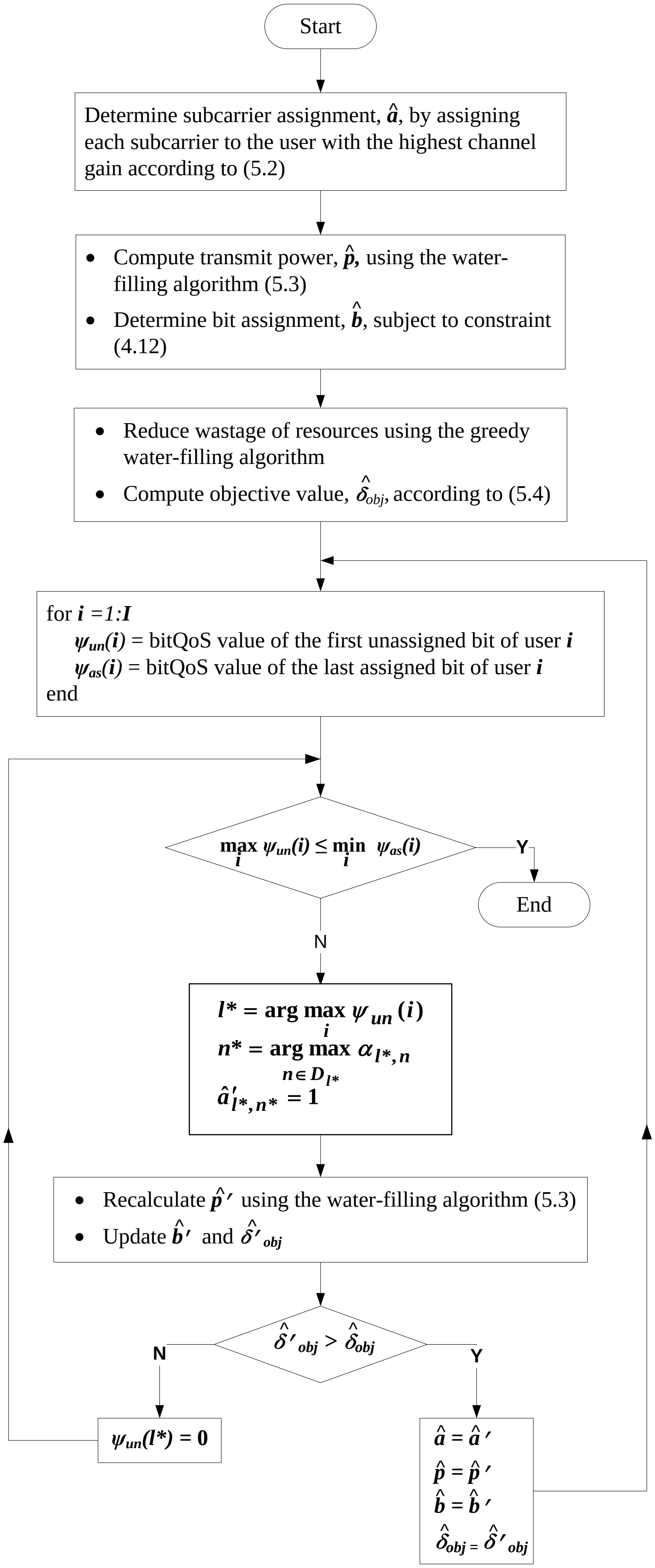

Start
Determine subcarrier assignment, $\hat{\boldsymbol{a}}$, by assigning each subcarrier to the user with the highest channel gain according to (5.2)
Compute transmit power, $\hat{\boldsymbol{p}}$, using the water-filling algorithm (5.3)
Determine bit assignment, $\hat{\boldsymbol{b}}$, subject to constraint (4.12)
Reduce wastage of resources using the greedy water-filling algorithm
Compute objective value, $\hat{\delta}_{obj}$, according to (5.4)
for $\boldsymbol{i} = 1{:}I$
$\psi_{un}(\boldsymbol{i})$ = bitQoS value of the first unassigned bit of user $\boldsymbol{i}$
$\psi_{as}(\boldsymbol{i})$ = bitQoS value of the last assigned bit of user $\boldsymbol{i}$
end
$\max_{i} \psi_{un}(i) \le \min_{i} \psi_{as}(i)$
Y
End
N
$l^* = \arg\max_{i} \psi_{un}(i)$
$n^* = \arg\max_{n \in D_{l^*}} \alpha_{l^*,n}$
$\hat{a}'_{l^*,n^*} = 1$
Recalculate $\hat{p}'$ using the water-filling algorithm (5.3)
Update $\hat{b}'$ and $\hat{\delta}'_{obj}$
$\hat{\delta}'_{obj} > \hat{\delta}_{obj}$
N
Y
$\psi_{un}(l^*) = 0$
$\hat{a} = \hat{a}'$
$\hat{p} = \hat{p}'$
$\hat{b} = \hat{b}'$
$\hat{\delta}_{obj} = \hat{\delta}'_{obj}$

**Figure 5.1:** WFH-FM Flow Chart

signed bit in the data buffer of user $i$ and let $\psi_{as}(i)$ denote the bitQoS value of the last assigned bit in the data buffer of user $i$, that is $\psi_{un}(i) = \max_{bit(i,j,z) \in \mathcal{S}_{un}(i)} f(\boldsymbol{\theta}_i^{j,z})$ and $\psi_{as}(i) = \min_{bit(i,j,z) \in \mathcal{S}_{as}(i)} f(\boldsymbol{\theta}_i^{j,z})$, where

$$\mathcal{S}_{un}(i) = \{bit(i,j,z) | \sum_n \hat{b}_{i,n}^{j,z} = 0, j \in \mathcal{J}_i, z \in \{1, \ldots, B_i^j\}\} \tag{5.5}$$

denotes the set of bits of user $i$ that have not yet been assigned based on the current intermediate assignments, and

$$\mathcal{S}_{as}(i) = \{bit(i,j,z) | \sum_n \hat{b}_{i,n}^{j,z} = 1, j \in \mathcal{J}_i, z \in \{1, \ldots, B_i^j\}\} \tag{5.6}$$

denotes the set of bits of user $i$ that have been assigned based on the current intermediate assignments. The term $bit(i,j,z)$ refers to the bit $z$ of user $i$, flow $j$. At each iteration, the user with the largest unassigned bitQoS-valued bit, $l^* = \arg\max_i \psi_{un}(i)$, will be assigned a subcarrier in an attempt to increase the current intermediate objective value $\hat{\delta}_{obj}$ even though power may be less efficiently used as this user may be experiencing a lower channel quality on this subcarrier. The subcarrier is chosen by $n^* = \arg\max_{n \in D_{l*}} \alpha_{l^*,n}$, where $D_{l^*} = \{n \in \mathcal{N} | \hat{a}_{l^*,n} = 0\}$ denotes the set of subcarriers that have not yet been assigned to user $l^*$. If the subcarrier $n^*$ was previously assigned to another user, then subcarrier $n^*$ is unassigned from that user and the corresponding assigned bits are put back to the data buffers. Based on this new subcarrier assignment variable, $\hat{a}_{i,n}^{'}$, the transmit power, $\hat{p}_{i,n}^{'}$, is recalculated using the water-filling algorithm. The bit assignment variable, $\hat{b}_{i,n}^{'j,z}$, and current intermediate objective value, $\hat{\delta}_{obj}^{'}$, are also updated accordingly. As this subcarrier reassignment may cause a decrease in $\hat{\delta}_{obj}$, this subcarrier reassignment is only performed if $\hat{\delta}_{obj}^{'} > \hat{\delta}_{obj}$. Otherwise, $\psi_{un}(l^*)$ is temporarily set to $0$, and a new user with the next largest unassigned

QoS-valued bit is selected. This step repeats until

$$\max_{i \in \mathcal{I}} \psi_{un}(i) \leq \min_{i \in \mathcal{I}} \psi_{as}(i). \tag{5.7}$$

It can be shown that the number of iterations required for Step 2 in the worst case is $IN$ iterations.

### 5.2.2 WFH-NFM Scheduling Policy

WFH-NFM is identical to WFH-FM with the exception that application bits from the different flows of a user cannot be assigned to the same subcarrier, i.e., each subcarrier assigned to the user can only carry bits from a single application flow of that user. As such, the bits from all flows of a user are not combined into one queue as in WFH-FM, but rather each application flow $j$ of user $i$ in WFH-NFM is regarded as a separate user with the same channel gain $\alpha_{i,n}$. Specifically, OP4.2 is modified as follows: we replace constraint (4.13) with

$$\sum_{i=1}^{I} \sum_{j=1}^{J_i} a_{i,n}^{j} \leq 1 \quad \forall n. \tag{5.8}$$

In addition, the variables $a_{i,n}$, $p_{i,n}$ and $c_{i,n}$ take dependence on $j$ and are replaced with $a_{i,n}^{j}$, $p_{i,n}^{j}$ and $c_{i,n}^{j}$ in WFH-NFM.

## 5.3 Multi-user BitQoS-aware Bit-loading

To evaluate the performance of the bitQoS-aware RA framework with flow merging and with no flow merging, we propose the following bit-loading-based adaptive, joint subcarrier, power and bit allocation algorithms, hereafter referred to as *Multi-user BitQoS-aware Bit-loading with Flow Merging* (BABL-FM) to solve the optimization problem, OP4.2, and *Multi-user BitQoS-aware Bit-loading with No Flow Merging* (BABL-NFM) to solve the optimization problem, OP4.1. The goal of the BABL scheduling policies is to jointly determine the subcarrier, power and bit assignments using bit-loading in an effort to maximize the total bitQoS-weighted throughput subject to the total transmit power constraint. This is accomplished by iteratively assigning the largest unassigned bitQoS-valued bits, one bit at a time to the subcarrier requiring the least amount of power, until the total BS transmit power, $P_{total}$, is depleted or all bits in the user data buffers have been assigned.

### 5.3.1 BABL-FM Scheduling Policy

At each scheduling decision time $k$, we run the following resource allocation algorithm. To ease the notational burden, we omit the time index $k$ from the equations in this section. The bits from all application flows of each user $i$ are merged into one queue, i.e., $J_i = 1$, and sorted in decreasing order based on their bitQoS values. A flow chart for the BABL-FM scheduling policy is shown in Fig. 5.2.

For each bit assignment iteration, we determine the largest unassigned bitQoS-valued bit in the data buffer as

$$bit(i^*, j^*, z^*) = \arg \max_{bit(i,j,z): \sum_n \hat{b}_{i,n}^{j,z} = 0} \psi_i^{j,z}, \tag{5.9}$$

where $bit(i, j, z)$ refers to bit $z$ of user $i$, flow $j$. The term $\hat{b}_{i,n}^{j,z}$ is used to denote the current intermediate bit assignment variable which may be different from the optimal bit assignment variable, $b_{i,n}^{j,z}$, for OP4.2. The power required to transmit this bit is computed for each subcarrier $n \in \mathcal{N}$ and is denoted by the temporary variable, $p_{i^*,n}^{'j^*,z^*}$. Depending on the current intermediate subcarrier assignment variable, $\hat{a}_{i,n}$, the power, $p_{i^*,n}^{'j^*,z^*}$, is determined by one of the following three cases:

Case 1: Subcarrier $n$ was previously not assigned to any user $i$ ($\hat{a}_{i,n} = 0 \; \forall i \in \mathcal{I}$): The power, $p_{i^*,n}^{'j^*,z^*}$, required to transmit $bit(i^*, j^*, z^*)$ on subcarrier $n$ is

$$p_{i^*,n}^{'j^*,z^*} = \frac{\zeta \sigma_0^2}{|\alpha_{i^*,n}|^2}. \tag{5.10}$$

Case 2: Subcarrier $n$ was previously assigned to user $i^*$ ($\hat{a}_{i^*,n} = 1$): The power, $p_{i^*,n}^{'j^*,z^*}$, required to transmit the additional $bit(i^*, j^*, z^*)$ on subcarrier $n$ is

$$\begin{aligned} p'^{j^*,z^*}_{i^*,n} &= \frac{(2^{\hat{c}_{i^*,n}+1} - 1)\zeta\sigma_0^2}{|\alpha_{i^*,n}|^2} - \frac{(2^{\hat{c}_{i^*,n}} - 1)\zeta\sigma_0^2}{|\alpha_{i^*,n}|^2} \\ &= \frac{\zeta\sigma_0^2}{|\alpha_{i^*,n}|^2}(2^{\hat{c}_{i^*,n}+1} - 2^{\hat{c}_{i^*,n}}), \end{aligned} \quad (5.11)$$

where $\hat{c}_{i^*,n}$ denotes the number of bits of user $i^*$ that have already been assigned to subcarrier $n$.

Case 3: Subcarrier $n$ was previously assigned to another user $l$ ($\hat{a}_{l,n} = 1$): As each subcarrier can only be assigned to at most one user based on constraint (4.13), allocating $bit(i^*, j^*, z^*)$ to subcarrier $n$ will first require reallocating the bits of user $l$ that were previously assigned to subcarrier $n$ to other subcarriers. We define

$$\mathcal{S}_{l,n} = \{bit(l, j, z)|\hat{b}^{j,z}_{l,n} = 1, j \in \mathcal{J}_l, z \in \{1, \dots, B^j_l\}\} \quad (5.12)$$

to denote the set of bits of user $l$ currently assigned to subcarrier $n$. To prevent nested bit reallocations, we restrict the reallocation of bits in $\mathcal{S}_{l,n}$ only to subcarriers that are either unassigned or previously assigned to user $l$. We define

$$\Omega_l = \{m \in \mathcal{N}|\hat{a}_{i,m} = 0 \ \forall i \in \mathcal{I} \text{ or } \hat{a}_{l,m} = 1\} \quad (5.13)$$

to denote the set of subcarriers that bits in $\mathcal{S}_{l,n}$ can be reallocated to. The bit reallocations are done iteratively in a FIFO manner by assigning bits in $\mathcal{S}_{l,n}$, one bit at a time, to the subcarriers in $\Omega_l$. For each $bit(l, j, z) \in \mathcal{S}_{l,n}$, the power, $p'^{j,z}_{l,m}$, required to reallocate the bit is computed using either (5.10) or (5.11) for all $m \in \Omega_l$. The subcarrier that requires the least power, $m^* = \arg\min_{m \in \Omega_l} p'^{j,z}_{l,m}$, is selected. This procedure repeats until all the bits in $\mathcal{S}_{l,n}$ have been reallocated. The power, $\hat{p}_{l,n}$, previously assigned to user $l$, subcarrier $n$ is reclaimed and $bit(i^*, j^*, z^*)$ is assigned to subcarrier $n$. The

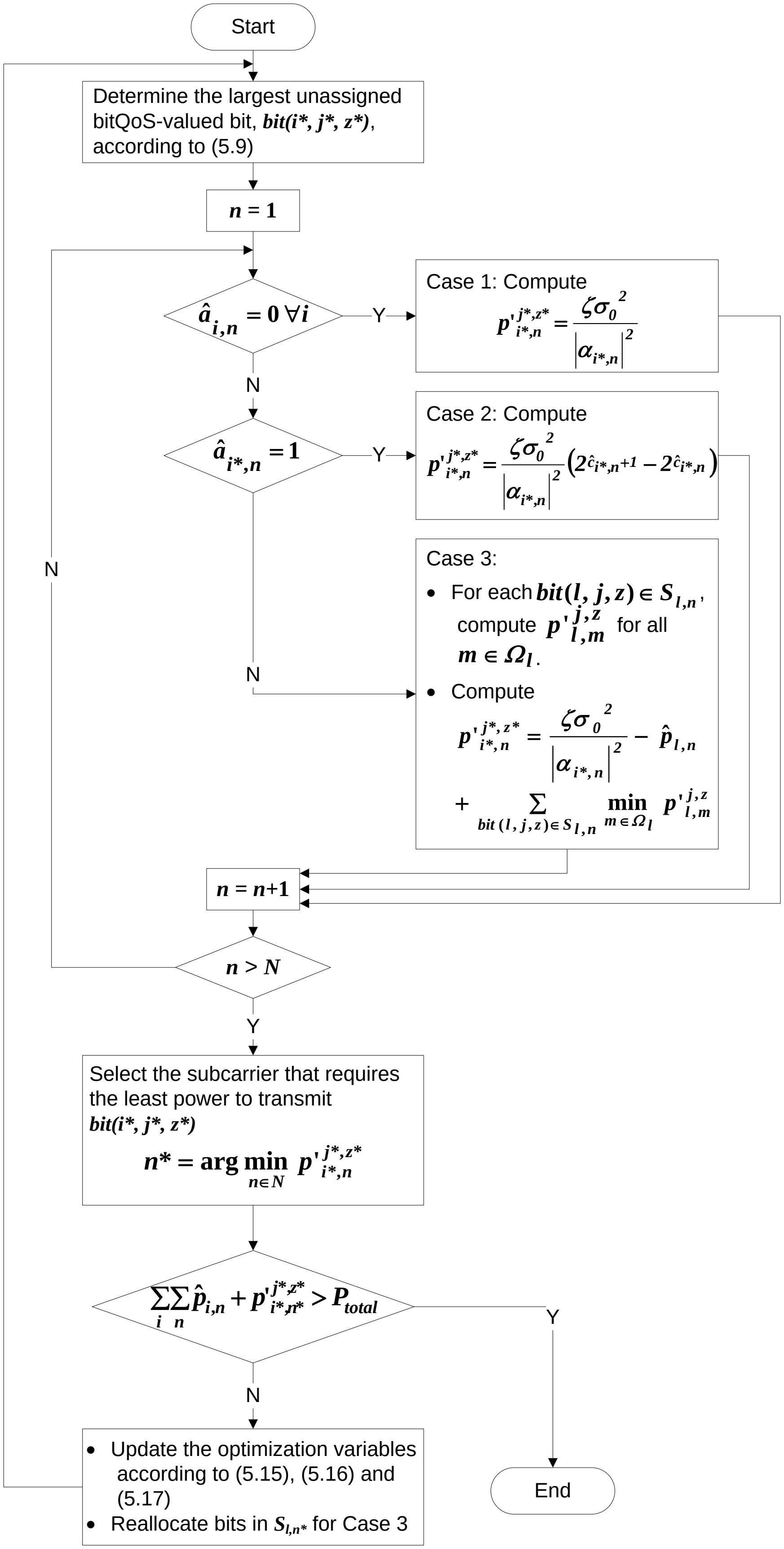

Start
Determine the largest unassigned bitQoS-valued bit, $bit(i^*, j^*, z^*)$, according to (5.9)
$n = 1$
$\hat{a}_{i,n} = 0 \; \forall i$
Y
Case 1: Compute
$p'^{j^*,z^*}_{i^*,n} = \frac{\zeta\sigma_0^2}{|\alpha_{i^*,n}|^2}$
N
$\hat{a}_{i^*,n} = 1$
Y
Case 2: Compute
$p'^{j^*,z^*}_{i^*,n} = \frac{\zeta\sigma_0^2}{|\alpha_{i^*,n}|^2}\left(2^{\hat{c}_{i^*,n}+1} - 2^{\hat{c}_{i^*,n}}\right)$
N
N
Case 3:
For each $bit(l, j, z) \in S_{l,n}$, compute $p'^{j,z}_{l,m}$ for all $m \in \Omega_l$.
Compute
$p'^{j^*,z^*}_{i^*,n} = \frac{\zeta\sigma_0^2}{|\alpha_{i^*,n}|^2} - \hat{p}_{l,n} + \sum_{bit(l,j,z)\in S_{l,n}} \min_{m\in\Omega_l} p'^{j,z}_{l,m}$
$n = n+1$
$n > N$
Y
Select the subcarrier that requires the least power to transmit $bit(i^*, j^*, z^*)$
$n^* = \arg\min_{n\in N} p'^{j^*,z^*}_{i^*,n}$
$\sum_i \sum_n \hat{p}_{i,n} + p'^{j^*,z^*}_{i^*,n^*} > P_{total}$
Y
N
Update the optimization variables according to (5.15), (5.16) and (5.17)
Reallocate bits in $S_{l,n^*}$ for Case 3
End

**Figure 5.2:** BABL-FM Flow Chart

power, $p'^{j^*,z^*}_{i^*,n}$, required to transmit $bit(i^*, j^*, z^*)$ on subcarrier $n$ is

$$p'^{j^*,z^*}_{i^*,n} = \frac{\zeta\sigma_0^2}{|\alpha_{i*,n}|^2} - \hat{p}_{l,n} + \sum_{bit(l,j,z)\in\mathcal{S}_{l,n}} \min_{m\in\Omega_l} p'^{j,z}_{l,m}. \tag{5.14}$$

Based on the above three possible cases, the subcarrier that requires the least power to transmit $bit(i^*, j^*, z^*)$ is selected as $n^* = \arg\min_{n\in\mathcal{N}} p'^{j^*,z^*}_{i^*,n}$ and this bit assignment is performed if $\sum_i \sum_n \hat{p}_{i,n} + p'^{j^*,z^*}_{i^*,n^*} \leq P_{total}$. The current intermediate optimization variables are then updated as follows:

$$\begin{aligned}
\hat{a}_{i^*,n^*} &= 1 && (5.15)\\
\hat{b}^{j^*,z^*}_{i^*,n^*} &= 1 && (5.16)\\
\hat{p}_{i^*,n^*} &= \begin{cases} \dfrac{\zeta\sigma_0^2}{|\alpha_{i^*,n^*}|^2}, & \text{for Case 1;} \\ \dfrac{(2^{\hat{c}_{i^*,n^*}+1} - 1)\zeta\sigma_0^2}{|\alpha_{i^*,n^*}|^2}, & \text{for Case 2;} \\ \dfrac{\zeta\sigma_0^2}{|\alpha_{i^*,n^*}|^2}, & \text{for Case 3,} \end{cases} && (5.17)
\end{aligned}$$

along with the reallocation of the bits in $\mathcal{S}_{l,n^*}$ for Case 3 as necessary. The next largest bitQoS-valued bit is then selected for the next bit assignment iteration. This iterative algorithm repeats until

$$\sum_i \sum_n \hat{p}_{i,n} + p'^{j^*,z^*}_{i^*,n^*} > P_{total} \tag{5.18}$$

or that all the bits in the data buffers of all users have been assigned.

### 5.3.2 BABL-NFM Scheduling Policy

BABL-NFM is identical to BABL-FM with the exception that application bits from the different flows of a user cannot be assigned to the same subcarrier, i.e., each subcarrier assigned to the user can only carry bits from a single application flow of that user. As such, the bits from all flows of a user are not combined into one queue as in BABL-FM, but rather each application flow $j$ of user $i$ in BABL-NFM is regarded as a separate user with the same channel gain $\alpha_{i,n}$. Specifically, OP4.2 is modified as follows: we replace constraint (4.13) with

$$\sum_{i=1}^{I}\sum_{j=1}^{J_i} a_{i,n}^{j} \leq 1 \quad \forall n. \tag{5.19}$$

In addition, the variables $a_{i,n}$, $p_{i,n}$ and $c_{i,n}$ take dependence on $j$ and are replaced with $a_{i,n}^{j}$, $p_{i,n}^{j}$ and $c_{i,n}^{j}$ in BABL-NFM.

## 5.4 Simulation Results

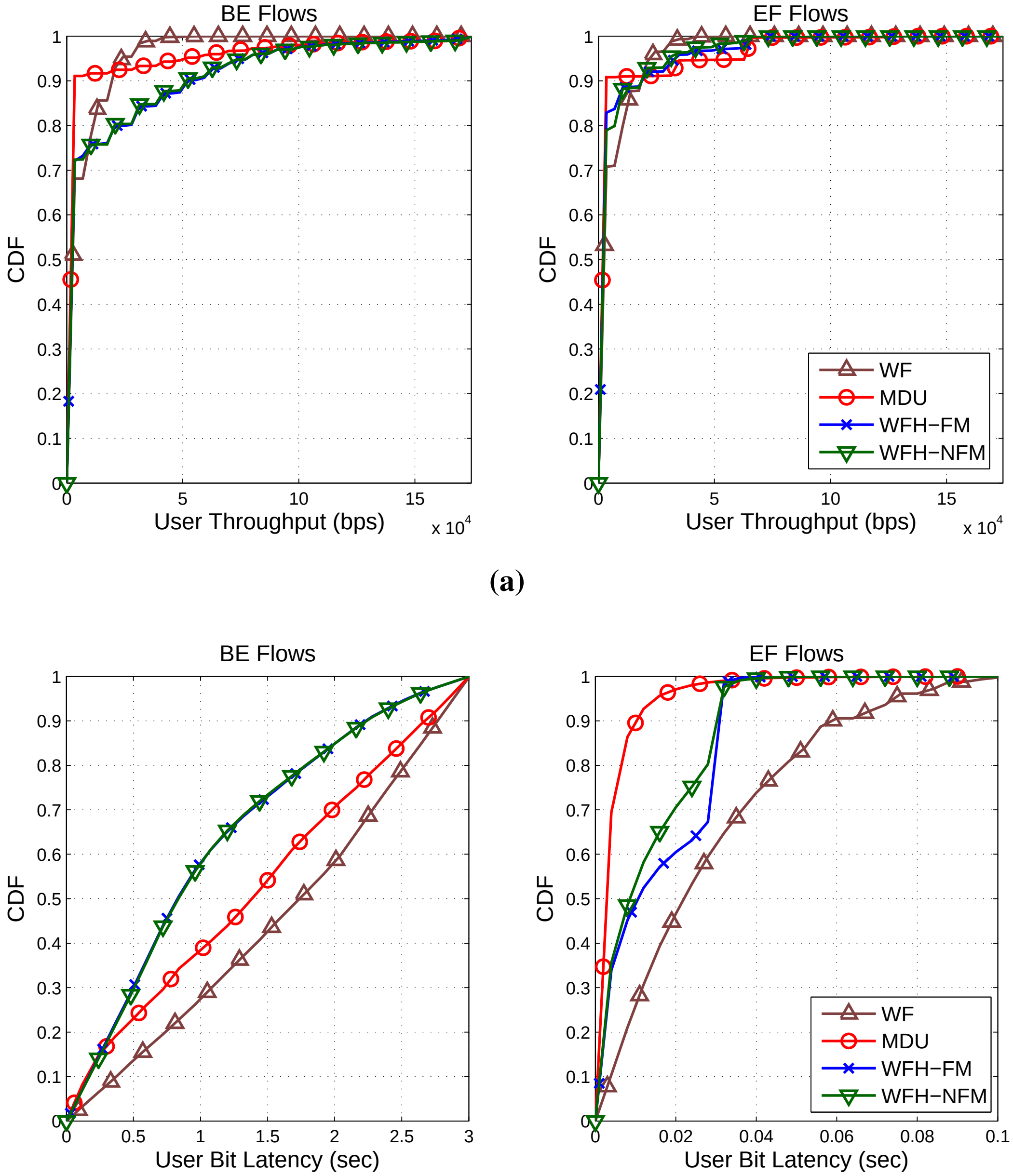
The WFH-FM, WFH-NFM, BABL-FM and BABL-NFM scheduling policies described in Sections 5.2.1, 5.2.2, 5.3.1 and 5.3.2, respectively, were simulated in Matlab using the system model described in Section 4.2. In the simulation, it is assumed that each user has 1 BE flow and 1 EF flow. The parameter values used in our simulation are listed in Tables 4.1 and 4.2.

### 5.4.1 WFH Simulation Results

We next discuss the performance of WFH-FM and WFH-NFM and the comparative schemes described in Section 4.4.3 under two system loading scenarios: A) heavy load and B) different loads. Simulation results were obtained for mixed-traffic scenarios for $I = \{4, 6, 8\}$ with a simulation length of $K = \{18000, 10000, 12000\}$ OFDM symbols, respectively.

#### WFH-FM performance under heavy load

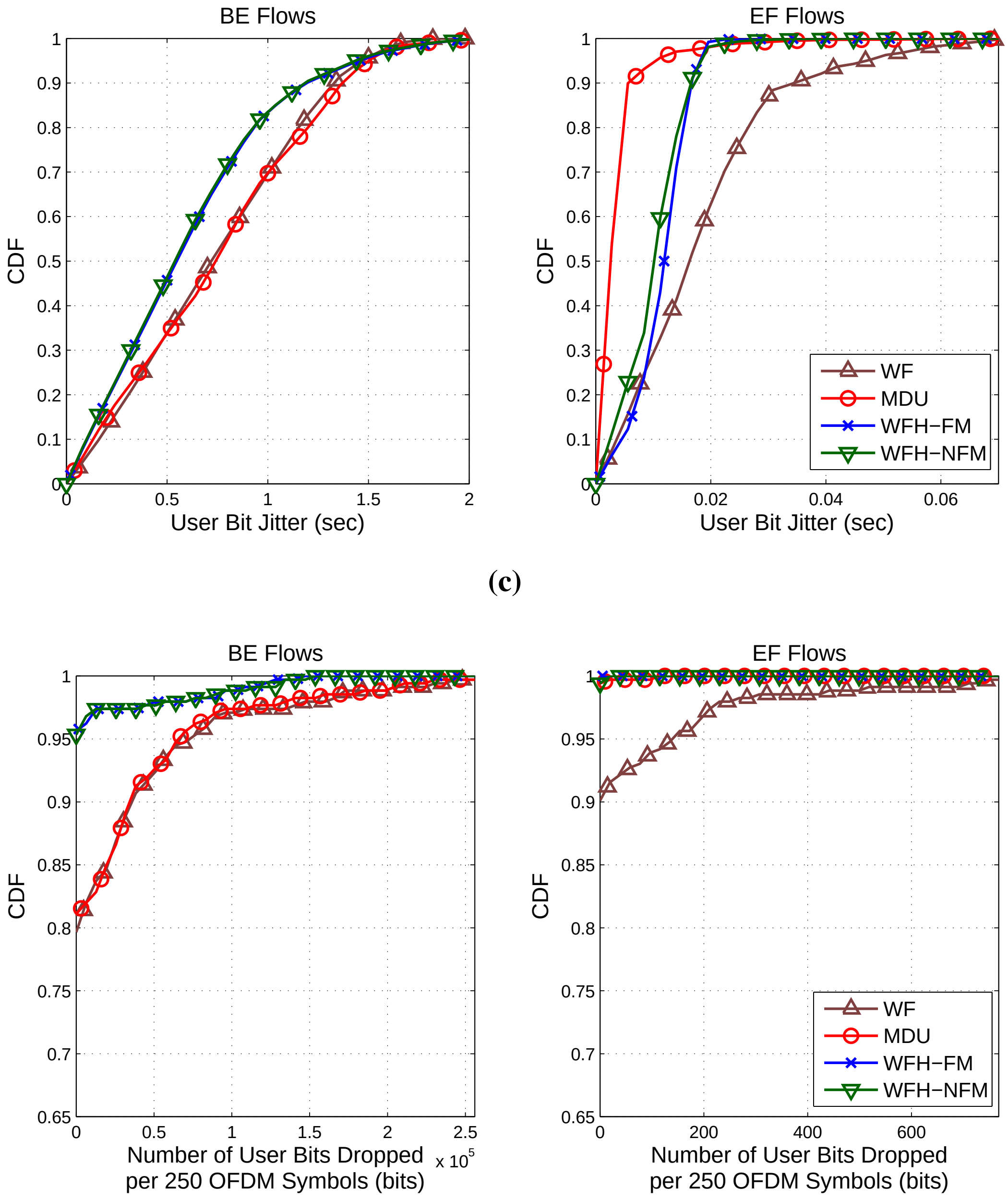
The CDF plots for user throughout, user bit latency, user bit jitter and number of user bits dropped for a system with $I = 8$, $N = 18$, 1 BE and 1 EF flow for each user are shown in Fig. 5.3. The CDF plots are averaged over $I$ users and obtained from $TP_i^j(k)$, $LT_i^{j,z}$, $JT_i^{j,z}$ and $BD_i^j(k)$, respectively, where the term, $JT_i^{j,z} = |LT_i^{j,z} - LT_{user}(i)|$. Figs. 5.3a and 5.3d show that WFH-FM not only has the highest user throughput for BE but also the lowest number of user bits dropped for both BE and EF traffic. From Fig. 5.3a, it can be seen that the MDU user EF throughput is slightly higher than that of WFH-FM; however, the MDU user BE throughput and MDU number of user BE bits dropped are significantly



**(a)**

**(b)**

**Figure 5.3:** WFH: Performance for a System with $I = 8$, $N = 18$, 1 BE and 1 EF Flow for each User (a) CDF of User Throughput (b) CDF of User Bit Latency



**(c)**

**(d)**

**Figure 5.3:** WFH: Performance for a System with $I = 8$, $N = 18$, 1 BE and 1 EF Flow for each User (Continued) (c) CDF of User Bit Jitter (d) CDF of Number of User Bits Dropped per 250 OFDM Symbols

worse compared to WFH-FM as the MDU scheduling policy strictly favors EF traffic in a mixed-traffic environment. WF, on the other hand, achieves the lowest user EF throughput and also the highest number of user EF bits dropped as it does not have QoS provisioning to schedule EF traffic in an attempt to meet the scheduling delay threshold. In terms of user bit latency, Fig. 5.3b shows that WFH-FM has the lowest user BE bit latency and the second lowest user EF bit latency. Only MDU has a better user EF bit latency than WFH-FM due to its strict bias towards EF traffic in a mixed-traffic environment. WFH-FM does not achieve the lowest user bit latency for EF traffic because the finesse control of the bitQoS-aware RA framework trades off a longer user EF scheduling delay (albeit within the packet drop threshold) for gains in both the user BE throughput (highest) and number of user BE and EF bits dropped (lowest). Similar to user bit latency, we see from Fig. 5.3c that WFH-FM has the lowest user BE bit jitter and the second lowest user EF bit jitter. The simulation results confirm the performance gains of the proposed WFH-FM scheduling policy which adopted the bitQoS-aware RA framework against the other comparative scheduling policies.

### WFH-FM performance under different loads

The average system throughput of the scheduling policies with no flow merging (WF, WF-FB, MDU and WFH-NFM) and the scheduling policy with flow merging (WFH-FM) as a function of $I$ are shown in Fig. 5.4. We see from Fig. 5.4 that the analytical throughput agrees very closely with the simulation results of WF-FB and that WFH-FM achieves the highest overall system throughput when compared to WF and MDU. While the objective function of WFH-FM is to maximize total bitQoS-weighted throughput, WFH-FM provides a good overall system throughput in part due to the adoption of the greedy multi-user water-filling in the first step of WFH-FM. The average system throughput of WFH-FM increases monotonically with $I$, for $I = \{4, 6, 8\}$.

The plots in Fig. 5.5 show the average user throughput, average user latency, average user jitter and average user packet drop probability for $I = \{4, 6, 8\}$ users. It can be observed that the performance of WFH-FM relative to the comparative scheduling policies is, in general,

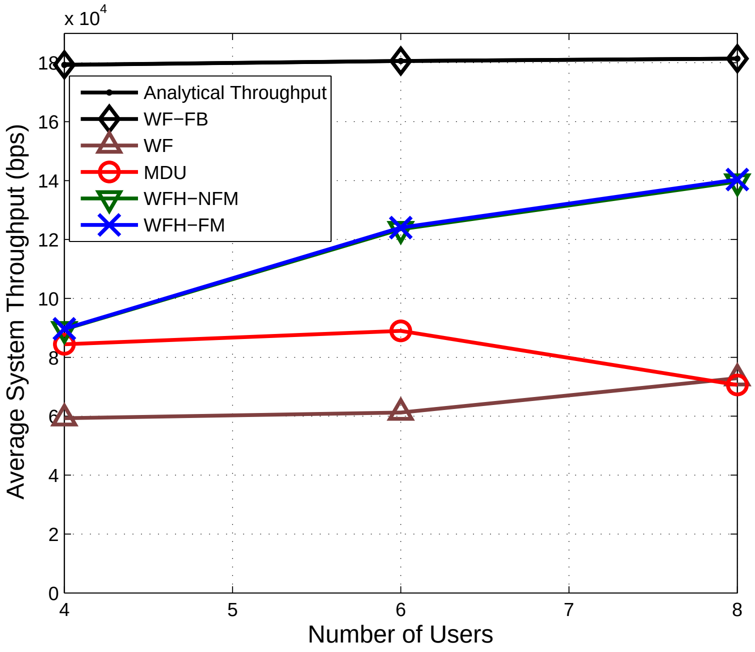
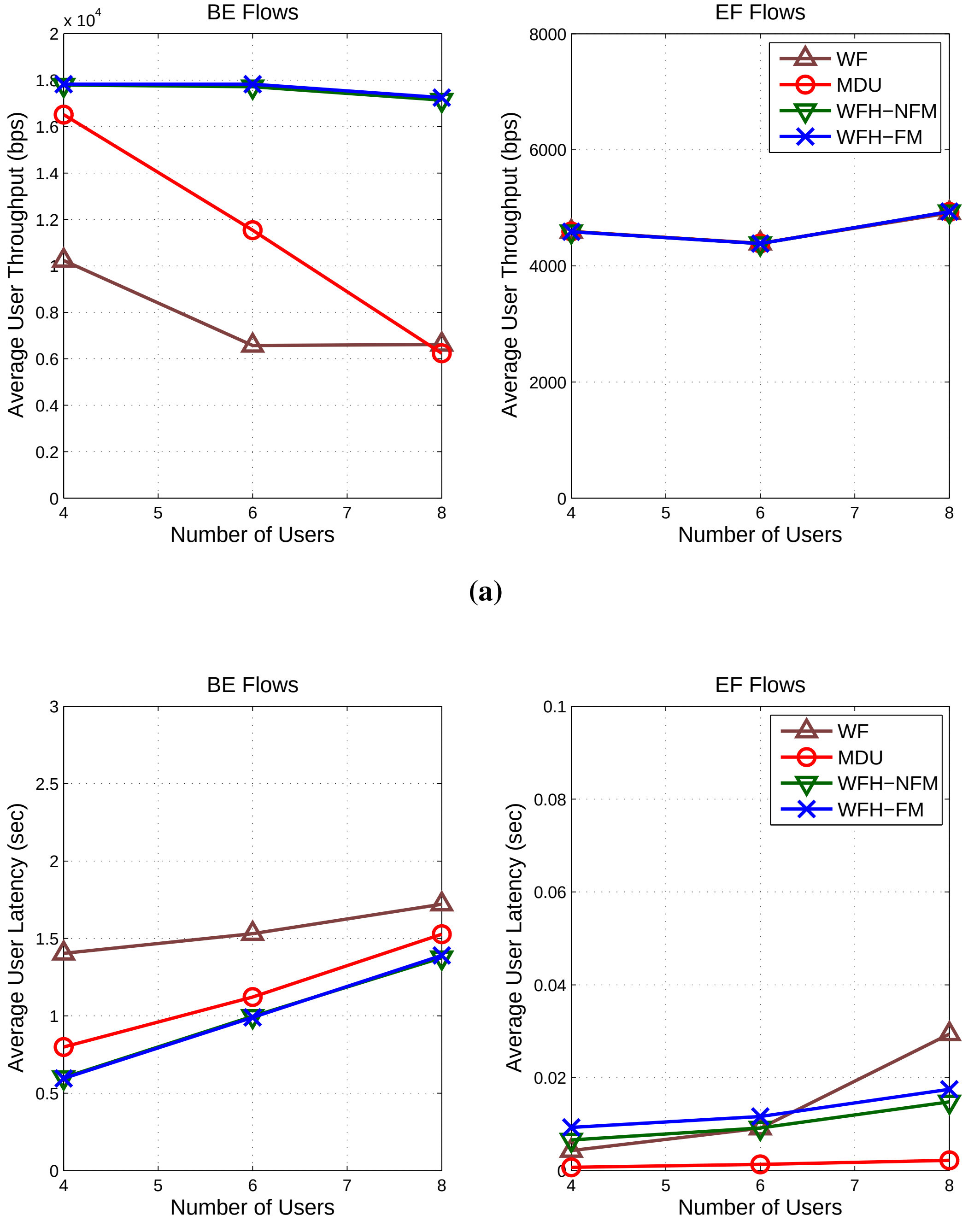


**Figure 5.4:** WFH: Average System Throughput under Different Loads

insensitive to the different loads across all the QoS metrics considered. In Fig. 5.5a, we see that the user BE throughput decreases monotonically and the user EF throughput is relatively constant for all scheduling polices as $I$ (system loading) increases. Figs. 5.5a and 5.5d show that WFH-NFM/FM not only have the highest user throughput but also the lowest user packet drop probability for both BE and EF traffic. While the MDU user EF throughput is close to that of WFH-NFM/FM, the MDU user BE throughput and MDU user BE packet drop probability are significantly worse than WFH-NFM/FM as the MDU scheduling policy strictly favors EF traffic in a mixed-traffic environment. On the other hand, WF has the lowest user BE throughput and also the highest user EF packet drop probability as it does not have QoS provisioning to meet the scheduling delay thresholds.

In terms of user latency, Fig. 5.5b shows that WFH-NFM/FM have the lowest user BE latency. MDU has a lower user EF latency than WFH-NFM/FM due to its strict bias towards EF traffic in a mixed-traffic environment. However, the MDU user EF latency gain comes at the expense of the user BE throughput and user BE packet drop probability. WF, with no QoS provisions, suffers the highest user BE latency. WFH-NFM/FM do not achieve the lowest user latency for EF traffic as the bitQoS-aware scheduling trades off a longer user scheduling delay (albeit within the scheduling delay threshold) for gains in both the user throughput (highest) and user packet drop probability (lowest) for both BE and EF traffic. This trade-off is possible since in OFDM, data is loaded onto subcarriers in units of bits and the latency QoS is satisfied as long as the bit waiting time does not exceed the scheduling delay threshold. By applying the bitQoS function at the bit-level as proposed, system providers can trade off the bit waiting time for a reduction in the user packet drop probability by prioritizing which bit to transmit based on its closeness to the scheduling delay threshold. This finer resolution of control provides an additional flexibility to push back the scheduling of bits that are not as close to the scheduling delay threshold (i.e., by increasing the bit waiting time) so as to allow the servicing of more "urgent" bits when necessary. As long as this push-back does not cause the bit waiting time to exceed the scheduling delay threshold, bits will be serviced within their scheduling delay thresholds, resulting in a simultaneous increase in user throughput and a reduction in user packet drop probability. The WFH-NFM/FM user EF latency are also influenced by the bitQoS function (4.4) which explicitly gives bits from the BE (delay-tolerant) flows a higher urgency when $w_i^{BE,z}(k) \leq \eta_{EF}$ so as to reduce the BE buffer backlog, if necessary.

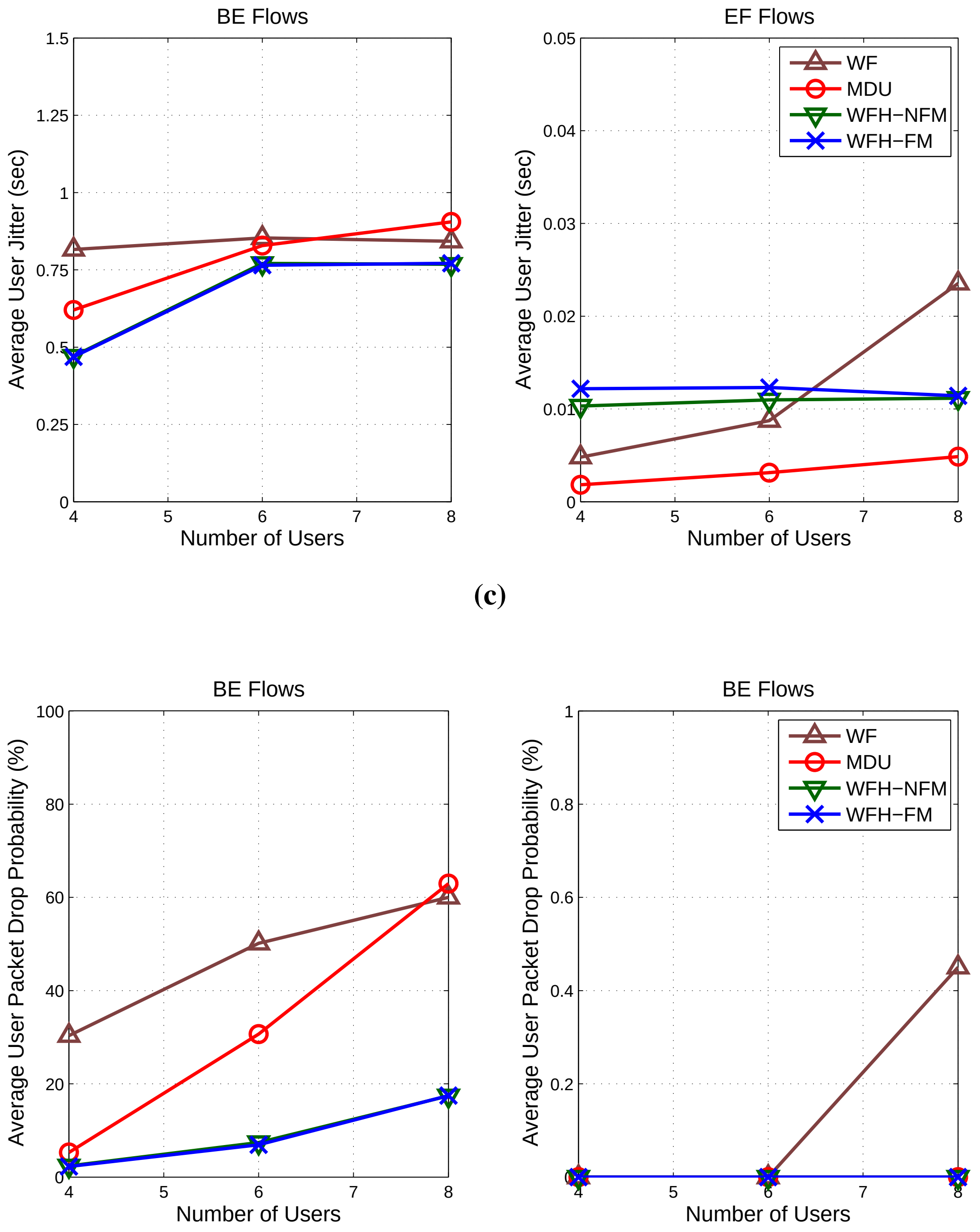
Similar to user latency, we see from Fig. 5.5c that WFH-NFM/FM have the lowest user BE jitter. For delay and jitter sensitive applications (EF flows), we note from Figs. 5.5b and 5.5c, that the plots for WFH-NFM/FM are flatter than WF as $I$ varies, i.e., user EF latency and user EF jitter are less sensitive to different system loads. This is particularly beneficial for the sizing of input buffers in mobile devices for delay and jitter sensitive applications. As shown in Fig. 5.5d, WFH-NFM/FM are able to maintain the lowest user packet drop probability for both BE and EF traffic across the different system loads and thus provides the highest user throughput for both BE and EF traffic among the comparative scheduling policies.



**(a)**

**(b)**

**Figure 5.5:** WFH: Performance for Systems under Different Loads (a) Average User Throughput (b) Average User Latency



**(c)**

**(d)**

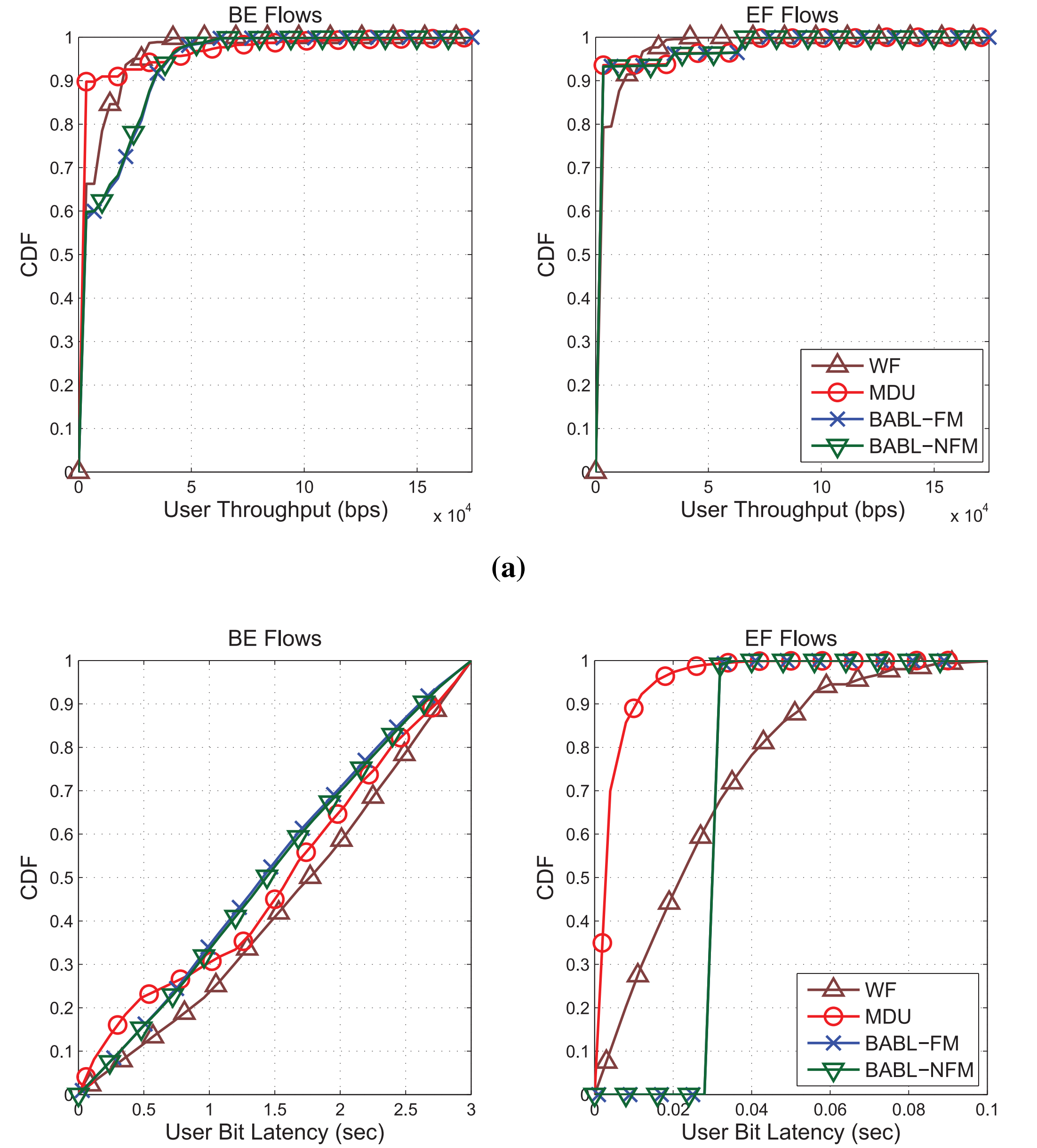
**Figure 5.5:** WFH: Performance for Systems under Different Loads (Continued) (c) Average User Jitter (d) Average User Packet Drop Probability

### 5.4.2 BABL Simulation Results

We next discuss the performance of BABL-FM and BABL-NFM and the comparative schemes described in Section 4.4.3 under two system loading scenarios: A) heavy load and B) different loads. Simulation results were obtained for mixed-traffic scenarios for $I = \{4, 6, 8\}$ with a simulation length of $K = \{15000, 11000, 4500\}$ OFDM symbols, respectively.

**BABL-FM performance under heavy load**

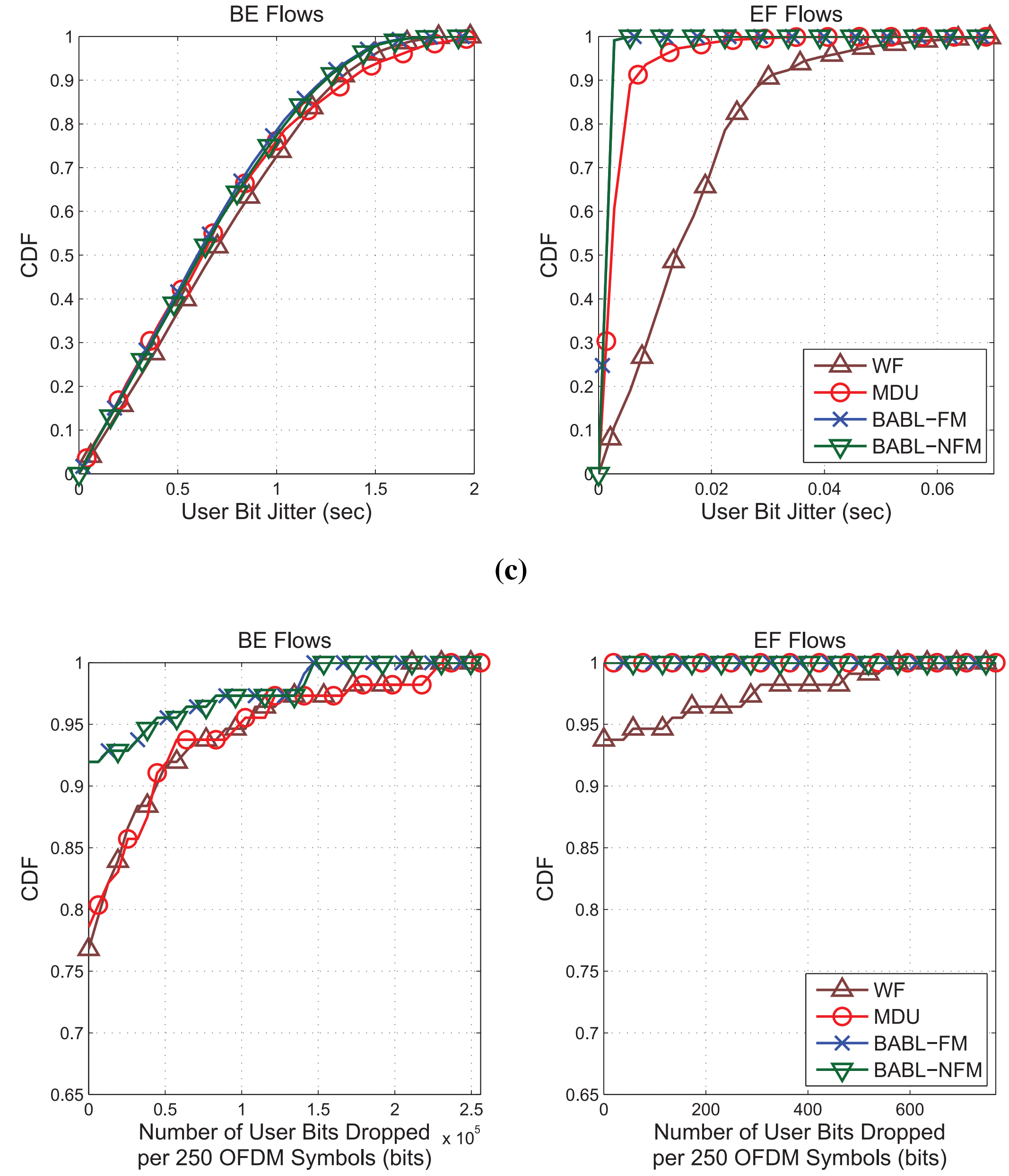
The CDF plots of user throughout, user bit latency, user bit jitter and number of user bits dropped for a system with $I$ = 8, $N$ = 18, 1 BE and 1 EF flow for each user are shown in Fig. 5.6. The CDF plots are averaged over $I$ users and obtained from $TP_i^j(k)$, $LT_i^{j,z}$, $JT_i^{j,z}$ and $BD_i^j(k)$, respectively, where the term, $JT_i^{j,z} = |LT_i^{j,z} - LT_{user}(i)|$. Figs. 5.6a and 5.6d show that BABL-FM not only has the highest user throughput but also the lowest number of user bits dropped for both BE and EF traffic. From Fig. 5.6a, it can be seen that the MDU user EF throughput is close to that of BABL-FM; however, the MDU user BE throughput and MDU number of user BE bits dropped are significantly worse compared to BABL-FM as the MDU scheduling policy strictly favors EF traffic in a mixed-traffic environment. WF, on the other hand, achieves the lowest user EF throughput and also the highest number of user EF bits dropped as it does not have QoS provisioning to schedule EF traffic in an attempt to meet the scheduling delay threshold. In terms of user bit latency, Fig. 5.6b shows that BABL-FM has the lowest user BE bit latency. However, MDU has a lower user EF bit latency than BABL-FM due to its strict bias towards EF traffic in a mixed-traffic environment but the MDU user EF bit latency gain comes at the expense of the user BE throughput and number of user BE bits dropped. BABL-FM does not achieve the lowest user bit latency for EF traffic as the bitQoS-aware scheduling trades off a longer user EF scheduling delay (albeit within the scheduling delay threshold) for gains in both the user throughput (highest) and number of user bits dropped (lowest) for both BE and EF traffic. The BABL-FM user EF bit latency is also influenced by the bitQoS function (4.4) which explicitly gives bits from the BE (delay-tolerant) flows a higher urgency when $w_i^{BE,z}(k) \leq \eta_{EF}$ so as to reduce the buffer backlog,



**(a)**

**(b)**

**Figure 5.6:** BABL: Performance for a System with $I = 8$, $N = 18$, 1 BE and 1 EF Flow for each User (a) CDF of User Throughput (b) CDF of User Bit Latency



**(c)**

**(d)**

**Figure 5.6:** BABL: Performance for a System with $I = 8$, $N = 18$, 1 BE and 1 EF Flow for each User (Continued) (c) CDF of User Bit Jitter (d) CDF of Number of User Bits Dropped per 250 OFDM Symbols

resulting in the EF traffic being allocated with $LT_i^{j,z}$ around $\eta_{EF}$ as shown in Fig. 5.6b. In addition, we note that since BABL-FM has a negligible number of user EF bits dropped, we can further improve the BABL-FM user BE throughput by either increasing $\eta_{EF}$ to a value closer to $T_{EF}$ or increasing the priority of BE traffic by adjusting the parameter values of the BE bitQoS function. In contrast, WF, with no QoS provisions, has the highest user BE bit latency and the second highest user EF bit latency. We see from Figs. 5.6c and 5.6d that BABL-FM has the lowest user bit jitter and lowest number of user bits dropped for both BE and EF traffic.

### BABL-FM performance under different loads

The average system throughput of the various scheduling policies (WF, WF-FB, MDU, BABL-FM and BABL-NFM) shown in Fig. 5.7 were obtained by simulation as a function of $I$ based on the system model described in Section 4.2 with the simulation parameter values listed in Tables 4.1 and 4.2. We see from Fig. 5.7 that the analytical throughput agrees very closely with the simulation results of WF-FB and that BABL-FM achieves the highest average system throughput when compared to WF and MDU. While the objective function of BABL-FM is to maximize the total bitQoS-weighted throughput, it provides a good average system throughput in part due to BABL-FM iteratively assigning the largest unassigned bitQoS-valued bit to the subcarrier requiring the least amount of power. The average system throughput of BABL-FM increases monotonically with $I$, for $I = \{4, 6, 8\}$.

The plots in Fig. 5.8 show the average user throughput, average user latency, average user jitter and average user packet drop probability, obtained by averaging $TP_{user}(i)$, $LT_{user}(i)$, $JT_{user}(i)$ and $PDP_{user}(i)$ over $I$ users, respectively, for $I = \{4, 6, 8\}$. We see from Fig. 5.8 that the performance of BABL-FM relative to the comparative scheduling policies is, in general, insensitive to the different loads across all the QoS metrics considered. In Fig. 5.8a, we see that the user BE throughput increases monotonically only for BABL-FM as $I$ (system loading) increases and the user EF throughput is constant for all scheduling polices. In terms of user latency, we see from Fig. 5.8b that BABL-FM achieves the lowest user BE latency

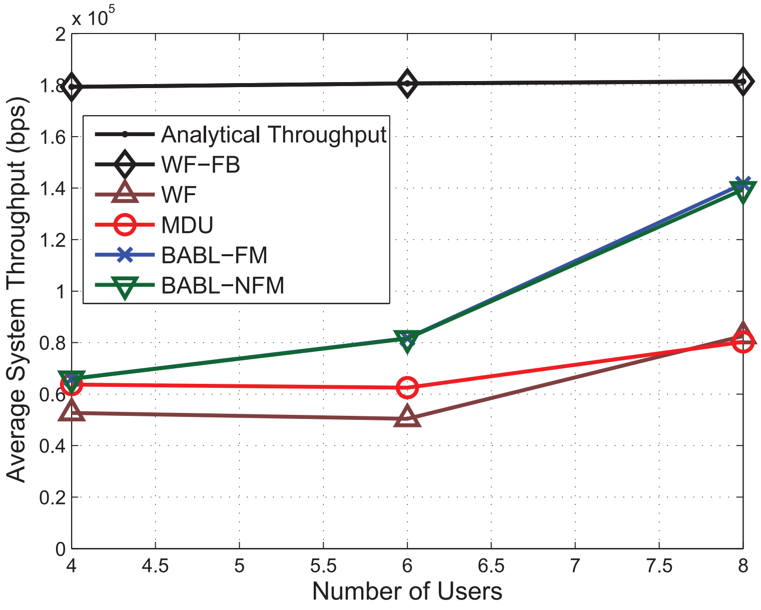
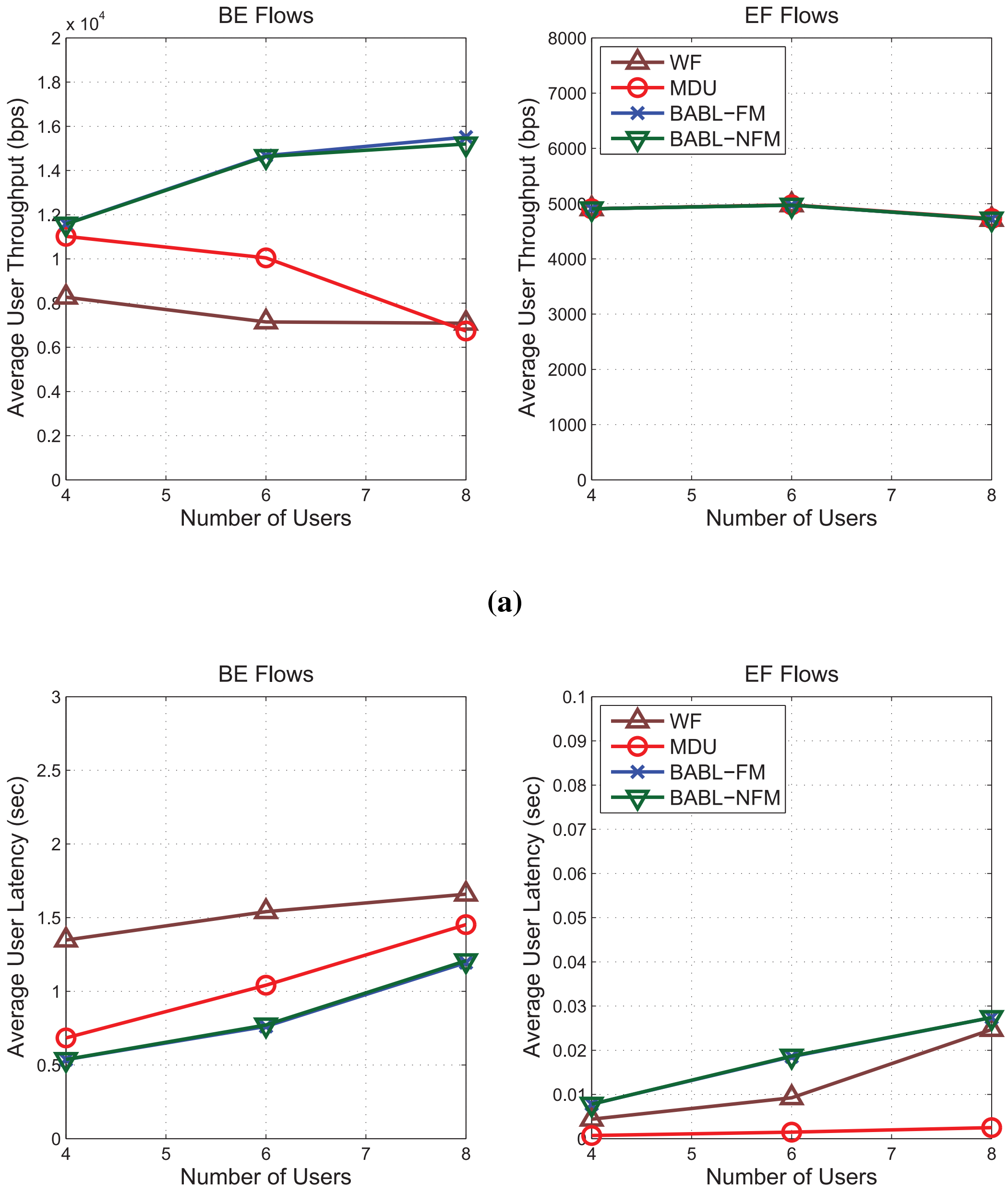


**Figure 5.7:** BABL: Average System Throughput under Different Loads

and the highest user EF latency across all $I$ in part due to the mixed-traffic bitQoS function (4.4) allowing BE traffic to reduce backlog when $w_i^{BE,z}(k) \leq \eta_{EF}$. In terms of user jitter, we see from Fig. 5.8c that BABL-FM achieves the lowest user BE jitter whereas the user EF jitter decreases as $LT_i^{EF,z}$ clusters around $\eta_{EF}$ when $I$ increases. In addition, BABL-FM is also able to maintain the lowest BE and EF user packet drop probabilities across the different system loads as shown in Fig. 5.8d and thus provides the highest user throughput for both BE and EF traffic among the comparative scheduling policies. Note that Figs. 5.7, 5.6 and 5.8 do not show the optimal solution to OP4.2 using the commercial MINLP optimization solver package due to the prohibitive computation time required.

## 5.5 Conclusion

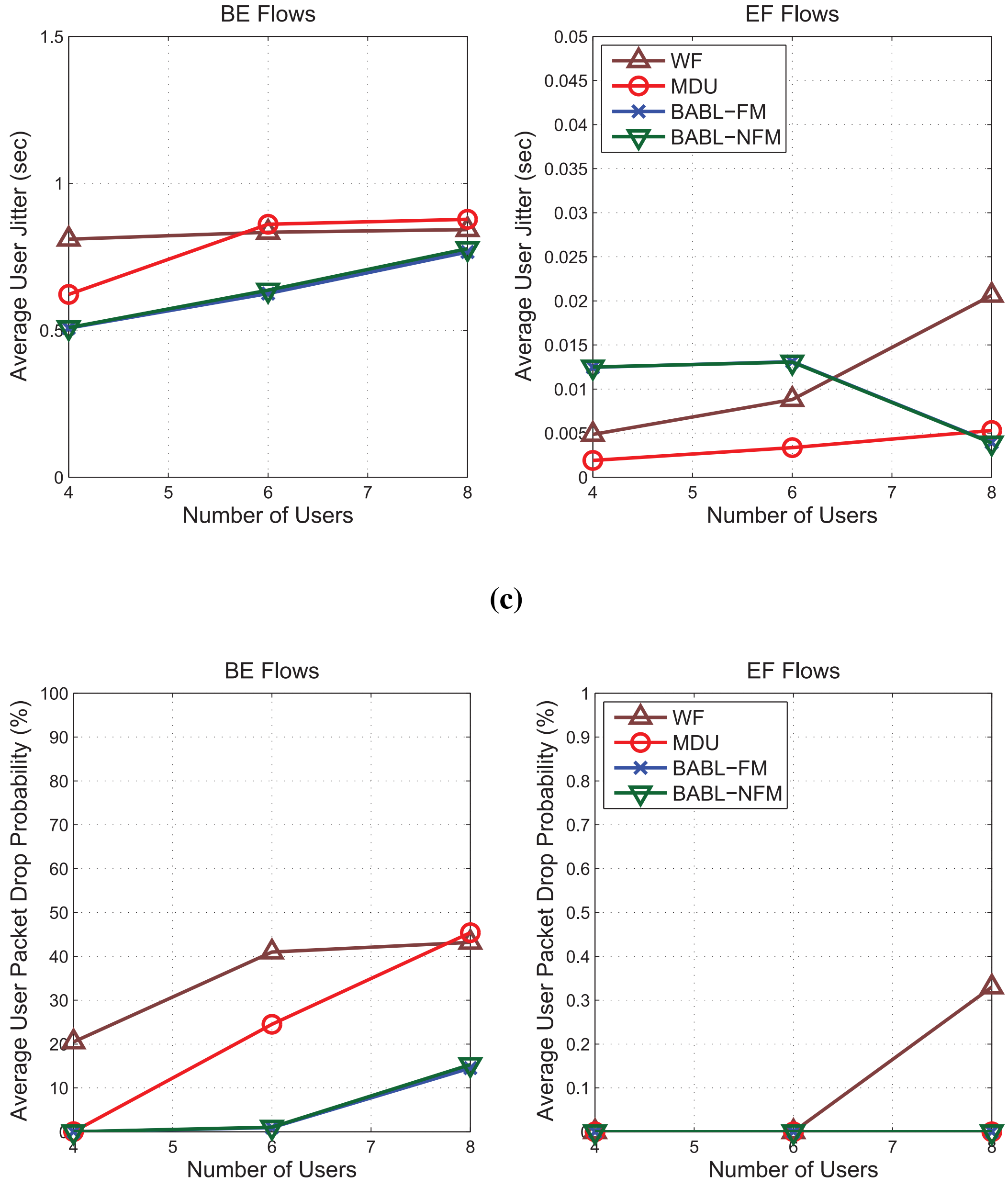
A bitQoS-aware RA framework that adaptively matches the QoS requirements of the user application bits to the characteristics of the OFDM subcarriers was proposed for multi-user OFDM systems. The performance gains achievable from the proposed framework are demonstrated using suboptimal water-filling-based WFH and bit-loading-based BABL scheduling policies. The results show that with the finesse bit-level control provided by the proposed



**(a)**

**(b)**

**Figure 5.8:** BABL: Performance for Systems under Different Loads (a) Average User Throughput (b) Average User Latency



**(c)**

**(d)**

**Figure 5.8:** BABL: Performance for Systems under Different Loads (Continued) (c) Average User Jitter (d) Average User Packet Drop Probability

framework, it is possible to simultaneously achieve both an increase in throughput and a reduction in packet drop probability in a mixed-traffic environment at the cost of a longer (albeit within the packet drop threshold) scheduling delay. This flexibility comes from the realization that in OFDM, data is loaded onto subcarriers in units of bits and the latency QoS is satisfied as long as the bit waiting time does not exceed the scheduling delay threshold. By applying the bitQoS function at the bit-level as proposed, system providers can trade off the bit waiting time for a reduction in the number of dropped packets by prioritizing which bit to transmit based on its closeness to the scheduling delay threshold. This finer resolution of control provides an additional flexibility to push back the scheduling of bits that are not as close to the scheduling delay threshold (i.e., by increasing the bit waiting time) so as to allow the servicing of more "urgent" bits when necessary. As long as this push-back does not cause the bit waiting time to exceed the scheduling delay threshold, bits will be serviced within their scheduling delay thresholds, resulting in a simultaneous increase in user throughput and a reduction in the number of user bits dropped.

Simulation results, obtained using the proposed WFH and BABL scheduling policies, show that the proposed bitQoS-aware RA framework is able to provide a substantial improvement in user throughput and user packet drop probability compared to scheduling policies that do not take QoS provisions into account such as WF and policies that consider only application flow QoS requirements such as MDU. In particular, WFH and BABL are also able to achieve the highest average system throughput across all considered system loads. In addition, it was found that in a multi-application system, the performance gains by allowing bits from different application flows of a user to be merged into a single subcarrier for transmission are small and should only be used if such gains, at the expense of the increased scheduling signaling overhead, are warranted. However, it provides the service providers the option to choose, based on computational resource availability, whether to let the BS fully take on the scheduling task with less scheduling signaling overhead as in WFH/BABL-NFM or let the MS share the computational burden with the BS at the expense of increased scheduling signaling overhead as in WFH/BABL-FM.

# Chapter 6

# Scheduling Signaling Overhead in BitQoS-aware Resource Allocation Framework [5]

## 6.1 Introduction

A novel bitQoS-aware RA framework is proposed in Chapter 4 which allows the exploitation of both multi-application and multi-bit diversities (in addition to multi-user and multi-channel diversities) in mixed-traffic OFDMA systems. It is shown in Chapter 5 that with the finesse control of bitQoS-aware scheduling, it is possible to simultaneously achieve both an increase in user throughput and a reduction in user packet drop probability by accepting a within packet drop threshold increase in user latency. However, as the granularity of RRM scheduling algorithms increases to more closely meet the different QoS requirements of multiple concurrent user application flows, the potential scheduling gain comes at the cost of an increased scheduling signaling overhead. This is due to the fact that the mapping between the application bits and the OFDM subcarriers need to be signaled using the control channel accompanying the data channel so that the receiver is able to extract the application bits from

[5] The material in this chapter is based on: C. E. Huang and C. Leung, "Scheduling signaling overhead in bitQoS-aware multi-flow OFDM systems," submitted.

the assigned OFDM subcarriers.

Only a few of the numerous published papers on RRM scheduling algorithms consider the scheduling signaling overhead. In [70], the compression of signaling information for adaptive multi-carrier systems is studied. It is shown that efficient compression schemes can reduce the amount of signaling information and increase system transmission efficiency. In [71], the authors attempt to reduce the scheduling signaling overhead by exploiting the correlation of the scheduling information in time. This is achieved by changing the subcarrier assignments in successive scheduling intervals only if the gain in system throughput is larger than the signaling overhead incurred with the reassignment. In [72], an algorithm for OFDMA downlink scheduling under a control signaling cost constraint is proposed. The authors formulate the subcarrier assignment as a combinatorial optimization problem with the objective of finding the subcarrier assignment that maximizes the system throughput while penalizing the cost for transmitting the scheduling information. In [73], a new scheme for encoding the scheduling information which exploits the correlation among different users' scheduling assignments is proposed to reduce the amount of scheduling information that needs to be transmitted. The scheme assumes that users with a high SNR can decode the scheduling information intended for all other users with a lower SNR and thus the scheduling information can be encoded differentially. In this chapter, we formulate a scheduling signaling overhead model to analyze the scheduling signaling overhead associated with the proposed bitQoS-aware RA framework and consider different schemes to compress the scheduling signaling information. The effective system throughput gains of the bitQoS-aware RA framework are determined so as to assess the tradeoff between the scheduling gain and signaling overhead.

This chapter is organized as follows: in Section 6.2, the scheduling signaling overhead model is presented and the required scheduling signaling information is described in Section 6.3. The entropy of the scheduling signaling information is evaluated in Section 6.4 and different schemes to compress the scheduling signaling information bits are described in Section 6.5. The simulation results are presented in Section 6.6 and the main findings are summarized in Section 6.7.

## 6.2 Scheduling Signaling Overhead Model

We examine the scheduling signaling overhead incurred by the proposed bitQoS-aware RA framework based on the control signaling evaluation model proposed in [74], where the authors compared the effects of different scheduling granularity, scheduling policies and control signaling transmission strategies in OFDMA systems. Specifically, we look at the case where the assignment of subcarriers is at a per-resource-element basis (i.e., each of the $N$ subcarriers can be assigned to different flows/users and the subcarrier assignment is updated at every OFDM symbol). It is assumed that the scheduling signaling information is compressed (when necessary) and broadcast to all $I$ users and that the scheduling signaling information bits for each OFDM symbol are transmitted with the application bits at each scheduling decision time $k$. In practice, the scheduling signaling information broadcast message needs to be encoded such that the user with the weakest channel condition is able to decode it. For simplicity, we also assume that subcarrier resources are pre-reserved for the transmission of the scheduling signaling information, i.e., there is no need to reallocate the application bits to take into account the transmission of the scheduling signaling information bits.

## 6.3 Scheduling Signaling Information

We consider the required scheduling signaling information bits at each scheduling decision time $k$ for 1) scheduling policies with no flow merging (NFM), 2) scheduling policies with flow merging (FM) and 3) scheduling policies with flow merging - grouped sorted (FMGS). The three types of scheduling policies are illustrated in Fig. 6.1.

### 6.3.1 Scheduling Policies with No Flow Merging

The scheduling decision at each scheduling decision time $k$ is represented by a $1 \times N$ subcarrier-to-flow vector, $\boldsymbol{U}^{NFM}(k) \triangleq \{u_n^{NFM}(k), \forall n \in \mathcal{N}\}$. The $n$-th element, $u_n^{NFM}(k)$, of the subcarrier-to-flow vector, $\boldsymbol{U}^{NFM}(k)$, is an integer from the set $\mathcal{J}_{sys} = \{1, \ldots, J_{sys}\}$ that indicates the flow $j$ of user $i$ to which subcarrier $n$ is assigned at time $k$. The set, $\mathcal{J}_{sys}$,

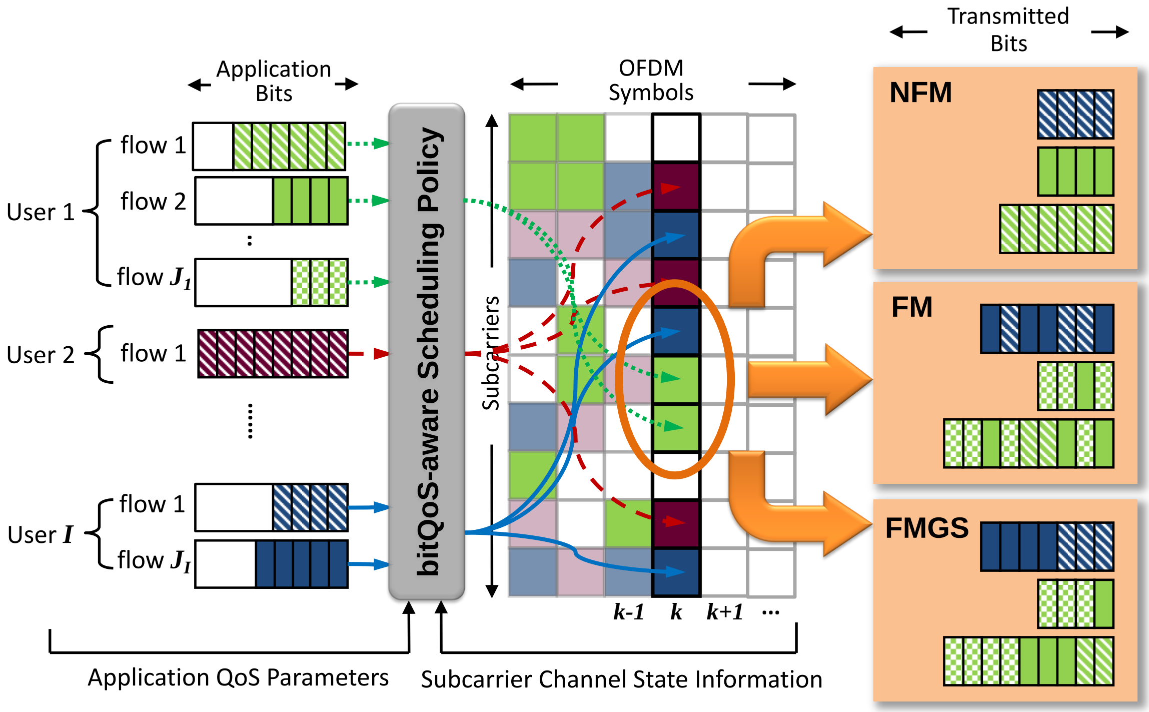


**Figure 6.1:** Mapping of application bits to OFDM subcarriers with different bitQoS-aware scheduling policies. NFM: Each subcarrier can only carry bits from a single application flow of a user. FM: Each subcarrier can carry bits from more than one application flow of a user. FMGS: Each subcarrier can carry bits from more than one application flow of a user. In addition, the bits on each subcarrier are grouped in a FIFO fashion by application flows and sorted in an ascending order by the flow index, $j$, prior to transmission.

contains the indices to all $flow(i,j), \forall i \in \mathcal{I}, j \in \mathcal{J}_i$ in the system and the term, $J_{sys} \triangleq \sum_{i=1}^{I} J_i$, denotes the total number of flows in the system.

### 6.3.2 Scheduling Policies with Flow Merging

The scheduling decision at each scheduling decision time $k$ is represented by a $1 \times N$ subcarrier-to-user vector, $\boldsymbol{U}^{FM}(k) \triangleq \{u_n^{FM}(k), \forall n \in \mathcal{N}\}$ and $N$ $1 \times M_n(k)$ bit-to-flow vectors, $\boldsymbol{V}_n^{FM}(k) \triangleq \{v_{n,z}^{FM}(k), \forall z = 1, \ldots, M_n(k)\}, \forall n \in \mathcal{N}$, where $M_n(k)$ is the total number of bits carried by subcarrier $n$ at time $k$. The $n$-th element, $u_n^{FM}(k)$, of the subcarrier-to-user vector, $\boldsymbol{U}^{FM}(k)$, is an integer from the set $\mathcal{I}$ that indicates the user $i$ to which subcarrier $n$ is assigned at time $k$. The $z$-th element, $v_{n,z}^{FM}(k)$, of the bit-to-flow vector for each subcarrier $n$,

$\boldsymbol{V}_n^{FM}(k)$, is an integer from the set $\mathcal{J}_{u_n^{FM}}$ that indicates the flow of user $u_n^{FM}(k)$ to which the $z$-th bit on subcarrier $n$ is assigned at time $k$.

### 6.3.3 Scheduling Policies with Flow Merging - Grouped Sorted

However, scheduling policies with flow merging, as described above, makes no assumption about the ordering of bits on each subcarrier $n$. While FM allows the BS to merge bits from different application flows without restriction, a significant amount of scheduling signaling overhead is incurred to communicate the mapping between the application bits on each OFDM subcarrier and application flows. To reduce the scheduling signaling overhead without decreasing the performance gains provided by the bitQoS-aware RA framework with flow merging, we can require the bits scheduled on each subcarrier to be grouped in a FIFO fashion by application flows and sorted in an ascending order by the flow index, $j$, prior to transmission. Hence, instead of having the BS signal the mapping between the application bits on each OFDM subcarrier and application flows for every single scheduled bit to be transmitted to the MS, the BS only needs to signal the number of consecutive bits belonging to each application flow $j$ on each subcarrier $n$. The scheduling decision at each scheduling decision time $k$ is represented by a $1 \times N$ subcarrier-to-user vector, $\boldsymbol{U}^{FMGS}(k) \triangleq \{u_n^{FMGS}(k), \forall n \in \mathcal{N}\}$ and $N$ $1 \times J_{u_n^{FMGS}(k)}$ bit-to-flow vectors, $\boldsymbol{V}_n^{FMGS}(k) \triangleq \{\tau_j^{FMGS}(k), \forall j \in \mathcal{J}_{u_n^{FMGS}(k)}\}$, $\forall n \in \mathcal{N}$. The $n$-th element, $u_n^{FMGS}(k)$, of the subcarrier-to-user vector, $\boldsymbol{U}^{FMGS}(k)$, is an integer from the set $\mathcal{I}$ that indicates the user $i$ to which subcarrier $n$ is assigned at time $k$. The $j$-th element, $\tau_j^{FMGS}(k)$, of the bit-to-flow vector for each subcarrier $n$, $\boldsymbol{V}_n^{FMGS}(k)$, is an integer from the set $\{0, \ldots, M_n(k)\}$ that indicates the number of bits belonging to flow $j$ at time $k$, where $\sum_{j=1}^{J_{u_n^{FMGS}(k)}} \tau_j^{FMGS}(k) = M_n(k)$.

It is assumed that each user is assigned its user index $i \in \mathcal{I}$ and its range of application flow indices $\{(i-1)J_{max}+1, \ldots, iJ_{max}\} \in \mathcal{J}_{sys}$ in the system during call admission. The term, $J_{max}$, denotes the maximum number of application flows a user can have as defined in [27, 28, 33] and the range of application flow indices for user $i$ is $\{1, \ldots, J_i\}$.

## 6.4 Scheduling Signaling Information Entropy

To gain some insight into the amount of scheduling signaling information bits incurred by scheduling policies with no flow merging, scheduling policies with flow merging and scheduling policies with flow merging - grouped sorted, we evaluate the entropies assuming a simplified model for the signaling information bits. Since the statistics of the scheduling decisions are not readily available for the considered scheduling policies, we determine an entropy upperbound (regardless of scheduling policy) by assuming that each subcarrier $n \in \mathcal{N}$ is independently and equally likely to be assigned to any flow $j \in \mathcal{J}_{sys}$ for scheduling policies with no flow merging or any user $i \in \mathcal{I}$ for scheduling policies with flow merging. Furthermore, for scheduling policies with flow merging and for scheduling policies with flow merging - grouped sorted, each bit $z, \forall z = 1, \ldots, M_n(k)$, on subcarrier $n$ is independently and equally likely to be mapped to any flow $j \in \mathcal{J}_{u_n^{FM}(k)}$ and $j \in \mathcal{J}_{u_n^{FMGS}(k)}$, respectively. In Section 6.6.2, we show that the entropy results obtained from this simplified model are useful in explaining the compressed scheduling signaling overhead results obtained by simulation.

Depending on whether flow merging is allowed, the entropy of the scheduling signaling information is determined by enumerating all the possible values that the pertinent vectors $\boldsymbol{U}^{NFM}(k)$, $\boldsymbol{U}^{FM}(k)$ and $\boldsymbol{V}_n^{FM}(k)$, and $\boldsymbol{U}^{FMGS}(k)$ and $\boldsymbol{V}_n^{FMGS}(k)$ can take on. All the possible assignment combinations are represented in a table which is assumed to be known at both the BS and MSs. At each scheduling decision time $k$, the index of the assignment combination corresponding to the scheduling decision is transmitted with the application bits. The number of bits required to represent the assignment combination index is determined by assuming that every assignment combination is equally likely. We determine the entropies for 1) scheduling policies with no flow merging, 2) scheduling policies with flow merging and 3) scheduling policies with flow merging - grouped sorted.

### 6.4.1 Scheduling Policies with No Flow Merging

For scheduling policies with no flow merging, each subcarrier $n$ assigned to user $i$ can only carry bits from a single application flow $j$ of that user, i.e., each subcarrier $n$ is assigned to one of $J_{sys}$ application flows. Hence, there are $(J_{sys})^N$ possible ways to assign all $N$ subcarriers to $J_{sys}$ flows. Assuming all $(J_{sys})^N$ possible assignments of subcarriers to application flows are equally likely, the entropy of the scheduling signaling information for NFM is given by

$$H^{NFM}(k) = \log_2 (J_{sys})^N = N \log_2 J_{sys}. \tag{6.1}$$

### 6.4.2 Scheduling Policies with Flow Merging

For scheduling policies with flow merging, each subcarrier $n$ assigned to user $i$ can carry up to $M_n(k)$ bits from any of $J_i$ flows of user $i$. There are $I^N$ possible ways to assign all $N$ subcarriers to $I$ users. Assuming that all $I^N$ possible assignments of subcarriers to users are equally likely, the corresponding entropy of $\boldsymbol{U}^{FM}(k)$ is given by $\log_2 I^N$. In addition, we assume that each bit, $z$, $\forall z = 1, \dots, M_n(k)$, on subcarrier $n$ is equally likely to be mapped to any flow $j \in \mathcal{J}_{u_n^{FM}(k)}$ and the mapping of the bits are independent from one bit to another. Hence, there are $(J_{u_n^{FM}(k)})^{M_n(k)}$ possible ways to map all $M_n(k)$ bits to $J_{u_n^{FM}(k)}$ flows. Assuming all $(J_{u_n^{FM}(k)})^{M_n(k)}$ possible mappings of bits to application flows are equally likely, the entropy of $\boldsymbol{V}_n^{FM}(k)$, $\forall n \in \mathcal{N}$ is given by $\log_2 \prod_{n=1}^{N} (J_{u_n^{FM}(k)})^{M_n(k)}$. Hence, the entropy of the scheduling signaling information for FM is given by

$$H^{FM}(k) = \log_2 I^N + \log_2 \prod_{n=1}^{N} (J_{u_n^{FM}(k)})^{M_n(k)} = N \log_2 I + \sum_{n=1}^{N} M_n(k) \log_2 J_{u_n^{FM}(k)}. \tag{6.2}$$

### 6.4.3 Scheduling Policies with Flow Merging - Grouped Sorted

The subcarrier-to-user vector for FMGS, $\boldsymbol{U}^{FMGS}(k)$, is determined identically as $\boldsymbol{U}^{FM}(k)$. Hence, the corresponding entropy of $\boldsymbol{U}^{FMGS}(k)$ is also given by $\log_2 I^N$. Determining the possible values of $\boldsymbol{V}_n^{FMGS}(k)$ is equivalent to finding the possible ways of distributing $M_n(k)$ indistinguishable balls into $J_{u_n^{FMGS}(k)}$ distinguishable urns [75]. This gives a total of $\binom{M_n(k) + J_{u_n^{FMGS}(k)} - 1}{J_{u_n^{FMGS}(k)} - 1}$ possible values. Assuming that all $\binom{M_n(k) + J_{u_n^{FMGS}(k)} - 1}{J_{u_n^{FMGS}(k)} - 1}$ possible values are equally likely, the corresponding entropy of $\boldsymbol{V}_n^{FMGS}(k)$, $\forall n \in \mathcal{N}$ is given by $\log_2 \prod_{n=1}^{N} \binom{M_n(k) + J_{u_n^{FMGS}(k)} - 1}{J_{u_n^{FMGS}(k)} - 1}$. Hence, the entropy of the scheduling signaling information for FMGS is given by

$$\begin{aligned} H^{FMGS}(k) &= \log_2 I^N + \log_2 \prod_{n=1}^{N} \binom{M_n(k) + J_{u_n^{FMGS}(k)} - 1}{J_{u_n^{FMGS}(k)} - 1} \\ &= N \log_2 I + \sum_{n=1}^{N} \log_2 \binom{M_n(k) + J_{u_n^{FMGS}(k)} - 1}{J_{u_n^{FMGS}(k)} - 1}. \end{aligned} \tag{6.3}$$

## 6.5 Compression of Scheduling Signaling Information

We consider two different schemes to compress the scheduling signaling information bits: 1) Run-Length Encoding (RLE) [76] and 2) Lempel-Ziv-Welch (LZW) [77]. Note that RLE/LZW compression of the scheduling signaling information bits is performed only if the compression reduces the number of scheduling signaling information bits; otherwise, the scheduling signaling information bits are transmitted uncompressed. An additional bit is added and transmitted along with the scheduling signaling information bits to indicate whether or not compression is performed.

### 6.5.1 Run-length Encoding

RLE is particularly efficient for short data blocks with long consecutively repeating data values and has a low implementation complexity. RLE compresses a data block by representing each run of data (i.e., a data sequence in which the same data value occurs in consecutive elements) by a single data value, called the run value, and the number of consecutively repeating data values, called the run length. The number of bits, $\Upsilon_{\boldsymbol{w}}^{RLE}$, required to represent a data block $\boldsymbol{w}$ of length $L$ with elements from an alphabet of cardinality $R$ using RLE is given by [76]

$$\Upsilon_{\boldsymbol{w}}^{RLE} = Q\lceil \log_2 R \rceil + \sum_{q=1}^{Q} \lceil log_2(L - \sum_{x=0}^{q-1} l_x) \rceil, \tag{6.4}$$

where $Q$ denotes the total number of runs in $\boldsymbol{w}$, $l_x$ denotes the run length of the $x$-th run and $l_0 = 0$. The first term in the right-hand side of (6.4) corresponds to the number of bits required to represent the run values and the second term corresponds to the number of bits required to represent the run lengths.

### 6.5.2 Lempel-Ziv-Welch

LZW is useful for data blocks with repeated patterns and is more efficient for long data blocks as the initial part of the compression algorithm builds a dictionary and has low compression efficiency. The dictionary is initialized to contain all the possible single-character strings of the input data block. LZW then scans through the input data block for successively longer sub-string that are not yet defined in the dictionary. When such a sub-string is found, the index for the sub-string less the last character (i.e., the longest sub-string that is in the dictionary) is sent to the output and the sub-string including the last character is added to the dictionary with the next available code. The last input data character is then used as the new starting point for the next scan. The dictionary building process repeats and successively longer data strings are added to the dictionary and made available for subsequent encoding as single output values. The number of bits, $\Upsilon_{\boldsymbol{w}}^{LZW}$, required to represent a data block $\boldsymbol{w}$ using LZW is obtained using simulation.

## 6.6 Simulation Results

To evaluate the scheduling signaling overhead of the bitQoS-aware resource allocation framework, we adopt the water-filling-based iterative subcarrier-power-bit allocation algorithm proposed in Section 5.2 for the following scheduling policies: 1) *Multi-user Water-filling with Heuristics with No Flow Merging* (WFH-NFM) where each subcarrier can only carry bits from a single application flow of a user, 2) *Multi-user Water-filling with Heuristics with Flow Merging* (WFH-FM) where each subcarrier can carry bits from more than one application flow of a user and 3) *Multi-user Water-filling with Heuristics with Flow Merging - Grouped Sorted* (WFH-FMGS) where each subcarrier can carry bits from more than one application flow of a user and in addition, the bits on each subcarrier are grouped in a FIFO fashion by application flows and sorted in an ascending order by the flow index, $j$, prior to transmission.

To provide a comparative performance assessment of the WFH-NFM/FM/FMGS scheduling policies, we consider the WF and MDU scheduling policies described in Section 4.4.3. The scheduling policies were simulated in Matlab using the system model described in Section 4.2. In the simulation, it is assumed that each user has 1 BE flow and 1 EF flow. The parameter values used in our simulation are listed in Tables 4.1 and 4.2. Simulation results were obtained for mixed-traffic scenarios with $I = \{4, 6, 8\}$.

### 6.6.1 Entropy of Scheduling Signaling Overhead

The entropy of the scheduling signaling overhead based on the entropy model described in Section 6.4 for scheduling policies with NFM, FM and FMGS are shown in Fig. 6.2 as a function of the number, $I$, of users in the system. It can be seen that the entropy increases with $I$. As expected, the entropy for NFM is the lowest since only the subcarrier-to-flow vector, $\boldsymbol{U}^{NFM}(k)$, is transmitted; in FM/FMGS, both the subcarrier-to-user vector, $\boldsymbol{U}^{FM}(k)$/$\boldsymbol{U}^{FMGS}(k)$, and bit-to-flow vectors, $\boldsymbol{V}_n^{FM}(k)$/$\boldsymbol{V}_n^{FMGS}(k)$, $\forall n \in \mathcal{N}$, are transmitted. The entropy for FM is much higher than that for FMGS; this is due to the fact that the bits on a subcarrier $n$ for FM are not grouped by application flows (i.e., every element, $v_{n,z}^{FM}(k)$, of $\boldsymbol{V}_n^{FM}(k)$ can take on values in the set $\mathcal{J}_i$ with equal probability), representing the maximum entropy for the bit-to-flow vectors. By grouping and sorting the bits carried on a subcarrier by their application flows and flow index respectively as in FMGS, the entropy can be greatly reduced.

### 6.6.2 Compressed Scheduling Signaling Overhead

Fig. 6.3 shows the compressed scheduling signaling overhead for the various scheduling policies (WF, MDU, WFH-NFM, WFH-FM and WFH-FMGS) as a function of $I$. These results were obtained from simulation, based on the scheduling signaling overhead compression

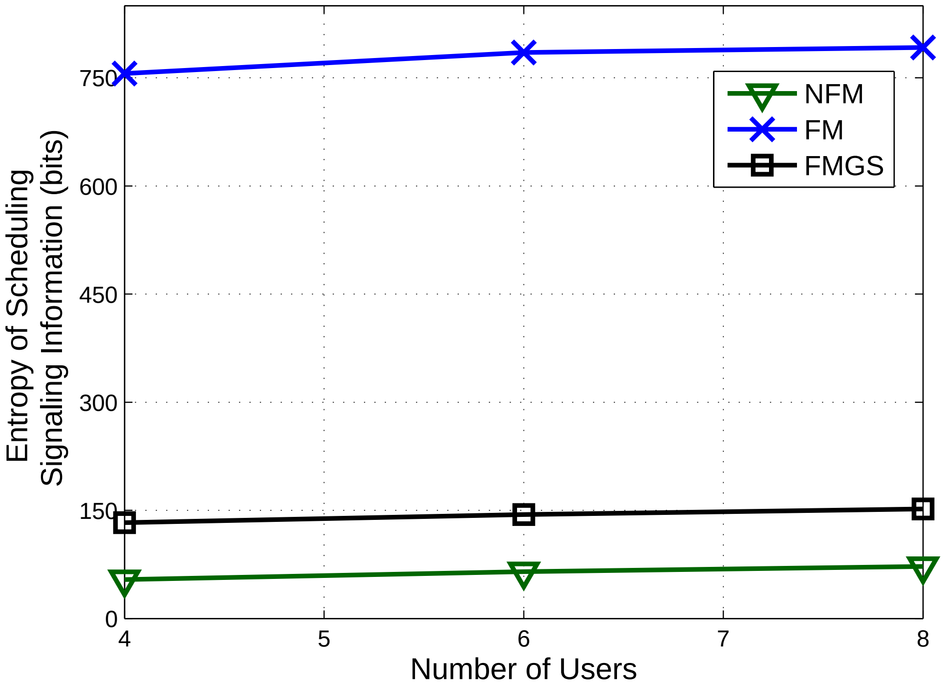


**Figure 6.2:** Entropy of Scheduling Signaling Overhead

schemes (RLE and LZW) described in Section 6.5. It can be observed from Fig. 6.3 that the compressed scheduling signaling overhead increases with the number, $I$, of users in the system for all the scheduling policies and compression schemes. These results are consistent with the entropy analysis results shown in Fig. 6.2. Regardless of the compression scheme used, WFH-FM/FMGS, which allows flow merging, incurs the highest scheduling signaling overhead when compared to WF, MDU and WFH-NFM, which do not allow flow merging. This is due to the fact that in general, as more constraints are imposed upon the scheduling problem, the amount of scheduling signaling overhead required decreases. In this case, for WF, MDU and WFH-NFM, with the no flow merging constraint, it eliminates the need to transmit the bit-to-flow mapping information and thus results in a lower scheduling signaling overhead.

Among the scheduling policies that do not allow flow merging, WF has the highest scheduling signaling overhead regardless of the compression scheme used. This is due to the fact that WF has no QoS provisioning and assigns each subcarrier to the user that has the highest channel gain for that subcarrier. Given that the channel gains are i.i.d., each subcarrier is equally likely to be assigned to any of the users $i \in \mathcal{I}$; this results in a higher entropy for $\boldsymbol{U}^{NFM}(k)$. On the other hand, a lower entropy for $\boldsymbol{U}^{NFM}(k)$ is expected for MDU and WFH-NFM since they consider QoS at the flow-level and bit-level, respectively; as such, each subcarrier is no longer equally likely to be assigned to any of the users $i \in \mathcal{I}$. We also see that MDU has a lower scheduling signaling overhead than WFH-NFM regardless of the compression scheme used. This is because MDU strictly favors EF traffic in a mixed-traffic environment, which effectively reduces the total number of flows scheduled by MDU to $J_{sys}/2$ (assuming an equal number of BE and EF flows).

For both WF and WFH-NFM, LZW provides a lower compressed scheduling signaling overhead than RLE. Since the chance of getting a long run length in $\boldsymbol{U}^{NFM}(k)$ decreases with increasing $I$, the gap between RLE and LZW widens when $I$ increases. For MDU, RLE gives a slightly lower compressed scheduling signaling overhead than LZW. This is due to the fact that MDU mostly schedules only $J_{sys}/2$ flows, which increases the chance of getting a consecutively repeating sequence in $\boldsymbol{U}^{NFM}(k)$, allowing RLE to achieve efficient compression.

With either RLE or LZW, there is little difference in the compressed scheduling signaling overhead for WFH-FM and WFH-FMGS. This is because the number of bits in one application data packet (128 bits) is greater than the number of bits that can be carried by a subcarrier and that all bits in one application data packet have identical bitQoS values. Hence, the bits carried by a subcarrier typically come from the same application data packet of an application flow. As such, the grouping and sorting of bits as described in Section 6.3.3 has little effect on the scheduling signaling overhead incurred by bit-to-flow vectors for WFH-FM. This observation also implies that the bit-to-flow vectors, $\boldsymbol{V}_n^{FM}(k)$, $\forall n \in \mathcal{N}$, which constitutes most of the scheduling signaling overhead of WFH-FM, correspond mostly to short and consecutively repeating data sequences (i.e., consecutive bits assigned to the same flow). This explains why in Fig. 6.3 RLE gives a much lower scheduling signaling overhead than LZW as RLE is more efficient for such data blocks.

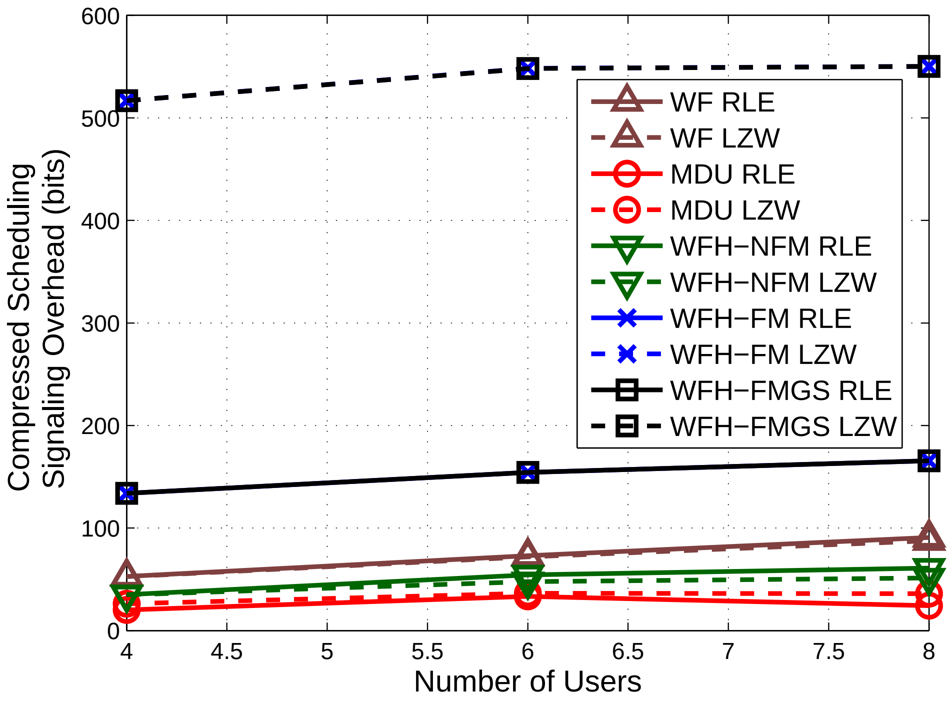


**Figure 6.3:** Compressed Scheduling Signaling Overhead of the Various Scheduling Policies using RLE and LZW

### 6.6.3 Effective Throughput

We note that although Fig. 6.3 shows that WFH-FM has the largest compressed scheduling signaling overhead, it also has the highest average system throughput, compared to WF and MDU, as shown in Fig. 5.4. Hence, to determine the viability of the bitQoS-aware RA framework, we define the effective throughput to account for the scheduling signaling overhead for each scheduling policy as

$$TP_{eff} = \begin{cases} \dfrac{\sum_i \sum_j \sum_k TP_i^j(k) - \sum_k \Upsilon^{\Xi}_{\boldsymbol{U}^{NFM}(k)}}{K} & \text{for WF, MDU and WFH-NFM,} \\ \dfrac{\sum_i \sum_j \sum_k TP_i^j(k) - \sum_k \Upsilon^{\Xi}_{\boldsymbol{U}^{FM}(k)} - \sum_n \sum_k \Upsilon^{\Xi}_{\boldsymbol{V}_n^{FM}(k)}}{K} & \text{for WFH-FM,} \\ \dfrac{\sum_i \sum_j \sum_k TP_i^j(k) - \sum_k \Upsilon^{\Xi}_{\boldsymbol{U}^{FMGS}(k)} - \sum_n \sum_k \Upsilon^{\Xi}_{\boldsymbol{V}_n^{FMGS}(k)}}{K} & \text{for WFH-FMGS,} \end{cases} \tag{6.5}$$

where $\Xi$ is either RLE or LZW depending on the compression scheme used as defined in Section 6.5. The effective throughput gain of scheduling policy $X$ over scheduling policy $Y$ is defined as $G^{X,Y}_{TP_{eff}} = \dfrac{TP^X_{eff} - TP^Y_{eff}}{TP^Y_{eff}}$. The effective throughput gains of WFH-NFM, WFH-FM and WFH-FMGS over the comparative scheduling policies for $I = \{4, 6, 8\}$ are listed in Table 6.1. We see that WFH-NFM, WFH-FM and WFH-FMGS have higher effective throughputs than WF and MDU with RLE compression, i.e., the bitQoS-aware RA framework provides an increased average system throughput even when the scheduling signaling overhead is taken into account. However, when LZW is used, scheduling policies with flow merging (WFH-FM and WFH-FMGS) do not yield a higher effective throughput over WF and MDU due to the inefficiency of LZW to compress the short and consecutively repeating data sequences of the bit-to-flow vectors.

It can also be seen that WFH-FM/FMGS has a lower effective throughput compared to WFH-NFM regardless of the compression scheme used, i.e., allowing flow merging in the bitQoS-aware RA framework yields no system throughput improvement in this case. This result is due to the fact that, with the per-resource-element scheduling granularity of 1 OFDM symbol × 1 subcarrier considered in this chapter, the number of bits in one application layer PDU is typically much greater than the number of bits that can be carried by a subcarrier. As a result, very little flow merging actually takes place and the performance gain from flow merging is minimal.

## 6.7 Conclusion

The viability of the proposed bitQoS-aware RA framework which adaptively matches the QoS requirements of the user application bits to the characteristics of the OFDM subcarriers, with and with no flow merging, was analyzed by taking the associated scheduling signaling overhead into account. A model is formulated to analyze the associated scheduling signaling overhead and the performance gains achievable with the bitQoS-aware RA framework are quantified. The entropy analysis shows that scheduling policies with flow merging incur a significantly higher scheduling signaling overhead compared to scheduling policies that do not allow flow merging. However, the scheduling signaling overhead for scheduling policies with flow merging can be greatly reduced by grouping and sorting the bits carried on the subcarrier by their application flows and flow indices respectively. Simulation results further show that despite the increase in the scheduling signaling overhead for scheduling policies with flow merging, the proposed bitQoS-aware RA framework is still able to provide a higher effective throughput gain compared to scheduling policies that do not take QoS provisions into account such as WF and policies that consider only flow-level QoS requirements such as MDU, when RLE compression of the scheduling signaling information is performed.

**Table 6.1:** Effective Throughput Gains of WFH-NFM, WFH-FM and WFH-FMGS for $I = \{4, 6, 8\}$, $N = 18$, 1 BE and 1 EF Flow for each User

| $I = 4$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $G^{X,Y}_{TP_{eff}} \times 100\%$ | | RLE | | | LZW | | |
| | | Scheduling Policy $X$ | | | | | |
| | | WFH-NFM | WFH-FM | WFH-FMGS | WFH-NFM | WFH-FM | WFH-FMGS |
| Scheduling Policy $Y$ | WF | 75.23 | 21.91 | 21.94 | 75.03 | -185.55 | -185.56 |
| | MDU | 1.73 | -29.23 | -29.21 | 3.77 | -150.72 | -150.73 |
| | WFH-NFM | 0.00 | -30.43 | -30.41 | 0.00 | -148.87 | -148.88 |
| | WFH-FM | 43.74 | 0.00 | 0.02 | 304.61 | 0.00 | 0.02 |
| | WFH-FMGS | 43.70 | -0.02 | 0.00 | 304.56 | -0.02 | 0.00 |

| $I = 6$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $G^{X,Y}_{TP_{eff}} \times 100\%$ | | RLE | | | LZW | | |
| | | Scheduling Policy $X$ | | | | | |
| | | WFH-NFM | WFH-FM | WFH-FMGS | WFH-NFM | WFH-FM | WFH-FMGS |
| Scheduling Policy $Y$ | WF | 148.33 | 96.06 | 96.09 | 150.29 | -114.70 | -114.67 |
| | MDU | 35.63 | 7.08 | 7.10 | 38.82 | -108.15 | -108.13 |
| | WFH-NFM | 0.00 | -21.05 | -21.03 | 0.00 | -105.87 | -105.86 |
| | WFH-FM | 26.66 | 0.00 | 0.02 | 1803.14 | 0.00 | -0.21 |
| | WFH-FMGS | 26.64 | -0.02 | 0.00 | 1806.66 | 0.21 | 0.00 |

| $I = 8$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $G^{X,Y}_{TP_{eff}} \times 100\%$ | | RLE | | | LZW | | |
| | | Scheduling Policy $X$ | | | | | |
| | | WFH-NFM | WFH-FM | WFH-FMGS | WFH-NFM | WFH-FM | WFH-FMGS |
| Scheduling Policy $Y$ | WF | 130.69 | 92.07 | 92.09 | 131.39 | -54.98 | -54.99 |
| | MDU | 86.07 | 54.92 | 54.94 | 96.02 | -61.86 | -61.87 |
| | WFH-NFM | 0.00 | -16.74 | -16.73 | 0.00 | -80.54 | -80.55 |
| | WFH-FM | 20.11 | 0.00 | 0.01 | 413.94 | 0.00 | -0.03 |
| | WFH-FMGS | 20.09 | -0.01 | 0.00 | 414.10 | 0.03 | 0.00 |

# Chapter 7

# Continuous and Discrete Rate Adaptation in BitQoS-aware Resource Allocation Framework [6]

## 7.1 Introduction

Given the promising performance gains and viability of the bitQoS-aware RA framework, even when scheduling signaling overhead is taken into account, in this chapter, we focus on developing more efficient algorithms and use the technique of Lagrange multipliers [78] to find the optimal solution to the bitQoS-aware resource allocation problem which, in addition to considering subcarrier assignments and power allocations, further involves discrete bit assignments for control of bit-level QoS requirements. This differs from the works presented in [63, 79] which only consider subcarrier assignments and power allocations to meet given rate requirements. The Lagrange multiplier technique provides an approach for finding the maxima/minima of a function subject to constraints and yields necessary conditions for optimality in equality constrained problems. To take inequality constraints into account, the technique of Lagrange multipliers is generalized by the KKT conditions [80, 81], which are necessary

[6] The material in this chapter is based on: C. E. Huang and C. Leung, "On the optimality of bitQoS-aware resource allocation in OFDMA systems," submitted.

conditions for a solution in non-linear programming to be optimal.

The technique of Lagrange multipliers has been applied to radio resource allocation problems in [36, 37, 41, 63, 79]. In [63], the authors investigate resource allocation in a multi-user OFDM system with homogeneous traffic and formulate the problem with the objective of minimizing the overall transmit power while satisfying a minimum discrete rate requirement for each user. By relaxing the subcarrier assignments to allow time-sharing, an iterative algorithm based on properties of the Lagrangian formulation is proposed to obtain a suboptimal solution to the original combinatorial resource allocation problem. In [79], the authors extended the time-sharing technique for subcarrier assignment used in [63] to the scheduling of heterogeneous traffic in a multi-user OFDM system. The authors converted the problem into a convex programming problem and proposed an iterative algorithm with polynomial complexity to obtain the optimal subcarrier and power allocation.

Since the bitQoS-aware resource allocation optimization problem is non-deterministic polynomial-time hard (NP-hard), we first look at a reduced-complexity form of the problem, obtained by transforming the joint subcarrier, power and bit allocation problem into a convex optimization problem through a variable transformation and the relaxation of the integer constraints for both the subcarrier and bit assignment variables. Using the KKT conditions, we establish necessary and sufficient optimality conditions and develop an iterative algorithm to obtain the optimal solution. We show that the solution to this relaxed problem follows a bitQoS-based multi-level water-filling principle whereby the water levels of the subcarriers assigned to a user are determined by the bitQoS values of the bits in the user data buffer. These water levels may be different from one user to another, in contrast to a constant water level for all users in the classical throughput maximization water-filling solution. Since the solution to this relaxed problem contains non-discrete bit assignments, it can be interpreted as assignments for systems with continuous rate adaptation and provides an upper bound on the objective value of the original unrelaxed problem. For systems with discrete rate adaptation, we leverage the results of the continuous rate solution and propose an efficient iterative algorithm to compute the solution to the original resource allocation problem with discrete bit assignments.

This chapter is organized as follows: in Section 7.2, the reduced-dimensionality bitQoS-aware RA framework is described. In Section 7.3, we formulate the resource allocation problem as a convex optimization problem and present necessary and sufficient conditions for the optimal solution to the continuous rate adaptation problem. An iterative algorithm to obtain the optimal solution is also presented. In Section 7.4, an algorithm which leverages on the iterative algorithm for the continuous rate adaptation problem is presented for the discrete rate adaptation problem. The simulation framework and results are discussed in Section 7.5 and the main findings are summarized in Section 7.6.

## 7.2 Reduced-dimensionality BitQoS-aware Resource Allocation Framework

In Chapter 4, we formulate the proposed bitQoS-aware RA framework as an optimization problem, OP4.2, with the objective of finding the joint subcarrier, power and bit assignment to maximize the total bitQoS-weighted throughput, subject to the total transmit power constraint, $P_{total}$. However, we note that since the mapping between the application bits and OFDM subcarriers does not affect the objective value of the optimal solution, we can reduce the dimensionality of the optimization problem OP4.2 by substituting $\sum\limits_{n=1}^{N} b_{i,n}^{j,z}$ with $b_i^{j,z}$. The new optimization variable, $b_i^{j,z}$, takes on the value 1 if bit $z$ of user $i$, flow $j$ is transmitted on any subcarrier(s) assigned to user $i$ and 0 otherwise. We can thus rewrite OP4.2 as follows

$$\text{OP7.1:} \quad \max_{\substack{a_{i,n}\in\{0,1\}\\ p_{i,n}\in[0,P_{total}]\\ b_i^{j,z}\in\{0,1\}}} \quad \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{z=1}^{B_i^j} \psi_i^{j,z} b_i^{j,z} \tag{7.1}$$

$$\text{subject to} \quad \sum_{i}\sum_{n} p_{i,n} a_{i,n} = P_{total} \tag{7.2}$$

$$\sum_{j}\sum_{z} b_i^{j,z} \le \sum_{n} \log_2\left(1 + \frac{p_{i,n}|\alpha_{i,n}|^2}{\zeta\sigma_0^2}\right) a_{i,n} \quad \forall i \tag{7.3}$$

$$\sum_{i} a_{i,n} = 1 \qquad \forall n \tag{7.4}$$

$$b_i^{j,z} \le 1 \qquad \forall i,j,z, \tag{7.5}$$

where the variables are as defined in Chapter 4.

## 7.3 Subcarrier, Power and Bit Allocation with Continuous Rate Adaptation

The reduced-dimensionality optimization problem OP7.1 is a MINLP problem whose solution is still computationally complex given the large number of subcarriers and users in a practical system. To make the problem computationally tractable, we adopt the integer constraint relaxation technique used in [63, 79], where it is assumed that the discrete subcarrier assignment variable can take on real values in $[0, 1]$. For our problem formulation with two discrete optimization variables, we attempt to convexify OP7.1 by relaxing the integer constraints for both $a_{i,n}$ and $b_i^{j,z}$, allowing them to take on real values in $[0, 1]$ and defining a new optimization variable $\pi_{i,n} = p_{i,n}a_{i,n}$. The optimization problem OP7.1 becomes

$$\text{OP7.2:} \quad \max_{\substack{a_{i,n}\in[0,1] \\ \pi_{i,n}\in[0,P_{total}a_{i,n}] \\ b_i^{j,z}\in[0,1]}} \quad \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{z=1}^{B_i^j} \psi_i^{j,z} b_i^{j,z} \tag{7.6}$$

$$\text{subject to} \quad P_{total} - \sum_i \sum_n \pi_{i,n} = 0 \tag{7.7}$$

$$\sum_n \log_2\left(1 + \frac{|\alpha_{i,n}|^2}{\zeta\sigma_0^2}\frac{\pi_{i,n}}{a_{i,n}}\right) a_{i,n} - \sum_j \sum_z b_i^{j,z} \geq 0 \quad \forall i \tag{7.8}$$

$$1 - \sum_i a_{i,n} = 0 \qquad \forall n \tag{7.9}$$

$$1 - b_i^{j,z} \geq 0 \qquad \forall i, j, z. \tag{7.10}$$

By evaluating the Hessian matrix of the functions on the LHS of (7.8), it can be shown that the Hessian matrix is negative semi-definite at any point in the convex constraint set $\mathcal{X} = \{a_{i,n} \in [0,1], \pi_{i,n} \in [0, P_{total}a_{i,n}], b_i^{j,z} \in [0,1]\}$, i.e., the functions in the LHS of (7.8) are concave (see proof in Appendix C). In addition, since the functions in (7.6) and the LHS of (7.10) are also concave and the functions in the LHS of (7.7) and (7.9) are affine, OP7.2 is a concave optimization problem [82, 83] in $\mathcal{X}$. Hence, the KKT conditions, which are necessary conditions for a solution to be optimal, are also sufficient for optimality in this case.

Using the technique of Lagrange multipliers [78], the Lagrangian for OP7.2 is

$$\begin{aligned} L \;=\; & \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{z=1}^{B_i^j} \psi_i^{j,z} b_i^{j,z} \\ +\; & \beta(P_{total} - \sum_i \sum_n \pi_{i,n}) \\ +\; & \sum_i \gamma_i \left[\sum_n \log_2\left(1 + \frac{|\alpha_{i,n}|^2}{\zeta\sigma_0^2}\frac{\pi_{i,n}}{a_{i,n}}\right) a_{i,n} - \sum_j \sum_z b_i^{j,z}\right] \\ +\; & \sum_n \mu_n (1 - \sum_i a_{i,n}) \\ +\; & \sum_i \sum_j \sum_z \lambda_i^{j,z}(1 - b_i^{j,z}), \end{aligned} \tag{7.11}$$

where $\beta$, $\gamma_i$, $\mu_n$ and $\lambda_i^{j,z}$ are the Lagrange multipliers for the constraints (7.7), (7.8), (7.9) and (7.10), respectively. The necessary and sufficient conditions for the optimal solution to OP7.2, $\{a^*_{i,n}, \pi^*_{i,n}, b_i^{j,z^*}, \beta^*, \gamma_i^*, \mu_n^*, \lambda_i^{j,z^*}\}$, if it exists, are

Primal feasibility:

$$P_{total} - \sum_i \sum_n \pi^*_{i,n} = 0 \tag{7.12}$$

$$\sum_n \log_2 \left(1 + \frac{|\alpha_{i,n}|^2}{\zeta\sigma_0^2} \frac{\pi^*_{i,n}}{a^*_{i,n}}\right) a^*_{i,n} - \sum_j \sum_z b_i^{j,z^*} \geq 0 \quad \forall i \tag{7.13}$$

$$1 - \sum_i a^*_{i,n} = 0 \qquad \forall n \tag{7.14}$$

$$1 - b_i^{j,z^*} \geq 0 \qquad \forall i, j, z \tag{7.15}$$

Dual feasibility:

$$\beta^* \geq 0, \gamma_i^* \geq 0, \mu_n^* \geq 0, \lambda_i^{j,z^*} \geq 0 \tag{7.16}$$

Stationarity:

$$\frac{\partial L}{\partial a^*_{i,n}} \begin{cases} < 0 & \text{if } a^*_{i,n} = 0 \\ = 0 & \text{if } 0 < a^*_{i,n} < 1 \qquad \forall i, n \\ > 0 & \text{if } a^*_{i,n} = 1 \end{cases} \tag{7.17}$$

$$\frac{\partial L}{\partial \pi^*_{i,n}} \begin{cases} < 0 & \text{if } \pi^*_{i,n} = 0 \\ = 0 & \text{if } 0 < \pi^*_{i,n} < P_{total} a^*_{i,n} \qquad \forall i, n \\ > 0 & \text{if } \pi^*_{i,n} = P_{total} a^*_{i,n} \end{cases} \tag{7.18}$$

$$\frac{\partial L}{\partial b_i^{j,z^*}} \begin{cases} < 0 & \text{if } b_i^{j,z^*} = 0 \\ = 0 & \text{if } 0 < b_i^{j,z^*} < 1 \qquad \forall i, j, z \\ > 0 & \text{if } b_i^{j,z^*} = 1 \end{cases} \tag{7.19}$$

Complementary slackness:

$$\gamma_i^* \left[ \sum_n \log_2 \left(1 + \frac{|\alpha_{i,n}|^2}{\zeta\sigma_0^2} \frac{\pi^*_{i,n}}{a^*_{i,n}}\right) a^*_{i,n} - \sum_j \sum_z b_i^{j,z^*} \right] = 0 \tag{7.20}$$

$$\lambda_i^{j,z^*}(1 - b_i^{j,z^*}) = 0 \tag{7.21}$$

Note that the inequalities in (7.17), (7.18) and (7.19) are obtained by considering the case where the optimal solution occurs at a boundary point of the constraint set $\mathcal{X}$. Since we are attempting to maximize (7.11), the partial derivatives of $L$ in (7.17), (7.18) and (7.19) will be negative if the optimal solution occurs at the lower limit of $a_{i,n}$, $\pi_{i,n}$ and $b_i^{j,z}$, respectively, and positive otherwise. We next derive the analytical expressions for the optimal power allocation, subcarrier and bit assignments.

### 7.3.1 Optimal Power Allocation

In this section, we determine the optimal power allocation for a given subcarrier and bit assignment. The optimal power allocation, $p^*_{i,n}$, is obtained by differentiating the Lagrangian, $L$, in (7.11) with respect to $\pi_{i,n}$ and substituting the result into the KKT condition (7.18). Specifically, for a given subcarrier assignment, $a_{i,n}$, where $a_{i,n} \neq 0$, and bit assignment, $b_i^{j,z}$, we have

$$\left.\frac{\partial L}{\partial \pi_{i,n}}\right|_{\pi_{i,n}=\pi^*_{i,n}} = -\beta + \frac{\gamma_i a_{i,n}}{\ln 2}\frac{\alpha^2_{i,n}}{\zeta\sigma_0^2 a_{i,n} + \alpha^2_{i,n}\pi^*_{i,n}} \begin{cases} < 0 & \text{if } \pi^*_{i,n} = 0 \\ = 0 & \text{if } 0 < \pi^*_{i,n} < P_{total} a_{i,n} \\ > 0 & \text{if } \pi^*_{i,n} = P_{total} a_{i,n} \end{cases} \quad \forall i, n \cdot \tag{7.22}$$

Since $p^*_{i,n} = \pi^*_{i,n}/a_{i,n}$, the three cases in (7.22) can be rewritten as

$$p^*_{i,n} = \frac{\pi^*_{i,n}}{a_{i,n}} = \begin{cases} 0 & \text{if } \dfrac{\gamma_i}{\beta \ln 2} < \dfrac{\zeta\sigma_0^2}{\alpha^2_{i,n}} \\ \dfrac{\gamma_i}{\beta \ln 2} - \dfrac{\zeta\sigma_0^2}{\alpha^2_{i,n}} & \text{if } \dfrac{\zeta\sigma_0^2}{\alpha^2_{i,n}} \le \dfrac{\gamma_i}{\beta \ln 2} \le \dfrac{\zeta\sigma_0^2}{\alpha^2_{i,n}} + P_{total} \\ P_{total} & \text{if } \dfrac{\gamma_i}{\beta \ln 2} > \dfrac{\zeta\sigma_0^2}{\alpha^2_{i,n}} + P_{total} \end{cases}, \qquad \forall i, n, \tag{7.23}$$

where the term, $\beta$, is chosen such that $\{\pi^*_{i,n}\}$ will satisfy the total power constraint (7.7).

Equation (7.23) shows that the optimal power allocation is similar to the classical water-filling algorithm [84] where $\dfrac{\gamma_i}{\beta \ln 2}$ is the equivalence of the water level and $\dfrac{\zeta\sigma_0^2}{\alpha^2_{i,n}}$ the noise floor of user $i$, subcarrier $n$. The main difference is that in (7.23), the water level on each subcarrier $n$ is determined by $\gamma_i$ of the user to which subcarrier $n$ is assigned, whereas in the classical water-filling algorithm, the water level is constant for all subcarriers. In addition, we will show in Section 7.3.3 that $\gamma_i$ is in fact related to the bitQoS values of user $i$ in the proposed bitQoS-aware RA framework. Hence, the optimal power allocation in (7.23) can be interpreted as a generalized bitQoS-based multi-level water-filling solution. Specifically, for the case where the bitQoS values are identical $\forall i, j, z$, the optimization problem OP7.2 reduces to a throughput maximization problem, and (7.23) becomes the classical water-filling solution where $\gamma_i$ is identical for every user $i$.

## 7.3.2 Optimal Subcarrier Assignment

In this section, we determine the optimal subcarrier assignment for a given power allocation and bit assignment. The optimal subcarrier assignment, $a^*_{i,n}$, is obtained by differentiating the Lagrangian, $L$, in (7.11) with respect to $a_{i,n}$ and substituting the result into the KKT condition (7.17). Specifically, for a given power allocation, $p_{i,n}$, and bit assignment, $b_i^{j,z}$, for $a^*_{i,n} \neq 0$, we have

$$\begin{aligned}\left.\frac{\partial L}{\partial a_{i,n}}\right|_{a_{i,n}=a^*_{i,n}} &= \gamma_i \left[\log_2\left(1+\frac{\alpha^2_{i,n}}{\zeta\sigma_0^2}\frac{\pi_{i,n}}{a^*_{i,n}}\right) - \frac{1}{\ln 2}\frac{\frac{\alpha^2_{i,n}}{\zeta\sigma_0^2}\frac{\pi_{i,n}}{a^*_{i,n}}}{1+\frac{\alpha^2_{i,n}}{\zeta\sigma_0^2}\frac{\pi_{i,n}}{a^*_{i,n}}}\right] - \mu_n \\ &\begin{cases} = 0 & \text{if } 0 < a^*_{i,n} < 1 \\ > 0 & \text{if } a^*_{i,n} = 1 \end{cases} \qquad \forall i, n. \end{aligned} \tag{7.24}$$

By substituting the power allocation, $p_{i,n}$, from (7.23) into (7.24), we obtain

$$H_{i,n}(\gamma_i) \begin{cases} = \mu_n & \text{if } 0 < a^*_{i,n} < 1 \\ > \mu_n & \text{if } a^*_{i,n} = 1 \end{cases} \qquad \forall i, n, \tag{7.25}$$

where the function $H_{i,n}(\gamma_i)$ is defined as

$$H_{i,n}(\gamma_i) = \begin{cases} 0 & \text{if } \dfrac{\gamma_i}{\beta \ln 2} < \dfrac{\zeta \sigma_0^2}{\alpha_{i,n}^2} \\ \gamma_i \left[ \log_2 \left( \dfrac{\alpha_{i,n}^2 \gamma_i}{\zeta \sigma_0^2 \beta \ln 2} \right) - \dfrac{1}{\ln 2} \left( 1 - \dfrac{\zeta \sigma_0^2 \beta \ln 2}{\alpha_{i,n}^2 \gamma_i} \right) \right] & \\ & \text{if } \dfrac{\zeta \sigma_0^2}{\alpha_{i,n}^2} \le \dfrac{\gamma_i}{\beta \ln 2} \le \dfrac{\zeta \sigma_0^2}{\alpha_{i,n}^2} + P_{total} \\ \gamma_i \left[ \log_2 \left( 1 + \dfrac{\alpha_{i,n}^2}{\zeta \sigma_0^2} P_{total} \right) - \dfrac{1}{\ln 2} \dfrac{\frac{\alpha_{i,n}^2}{\zeta \sigma_0^2} P_{total}}{1 + \frac{\alpha_{i,n}^2}{\zeta \sigma_0^2} P_{total}} \right] & \\ & \text{if } \dfrac{\gamma_i}{\beta \ln 2} > \dfrac{\zeta \sigma_0^2}{\alpha_{i,n}^2} + P_{total} \end{cases} . \tag{7.26}$$

In the case where $a_{i,n}^* \in (0,1)$ for an arbitrary user $i$ on subcarrier $n$, Constraint (7.9) mandates that there will be time-sharing on subcarrier $n$ (i.e., $a_{i,n}^* \in (0,1)$ for more than one user) since $\sum_i a_{i,n}^* = 1$. From (7.25), we thus have $H_{i,n}(\gamma_i) = \mu_n$ for every user $i \in \{i \in \mathcal{I} | a_{i,n}^* \in (0,1)\}$, which implies that all users sharing subcarrier $n$ must have the same $H_{i,n}(\gamma_i)$. However, since $H_{i,n}(\gamma_i)$ is a function of $\alpha_{i,n}$ and $\{\alpha_{i,n}\}$ are outcomes of independent and real-valued random variables modeling Rayleigh fading, it is highly unlikely that the value of $H_{i,n}(\gamma_i)$ for two or more users will be identical. Hence, the case of $a_{i,n}^* \in (0,1)$ in (7.25) is unlikely. Using (7.9) and (7.25), we see that there will only be one user $i^*$ for which $a_{i^*,n}^* = 1$ and that subcarrier $n$ will be assigned to the user $i^*$ with the largest $H_{i,n}(\gamma_i)$. For all other users $i \neq i^*$, $a_{i,n}^* = 0$. In other words,

$$a_{i,n}^* = \begin{cases} 1 & \text{if } i = i^* \\ 0 & \text{if } i \neq i^* \end{cases} \qquad \forall n, \tag{7.27}$$

where

$$i^* = \arg\max_i H_{i,n}(\gamma_i). \tag{7.28}$$

In the unlikely event that multiple users have identical $H_{i,n}(\gamma_i)$ values for subcarrier $n$, then $i^*$ is chosen from among these users with equal probabilities. Since $H_{i,n}(\gamma_i)$ in (7.26) plays an integral role in determining the optimal subcarrier assignment, we examine its properties with respect to (w.r.t.) $\alpha_{i,n}^2$ and $\gamma_i$:

1) By differentiating $H_{i,n}(\gamma_i)$ w.r.t. $\alpha_{i,n}^2$ for all three cases in (7.26), it can be shown that $H_{i,n}(\gamma_i)$ is a monotonically increasing function of $\alpha_{i,n}^2$ since $\dfrac{\partial H_{i,n}(\gamma_i)}{\partial \alpha_{i,n}^2} \geq 0$. As each subcarrier $n$ is assigned to the user $i^*$ with the largest $H_{i,n}(\gamma_i)$ according to (7.28), the subcarrier will be assigned to the user with the highest channel gain on that subcarrier when the effect of $\gamma_i$ is not considered.

2) Similarly, by differentiating $H_{i,n}(\gamma_i)$ w.r.t. $\gamma_i$ for all three cases in (7.26), it can be shown that $H_{i,n}(\gamma_i)$ is also a monotonically increasing function of $\gamma_i$. As will be shown in Section 7.3.3, $\gamma_i$ is related to the bitQoS values of user $i$. Hence, users with higher bitQoS-valued bits are also more likely to have higher $H_{i,n}(\gamma_i)$ values, resulting in a higher chance of being assigned the subcarrier $n$.

From property 1), we note that the optimal subcarrier assignment in (7.27) agrees with the water-filling solution for throughput maximization in a multi-user OFDM system [30]. Property 2) shows that the optimal subcarrier assignment in (7.27) takes into account the bitQoS values from our proposed bitQoS-aware RA framework. Given that $H_{i,n}(\gamma_i)$ is a monotonically increasing function of both $\alpha_{i,n}^2$ and $\gamma_i$, the optimal subcarrier assignment (7.27) will assign a subcarrier to the user with good channel condition and high bitQoS-valued bits.

### 7.3.3 Optimal Bit Assignment

In this section, we determine the optimal bit assignment for a given power allocation and subcarrier assignment. The optimal bit assignment, $b_i^{j,z*}$, is obtained by differentiating the Lagrangian, $L$, in (7.11) with respect to $b_i^{j,z}$ and substituting the result into the KKT condition (7.19). Specifically, for a given power allocation, $p_{i,n}$, and subcarrier assignment, $a_{i,n}$, we have

$$\left.\frac{\partial L}{\partial b_i^{j,z}}\right|_{b_i^{j,z}=b_i^{j,z*}} = \psi_i^{j,z} - \gamma_i - \lambda_i^{j,z} \begin{cases} < 0 & \text{if } b_i^{j,z*} = 0 \\ = 0 & \text{if } b_i^{j,z*} \in (0,1) \\ > 0 & \text{if } b_i^{j,z*} = 1 \end{cases} \qquad \forall i, j, z. \tag{7.29}$$

From (7.21), we see that if $b_i^{j,z*} = 0$, $\lambda_i^{j,z} = 0$ and by solving (7.29) for $\gamma_i$, we can obtain

$$\gamma_i \begin{cases} > \psi_i^{j,z} & \text{if } b_i^{j,z*} = 0 \\ = \psi_j^{j,z} - \lambda_i^{j,z} & \text{if } b_i^{j,z*} \in (0,1) \\ < \psi_j^{j,z} - \lambda_i^{j,z} & \text{if } b_i^{j,z*} = 1 \end{cases} \qquad \forall i, j, z. \tag{7.30}$$

The term, $\lambda_i^{j,z}$, is a Lagrange multiplier which takes on a value in $[0, \psi_i^{j,z} - \gamma_i]$. We see from (7.30) that $\gamma_i$ (and the associated water-level, $\frac{\gamma_i}{\beta \ln 2}$) for each user $i$ is related to the bitQoS values, $\psi_i^{j,z}$, of the assigned and unassigned bits of that user. Specifically, in the case of unassigned bits ($b_i^{j,z*} = 0$), $\gamma_i > \psi_i^{j,z}$, i.e., $\gamma_i$ should take on a value that is greater than the bitQoS values of all the unassigned bits in the data buffer of user $i$. In the case of assigned bits ($0 < b_i^{j,z*} \leq 1$), $\gamma_i \leq \psi_i^{j,z} - \lambda_i^{j,z}$, i.e., $\gamma_i$ should take on a value that is less than or equal to the bitQoS values of all the assigned bits in the data buffer of user $i$. This relationship in (7.30) appears to be counter-intuitive, as increasing $\gamma_i$ (and the associated allocated power, $p_{i,n}$ (7.23)) does not increase the number of assigned bits, since bits can only be assigned when $\psi_i^{j,z} \geq \gamma_i$. However, if we take into account the KKT conditions collectively, in particular the primal condition (7.14) and the stationarity condition (7.19) (from which (7.30) is derived), we see that these two KKT conditions drive the assignment of bits in opposing directions such that the optimal solution (satisfying all the KKT conditions), if it exists, strives to assign the highest bitQoS-valued bits requiring the least amount of power (i.e., bits with large $\psi_i^{j,z}$, subcarriers with large $\alpha_{i,n}$ and subcarriers requiring small $p_{i,n}$).

### 7.3.4 Optimal Joint Subcarrier, Power and Bit Allocation

In this section, we determine the optimal solution to OP7.2 which must simultaneously satisfy all of the KKT conditions (7.12)-(7.21) for continuous rate adaptation. Using the optimal power, subcarrier and bit allocations in (7.23), (7.27) and (7.30), we propose an iterative KKT-driven algorithm, hereafter referred to as *KKT for Continuous Rate Adaptation* (KKT-CRA) to numerically obtain the optimal solution. The algorithm is outlined in Algorithm 1.

**Algorithm 1** KKT-CRA

```
Initialize $\epsilon$ and $\gamma_i$ to some small number for all $i$
Sort bits in each user data buffer by their bitQoS values in a descending order
repeat
    1) Perform subcarrier assignment
        1.1) Compute $H_{i,n}(\gamma_i)$ for all $i$ and $n$ according to (7.26)
        1.2) Update subcarrier assignment $a_{i,n}$ according to (7.27) with
                $i^*(n) = \arg\max_i H_{i,n}(\gamma_i)$        for all $n$
    2) Perform power allocation $p_{i,n}$ for all $i$ and $n$ according to (7.23)
      where $\beta$ is determined using bisection subject to constraint (7.7)
    3) Perform bit assignment
        3.1) Compute the throughput limits $c_{i,n}$ according to
                $c_{i,n} = \log_2\left(1 + \frac{|\alpha_{i,n}|^2}{\zeta\sigma_0^2} p_{i,n}\right)$        for all $i$ and $n$
        3.2) Assign bits of each user $i$ in a FIFO manner to the subcarriers in $\mathcal{V}_i = \{n \in \mathcal{N} | a_{i,n} = 1\}$ , one subcarrier at a time, until all the bits of user $i$ are assigned or the throughput limits $c_{i,n}, \forall n \in \mathcal{V}_i$ are reached.
    4) Update $\gamma_i$ with iteration step size $\delta$
        4.1) Determine the bitQoS value, $\psi^{HOL}(i)$, of the first unassigned bit in the data buffer of each user $i$
        4.2) Update $\gamma_i$ according to
                $\gamma_i = (1-\delta)\gamma_i + \delta\psi^{HOL}(i)$     for all $i$
until $\gamma_i + \epsilon > \psi^{HOL}(i)$ for all $i$
```

The algorithm begins with initializing $\gamma_i, \forall i \in \mathcal{I}$ to a value less than $\psi^{min}$, where $\psi^{min} = \min \psi_i^{j,z}$ denotes the smallest bitQoS value of all bits in the data buffers of all users. The bits from all application flows of each user $i$ are merged into one queue, i.e., $J_i = 1$, and sorted in decreasing order based on their bitQoS values. At each iteration, using the current values of $\gamma_i, \forall i$, power and subcarriers are allocated according to (7.23) and (7.27) respectively and the corresponding number of bits, $c_{i,n}, \forall i, n$, that user $i$ can transmit on subcarrier $n$ is

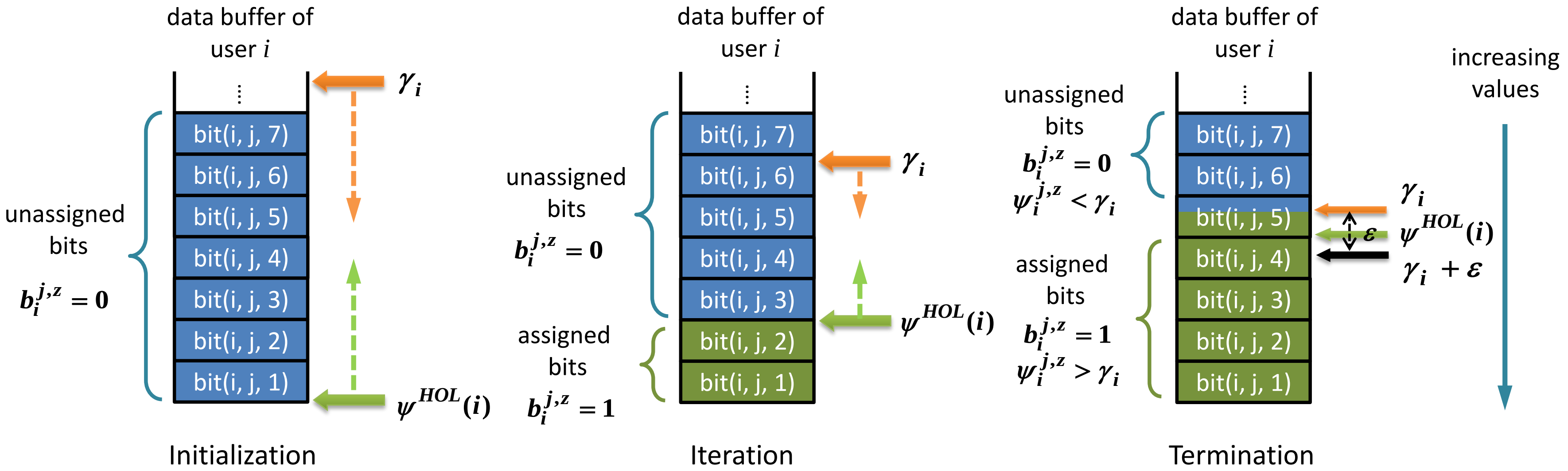


**Figure 7.1:** Relationship between $\gamma_i$ and $\psi_i^{j,z}$

determined according to (4.1). The bits of each user $i$ are assigned in a FIFO manner to the subcarriers in $\mathcal{V}_i$, one subcarrier at a time, where $\mathcal{V}_i = \{n \in \mathcal{N} | a_{i,n} = 1\}$, until either all the bits of user $i$ have been assigned or the throughput limits $c_{i,n}, \forall n \in \mathcal{V}_i$ have been reached. At the end of each iteration, the value of $\gamma_i$ is updated according to $\gamma_i = (1 - \delta)\gamma_i + \delta\psi^{HOL}(i)$ with an iteration step size, $\delta \in (0, 1)$, where $\psi^{HOL}(i)$ denotes the bitQoS value of the first unassigned bit in the data buffer of user $i$ defined as $\psi^{HOL}(i) = \max\limits_{bit(i,j,z) \in \mathcal{S}_{un}(i)} \psi_i^{j,z}$, and $\mathcal{S}_{un}(i) = \{bit(i, j, z) | b_i^{j,z} = 0, j \in \mathcal{J}_i, z \in \{1, \ldots, B_i^j\}\}$ denotes the set of bits of user $i$ that have not yet been assigned based on the current bit assignment. The term $bit(i, j, z))$ refers to bit $z$ of user $i$, flow $j$. The iteration repeats until $\gamma_i + \epsilon > \psi^{HOL}(i), \forall i$, where $\epsilon \in \mathbb{R}_+$ is the termination tolerance. This relationship between $\gamma_i$ and $\psi_i^{j,z}$ is illustrated in Fig. 7.1. Note that in the process of updating $\gamma_i$ and the associated subcarrier power allocation $p_{i,n}$, $c_{i,n}$ may take on a non-integer value, hence yielding a continuous rate adaptation solution. Since the KKT-CRA algorithm works by iteratively increasing the values of $\gamma_i, \forall i$ monotonically towards $\psi^{HOL}(i)$, it will always converge and the solution will satisfy the KKT conditions with appropriate values of $\delta$ and $\epsilon$. In addition, the choice of values for $\delta$ and $\epsilon$ will also determine the number of iterations for KKT-CRA to converge and the closeness of the obtained solution to the optimal solution. They can be varied to achieve a tradeoff between the closeness to optimality and computation time.

## 7.4 Subcarrier, Power and Bit Allocation with Discrete Rate Adaptation

In the previous section, we adopted the integer constraint relaxation technique and proposed an optimal joint subcarrier, power and bit allocation algorithm for the bitQoS-aware RA framework with continuous rate adaptation. This relaxation on $a_{i,n}$ and $b_i^{j,z}$ allows time-sharing of a subcarrier as well as permits subcarriers to transmit a non-integer number of bits. Simply quantizing the solution from the continuous rate case does not necessarily yield the optimal solution to the discrete rate case. As a result, the optimal solution obtained by KKT-CRA may not provide a feasible solution for OP7.1, but gives an upperbound to the maximum achievable bitQoS-weighted throughput for OP7.1. In this section, we leverage on the KKT conditions and KKT-CRA presented in Section 7.3 and propose a bit-loading-based, iterative joint subcarrier, power and bit allocation algorithm, hereafter referred to as *KKT with Discrete Rate Adaptation* (KKT-DRA) to numerically obtain a solution to OP7.1, with discrete $a_{i,n} \in \{0, 1\}$ and $b_i^{j,z} \in \{0, 1\}$ solutions. This algorithm is outlined in Algorithm 2.

As we have shown in Section 7.3.2, time-sharing of a subcarrier, i.e., $a_{i,n} \in (0, 1)$, is unlikely; hence by applying the same argument in KKT-DRA, we obtain $a^*_{i,n} \in \{0, 1\}$. The key difference to obtain discrete $b_i^{j,z}$ solutions in KKT-DRA is that the power allocation and bit assignment are done iteratively, one bit at a time, by bit-loading a bit to the subcarrier that maximizes the gain in bitQoS value while requiring the least amount of power instead of using the water-filling algorithm as in KKT-CRA. Specifically, the algorithm begins by initializing the power allocation to each subcarrier to zero, i.e., $p_{i,n} = 0, \forall i, n$. At each bit-loading iteration, the incremental power, $\Delta p(n)$, required to transmit an additional bit from user $i$ (for which $a_{i,n} = 1$) on subcarrier $n$ and the increase in bitQoS value, $\Delta\psi(n)$, for transmitting that bit are calculated for all subcarriers $n$. The subcarrier that achieves the highest bitQoS value increase per unit power is selected, i.e., $n^* = \arg\max_{n \in \mathcal{N}} \dfrac{\Delta\psi(n)}{\Delta p(n)}$, and the corresponding bit assignment is performed. This iterative bit-loading algorithm repeats until the total transmit power, $P_{total}$, is reached or all the bits in the data buffers of all users have been assigned. The differences in power allocation and bit assignment between KKT-DRA and KKT-CRA are illustrated in Fig. 7.2. However, since the KKT conditions are sufficient for optimality only for convex optimization problems, the discrete rate adaptation solutions obtained using KKT-DRA may not be optimal for OP7.1. Nonetheless, we will later show in Section 7.5.1 that the reduced-complexity KKT-DRA algorithm provides a solution that closely approximates the optimal solution obtained using a commercial MINLP optimization solver package.

**Algorithm 2** KKT-DRA

```
Initialize $\delta$ and $\gamma_i$ to some small number for all $i$
Sort bits in each user data buffer by their bitQoS values in a descending order
repeat
    1) Perform subcarrier assignment
        1.1) Compute $H_{i,n}(\gamma_i)$ for all $i$ and $n$ according to (7.26)
        1.2) Update subcarrier assignment $a_{i,n}$ according to (7.27) with
                $i^*(n) = \arg\max_i H_{i,n}(\gamma_i) \qquad \text{for all } n$
    2) Perform discrete power and bit allocation using bit-loading
        $\hat{p}_{i,n} = 0$ for all $i$ and $n$
        $\hat{b}^{j,z}_{i,n} = 0$ for all $i$, $j$, $z$ and $n$
        $r(n) = 0$ for all $n$; $C(i) = 0$ for all $i$
        $p_{used} = 0$; $p_{inc} = 0$
    while $p_{used} + p_{inc} \leq P_{total}$ do
            $p_{used} = p_{used} + p_{inc}$
        for $n = 1 : N$ do
            $i^*(n) = \arg\max_{i \in \mathcal{I}} \hat{a}_{i,n}$
            $\Delta p(n) = \dfrac{2^{r(n)+1} - 2^{r(n)}}{SNR(i^*(n), n)}$
            $\Delta\psi(n) = \psi_i^{1,C(i^*(n))+1}$
        end for
        $n^* = \arg\max_{n \in \mathcal{N}} \dfrac{\Delta\psi(n)}{\Delta p(n)}$
        $p_{inc} = \Delta p(n^*)$
        if $p_{used} + p_{inc} \leq P_{total}$ then
            $r(n^*) = r(n^*) + 1$
            $C(i^*(n^*)) = C(i^*(n^*)) + 1$
            $\hat{p}_{i^*(n^*),n^*} = \hat{p}_{i^*(n^*),n^*} + p_{inc}$
            $\hat{b}^{j,C(i^*(n^*))}_{i^*(n^*),n^*} = 1$
        end if
    end while
    3) Determine $\beta$ based on

            $$\sum_i \sum_n a_{i,n} \left( \frac{\gamma_i}{\beta \ln 2} - \frac{\zeta \sigma_0^2}{\alpha_{i,n}^2} \right)^+ \leq P_{total} \qquad (7.31)$$

    using bisection
    4) Update $\gamma_i$ with iteration step size $\delta$
        4.1) Determine the bitQoS value, $\psi^{HOL}(i)$, of the first unassigned bit in the data
    buffer of each user $i$
        4.2) Update $\gamma_i$ according to
                $\gamma_i = (1 - \delta)\gamma_i + \delta \psi^{HOL}(i) \qquad \text{for all } i$
until $\gamma_i + \epsilon > \psi^{HOL}(i)$ for all $i$
```

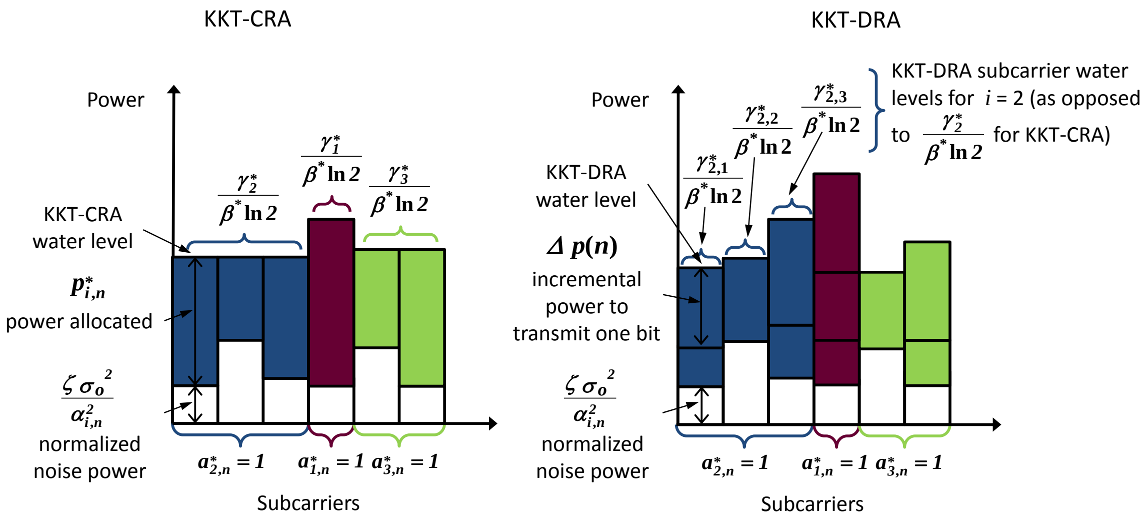

KKT-CRA
Power
KKT-CRA water level
$p^*_{i,n}$ power allocated
$\frac{\zeta \sigma_o^2}{\alpha^2_{i,n}}$ normalized noise power
$\frac{\gamma_2^*}{\beta^* \ln 2}$
$\frac{\gamma_1^*}{\beta^* \ln 2}$
$\frac{\gamma_3^*}{\beta^* \ln 2}$
$a^*_{2,n} = 1$
$a^*_{1,n} = 1$
$a^*_{3,n} = 1$
Subcarriers
KKT-DRA
Power
KKT-DRA water level
$\Delta\, p(n)$ incremental power to transmit one bit
$\frac{\zeta \sigma_o^2}{\alpha^2_{i,n}}$ normalized noise power
$\frac{\gamma_{2,1}^*}{\beta^* \ln 2}$
$\frac{\gamma_{2,2}^*}{\beta^* \ln 2}$
$\frac{\gamma_{2,3}^*}{\beta^* \ln 2}$
KKT-DRA subcarrier water levels for $i = 2$ (as opposed to $\frac{\gamma_2^*}{\beta^* \ln 2}$ for KKT-CRA)
$a^*_{2,n} = 1$
$a^*_{1,n} = 1$
$a^*_{3,n} = 1$
Subcarriers

**Figure 7.2:** Differences in Power Allocation and Bit Assignment between KKT-CRA and KKT-DRA

## 7.5 Simulation Results

In this section, we present simulation results to illustrate the performance of the proposed algorithms for the bitQoS-aware RA framework: KKT-CRA, for the continuous rate adaptation problem OP7.2 and KKT-DRA, for the discrete rate adaptation problem OP7.1. The simulation was performed using Matlab and the system model described in Section 4.2. In the simulation, it is assumed that each user has one flow with full buffer. As the focus of this chapter is to develop efficient and practical algorithms to obtain near-optimal solutions, the generation of realistic bitQoS values as in Chapter 5 is deemed to be unnecessary. Instead, the bitQoS values of the bits in each data buffer are randomly generated from a continuous uniform distribution where the range is varied to represent bitQoS values of different traffic class types [85]. To study the effect of the variability of bitQoS values on the algorithms, we consider two different bitQoS generation schemes: *Same Maximum BitQoS* (SMB), in which the bitQoS values of all application flows, $\psi_i^{j,z} \sim \mathcal{U}(0,50), \forall i,j,z$, are generated in the range between 0 and 50 and *Varying Maximum BitQoS* (VMB), in which the bitQoS values of user $i$, flow $j$, $\psi_i^{j,z} \sim \mathcal{U}(0,\Psi_i^j), \forall i,j,z$, are generated in the range between 0 and $\Psi_i^j$, where $\Psi_i^j \sim \mathcal{U}(0,50)$ is the maximum bitQoS value for user $i$, flow $j$. The SMB scheme represents a traffic mix with mild variation between applications and the VMB scheme represents a traffic mix with diverse servicing priorities between applications.

To provide a comparative performance assessment of the proposed algorithms, we compare the solutions from KKT-CRA and KKT-DRA with the optimal results obtained using a commercial MINLP optimization solver package. We present simulation results to illustrate various aspects of the KKT-CRA and KKT-DRA algorithms in terms of A) optimality and computation time, B) sensitivity of KKT-CRA and KKT-DRA to the iteration step size, $\delta$, and termination tolerance, $\epsilon$, and C) performance comparison of KKT-CRA and KKT-DRA to the classical greedy multi-user water-filling algorithm. The simulation results were obtained by averaging over 10,000 independent trials, each representing a scheduling decision with a different realization of $\psi_i^{j,z}, \forall i, j, z$ and $\alpha_{i,n}, \forall i, n$. Due to the long computation times required, the optimal results in Sections 7.5.1 and 7.5.2 were obtained over 20 independent trials using the commercial MINLP optimization solver package.

### 7.5.1 Optimality and Computation Time

We demonstrate the optimality of the KKT-CRA and KKT-DRA by comparing them with the optimal continuous rate adaptation (OPT-CRA) and optimal discrete rate adaptation (OPT-DRA) solutions obtained by a commercial MINLP optimization solver package, which uses the branch-and-bound approach [86]. Due to the NP-hard nature of OP7.1, where branch-and-bound has a worst case complexity of $O(2^{RIN})$, we simulate the system with $I = \{2, 3, 4, 5, 6\}$, $N = 6$, $\delta = 0.3$, $\epsilon = 10^{-4}$ and $SNR \triangleq \frac{p_{i,n}|\alpha_{i,n}|^2}{\zeta\sigma_0^2} = 15$ dB for the results in this subsection so that the OPT-CRA and OPT-DRA solutions can be obtained within a reasonable amount of time.

We show in Fig. 7.3 the average objective value obtained by OPT-CRA, KKT-CRA, OPT-DRA and KKT-DRA over an identical set of 20 independent trials for $I = \{2, 3, 4, 5, 6\}$ using the SMB bitQoS generation scheme. As stated in Section 7.3, since OP7.2 is a convex optimization problem, satisfaction of the KKT conditions (7.12)-(7.21) is sufficient for the continuous rate adaptation solution to be optimal. This is shown in Fig. 7.3 where the average objective value obtained by KKT-CRA is identical to the optimal solution obtained by OPT-

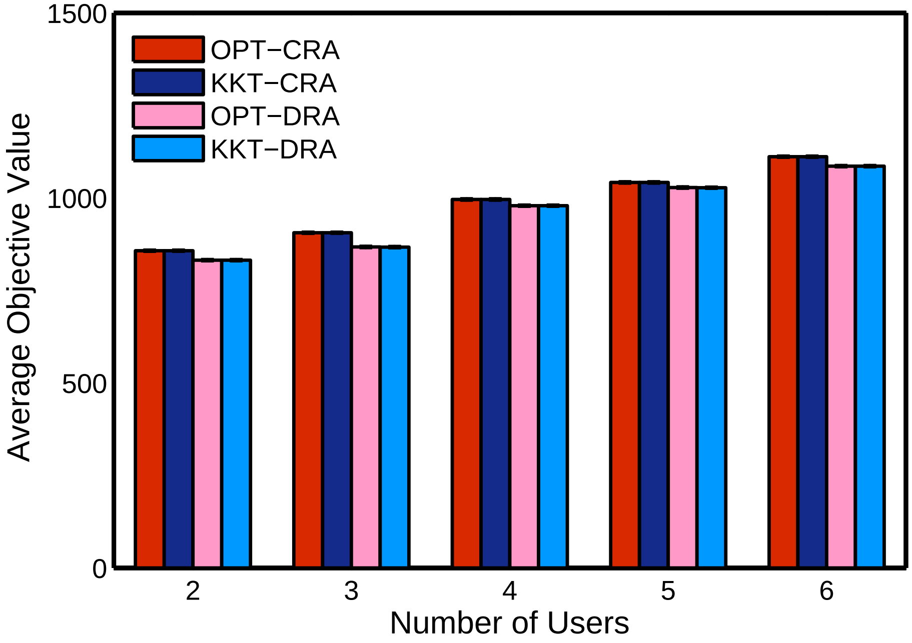


**Figure 7.3:** Average Objective Value as a Function of $I$ ($N = 6$, $\delta = 0.3$, $\epsilon = 10^{-4}$ and $SNR = 15$ dB)

CRA and provides an upperbound to the maximum achievable bitQoS-weighted throughput for OP7.1. In the case of discrete rate adaptation, the average objective value obtained by KKT-DRA is slightly lower than OPT-CRA/KKT-CRA (objective value upperbound). While the optimality of KKT-DRA cannot be proved, Fig. 7.3 shows that the KKT-DRA solution is almost identical to the optimal solution obtained by OPT-DRA. It is important to note that both the KKT-CRA and KKT-DRA solutions are obtained with reduced complexity (shown in Section 7.3.4 and Section 7.4, respectively) as compared to the optimal solutions, OPT-CRA and OPT-DRA. This is reflected in Fig. 7.4, where the computation times of both KKT-CRA and KKT-DRA are orders of magnitude lower than those of OPT-CRA and OPT-DRA.

### 7.5.2 Sensitivity to Iteration Step Size and Termination Tolerance

We next study the sensitivities of the objective values and computation times of KKT-CRA/DRA to the iteration step size, $\delta$, and the termination tolerance, $\epsilon$. We consider a system with $I = 3$, $N = 6$ and $SNR = 15$ dB using both the SMB and VMB bitQoS generation

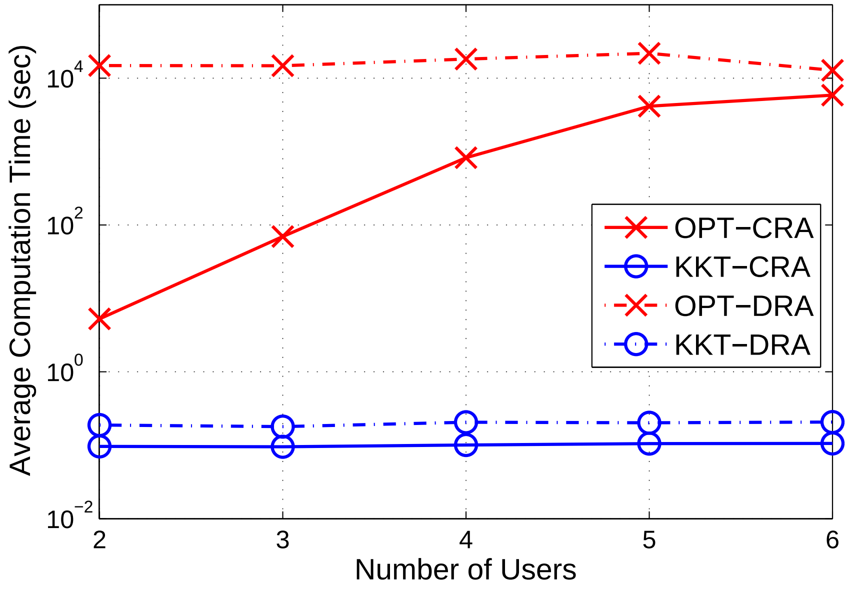


**Figure 7.4:** Average Computation Time as a Function of $I$ ($N = 6$, $\delta = 0.3$, $\epsilon = 10^{-4}$ and $SNR = 15$ dB)

schemes. The average objective value deviations of the KKT-CRA/DRA solutions from the OPT-CRA/DRA solutions and the average computation times of KKT-CRA/DRA as functions of $\delta \in [0, 1]$ and of $\epsilon \in [0, 50]$ are shown in Figs. 7.5 and 7.6, respectively. The objective value deviation is defined as $(\delta_{obj}^{OPT-CRA/DRA} - \delta_{obj}^{KKT-CRA/DRA})/\delta_{obj}^{OPT-CRA/DRA} \times 100\%$, where $\delta_{obj}^{OPT-CRA/DRA}$ and $\delta_{obj}^{KKT-CRA/DRA}$ denote the objective values obtained by OPT-CRA/DRA and KKT-CRA/DRA, respectively.

We see from Fig. 7.5 that the objective value deviations of KKT-CRA/DRA from OPT-CRA/DRA respectively are minimal for $\delta \in [0, 0.95]$ for both the SMB and VMB bitQoS generation schemes. KKT-CRA/DRA have average deviations of $1.3 \times 10^{-3}\%$ and $3.9 \times 10^{-2}\%$ from OPT-CRA/DRA respectively using the SMB bitQoS generation scheme, and average deviations of $0.48\%$ and $1.42\%$ respectively using the VMB bitQoS generation scheme. In general, using a high value of $\delta$ results in a faster termination (fewer iterations) of the KKT-CRA/DRA algorithms. Since the power allocation (7.23), subcarrier assignment (7.26) and bit assignment (7.30) are all dependent on $\gamma_i$, and the bit assignment is also dependent on $\psi_i^{j,z}$ of the assigned and unassigned bits in (7.30), increasing $\gamma_i$ too rapidly with a high value of $\delta$ may cause the algorithm to approach the thresholds in (7.30) too rapidly and lead to subcarrier/bit assignments and power allocations which are quite far from optimal. As the bitQoS values of the bits that are selected for transmission between application flows are more diverse for the VMB bitQoS generation scheme, terminating KKT-CRA/DRA prematurely (possibly in the first iteration) will result in a solution with a high objective value deviation. This is especially noticeable for $\delta$ values close to 1 as shown in Fig. 7.5 since $\gamma_i$ is updated to a value that is close to $\psi^{HOL}(i)$ very quickly. On the other hand, we can see that the effect of a high $\delta$ value is less pronounced with the SMB bitQoS generation scheme since the bitQoS values of the bits selected for transmission among application flows are less diverse. Hence, even if $\gamma_i$ is set to $\psi^{HOL}(i)$ in the first iteration of the algorithms, the associated water-level, $\gamma_i/(\beta \ln 2)$, will be similar among users, resulting in a classical water-filling solution.

In terms of computation time, Fig. 7.5 shows that, regardless of the bitQoS generation scheme, the computation time decreases as $\delta$ increases for both KKT-CRA/DRA. This is due to the fact that the number of iterations, $D$, performed in the main loop of KKT-CRA/DRA is inversely related to $\delta$. It can be seen that KKT-DRA has a higher computation time than KKT-CRA for both SMB and VMB, as within each main loop, the power allocation in KKT-DRA is performed iteratively on a bit-by-bit basis whereas in KKT-CRA, the power allocation is obtained by (7.18) using the bisection algorithm. From Fig. 7.5, it is recommended that $\delta$ be set to a value of around 0.3 in this simulation setup, which is a compromise between a desired small $\delta$ value to achieve a low objective value deviation and the computation time required for both the SMB and VMB bitQoS generation schemes.

It can be seen from Fig. 7.6 that the objective value deviations of KKT-CRA/DRA from OPT-CRA/DRA respectively are larger as the termination tolerance $\epsilon$ varies between 0 and 50 compared to as the iteration step size $\delta$ varies between 0 and 1 for both the SMB and VMB bitQoS generation schemes. KKT-CRA/DRA have average deviations of $0.18\%$ and $0.23\%$ from OPT-CRA/DRA respectively using the SMB bitQoS generation scheme and average deviations of $7.68\%$ and $6.23\%$ respectively using the VMB bitQoS generation scheme. KKT-CRA/DRA is more sensitive to $\epsilon$ since $\epsilon$ determines how close $\gamma_i$ approaches $\psi^{HOL}(i)$ when the algorithm terminates. A high value of $\epsilon$ can cause the algorithm to terminate prematurely

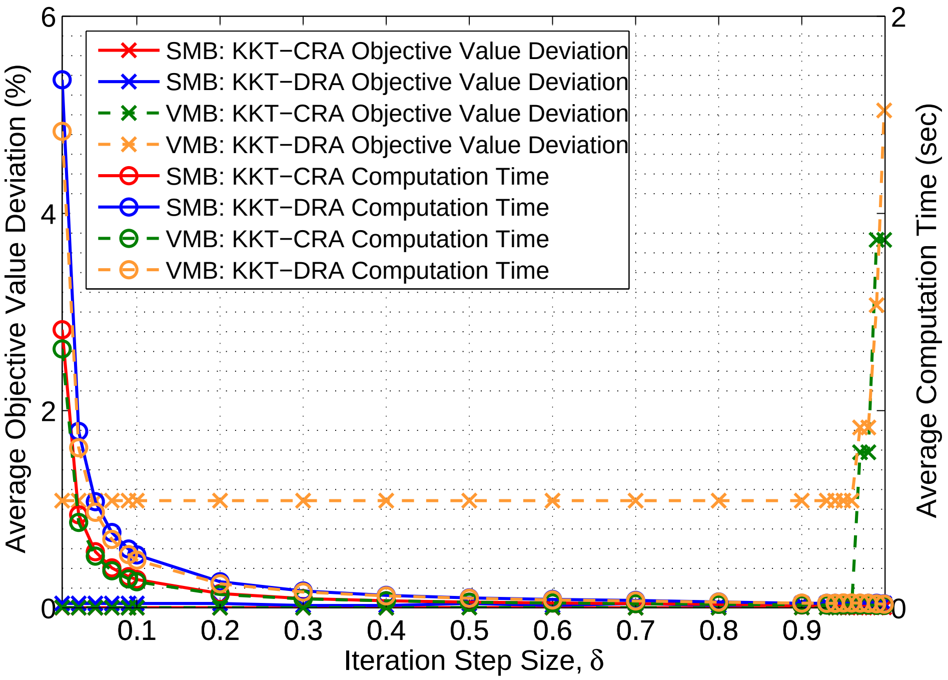


**Figure 7.5:** Sensitivity of KKT-CRA and KKT-DRA to $\delta$ ($I = 3$, $N = 6$, $SNR = 15$ dB and $\epsilon = 1$)

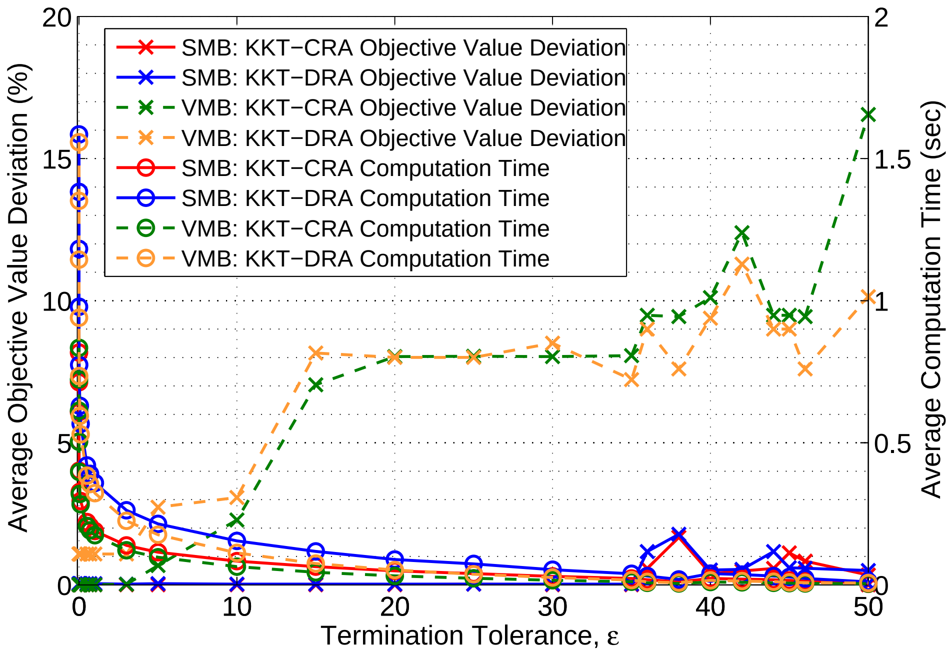


**Figure 7.6:** Sensitivity of KKT-CRA and KKT-DRA to $\epsilon$ ($I = 3$, $N = 6$, $SNR = 15$ dB and $\delta = 0.3$)

with a $\gamma_i$ value that is less than the intended value of $\psi^{HOL}(i)$ in the terminating conditions of KKT-CRA and KKT-DRA ($\gamma_i + \epsilon > \psi^{HOL}(i)$), and lead to allocations and assignments that are quite far from optimal. The objective value deviation becomes pronounced when the value of $\epsilon$ is $\geq \min_i \psi^{HOL}(i)$ as shown in Fig. 7.6 ($\epsilon > 35$ for SMB and $\epsilon > 5$ for VMB). In particular, since the bitQoS values of the bits selected for transmission are more diverse when the VMB bitQoS generation scheme is used, the objective value deviation spans a larger range of $\epsilon \in [5, 50]$ as compared to $\epsilon \in [35, 50]$ when using the SMB bitQoS generation scheme.

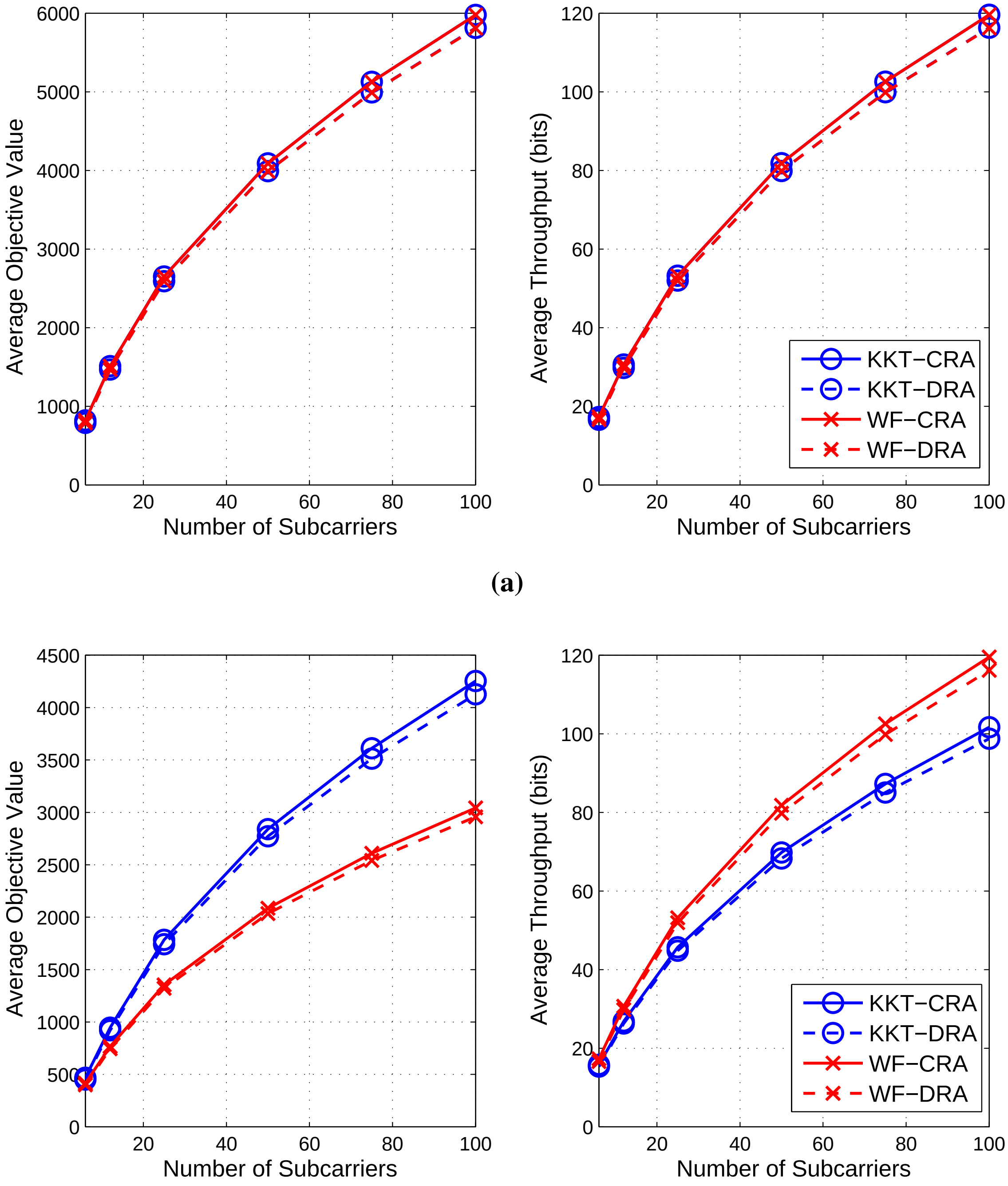
In terms of computation time, we see from Fig. 7.6 that, regardless of bitQoS generation scheme, the computation time of both KKT-CRA and KKT-DRA decreases as $\epsilon$ increases since a large value of $\epsilon$ will terminate the algorithm more quickly. Hence, as with $\delta$, a small value of $\epsilon$ is preferred while at the same time maintaining computational efficiency. It can also be seen that KKT-DRA has a higher computation time than KKT-CRA for both SMB and VMB since the power allocation in KKT-DRA is performed iteratively on a bit-by-bit basis whereas in KKT-CRA, the power allocation is obtained by (7.18) using the bisection algorithm. It is recommended that $\epsilon$ be set to a value that is smaller than the difference between the bitQoS values of any two consecutive different bitQoS-valued bits.

By selecting the values of $\delta$ and $\epsilon$ appropriately, the KKT-CRA/DRA algorithms can be tuned to meet the computation time requirement with a pre-determined objective value deviation bound.

### 7.5.3 Performance Comparison of KKT-CRA and KKT-DRA to the Greedy Multi-user Water-filling Algorithm

In this section, we compare the performance of the bitQoS-aware resource allocation using the proposed KKT-CRA and KKT-DRA algorithms to the classical greedy multi-user water-filling algorithm [30], hereafter referred to as WF-CRA for continuous rate adaptation and WF-DRA for discrete rate adaptation. WF-CRA/DRA assign each subcarrier to the user that has the best channel gain for that subcarrier, and the transmit power is distributed over the subcarriers using the water-filling algorithm [84]. The purpose of comparing KKT-CRA/DRA to WF-CRA/DRA is to study the effect of KKT-CRA/DRA where the subcarrier assignments and power allocations are dependent on both the channel gains and the bitQoS values of the bits in the user data buffers as opposed to just the channel gains in the well-studied WF-CRA/DRA which do not take QoS requirements into account but attempt to maximize the overall throughput of the system. We simulate the system with a number of subcarriers, $N = \{6, 12, 25, 50, 75, 100\}$, to represent the number of resource blocks (RBs) of the different practical LTE transmission bandwidth configurations [87, 88]. The corresponding number of users in the system are $I = \{2, 4, 9, 17, 25, 34\}$ respectively using both the SMB and VMB bitQoS generation schemes.

We see from Fig. 7.7a that the average throughput and average objective values for WF-CRA/DRA and KKT-CRA/DRA are essentially identical for the SMB bitQoS generation scheme. This is due to the fact that the bitQoS values of the bits selected for transmission do not vary widely. As such, the bitQoS values can essentially be neglected and OP4.2 becomes a throughput maximization problem subject to a total power constraint where the optimal channel assignment (7.26) is solely determined by $\alpha_{i,n}$. On the other hand, for the VMB bitQoS generation scheme, we see from Fig. 7.7b, that while WF-CRA/DRA have higher average throughputs than KKT-CRA/DRA, KKT-CRA/DRA have higher average objective values as they attempt to maximize the bitQoS-weighted throughput. This is due to the fact that the bitQoS values of the bits selected for transmission are more diverse and hence, the optimal channel assignment (7.26) for KKT-CRA/DRA depends on both $\alpha_{i,n}$ and $\psi_i^{j,z}$ instead of just $\alpha_{i,n}$ for WF-CRA/DRA. In terms of computation time, we see from Fig. 7.8 that while KKT-CRA/DRA incurs a computation time increase over WF-CRA/DRA for both the SMB and VMB bitQoS generation schemes, the increase is small for KKT-DRA.



**(a)**

**(b)**

**Figure 7.7:** Comparison of Average Objective Value and Average Throughput between KKT-CRA/DRA and WF-CRA/DRA for (a) SMB and (b) VMB

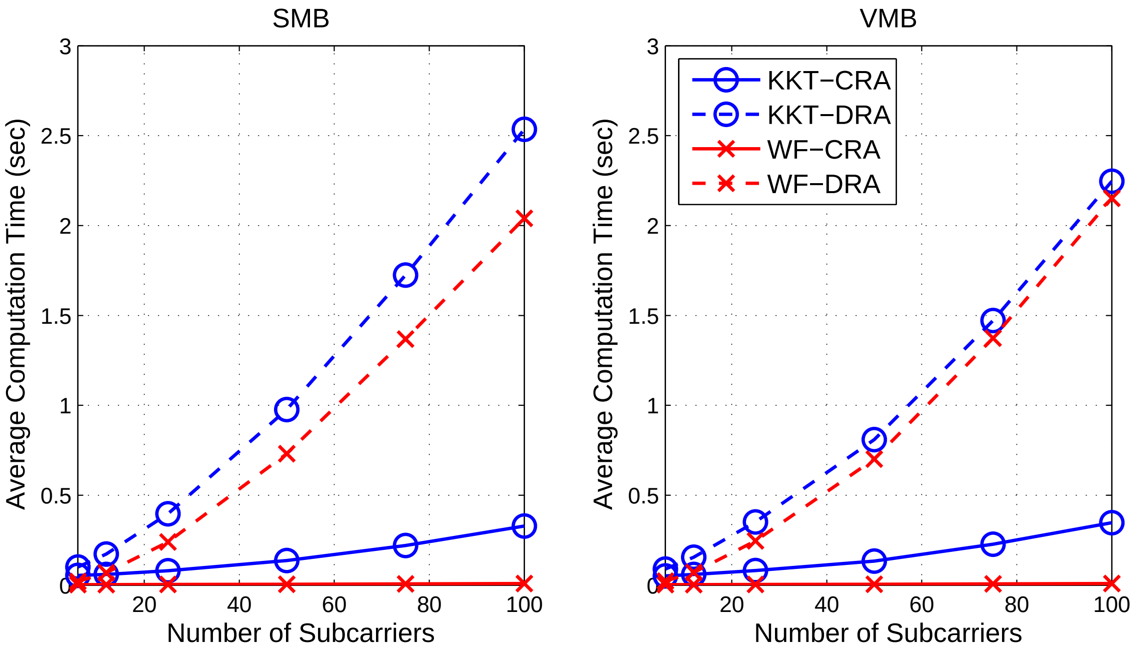


**Figure 7.8:** Comparison of Average Computation Time between KKT-CRA/DRA and WF-CRA/DRA

## 7.6 Conclusion

Optimality conditions and efficient algorithms for the proposed bitQoS-aware RA framework were presented in this chapter for deployment consideration in practical OFDMA systems. The MINLP bitQoS-aware RA problem (NP-hard) was transformed into a convex optimization problem for continuous rate adaptation through a variable transformation and the relaxation of integer constraints for both the subcarrier and bit assignment variables. Using the KKT conditions, we established necessary and sufficient optimality conditions for the continuous rate adaptation problem and showed that the optimal subcarrier assignments and power allocations are dependent on both the channel gains and the bitQoS values of the bits in the user data buffers. In addition, the optimal power allocation can be interpreted as a bitQoS-based multi-level water-filling solution. Efficient KKT-based algorithms, KKT-CRA and KKT-DRA, were developed to obtain the optimal and near-optimal solutions to the joint subcarrier, power and bit allocation problem with continuous and discrete rate adaptation, respectively. The solutions obtained using the lower complexity KKT-CRA and KKT-DRA algorithms were compared with the optimal solutions, OPT-CRA and OPT-DRA, obtained using a commercial MINLP optimization solver package. The simulation results show that KKT-CRA yield identical solutions to OPT-CRA. While the optimality of the KKT-DRA solutions cannot be proved, the KKT-DRA solutions are shown to be almost identical to those of OPT-DRA. By appropriately selecting the parameters, $\delta$ and $\epsilon$, the KKT-CRA and KKT-DRA algorithms can be tuned to tradeoff the computation time against the closeness of the solution to the optimal value.

# Chapter 8

# Computational Complexity and Practicality of BitQoS-aware Resource Allocation Framework [7]

## 8.1 Introduction

In this chapter, we assess the computational complexity of the scheduling policies proposed for the bitQoS-aware RA framework and evaluate their practicality for real-time resource allocation in LTE [27], an OFDM-based air interface.

## 8.2 Computational Complexity

To assess the computational complexity of the considered scheduling policies, we determine the number of operations performed at each scheduling decision time. The number of operations is determined by listing the pseudocode for each scheduling policy and counting the number of addition, assignment, comparison and multiplication operations associated with each line of code. To simplify the analysis, the exponential, ceiling and absolute value func-

[7] The material in this chapter is based on: C. E. Huang and C. Leung, "BitQoS-aware resource allocation for multi-user mixed-traffic OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2067-2082, Jun. 2012.
http://dx.doi.org/10.1109/TVT.2012.2189030

tions are treated as a multiplication operation. For each of the considered scheduling policies, the pseudocode and the associated number of operations performed per line of pseudocode are given in Appendix D and the number of operations performed by each scheduling policy is summarized in Table 8.1.

The term, $R$, is the total number of allocated bits and $B = \max_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} B_i^j$. The term, $L$, is associated with the number of iterations performed by the bisection algorithm, which can be approximated by $L \approx \log_2(\frac{P_{total}}{\varphi})$ [89], where $\varphi$ is the tolerance value on how close the bisection algorithm comes to the solution. The term, $\kappa$, is associated with the number of iterative subcarrier assignment, power allocation and update of the marginal utility performed by the MDU scheduling policy. The term, $D$, denotes the number of iterative updates of the Lagrange multiplier, $\gamma_i$, performed in the main loop of the KKT-CRA and KKT-DRA algorithms and it can be approximated by $D \approx \log_{\frac{1}{1-\delta}} \frac{\psi_{max}}{\epsilon}$, where $\psi^{max} = \max_{i,j,z} \psi_i^{j,z}$ denotes the largest bitQoS value of all bits in the data buffers of all users. The term, $Q$, denotes the number of iterations required by the bisection algorithm in KKT-CRA and KKT-DRA to determine the Lagrange multiplier, $\beta$, and it can be approximated by $Q \approx \log_2 \frac{y - x}{\varphi}$, where $x = \frac{\psi^{min}}{\ln 2(P_{total} + \frac{\zeta\sigma_0^2}{\min_{i,n} |\alpha_{i,n}|^2})}$ and $y = \frac{\psi^{max} \max_{i,n} |\alpha_{i,n}|^2}{\ln 2 \zeta \sigma_0^2}$ are the minimum and maximum values of $\beta$ respectively, which are obtained by solving for $\beta$ within the total transmit power constraint ($0 \leq \frac{\gamma_i}{\beta \ln 2} - \frac{\zeta\sigma_0^2}{|\alpha_{i,n}|^2} \leq P_{total}$). Due to the increased scheduling granularity, we note that the proposed bitQoS-aware scheduling policies, with the exception of KKT-CRA and KKT-DRA, generally have a higher computational complexity compared to both WF and MDU. Nonetheless, all of the proposed bitQoS-aware scheduling policies have polynomial-time complexities as compared to a worst case complexity of $O(2^{RIN})$ for the optimal solution to the discrete rate adaptation problem.

## 8.3 Practicality of BitQoS-aware Scheduling Policies

To evaluate the practicality of the proposed bitQoS-aware scheduling policies for real-time resource allocation, we consider the transmission bandwidth configurations [87, 88] of the

**Table 8.1:** Number of Operations Performed by Each Scheduling Policy - Part I

| Scheduling Policy | Addition Operations | Assignment Operations | Comparison Operations | Multiplication Operations | Big O Notation |
|---|---|---|---|---|---|
| WF | $2NI + 4N$<br>$+L(2N + 2)$<br>$+3$ | $4NI + 4N$<br>$+I + R + 4$<br>$+L(N + 4)$ | $3NI + 3N$<br>$+L(N + 2)$ | $2NI + 8N$<br>$+L(N + 3)$ | $O(NI + R + LN)$ |
| MDU | $2NI + 2I + \kappa(NI$<br>$+6N + 4I + 2)$<br>$+\kappa L(2N + 2)$ | $4NI + 2N + 5I + R$<br>$+\kappa(NI + 4N + 3I$<br>$+4) + \kappa L(N + 4)$ | $2NI + 3I + \kappa(NI$<br>$+4N + 2I + 1)$<br>$+\kappa L(N + 2)$ | $7NI + N + 5I$<br>$+\kappa(4NI + 7N + 5I)$<br>$+\kappa L(2N + 3)$ | $O(\kappa NI + R$<br>$+\kappa LN)$ |
| WFH-FM | $I^3N^2B + 2I^2N^2B+$<br>$INB + 2I^3N^2 + 4I^2N^2$<br>$+8I^2N + 2I^2 + 7IN$<br>$+2IB + J_{sys} + 5I + 4N$<br>$+3 + L(2I^2N^2 + 2I^2N$<br>$+2IN + 2N + 2I + 2)$ | $I^3N^2B + 5I^3N^2 + 2I^3N$<br>$+3I^2N^2 + I^2NR + I^2NB$<br>$+15I^2N + 6I^2 + 11IN+$<br>$IB \log B + 2IB + 16I + IR$<br>$+4N + R + 9 + L(I^2N^2$<br>$+4I^2N + IN + 4I + N + 4)$ | $I^3N^2 + 3I^3N + 5I^2N^2$<br>$+4I^2N + 6I^2 + 4INB$<br>$+IB \log B + 7IN + 4I$<br>$+3N + L(I^2N^2$<br>$+2I^2N + IN + 2I$<br>$+N + 2)$ | $3I^3N^2 + I^3NB$<br>$+2I^2N^2 + 8I^2N$<br>$+7I^2 + 6IB + 5IN$<br>$+5I + 7N + L(I^2N^2$<br>$+3I^2N + IN + 3I$<br>$+N + 3)$ | $O(I^3N^2B + J_{sys}$<br>$+IB \log B + IR$<br>$+2LI^2N^2)$ |
| WFH-NFM | $J_{sys}^3N^2B + 2J_{sys}^2N^2B$<br>$+J_{sys}NB + 2J_{sys}^3N^2$<br>$+4J_{sys}^2N^2 + 8J_{sys}^2N$<br>$+2J_{sys}^2 + 7J_{sys}N + 2IB$<br>$+5J_{sys} + 4N + 3$<br>$+L(2J_{sys}^2N^2 + 2J_{sys}^2N$<br>$+2J_{sys}N + 2N + 2J_{sys}$<br>$+2)$ | $J_{sys}^3N^2B + 5J_{sys}^3N^2$<br>$+2J_{sys}^3N + 3J_{sys}^2N^2$<br>$+J_{sys}^2NR + J_{sys}^2NB$<br>$+15J_{sys}^2N + 6J_{sys}^2 + 12J_{sys}N$<br>$+2IB + 15J_{sys} + J_{sys}R + 4N$<br>$+R + 9 + L(J_{sys}^2N^2$<br>$+4J_{sys}^2N + J_{sys}N + 4J_{sys}$<br>$+N + 4)$ | $J_{sys}^3N^2 + 3J_{sys}^3N$<br>$+5J_{sys}^2N^2 + 4J_{sys}^2N$<br>$+6J_{sys}^2 + 4J_{sys}NB$<br>$+7J_{sys}N + 4J_{sys} + 3N$<br>$+L(J_{sys}^2N^2 + 2J_{sys}^2N$<br>$+J_{sys}N + 2J_{sys} + N$<br>$+2)$ | $3J_{sys}^3N^2 + J_{sys}^3NB$<br>$+2J_{sys}^2N^2 + 8J_{sys}^2N$<br>$+7J_{sys}^2 + 5IB + J_{sys}B$<br>$+5J_{sys}N + 5J_{sys} + 7N$<br>$+L(J_{sys}^2N^2 + 3J_{sys}^2N$<br>$+J_{sys}N + 3J_{sys} + N$<br>$+3)$ | $O(J_{sys}^3N^2B$<br>$+J_{sys}R + IB$<br>$2LJ_{sys}^2N^2)$ |

**Table 8.2:** Number of Operations Performed by Each Scheduling Policy - Part II

| Scheduling Policy | Addition Operations | Assignment Operations | Comparison Operations | Multiplication Operations | Big O Notation |
|---|---|---|---|---|---|
| BABL-FM | $2IB + 2RBN^2 + RBN + 2RN + 4R$ | $2IB + 4NI + RBN^2 + 2RBN + RN + 9R + N + I + 1$ | $IB \log B + 3RBN^2 - 3RBN + 3RN + RI + R$ | $4IB + 3RBN^2 - 3RBN + 5NI + RN$ | $O(IB \log B + RBN^2 + RI + NI)$ |
| BABL-NFM | $2IB + 2RBN^2 + RBN + 2RN + 4R$ | $2IB + 4NJ_{sys} + RBN^2 + 2RBN + RN + 9R + N + I + 1$ | $3RBN^2 - 3RBN + 3RN + RJ_{sys} + R$ | $4IB + 3RBN^2 - 3RBN + 5NJ_{sys} + RN$ | $O(IB + RBN^2 + RJ_{sys} + NJ_{sys})$ |
| KKT-CRA | $2IB + J_{sys} + D[5IN + 6I + 4N + Q(N+2)]$ | $2IB + NI + 3I + 1 + D[IB + 5NI + 4I + 3N + 5 + Q(N+4)]$ | $IB \log B + D[5NI + 5I + 2N + Q(N+2)]$ | $5IB + 5NI + D[18NI + 5I + 8N + Q(4N+3)]$ | $O(IB \log B + J_{sys} + D(NI + IB + QN))$ |
| KKT-DRA | $2IB + J_{sys} + D[2NI + 6I + 4N + 2 + R(3N+5) + Q(2N+2)]$ | $2IB + NI + 3I + 1 + D[IBN + 3NI + 4I + 4N + 7 + R(2N+7) + Q(N+4)]$ | $IB \log B + D[4NI + 5I + 2N + R(2N+1) + Q(N+2)]$ | $5IB + 5NI + D[11NI + 2I + 8N + R(4N) + Q(4N+3)]$ | $O(IB \log B + J_{sys} + D(IBN + RN + QN))$ |

**Table 8.3:** Computation Time Calculation Parameter Values for LTE

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Subcarrier bandwidth | 15 kHz | $B$ | 44 bits |
| Number of subcarriers in a SB | 12 | $\varphi$ | 0.001 |
| SB bandwidth | 180 kHz | $L$ | 9.97 |
| SB time duration | 1 ms | $\kappa$ | $R$ |
| OFDM symbols per SB with extended cyclic prefix | 12 | $\delta$ | 0.3 |
| Symbol duration | 1/12 ms | $\epsilon$ | 3.0 |
| Modulation | 16 QAM | $\psi^{max}$ | 50.0 |
| $J_{sys}$ | $2I$ | | |

LTE air interface. In the forward link of LTE systems, subcarriers are grouped into resource blocks (RBs) of 12 adjacent subcarriers, each with a subcarrier bandwidth of 15 kHz. Each RB has a time slot duration of 0.5 ms, which corresponds to 6 or 7 OFDM symbols depending on whether an extended or normal cyclic prefix is used. The smallest resource unit which a scheduler can assign to a user in LTE is a Scheduling Block (SB) [27, 90], which consists of two consecutive RBs, resulting in a subframe time duration of 1 ms with a frequency block of 180 kHz. The LTE specifications define transmission bandwidth configurations ranging from 1.4 MHz to 20 MHz and the number of SBs and subcarriers depends on the overall transmission bandwidth of the system.

In our calculation of the computation times of the considered scheduling policies for LTE, a system with a loading of $\rho = 0.95$ is used, where $\rho = I(\lambda_{BE} + \lambda_{EF})/\mu$. Each user is assumed to have 1 BE and 1 EF flow and the term, $\mu$, denotes the service rate of the system. The number of instructions performed by each of the considered scheduling policies at each scheduling decision time is determined by summing the number of addition, assignment, comparison and multiplication operations listed in Table 8.1. We assume that the basic operations used (addition, assignment, comparison and multiplication) are effectively executed as a single instruction in a modern pipelined microprocessor architecture. The times

**Table 8.4:** LTE Transmission Bandwidth Configurations

| **LTE Transmission Bandwidth Configuration Parameter Values** | | | | | | |
|---|---|---|---|---|---|---|
| LTE Transmission Bandwidth Configuration | A | B | C | D | E | F |
| Bandwidth (MHz) | 1.4 | 3.0 | 5.0 | 10.0 | 15.0 | 20.0 |
| Number of subcarriers | 72 | 144 | 300 | 600 | 900 | 1200 |
| $N$ (Number of SBs) | 6 | 12 | 25 | 50 | 75 | 100 |
| $R$ (Number of bits per scheduling decision time) | 3456 | 6912 | 14400 | 28800 | 43200 | 57600 |
| $\mu$ (System service rate) (Mbps) | 3.456 | 6.912 | 14.4 | 28.8 | 43.2 | 57.6 |
| $I$ (Number of users for $\rho = 0.95$ ) | 75 | 150 | 313 | 626 | 939 | 1252 |

**Table 8.5:** Computation Times of the Considered Scheduling Policies

| **Number of Instructions Per Scheduling Decision Time** | | | | | | |
|---|---|---|---|---|---|---|
| LTE Transmission Bandwidth Configuration | A | B | C | D | E | F |
| WF | $9.02 \times 10^{3}$ | $2.78 \times 10^{4}$ | $1.03 \times 10^{5}$ | $3.77 \times 10^{5}$ | $8.24 \times 10^{5}$ | $1.44 \times 10^{6}$ |
| MDU | $1.66 \times 10^{7}$ | $1.09 \times 10^{8}$ | $8.83 \times 10^{8}$ | $6.68 \times 10^{9}$ | $2.21 \times 10^{10}$ | $5.20 \times 10^{10}$ |
| WFH-FM | $1.76 \times 10^{9}$ | $5.21 \times 10^{10}$ | $1.96 \times 10^{12}$ | $6.16 \times 10^{13}$ | $4.65 \times 10^{14}$ | $1.95 \times 10^{15}$ |
| WFH-NFM | $1.36 \times 10^{10}$ | $4.10 \times 10^{11}$ | $1.56 \times 10^{13}$ | $4.91 \times 10^{14}$ | $3.71 \times 10^{15}$ | $1.56 \times 10^{16}$ |
| BABL-FM | $4.67 \times 10^{7}$ | $3.82 \times 10^{8}$ | $3.50 \times 10^{9}$ | $2.82 \times 10^{10}$ | $9.52 \times 10^{10}$ | $2.26 \times 10^{11}$ |
| BABL-NFM | $4.70 \times 10^{7}$ | $3.83 \times 10^{8}$ | $3.50 \times 10^{9}$ | $2.82 \times 10^{10}$ | $9.53 \times 10^{10}$ | $2.26 \times 10^{11}$ |
| KKT-CRA | $2.31 \times 10^{5}$ | $6.97 \times 10^{5}$ | $2.53 \times 10^{6}$ | $9.23 \times 10^{6}$ | $2.01 \times 10^{7}$ | $3.51 \times 10^{7}$ |
| KKT-DRA | $2.47 \times 10^{6}$ | $8.99 \times 10^{6}$ | $3.70 \times 10^{7}$ | $1.44 \times 10^{8}$ | $3.22 \times 10^{8}$ | $5.71 \times 10^{8}$ |
| **Computation Times (ms) (Intel 990x)** | | | | | | |
| WF | $5.67 \times 10^{-5}$ | $1.75 \times 10^{-4}$ | $6.46 \times 10^{-4}$ | $2.37 \times 10^{-3}$ | $5.18 \times 10^{-3}$ | $9.08 \times 10^{-3}$ |
| MDU | $1.05 \times 10^{-1}$ | $6.88 \times 10^{-1}$ | $5.55 \times 10^{0}$ | $4.20 \times 10^{1}$ | $1.39 \times 10^{2}$ | $3.27 \times 10^{2}$ |
| WFH-FM | $1.11 \times 10^{1}$ | $3.28 \times 10^{2}$ | $1.23 \times 10^{4}$ | $3.87 \times 10^{5}$ | $2.92 \times 10^{6}$ | $1.23 \times 10^{7}$ |
| WFH-NFM | $8.57 \times 10^{1}$ | $2.58 \times 10^{3}$ | $9.79 \times 10^{4}$ | $3.09 \times 10^{6}$ | $2.33 \times 10^{7}$ | $9.80 \times 10^{7}$ |
| BABL-FM | $2.94 \times 10^{-1}$ | $2.41 \times 10^{0}$ | $2.20 \times 10^{1}$ | $1.77 \times 10^{2}$ | $5.99 \times 10^{2}$ | $1.42 \times 10^{3}$ |
| BABL-NFM | $2.95 \times 10^{-1}$ | $2.41 \times 10^{0}$ | $2.20 \times 10^{1}$ | $1.77 \times 10^{2}$ | $5.99 \times 10^{2}$ | $1.42 \times 10^{3}$ |
| KKT-CRA | $1.45 \times 10^{-3}$ | $4.39 \times 10^{-3}$ | $1.59 \times 10^{-2}$ | $5.80 \times 10^{-2}$ | $1.26 \times 10^{-1}$ | $2.21 \times 10^{-1}$ |
| KKT-DRA | $1.55 \times 10^{-2}$ | $5.56 \times 10^{-2}$ | $2.33 \times 10^{-1}$ | $9.90 \times 10^{-1}$ | $2.03 \times 10^{0}$ | $3.59 \times 10^{0}$ |

that the considered scheduling policies require to make a scheduling decision is determined assuming the use of an Intel Core i7 Extreme Edition 990x microprocessor [91], which is rated to perform 159,000 Million Instructions Per Second (MIPS) at 3.46 GHz. The parameter values used for the calculation of the computation times are presented in Table 8.3, where the term, $B$, is approximated by $T_{SB}(\lambda_{BE}+\lambda_{EF})$, and $T_{SB}$ is the time duration of a SB. The LTE transmission bandwidth configuration parameter values, the number of instructions executed by each scheduling policy and the computation times required for making each scheduling decision are presented in Tables 8.4 and 8.5. Note that for LTE, the term, $N$, can be viewed as the number of SBs rather than the number of subcarriers in the system.

As can be seen from the computation time results in Table 8.5, the Intel 990x is currently already capable of executing: KKT-CRA within a 1 ms SB for all LTE transmission bandwidth configurations; KKT-DRA up to LTE transmission bandwidth configuration D; BABL-FM/NFM for LTE transmission bandwidth configuration A; and WFH-FM/NFM for LTE transmission bandwidth configuration A when the system loading is reduced to $\rho = 0.45$. KKT-DRA incurs a higher computational complexity compared to KKT-CRA as at each iteration of the main loop where $\gamma_i$ is updated, KKT-DRA has to perform $RN$ bit-loading assignments instead of just $N$ water-filling operations as in the case of KKT-CRA. While the proposed bitQoS-aware scheduling policies are, in general, more computationally complex than the other considered scheduling policies (especially WF), the performance gains of the bitQoS-aware scheduling policies in user throughput and user packet drop probability (shown in Chapter 5) as well as in effective throughput gains (shown in Table 6.1) over scheduling policies, such as WF, that do not take QoS provisions into account and scheduling policies, such as MDU, that only consider flow-level QoS requirements demonstrate that the increased scheduling granularity and flexibility of the proposed bitQoS RA framework may be attractive in many situations. Given the computation times of KKT-CRA and KKT-DRA shown in Table 8.5, which can at least support up to LTE transmission bandwidth configuration D, we expect that with the additional technological advancements outlined below, the bitQoS-aware RA framework is practical and can be adopted in even higher LTE transmission bandwidth configurations.

1) Faster/dedicated processors: Forward-looking statements indicate that the upcoming Intel Core i7 Extreme Edition 3960x microprocessor utilizing the Sandy Bridge architecture will yield a 47% performance increase [92] over the Intel 990x and microprocessors utilizing the Ivy Bridge architecture (to be released in 2012) [93] will yield another 20% performance increase over the Intel 3960x. In addition, for timing critical components such as resource allocation at the BS, we would expect commercial grade microprocessors/dedicated DSPs to be used in commercial deployments.

2) Algorithm development: We expect more efficient algorithms to be developed to take advantage of the proposed bitQoS-aware RA framework for deployment. The efficiencies can come from multiple areas such as mathematical techniques to reduce algorithm complexity and/or tradeoffs made between performance and complexity.

3) Multiple parallel baseband processing modules: We note that it is not uncommon for BSs to use multiple parallel baseband processing modules for system scalability/flexibility as well as to handle large bandwidth systems, e.g., $4 \times 5$ MHz for a 20 MHz system.

## 8.4 Conclusion

In this chapter, we assessed the computational complexity of the proposed bitQoS-aware scheduling policies (WFH-FM/NFM, BABL-FM/NFM and KKT-CRA/DRA) by determining the number of operations performed at each scheduling decision time and evaluated the practicality of the proposed scheduling policies for real-time resource allocation by determining the computation time required to make a scheduling decision for the LTE air interface. We showed that the Intel 990x microprocessor is currently already capable of executing KKT-CRA within a 1 ms SB for all LTE transmission bandwidth configurations, KKT-DRA up to LTE transmission bandwidth configuration D, BABL-FM/NFM for LTE transmission bandwidth configuration A, and WFH-FM/NFM for LTE transmission bandwidth configuration A when the system loading is reduced to $\rho = 0.45$. In addition, we believe that with the rapid improvement in microprocessor performances, algorithm development and parallel processing modules, among technological advancements, the bitQoS-aware RA framework is practical and can be adopted in even higher LTE transmission bandwidth configurations.

# Chapter 9

# Conclusion

This chapter summarizes the main contributions of this thesis and provides suggestions for future research work.

## 9.1 Contributions

In this thesis, we have investigated RA and proposed scheduling policies for single-carrier and multi-carrier communication systems that service multiple users with different applications and different QoS requirements. The main contributions of this thesis are summarized as follows:

In Chapter 2, the performance gains of scheduling policies that exploit MFM in multi-application single-carrier CDMA communication systems were quantified in terms of user throughput, user latency and user packet drop probability. The gains of MFM results from wastage reduction in the physical layer encoder packet and multiplexing of packets with different latency tolerances in a scheduling period. Additional performance gains were achieved by the ACLS-FM scheduling policy through the integration of MFM with a cross layer design (physical, MAC and application layers) and the utilization of a packet urgency function to allow a packet from a delay-sensitive application flow to have its service priority raised when its waiting time exceeds a predetermined threshold.

In Chapter 3, an ACLS-FUM scheduling policy that integrates both MFM and PDM while jointly considering physical-layer time-varying channel conditions as well as application-layer QoS requirements in a mixed traffic environment was proposed and evaluated. Simulation results showed that ACLS-FUM is able to achieve substantial performance gains in user throughput, user latency, user jitter and user packet drop probability when compared to other well known scheduling policies. This improvement is achieved due to the ability to PDM the physical layer encoder packet using MUP transmission which not only improves the resource utilization (packing efficiency) by allowing delay-tolerant applications to fill up the unused physical layer encoder packet with higher priority, low-rate, latency-sensitive applications, but also provides an increase in the number of available time slots to support low-rate latency-sensitive applications, leading to increased system throughput and spectral efficiency.

In Chapters 4 and 5, a bitQoS-aware RA framework that exploits multi-application and multi-bit diversities by adaptively matching the QoS requirements of user application bits to the characteristics of the OFDM subcarriers was proposed for a multi-user OFDM system in a mixed-traffic environment. The simulation results, obtained using the proposed water-filling-based WFH scheduling policy and bit-loading-based BABL scheduling policy, showed that with the finesse bit-level control provided by the proposed bitQoS-aware RA framework, it is possible to achieve both an increase in throughput and a reduction in packet drop probability at the cost of a longer (albeit within the scheduling delay threshold) scheduling delay. This flexibility comes from the realization that in OFDM, data is loaded onto subcarriers in units of bits and the latency QoS is satisfied as long as the bit waiting time does not exceed the scheduling delay threshold. By applying the bitQoS function at the bit-level as proposed, system providers can trade off the bit waiting time for a reduction in the number of dropped packets by prioritizing which bit to transmit based on its closeness to the scheduling delay threshold. This finer resolution of control provides an additional flexibility to push back the scheduling of bits that are not as close to the scheduling delay threshold (i.e., by increasing the bit waiting time) so as to allow the servicing of more “urgent” bits when necessary. As long as this push-back does not cause the bit waiting time to exceed the scheduling delay threshold, bits will be serviced within their scheduling delay thresholds, resulting in a simultaneous increase in user throughput and a reduction in the number of user bits dropped. Both WFH and BABL were also able to achieve the highest average system throughput across all considered system loads when compared to scheduling policies that do not take QoS provisions into account such as WF and policies that consider only flow-level QoS such as MDU. In addition, it was found that in a multi-application system, the performance gains by allowing bits from different application flows of a user to be merged into a single subcarrier for transmission are small.

In Chapter 6, the viability of the proposed bitQoS-aware RA framework, with and with no flow merging, was analyzed by taking the associated scheduling signaling overhead into account. A model is formulated to analyze the associated scheduling signaling overhead and the performance gains achievable with the bitQoS-aware RA framework are quantified. The entropy analysis shows that scheduling policies with flow merging incur a significantly higher scheduling signaling overhead compared to scheduling policies that do not allow flow merging. However, the scheduling signaling overhead for scheduling policies with flow merging can be greatly reduced by grouping and sorting the bits carried on the subcarrier by their application flows and flow indices, respectively. Simulation results further show that despite the increase in the scheduling signaling overhead for scheduling policies with flow merging, the proposed bitQoS-aware RA framework is able to provide a higher effective throughput gain compared to scheduling policies that do not take QoS provisions into account such as WF and policies that consider only flow-level QoS requirements such as MDU, when RLE compression of the scheduling signaling information is performed.

In Chapter 7, optimality conditions and efficient algorithms for the proposed bitQoS-aware RA framework were presented for deployment consideration in practical OFDMA systems. The MINLP bitQoS-aware RA problem (NP-hard) was transformed into a convex optimization problem for continuous rate adaptation through a variable transformation and the relaxation of integer constraints for both the subcarrier and bit assignment variables. Using the KKT conditions, we established necessary and sufficient optimality conditions for the continuous rate adaptation problem and showed that the optimal subcarrier assignments and power allocations are dependent on both the channel gains and the bitQoS values of the bits in the user data buffers. In addition, the optimal power allocation can be interpreted as a bitQoS-based multi-level water-filling solution. Efficient KKT-based algorithms, KKT-CRA and KKT-DRA, were developed to obtain the optimal and near-optimal solutions to the joint subcarrier, power and bit allocation problem with continuous and discrete rate adaptation, respectively. The solutions obtained using the lower complexity KKT-CRA and KKT-DRA algorithms were compared with the optimal solutions, OPT-CRA and OPT-DRA, obtained using a commercial MINLP optimization solver package. The simulation results show that KKT-CRA yield identical solutions to OPT-CRA. While the optimality of the KKT-DRA solutions cannot be proved, the KKT-DRA solutions are shown to be almost identical to those of OPT-DRA. By appropriately selecting the parameters, $\delta$ and $\epsilon$, the KKT-CRA and KKT-DRA algorithms can be tuned to tradeoff the computation time against the closeness of the solution to the optimal value.

In Chapter 8, we assessed the computational complexity of the proposed bitQoS-aware scheduling policies (WFH-FM/NFM, BABL-FM/NFM and KKT-CRA/DRA) by determining the number of operations performed at each scheduling decision time and evaluated the practicality of the proposed scheduling policies for real-time resource allocation by determining the computation time required to make a scheduling decision for the LTE air interface. We showed that the Intel 990x microprocessor is currently already capable of executing KKT-CRA within a 1 ms SB for all LTE transmission bandwidth configurations, KKT-DRA up to LTE transmission bandwidth configuration D, BABL-FM/NFM for LTE transmission bandwidth configuration A, and WFH-FM/NFM for LTE transmission bandwidth configuration A when the system loading is reduced to $\rho = 0.45$. In addition, we believe that with the rapid improvement in microprocessor performances, algorithm development and parallel processing modules, among technological advancements, the bitQoS-aware RA framework is practical and can be adopted in even higher LTE transmission bandwidth configurations.

## 9.2 Future Work

In this thesis, we increase the flexibility and granularity of the RA algorithms by adopting an adaptive cross layer approach to exploit multi-application diversity in single-carrier communication systems and additionally, multi-bit diversity in multi-carrier communication systems. While the results show that the proposed algorithms can achieve a higher system throughput with substantial performance gains in the considered QoS metrics, the following summarizes some possible topics for future study.

### 9.2.1 Analysis and Determination of Scheduling Block Size

The bitQoS-aware RA framework in Chapter 4 is formulated as optimization problems with no flow merging and with flow merging. However, it is shown in the results in Chapters 5 and 6 that, with or without consideration of the scheduling signaling overhead, in a multi-application system, the performance gains achievable by allowing different application flows of a user to be merged into a single subcarrier for transmission are quite small. This is due to the fact that the scheduling block size considered in the simulations is on a per-resource-element basis (1 OFDM symbol × 1 subcarrier) and the number of bits in one application PDU is typically much greater than the number of bits that can be carried by a subcarrier. As a result, very little flow merging actually takes place and the performance gain from flow merging is minimal. It is expected that if we increase the scheduling block size to a per-resource-block basis (6/7 OFDM symbols × 12 subcarriers) as in LTE [27, 94], a higher throughput [74] and better QoS performance may be possible due to the further exploitation of the flow merging gain and bit-level scheduling. The higher throughput achieved may thus offset the additional scheduling signaling overhead that is incurred, especially for WFH-FMGS, and result in a higher effective throughput gain. As the potential flow merging gain is dependent on the scheduling block size, determining the appropriate scheduling block size is critical. Detailed analysis needs to be performed when determining the scheduling block size as factors such as dependencies among subcarrier channel gains (over time and across frequency) and the increased bit waiting times need to be taken into consideration and traded off with the potential flow merging gain.

### 9.2.2 Efficient and Optimal Solution to Discrete Rate Adaptation Problem

It is shown in Chapter 7 that KKT-DRA attains a near-optimal solution. However, optimality of the solution to the discrete rate adaptation problem cannot be claimed. Further studies should be undertaken to develop efficient algorithms (for practical importance) to obtain the optimal solution (for theoretical importance) to the discrete rate adaptation problem. Given that the number of bits to be transmitted on a subcarrier is discrete, the MINLP problem can be transformed into a Mixed-Integer Linear Programming (MILP) problem by replacing the non-linear $\log$ function for $c_{i,n}$ in OP4.2 with piece-wise linear representations [95] and replacing constraints (4.11) and (4.12) of OP4.2 accordingly. The problem formulation can thus be represented as follows:

$$\max_{\substack{a_{i,n}\in\{0,1\}\\ w_{i,n}^{d}\in[0,1]\\ b_{i,n}^{j,z}\in\{0,1\}}} \quad \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{z=1}^{B_i^j}\sum_{n=1}^{N} f(\boldsymbol{\theta}_i^{j,z})b_{i,n}^{j,z} \tag{9.1}$$

$$\text{subject to} \quad \sum_{i}\sum_{n}\sum_{d} w_{i,n}^{d}p_{i,n}^{d} \leq P_{total} \tag{9.2}$$

$$\sum_{j}\sum_{z} b_{i,n}^{j,z} \leq \sum_{d} d w_{i,n}^{d} \quad \forall i,n \tag{9.3}$$

$$\sum_{i} a_{i,n} \leq 1 \qquad \forall n \tag{9.4}$$

$$\sum_{n} b_{i,n}^{j,z} \leq 1 \qquad \forall i,j,z \tag{9.5}$$

$$\sum_{d} w_{i,n}^{d} \leq a_{i,n} \quad \forall i,n, \tag{9.6}$$

where the index $d$, $d \in \{0,1,2,...,D\}$ denotes the number of discrete bits a user can transmit on a subcarrier and $D$ is the maximum number of bits that can be transmitted by a subcarrier. The term, $p_{i,n}^{d}$, denotes the transmit power required to transmit $d$ bits of user $i$ on subcarrier $n$ and can be calculated *a priori* for every value of $d$ using $p_{i,n}^{d} = (2^d - 1)\zeta\sigma_0^2/\alpha_{i,n}^2$. The term, $w_{i,n}^{d}$, is a optimization variable which takes on a value between 0 and 1. Techniques for solving MILP problems should be explored and the optimality of the MILP solution to the MINLP problem needs to be established [96].

### 9.2.3 Alternative Formulations of BitQoS Function

In Chapter 7, it is shown that the solution to the bitQoS-aware RA framework is a multi-level water-filling solution where the optimal subcarrier assignment is dependent on both the channel gain and the bitQoS value of the user as opposed to just the channel gain in the classical water-filling solution. Since the proposed KKT-CRA and KKT-DRA scheduling policies are able to obtain the optimal and near-optimal solutions to the continuous and discrete rate adaptation problems, respectively, alternative formulations of the bitQoS function should be studied to take advantage of the bitQoS-aware RA framework to potentially address other critical issues in OFDM networks. As an example, given that the incremental power required to transmit additional bits on an OFDM subcarrier increases as bits are loaded onto a subcarrier, the bitQoS function can be formulated such that it takes into account both the bit latency and transmit power required, where the latency experienced by a bit can be traded-off for energy savings considerations in green communication systems. Trade-offs in terms of the system throughput and pertinent QoS metrics should be quantified along with the savings in energy.

### 9.2.4 Distributed Resource Allocation Algorithms

The RA algorithms proposed in this thesis are centralized scheduling policies. However, as the scheduling granularity increases, so does the computational complexity of the algorithms for systems with a large number of users and subcarriers. In addition to developing efficient and optimal algorithms as outlined in Section 9.2.2, distributed RA algorithms should also be studied to broaden the scope of the centralized scheduling policies considered in this thesis. In particular, computationally complex functions within the centralized RA algorithms need to be identified and segmented for distributed computing in an effort to reduce the computation burden on the computing server. In addition, since in a cellular system, the MS entity is most aware of its channel conditions and application QoS requirements, distributed RA may be performed using a game theoretic approach [97, 98] where multiple players (MSs) seek to maximize a utility function (e.g., bitQoS-weighted throughput) using one of several available strategic RA actions as opposed to a centralized RA being performed solely by the BS. The performance and trade-offs of such a distributed game theoretic RA approach can be evaluated against the centralized scheduling approach presented in this thesis, taking into account that the information received (e.g., CSI) may be imperfect.

# Bibliography

[1] A. K. Parekh and R. G. Gallager, “A generalized processor sharing approach to flow control in integrated services networks: The single-node case,” *IEEE/ACM Trans. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[2] H. Zhang, “Service disciplines for guaranteed performance service in packet-switching networks,” *Proc. IEEE*, vol. 83, no. 10, pp. 1374–1396, Oct. 1995.

[3] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, “Cross-layer design for wireless networks,” *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 74–80, Oct. 2003.

[4] H. Fattah and C. Leung, “An overview of scheduling algorithms in wireless multimedia networks,” *IEEE Wireless Commun.*, vol. 9, no. 5, pp. 76–83, Oct. 2002.

[5] G. Song and Y. Li, “Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks,” *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 127–134, Dec. 2005.

[6] Y. Cao and V. O. K. Li, “Scheduling algorithms in broad-band wireless networks,” *Proc. IEEE*, vol. 89, no. 1, pp. 76–87, Jan. 2001.

[7] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, “Enhancing throughput over wireless LANs using channel state dependent packet scheduling,” in *Proc. INFOCOM*, Mar. 1996, pp. 1133–1140.

[8] J. M. Holtzman, “CDMA forward link waterfilling power control,” in *Proc. VTC*, May 2000, pp. 1663–1667.

[9] A. Jalali, R. Padovani, and R. Pankaj, “Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system,” in *Proc. VTC*, May 2000, pp. 1854–1858.

[10] S. Shakkottai and A. L. Stolyar, “Scheduling for multiple flows sharing a time-varying channel: The exponential rule,” *Amer. Mathematical Soc. Translations*, vol. 207, no. 1, pp. 185–202, Dec. 2002.

[11] ——, “Scheduling algorithms for a mixture of real-time and non-real-time data in HDR,” in *Proc. ITC*, Sep. 2001, pp. 793–804.

[12] G. Barriac and J. M. Holtzman, “Introducing delay sensitivity into the proportional fair algorithm for CDMA downlink scheduling,” in *Proc. IEEE Int. Symp. Spread-Spectrum Tech. & Appl.*, Sep. 2002, pp. 652–656.

[13] A. Farrokh, F. Blomer, and V. Krishnamurthy, “A comparison of opportunistic scheduling algorithms for streaming media in high-speed downlink packet access (HSDPA),” *Lecture Notes in Computer Science*, vol. 3311, no. 1, pp. 130–142, Oct. 2004.

[14] C. Zhou, M. L. Honig, S. Jordan, and R. Berry, “Utility-based resource allocation for wireless networks with mixed voice and data services,” in *Proc. Int. Conf. Computer Comm. Networks*, Oct. 2002, pp. 485–488.

[15] H. Fattah and C. Leung, “A guaranteed quality of service wireless access scheme for CDMA networks,” in *Proc. PACRIM*, Aug. 2003, pp. 533–536.

[16] Y. P. Fallah and H. Alnuweiri, “Hybrid polling and contention access scheduling in IEEE 802.11e WLANs,” *J. Parallel and Distributed Computing*, vol. 67, no. 2, pp. 242–256, Feb. 2007.

[17] S. L. Kota, E. Hossain, R. Fantacci, and A. Karmouch, “Cross-layer protocol engineering for wireless mobile networks: Part 1,” *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 110–111, Dec. 2005.

[18] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1992.

[19] Z. J. Haas, “Design methodologies for adaptive and multimedia networks,” *IEEE Commun. Mag.*, vol. 39, no. 11, pp. 106–107, Nov. 2001.

[20] T. S. Rappaport, A. Annamalai, R. M. Buehrer, and W. H. Tranter, “Wireless communications: Past events and a future perspective,” *IEEE Commun. Mag.*, vol. 40, no. 5, pp. 148–161, May 2002.

[21] C. Verikoukis, L. Alonso, and T. Giamalis, “Cross-layer optimization for wireless systems: A european research key challenge,” *IEEE Commun. Mag.*, vol. 43, no. 7, pp. 1–3, Jul. 2005.

[22] V. Srivastava and M. Motani, “Cross-layer design: A survey and the road ahead,” *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 112–119, Dec. 2005.

[23] H. Jiang, W. Zhuang, and X. Shen, “Cross-layer design for resource allocation in 3G wireless networks and beyond,” *IEEE Commun. Mag.*, vol. 43, no. 12, pp. 120–126, Dec. 2005.

[24] R. Ferrus, L. Alonso, A. Umbert, X. Reves, and J. Perez, “Cross-layer scheduling strategy for UMTS downlink enhancement,” *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 24–28, Jun. 2005.

[25] K. B. Johnsson and D. C. Cox, “An adaptive cross-layer scheduler for improved QoS support of multiclass data services on wireless systems,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 334–343, Feb. 2005.

[26] V. Kawadia and P. R. Kumar, “A cautionary perspective on cross-layer design,” *IEEE Wireless Comm.*, vol. 12, no. 1, pp. 3–11, Feb. 2005.

[27] 3GPP TS 36.211 v9.1.0, “Physical Channels and Modulation (Release 9),” Mar. 2010.

[28] IEEE 802.16-2009, “Part 16: Air Interface for Broadband Wireless Access Systems,” May 2009.

[29] T. Keller and L. Hanzo, “Adaptive multicarrier modulation: A convenient framework for time-frequency processing in wireless communications,” *Proc. IEEE*, vol. 88, no. 5, pp. 611–640, May 2000.

[30] J. Jang and K. Lee, “Transmit power adaptation for multiuser OFDM systems,” *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp. 171–178, Feb. 2003.

[31] W. Rhee and J. Cioffi, “Increase in capacity of multiuser OFDM system using dynamic subchannel allocation,” in *Proc. IEEE VTC*, May 2000, pp. 1085–1089.

[32] Z. Shen, J. Andrews, and B. Evans, “Adaptive resource allocation in multiuser OFDM systems with proportional rate constraints,” *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 2726–2737, Nov. 2005.

[33] 3GPP2 C.S0024-200-C v1.0, “Physical Layer for cdma2000 High Rate Packet Data Air Interface Specification,” Apr. 2010.

[34] X. Wang, G. Giannakis, and A. Marques, “A unified approach to QoS-guaranteed scheduling for channel-adaptive wireless networks,” *Proc. IEEE*, vol. 95, no. 12, pp. 2410–2431, Dec. 2007.

[35] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, “Providing quality of service over a shared wireless link,” *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150–154, Feb. 2001.

[36] G. Song and Y. Li, “Cross-layer optimization for OFDM wireless networks-part I: theoretical framework,” *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 614–624, Mar. 2005.

[37] ——, “Cross-layer optimization for OFDM wireless networks-part II: algorithm development,” *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 625–634, Mar. 2005.

[38] G. Song, “Cross-layer resource allocation and scheduling in wireless multicarrier networks,” Ph.D. dissertation, Georgia Inst. Technol., 2005.

[39] S. Ryu, B. Ryu, H. Seo, and M. Shin, “Urgency and efficiency based packet scheduling algorithm for OFDMA wireless system,” in *Proc. IEEE ICC*, May 2005, pp. 2779–2785.

[40] W. Park, S. Cho, and S. Bahk, “Scheduler design for multiple traffic classes in OFDMA networks,” in *Proc. IEEE ICC*, Jun. 2006, pp. 790–795.

[41] M. Katoozian, K. Navaie, and H. Yanikomeroglu, “Utility-based adaptive radio resource allocation in OFDM wireless networks with traffic prioritization,” *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 66–71, Jan. 2009.

[42] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, “CDMA data QoS scheduling on the forward link with variable channel conditions,” *Bell Labs Tech. Memo.*, Apr. 2000.

[43] 3GPP2 C.S0024-A v2.0 (TIA-856-A-1), “cdma2000 High Rate Packet Data Air Interface Specification,” Aug. 2005.

[44] T. IS-95B, “Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular systems,” Dec. 1998.

[45] A. Bedekar, S. Borst, K. Ramanan, P. Whiting, and E. Yeh, “Downlink scheduling in CDMA data networks,” in *Proc. GLOBECOM*, Dec. 1999, pp. 2653–2657.

[46] 3GPP2 C.S0024-0 v4.0 (TIA-IS-856-2), “cdma2000 High Rate Packet Data Air Interface Specification,” Oct. 2002.

[47] R. Yallapragada and M. Naidu, “New enhancements in 3G technologies,” in *Proc. ICPWC*, Jan. 2005, pp. 182–187.

[48] N. Bhushan, C. Lott, P. Black, R. Attar, Y.-C. Jou, M. Fan, D. Ghosh, and J. Au, “CDMA2000 1xEV-DO Revision A: A physical layer and MAC layer overview,” *IEEE Commun. Mag.*, vol. 44, no. 2, pp. 37–49, Feb. 2006.

[49] C. Lott, N. Bhushan, D. Ghosh, R. Attar, J. Au, and M. Fan, “Reverse traffic channel MAC design of CDMA2000 1xEV-DO Revision A system,” in *Proc. VTC*, May 2005, pp. 1416–1421.

[50] 3GPP2 TSG-C WG 3, “cdma2000 Evaluation Methodology V6,” Dec. 2006.

[51] IETF RFC 2988, “Computing TCP’s Retransmission Timer,” Nov. 2000.

[52] ITU-T G.114, “One-way Transmission Time,” May 2003.

[53] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridg, MA, USA: MIT Press, 1990.

[54] Q. Bi, P.-C. Chen, Y. Yang, and Q. Zhang, “An analysis of VoIP service using 1xEV-DO Revision A system,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 1, pp. 36–45, Jan. 2006.

[55] T. IS-2000.2-A, “Physical Layer Standard for cdma2000 Spread Spectrum Systems,” Mar. 2001.

[56] M. Yavuz, S. Diaz, R. Kapoor, M. Grob, P. Black, Y. Tokgoz, and C. Lott, “VoIP over cdma2000 1xEV-DO Revision A,” *IEEE Commun. Mag.*, vol. 44, no. 2, pp. 88–95, Feb. 2006.

[57] R. Chang, “Synthesis of band-limited orthogonal signals for multichannel data transmission,” *Bell Sys. Tech. J.*, vol. 45, no. 10, pp. 1775–1796, Dec. 1966.

[58] A. Bahai, B. Saltzberg, and M. Ergen, *Multi-Carrier Digital Communications: Theory and Applications of OFDM*, 2nd ed. New York, NY, USA: Springer Verlag, 2004.

[59] A. Goldsmith, *Wireless Communications*. New York, NY, USA: Cambridge University Press, 2005.

[60] ITU-T G.993.1, “Very High Speed Digital Subscriber Line Transceivers,” Jun. 2004.

[61] P. W. C. Chan and R. S. K. Cheng, “Optimal power allocation in zero-forcing MIMO-OFDM downlink with multiuser diversity,” in *Proc. IST Mobile & Wireless Communications Summit*, Jun. 2005, pp. 1–5.

[62] Y. M. Tsang and R. Cheng, “Optimal resource allocation in SDMA/multi-input-single-output/OFDM systems under QoS and power constraints,” in *Proc. IEEE WCNC*, Mar. 2004, pp. 1595–1600.

[63] C. Wong, R. Cheng, K. Lataief, and R. Murch, “Multiuser OFDM with adaptive subcarrier, bit, and power allocation,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 10, pp. 1747–1758, Oct. 1999.

[64] Y. Li, “Pilot-symbol-aided channel estimation for OFDM in wireless systems,” *IEEE Trans. Veh. Technol.*, vol. 49, no. 4, pp. 1207–1215, Jul. 2000.

[65] X. Qiu and K. Chawla, “On the performance of adaptive modulation in cellular systems,” *IEEE Trans. Commun.*, vol. 47, no. 6, pp. 884–895, Jun. 1999.

[66] M. Alouini and A. Goldsmith, “Capacity of Rayleigh fading channels under different adaptive transmission and diversity-combining techniques,” *IEEE Trans. Veh. Technol.*, vol. 48, no. 4, pp. 1165–1181, Aug. 2002.

[67] I. Wong and B. Evans, “Optimal downlink OFDMA resource allocation with linear complexity to maximize ergodic rates,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 3, pp. 962–971, Mar. 2008.

[68] J. Huang, V. Subramanian, R. Agrawal, and R. Berry, “Downlink scheduling and resource allocation for OFDM systems,” in *Proc. Conf. on Info. Science and Systems*, Mar. 2006, pp. 1272–1279.

[69] X. Wang and G. Giannakis, “Resource allocation for wireless multiuser OFDM networks,” *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4359–4372, Jul. 2011.

[70] H. Nguyen, J. Brouet, V. Kumar, and T. Lestable, “Compression of associated signaling for adaptive multi-carrier systems,” in *Proc. IEEE VTC*, May 2004, pp. 1916–1919.

[71] J. Gross, H. Geerdes, H. Karl, and A. Wolisz, “Performance analysis of dynamic OFDMA systems with inband signaling,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 427–436, Mar. 2006.

[72] E. Larsson, “Optimal OFDMA downlink scheduling under a control signaling cost constraint,” *IEEE Trans. Commun.*, vol. 58, no. 10, pp. 2776–2781, Oct. 2010.

[73] R. Moosavi, J. Eriksson, and E. Larsson, “Differential signaling of scheduling information in wireless multiple access systems,” in *Proc. IEEE GLOBECOM*, Dec. 2010, pp. 1–6.

[74] R. Moosavi, J. Eriksson, E. Larsson, N. Wiberg, P. Frenger, and F. Gunnarsson, “Comparison of strategies for signaling of scheduling assignments in wireless OFDMA,” *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4527–4542, Nov. 2010.

[75] S. S. Epp, *Discrete Mathematics with Applications*, 3rd ed. Boston, MA, USA: Brooks Cole, 2003.

[76] S. Golomb, “Run-length encodings,” *IEEE Trans. Inf. Theory*, vol. 12, no. 3, pp. 399–401, Jul. 1966.

[77] T. Welch, “A technique for high-performance data compression,” *Computer*, vol. 17, no. 6, pp. 8–19, Jun. 1984.

[78] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Boston, MA, USA: Academic Press, 1982.

[79] M. Tao, Y. Liang, and F. Zhang, “Resource allocation for delay differentiated traffic in multiuser OFDM systems,” *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2190–2201, Jun. 2008.

[80] W. Karush, “Minima of functions of several variables with inequalities as side constraints,” Master’s thesis, Univ. of Chicago, 1939.

[81] H. Kuhn and A. Tucker, “Nonlinear programming,” in *Proc. Berkeley Symposium on Mathematical Statistics and Probability*, Jul. 1951, pp. 481–492.

[82] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[83] Z. Luo and W. Yu, “An introduction to convex optimization for communications and signal processing,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1426–1438, Aug. 2006.

[84] T. Cover and J. Thomas, *Elements of Information Theory*. New York, NY, USA: John Wiley & Sons, 1991.

[85] ITU-T Rec. Y.1541, “Network performance objectives for IP-based services,” Feb. 2006.

[86] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 2nd ed. New York, NY, USA: McGraw-Hill, 1995.

[87] 3GPP TS 36.104 v9.9.0, “Base Station (BS) radio transmission and reception (Release 9),” Sep. 2011.

[88] J. Zyren and W. McCoy, “Overview of the 3GPP long term evolution physical layer,” White Paper, Freescale Semiconductor, Inc., Jul. 2007.

[89] W. Press, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1992.

[90] R. Kwan, C. Leung, and J. Zhang, “Multiuser scheduling on the downlink of an LTE cellular system,” *Research Letters in Communications*, vol. 2008, no. 1, pp. 1–4, Jan. 2008.

[91] Intel Corporation. (2011) Intel Core i7-990X Processor Extreme Edition. [Online]. Available: http://ark.intel.com/products/52585

[92] ——. (2011) Intel Core i7-3960X Processor Extreme Edition. [Online]. Available: http://ark.intel.com/products/63696

[93] ——. (2012) Ivy Bridge Products. [Online]. Available: http://ark.intel.com/products/codename/29902/Ivy-Bridge

[94] A. Ghosh, J. Zhang, R. Muhamed, and J. Andrews, *Fundamentals of LTE*. Upper Saddle River, NJ, USA: Prentice Hall, 2010.

[95] D. Babayev, “Piece-wise linear approximation of functions of two variables,” *Journal of Heuristics*, vol. 2, no. 4, pp. 313–320, Apr. 1997.

[96] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*. New York, NY, USA: John Wiley & Sons, 1988.

[97] R. Myerson, *Game Theory: Analysis of Conflict*. Cambridge, MA, USA: Harvard University Press, 1997.

[98] D. Gesbert, S. Kiani, A. Gjendemsj, and G. Øien, “Adaptation, coordination, and distributed resource allocation in interference-limited wireless networks,” *Proc. IEEE*, vol. 95, no. 12, pp. 2393–2409, Dec. 2007.

[99] I. Gradshteyn, I. Ryzhik, and A. Jeffrey, *Table of integrals, series, and products*, 6th ed. San Diego, CA, USA: Academic Press, 2000.

# Appendix A

# Inductive Proof of MUP Throughput Gain

Consider two scheduling policies: single-user packet (SUP) and multi-user packet (MUP). We assume all packets are of the same size and each user has one flow (i.e. $J_i = 1$).

The SUP scheduling policy is formulated as follows:

Step 1: Let $Q_{SUP_i}(k)$ denote the priority of user $i$ at time $k$. It is determined by

$$Q_{SUP_i}(k) = EPSize_i(k),\ \forall i \in \mathcal{I}. \tag{A.1}$$

The user to be scheduled at time $k$ is determined as

$$i^*(k) = \arg\max_{i \in \mathcal{I}} Q_{SUP_i}(k). \tag{A.2}$$

Step 2: Packets are selected, one at a time in an iterative fashion, from the data queue of user $i^*$ and added to the physical layer encoder packet until either the physical layer encoder packet $EPSize_{i^*}(k)$ is filled or that there are no more packets in the data queue.

The MUP scheduling policy is formulated similarly as the SUP scheduling policy except that if there is any unfilled space in the physical layer encoder packet, the MUP scheduling policy may use MUP transmission mode and piggyback packets from other users until the physical layer encoder packet is full. Thus, by construction, the MUP scheduling policy always transmits at least as many bits as the SUP scheduling policy in each time slot. Let $\triangle_{SUP}(k)$ and $\triangle_{MUP}(k)$ denote the number of bits scheduled by the SUP and MUP scheduling policies respectively at time $k$. Hence,

$$\triangle_{MUP}(k) \geq \triangle_{SUP}(k) \quad \text{for} \quad k \geq 1. \tag{A.3}$$

Let $C_{SUP}(K)$ and $C_{MUP}(K)$ denote the total number of bits sent by the SUP and MUP scheduling policies respectively from $k = 1$ to $k = K$. They can be written as

$$C_{SUP}(K) = \triangle_{SUP}(1) + \triangle_{SUP}(2) + \ldots + \triangle_{SUP}(K) \tag{A.4}$$

$$C_{MUP}(K) = \triangle_{MUP}(1) + \triangle_{MUP}(2) + \ldots + \triangle_{MUP}(K). \tag{A.5}$$

We claim that $C_{MUP}(k) \geq C_{SUP}(k)$ for $k \geq 1$. This can be proven by induction as follows:

1. Base case: When $K = 1$, $C_{MUP}(1) = \triangle_{MUP}(1) \geq \triangle_{SUP}(1) = C_{SUP}(1)$.

2. Induction hypothesis: Assume that $C_{MUP}(k) \geq C_{SUP}(k)$.

3. Inductive step:

$$\begin{aligned} C_{MUP}(k+1) &= C_{MUP}(k) + \triangle_{MUP}(k+1) && \text{(A.6)} \\ &\geq C_{SUP}(k) + \triangle_{MUP}(k+1) && \text{(A.7)} \\ &\quad \text{(by induction hypothesis)} \\ &\geq C_{SUP}(k) + \triangle_{SUP}(k+1) && \text{(A.8)} \\ &\quad \text{(by (A.3))} \\ &= C_{SUP}(k+1). && \text{(A.9)} \end{aligned}$$

Hence, $C_{MUP}(k) \geq C_{SUP}(k)$ for $k \geq 1$. Q.E.D.

# Appendix B

# Proof of Monotonicity of LHS of (4.21) and Existence of Solution of (4.21)

From (4.21), we define

$$g(\gamma_0) \triangleq \sum_{n=1}^{N}\sum_{z=1}^{I}(-1)^z\binom{I}{z}\left[\frac{z}{E\{\Gamma_n\}}E_1\left(\frac{z\gamma_0}{E\{\Gamma_n\}}\right)-\frac{1}{\gamma_0}e^{-\frac{z\gamma_0}{E\{\Gamma_n\}}}\right]-1. \tag{B.1}$$

The first order derivative of $g(\gamma_0)$ with respect to $\gamma_0$ is

$$\frac{dg(\gamma_0)}{d\gamma_0}=\sum_{n=1}^{N}\sum_{z=1}^{I}(-1)^z\binom{I}{z}\left\{\frac{d}{d\gamma_0}\left[\frac{z}{E\{\Gamma_n\}}E_1\left(\frac{z\gamma_0}{E\{\Gamma_n\}}\right)\right]-\frac{d}{d\gamma_0}\left[\frac{1}{\gamma_0}e^{-\frac{z\gamma_0}{E\{\Gamma_n\}}}\right]\right\}. \tag{B.2}$$

Using the formula [99] $\frac{d}{dx}E_1(x)=-E_0(x)$, where $E_0(x)=\frac{e^{-x}}{x}$, we can rewrite (B.2) as

$$\frac{dg(\gamma_0)}{d\gamma_0}=\sum_{n=1}^{N}\sum_{z=1}^{I}(-1)^z\binom{I}{z}\frac{1}{\gamma_0^2}e^{-\frac{z\gamma_0}{E\{\Gamma_n\}}}=\frac{1}{\gamma_0^2}\sum_{n=1}^{N}h_n(\gamma_0), \tag{B.3}$$

where $h_n(\gamma_0) \triangleq \sum_{z=1}^{I} \binom{I}{z} (-1)^z (e^{-\frac{\gamma_0}{E\{\Gamma_n\}}})^z$. We also have

$$h_n(\gamma_0) = \begin{cases} \sum_{z=0}^{I} \binom{I}{z} (-1)^{I-z} (e^{-\frac{\gamma_0}{E\{\Gamma_n\}}})^z - 1, & \text{if } I \text{ is even} \\ -\sum_{z=0}^{I} \binom{I}{z} (-1)^{I-z} (e^{-\frac{\gamma_0}{E\{\Gamma_n\}}})^z - 1, & \text{if } I \text{ is odd.} \end{cases} \tag{B.4}$$

Recall the binomial formula where $(x+y)^n = \sum_{z=0}^{n} \binom{n}{z} x^{n-z} y^z$, then (B.4) becomes

$$h_n(\gamma_0) = (-1)^I (-1 + e^{-\frac{\gamma_0}{E\{\Gamma_n\}}})^I - 1. \tag{B.5}$$

Thus, for all $\gamma_0 > 0$ and $E\{\Gamma_n\} > 0$, we have $h_n(\gamma_0) \in [-1, 0)$ and $\frac{dg(\gamma_0)}{d\gamma_0} < 0$. In addition, given that $\lim_{\gamma_0 \to 0^+} g(\gamma_0) = +\infty > 0$ and $lim_{\gamma_0 \to^+ \infty} g(\gamma_0) = -1 < 0$, there exists a unique $\gamma_0$ for which $g(\gamma_0) = 0$. Since $g(\gamma_0)$ is a monotonically decreasing function of $\gamma_0$, $\forall I \geq 1$, $N \geq 1$, $\gamma_0 > 0$ and $E\{\Gamma_n\} > 0$, the value of $\gamma_0$ for which $g(\gamma_0) = 0$ can be found numerically using a bisection algorithm.

# Appendix C

# Proof of Concavity of LHS of (7.8)

From (7.8), we define

$$g(a_{i,n}, \pi_{i,n}, b_i^{j,z}) = \sum_n \log_2 \left(1 + \frac{|\alpha_{i,n}^2| \pi_{i,n}}{\zeta \sigma_0^2 a_{i,n}}\right) a_{i,n} - \sum_j \sum_z b_i^{j,z}. \tag{C.1}$$

The Hessian of the function $g$ at the point $\boldsymbol{x} = (a_{i,n}, \pi_{i,n}, b_i^{j,z})$ is given by

$$\begin{aligned} H(g)(\boldsymbol{x}) &= \begin{bmatrix} \dfrac{\partial^2 g}{\partial a_{i,n}^2} & \dfrac{\partial^2 g}{\partial a_{i,n} \partial \pi_{i,n}} & \dfrac{\partial^2 g}{\partial a_{i,n} \partial b_i^{j,z}} \\ \dfrac{\partial^2 g}{\partial \pi_{i,n} \partial a_{i,n}} & \dfrac{\partial^2 g}{\partial \pi_{i,n}^2} & \dfrac{\partial^2 g}{\partial \pi_{i,n} \partial b_i^{j,z}} \\ \dfrac{\partial^2 g}{\partial b_i^{j,z} \partial a_{i,n}} & \dfrac{\partial^2 g}{\partial b_i^{j,z} \partial \pi_{i,n}} & \dfrac{\partial^2 g}{\partial b_i^{j,z2}} \end{bmatrix} & \text{(C.2)} \\ &= \begin{bmatrix} -\dfrac{h_{i,n}^2 \pi_{i,n}^2}{\ln 2 a_{i,n}^3 \left(1 + \frac{h_{i,n}\pi_{i,n}}{a_{i,n}}\right)^2} & \dfrac{h_{i,n}^2 \pi_{i,n}}{\ln 2 a_{i,n}^2 \left(1 + \frac{h_{i,n}\pi_{i,n}}{a_{i,n}}\right)^2} & 0 \\ \dfrac{h_{i,n}^2 \pi_{i,n}}{\ln 2 a_{i,n}^2 \left(1 + \frac{h_{i,n}\pi_{i,n}}{a_{i,n}}\right)^2} & -\dfrac{h_{i,n}^2}{\ln 2 a_{i,n} \left(1 + \frac{h_{i,n}\pi_{i,n}}{a_{i,n}}\right)^2} & 0 \\ 0 & 0 & 0 \end{bmatrix} & \text{(C.3)} \end{aligned}$$

where $h_{i,n}$ denotes $\frac{|\alpha_{i,n}^2|}{\zeta\sigma_0^2}$. The eigenvalues of $H(g)(\boldsymbol{x})$, $\lambda_1 = 0$, $\lambda_2 = 0$ and $\lambda_3 = -h_{i,n}^2(a_{i,n}^2 + \pi_{i,n}^2)/(\ln 2a_{i,n}(a_{i,n} + h_{i,n}\pi_{i,n})^2)$, are obtained by solving $\det(H(g)(\boldsymbol{x}) - \lambda I) = 0$. Given that $h_{i,n} \geq 0$, $a_{i,n} \geq 0$ and $\pi_{i,n} \geq 0$, it can be shown that $H(g)(\boldsymbol{x})$ is a negative semi-definite matrix and hence, $g(\boldsymbol{x})$ is a concave function.

# Appendix D

# Computation Complexity Analysis

Legend:

1. *add* denotes the addition operation

2. *assgn* denotes the assignment operation

3. *comp* denotes the comparison operation

4. *mult* denotes the multiplication operation.

## D.1 WFH-FM

**Algorithm 3** WFH-FM (Part I)

```
\\ 1. Compute bitQoS values
for $i = 1 : I$ do                                              ▷ for $I$ times
    for $j = 1 : J_i$ do                                        ▷ for $J_i$ times
        for $z = 1 : B_i^j(k)$ do                               ▷ for $B_i^j$ times
            $w_i^{j,z}(k) = (k - b_i^{j,z}.arrivalTime) * T_s$    ▷ 1 add, 1 assgn, 1 mult
            $\psi_i^{j,z} = c_j * \pi_j * \gamma_j^{d_j*(w_i^{j,z}(k)-\eta_j)}$    ▷ 1 add, 1 assgn, 4 mult
        end for
    end for
end for
```

**Algorithm 3** WFH-FM (Part II)

```
$\\$ *2. Merge and sort bits by bitQoS values*
for $i = 1 : I$ do                                                   ▷ for $I$ times
    $\psi_i$ = merge the bits from all application flows of user $i$ and
                sort by $\psi_i^{j,z}$ in a descending order        ▷ $B \log B$ assign, $B \log B$ comp
    $B_i = sum_{j \in \mathcal{J}_i} B_i^j(k)$                      ▷ $J_i$ add, 1 assgn
end for

$\\$ *3. Throughput maximization*
$\\$ *3.1. Assign each subcarrier to the user with the highest channel gain*
for $n = 1 : N$ do                                                   ▷ for $N$ times
    $i^*(n) = \arg\max_{i \in \mathcal{I}} SNR_{i,n}$                ▷ 1 assgn, $I$ comp
    $SNR(n) = \frac{|\alpha_{i^*(n),n}|^2}{\zeta * \sigma_0^2}$      ▷ 5 mult, 1 assgn
    for $i = 1 : I$ do                                               ▷ for $I$ times
        if $i = i^*(n)$ then                                         ▷ 1 comp
            $\hat{a}_{i,n} = 1$                                      ▷ 1 assgn
        else
            $\hat{a}_{i,n} = 0$                                      ▷ 1 assgn
        end if
    end for
end for

$\\$ *3.2. Determine transmit power and bit assignment using the water-filling algorithm*
$(\hat{p}_{i,n}, \hat{c}_{i,n}, \hat{b}_{i,n}^{1,z}) = Waterfilling(P_{total}, SNR(n), \hat{a}_{i,n})$

$\\$ *3.3. Perform greedy water-filling subcarrier reassignment*
$\mathcal{U} = \varnothing$ and $C = 0$                              ▷ 2 assgn
for $i = 1 : I$ do                                                   ▷ for $I$ times
    $R(i) = \sum_n \hat{c}_{i,n}$                                    ▷ $N$ add, 1 assgn
    $C = C + \min(R(i), B_i)$                                        ▷ 1 add, 1 assgn, 1 comp
    if $R(i) > B_i$ then                                             ▷ 1 comp
        $\mathcal{U} = \{\mathcal{U}, i\}$                           ▷ 1 assgn
    end if
end for
$\mathcal{U}^c = \mathcal{I} - \mathcal{U}$ and $\Omega_{\mathcal{U}} = \varnothing$   ▷ 2 assgn, $I$ comp
for $i \in \mathcal{U}$ do                                           ▷ for up to $I$ times
    for $n = 1 : N$ do                                               ▷ for $N$ times
        if $a_{i,n} = 1$ then                                        ▷ 1 comp
            $\Omega_{\mathcal{U}} = \{\Omega_{\mathcal{U}}, n\}$     ▷ 1 assgn
        end if
    end for
end for
```

**Algorithm 3** WFH-FM (Part III)

```
loop    ▷ for up to $I$ times
    for $n \in \Omega_{\mathcal{U}}$ do    ▷ for up to $N$ times
        $i^*(n) = \arg\max_i a_{i,n}$    ▷ 1 assgn, $I$ comp
        for $i \in \mathcal{U}^c$ do    ▷ for up to $I$ times
            $\Delta C_{i,n} = \min\left(R(i) + \lfloor \log_2(1 + \hat{p}_{i^*(n),n} * \frac{|\alpha_{i,n}|^2}{\zeta\sigma_0^2})\rfloor, B_i\right) - \min\left(R(i), B_i\right)$
                ▷ 2 add, 1 assgn, 2 comp, 7 mult
        end for
    end for
    $(i^{'}, n^{'}) = \arg\max_{i,n} \Delta C_{i,n}$    ▷ 2 assgn, $IN$ comp
    $\Omega_{\mathcal{U}} = \Omega_{\mathcal{U}} - n^{'}$    ▷ 1 add, 1 assgn
    $\hat{a}^{'}_{i,n^{'}} = 0, \forall i \neq i^{'}$
    $\hat{a}^{'}_{i^{'},n^{'}} = 1$    ▷ $I$ assgn, $I$ comp
    $SNR(n^{'}) = \frac{|\alpha_{i^{'},n^{'}}|^2}{\zeta * \sigma_0^2}$    ▷ 1 assgn, 5 mult
    $(\hat{p}^{'}_{i,n}, \hat{c}^{'}_{i,n}, \hat{b}^{1,z}_{'i,n}) = Waterfilling(P_{total}, SNR(n), \hat{a}^{'}_{i,n})$
    $C^{'} = 0$    ▷ 1 assgn
    for $i = 1 : I$ do    ▷ for $I$ times
        $R(i) = \sum_n \hat{c}^{'}_{i,n}$    ▷ $N$ add, 1 assgn
        $C^{'} = C^{'} + \min(R(i), B_i)$    ▷ 1 add, 1 assgn, 1 comp
        if $R(i) \geq B_i$ then    ▷ 1 comp
            $\mathcal{U}^{c^{'}} = \mathcal{U}^{c^{'}} - i$    ▷ 1 add, 1 assgn
        end if
    end for
    if $C^{'} > C$ then    ▷ 1 comp
        $\hat{a}_{i,n} = \hat{a}^{'}_{i,n}, \forall i, n$    ▷ $IN$ assgn
        $\hat{p}_{i,n} = \hat{p}^{'}_{i,n}, \forall i, n$    ▷ $IN$ assgn
        $\hat{b}^{j,z}_{i,n} = \hat{b}^{'j,z}_{i,n}, \forall i, j, z, n$    ▷ $N \sum_i B_i$ assgn
        $C = C^{'}$    ▷ 1 assgn
    else
        break
    end if
end loop

\\ 3.4. Compute current intermediate objective value
$\hat{\delta}_{obj} = \sum_i \sum_i \sum_z \sum_n \psi_i^{j,z} \hat{b}^{j,z}_{i,n}$    ▷ $N \sum_i B_i$ add, 1 assgn, $\sum_i B_i$ mult
```

**Algorithm 3** WFH-FM (Part IV)

```
$\\$ *4. Iterative subcarrier reassignment*
loop                                                                   ▷ for up to $IN$ times
    for $i = 1 : I$ do                                                 ▷ for $I$ times
        $\mathcal{S}_{un}(i) = \{bit(i,j,z) | \sum_n \hat{b}_{i,n}^{j,z} = 0, j \in \mathcal{J}_i, z \in \{1, \ldots, B_i^j\}\}$
        $\mathcal{S}_{as}(i) = \{bit(i,j,z) | \sum_n \hat{b}_{i,n}^{j,z} = 1, j \in \mathcal{J}_i, z \in \{1, \ldots, B_i^j\}\}$
                                                                       ▷ $2NB_i$ add, $2B_i$ comp
        $\psi_{un}(i) = \max_{bit(i,j,z) \in \mathcal{S}_{un}(i)} \psi_i^{j,z}$            ▷ $B_i$ comp
        $\psi_{as}(i) = \min_{bit(i,j,z) \in \mathcal{S}_{an}(i)} \psi_i^{j,z}$            ▷ $B_i$ comp
    end for
    loop                                                               ▷ for up to $I - 1$ times
        if $\max_i \psi_{un}(i) \leq \min_i \psi_{as}(i)$ then                ▷ $2I + 1$ comp
            break
        end if
        $l^* = \arg\max_i \psi_{un}(i)$                                ▷ 1 assgn, $I$ comp
        $D_{l^*} = \varnothing$                                        ▷ 1 assgn
        for $n = 1 : N$ do                                             ▷ for $N$ times
            if $a_{i,n}^{'} = 0$ then                                  ▷ 1 comp
                $D_{l^*} = \{D_{l^*,n}\}$                              ▷ 1 assgn
            end if
        end for
        $n^* = \arg\max_{n \in D_{l^*}} \alpha_{l^*,n}$                ▷ 1 assgn, up to $N$ comp
        for $i = 1 : I$ do                                             ▷ for $I$ times
            $a_{i,n^*}^{'} = 0$                                        ▷ 1 assgn
        end for
        $a_{l^*,n^*}^{'} = 1$                                          ▷ 1 assgn
        $SNR(n^*) = \frac{|\alpha_{l^*,n^*}|^2}{\zeta * \sigma_0^2}$   ▷ 1 assgn, 5 mult
        $(\hat{p}_{i,n}^{'}, \hat{b'}_{i,n}^{1,z}) = Waterfilling(P_{total}, SNR)$
        $\hat{\delta}_{obj}^{'} = \sum_i \sum_i \sum_z \sum_n \psi_i^{j,z} \hat{b'}_{i,n}^{j,z}$     ▷ $N \sum_i B_i$ add, 1 assgn, $\sum_i B_i$ mult
        if $\hat{\delta}_{obj}^{'} > \hat{\delta}_{obj}$ then          ▷ 1 comp
            $\hat{a}_{i,n} = \hat{a}_{i,n}^{'}, \forall i, n$          ▷ $IN$ assign
            $\hat{p}_{i,n} = \hat{p}_{i,n}^{'}, \forall i, n$          ▷ $IN$ assign
            $\hat{b}_{i,n}^{j,z} = \hat{b}_{i,n}^{'j,z}, \forall i, j, z, n$     ▷ $N \sum_i B_i$ assign
            $\hat{\delta}_{obj} > \hat{\delta}_{obj}^{'}$              ▷ 1 assign
        else
            $\psi_{un}(l^*) = 0$                                       ▷ 1 assign
        end if
    end loop
end loop
```

**Algorithm 3** WFH-FM (Part V)

```
function WATERFILLING($P_{total}$, $SNR(n)$, $a_{i,n}$)
    $x = 0$                                                        ▷ 1 assgn
    $y = P_{total} + \max_{n \in \mathcal{N}} \frac{1}{SNR(n)}$                 ▷ 1 add, 1 assgn, $N$ comp
    for $n = 1 : N$ do                                               ▷ for $N$ times
        $px(n) = \max(0, x - \frac{1}{SNR(n)})$              ▷ 1 add, 1 assgn, 1 comp, 1 mult
        $py(n) = \max(0, y - \frac{1}{SNR(n)})$              ▷ 1 add, 1 assgn, 1 comp, 1 mult
    end for
    $fx = \sum_{n \in \mathcal{N}} px(n) - P_{total}$                          ▷ $N + 1$ add, 1 assgn
    $fy = \sum_{n \in \mathcal{N}} py(n) - P_{total}$                          ▷ $N + 1$ add, 1 assgn
    while $|x - y| > \epsilon$ do                                       ▷ for $L$ times
        $\lambda = \frac{x + y}{2}$                                 ▷ 1 add, 1 assgn, 1 mult
        for $n = 1 : N$ do                                           ▷ for $N$ times
            $p_n = \max(0, \lambda - \frac{1}{SNR(n)})$          ▷ 1 add, 1 assgn, 1 comp, 1 mult
        end for
        $f = \sum_{n \in \mathcal{N}} p_n - P_{total}$                         ▷ $N + 1$ add, 1 assgn
        if $fx * f > 0$ then                                          ▷ 1 comp, 1 mult
            $x = \lambda$                                                  ▷ 1 assgn
            $fx = f$                                                 ▷ 1 assgn
        else if $fy * f > 0$ then                                     ▷ 1 comp, 1 mult
            $y = \lambda$                                                  ▷ 1 assgn
            $fy = f$                                                 ▷ 1 assgn
        else
            break
        end if
    end while

    for $i = 1 : I$ do                                               ▷ for $I$ times
        $HOL = 0$                                                    ▷ 1 assgn
        for $n = 1 : N$ do                                           ▷ for $N$ times
            if $a_{i,n} = 1$ then                                        ▷ 1 comp
                $c_{i,n} = \lfloor \log_2(1 + SNR(n) * p_n) \rfloor$          ▷ 1 add, 1 assgn, 3 mult
                $p_{i,n} = p_n$                                              ▷ 1 assgn
                $b_{i,n}^{1,z} = 1$ for $z = HOL + 1 : HOL + c_{i,n}$               ▷ $c_{i,n}$ assgn
                $HOL = HOL + c_{i,n}$                                   ▷ 1 add, 1 assgn
            end if
        end for
    end for
    return $p_{i,n}, c_{i,n}, b_{i,n}^{1,z}$
end function
```

## D.2 BABL-FM

**Algorithm 4** BABL-FM (Part I)

```
for $i = 1 : I$ do                                                    ▷ for $I$ times
    for $j = 1 : J_i$ do                                              ▷ for $J_i$ times
        for $z = 1 : B_i^j(k)$ do                                     ▷ for $B_i^j$ times
            $w_i^{j,z}(k) = (k - b_i^{j,z}.arrivalTime) * T_s$
                                                                      ▷ 1 add, 1 mult, 1 assgn
            $\psi_i^{j,z} = c_j * \pi_j * \xi_j^{d_j * (w_i^{j,z}(k) - \eta_j)}$
                                                                      ▷ 1 add, 3 mult, 1 assgn
        end for
    end for
end for
for $i = 1 : I$ do                                                    ▷ for $I$ times
    $\psi_i$ = merge the bits from all application flows of user $i$ and sort by $\psi_i^{j,z}$ in a descending order
                                                                      ▷ $B \log B$ comp
end for
for $i = 1 : I$ do                                                    ▷ for $I$ times
    for $n = 1 : N$ do                                                ▷ for $N$ times
        $SNR_{i,n} = \frac{|\alpha_{i,n}|^2}{\zeta * \sigma_0^2}$                      ▷ 5 mult, 1 assgn
        $\hat{c}_{i,n} = 0$                                           ▷ 1 assgn
        $\hat{a}_{i,n} = 0$                                           ▷ 1 assgn
        $\hat{p}_{i,n} = 0$                                           ▷ 1 assgn
    end for
end for
$p_{sum} = 0$                                                         ▷ 1 assgn
$ch_{used}(n) = 0$ for all $n \in \mathcal{N}$                        ▷ $N$ assgn
$HOL(i) = 1$ for all $i \in \mathcal{I}$                              ▷ $I$ assgn
loop                                                                  ▷ for $R$ times
    $bit(i^*, j^*, z^*) = \arg\max_{i \in \mathcal{I}} \psi_i^{HOL(i)}$
                                                                      ▷ $I$ comp, 1 assgn
    for $n = 1 : N$ do                                                ▷ for $N$ times
        if $ch_{used}(n) = 0$ then                                    ▷ 1 comp
            $p_{i^*,n}^{\prime j^*,z^*} = \frac{1}{SNR_{i^*,n}}$                       ▷ 1 mult, 1 assgn
        else if $\hat{a}_{i^*,n} = 1$ then                            ▷ 1 comp
            $p_{i^*,n}^{\prime j^*,z^*} = \frac{2^{\hat{c}_{i^*,n}+1} - 2^{\hat{c}_{i^*,n}}}{SNR_{i^*,n}}$
                                                                      ▷ 2 add, 3 mult, 1 assgn
```

**Algorithm 4** BABL-FM (Part II)

```
        else
          for every bit $bit(l, j, z) \in \mathcal{S}_{l,n}$ do
                                                  ▷ for up to $B$ times
            for $m \in \Omega_l$ do               ▷ for up to $N - 1$ times
              if $ch_{used}(m) = 0$ then          ▷ 1 comp
                $p_{l,m}^{'j,z} = \frac{1}{SNR_{l,m}}$    ▷ 1 mult, 1 assgn
              else if $\hat{a}_{l,m} = 1$ then    ▷ 1 comp
                $p_{l,m}^{'j,z} = \frac{2^{\hat{c}_{l,m}+1} - 2^{\hat{c}_{l,m}}}{SNR_{l,m}}$
                                                  ▷ 2 add, 3 mult, 1 assgn
              end if
            end for
            $m^* = \arg\max_{m \in \Omega_l} p_{l,m}^{'j,z}$
                                                  ▷ $N - 1$ comp, 1 assgn
            $\hat{c}_{l,m^*} = \hat{c}_{l,m^*} + 1$    ▷ 1 add, 1 assgn
            $\hat{c}_{l,n} = \hat{c}_{l,n} - 1$        ▷ 1 add, 1 assgn
          end for
          $p_{i^*,n}^{'j^*,z^*} = \frac{1}{SNR_{i^*,n}} - \hat{p}_{l,n} + \sum_{bit(l,j,z) \in \mathcal{S}_{l,n}} p_{l,m^*}^{'j,z}$
                                                  ▷ $B + 2$ add, 1 mult, 1 assgn
        end if
      end for
      $n^* = \arg\min_{n \in \mathcal{N}} p_{i^*,n}^{'j^*,z^*}$    ▷ $N$ comp, 1 assgn
      if $p_{sum} + p_{i^*,n^*}^{'j^*,z^*} > P_{total}$ then    ▷ 1 add, 1 comp
        break
      else
        $\hat{a}_{i^*,n^*} = 1$                   ▷ 1 assgn
        $\hat{p}_{i^*,n^*} = p_{i^*,n^*}^{'j^*,z^*}$    ▷ 1 assgn
        $\hat{b}_{i^*,n^*}^{j^*,z^*} = 1$         ▷ 1 assgn
        $\hat{c}_{i^*,n^*} = \hat{c}_{i^*,n^*} + 1$    ▷ 1 add, 1 assgn
        $p_{sum} = p_{sum} + p_{i^*,n^*}^{'j^*,z^*}$   ▷ 1 add, 1 assgn
        $ch_{used}(n^*) = 1$                      ▷ 1 assgn
        $HOL(i) = HOL(i) + 1$                     ▷ 1 add, 1 assgn
      end if
    end loop
```

## D.3 KKT-CRA

**Algorithm 5** KKT-CRA (Part I)

```
\\ Compute bitQoS values
for $i = 1 : I$ do                                                          ▷ for $I$ times
    for $j = 1 : J_i$ do                                                    ▷ for $J_i$ times
        for $z = 1 : B_i^j(k)$ do                                           ▷ for $B_i^j$ times
            $w_i^{j,z}(k) = (k - b_i^{j,z}.arrivalTime) * T_s$              ▷ 1 add, 1 mult, 1 assgn
            $\psi_i^{j,z} = c_j * \pi_j * \gamma_j^{d_j*(w_i^{j,z}(k)-\eta_j)}$    ▷ 1 add, 4 mult, 1 assgn
        end for
    end for
end for

for $i = 1 : I$ do                                                          ▷ for $I$ times
    $\psi_i$ = merge the bits from all application flows of user $i$ and
                sort by $\psi_i^{j,z}$ in a descending order                ▷ $B \log B$ comp
    $B_i = sum_{j \in \mathcal{J}_i} B_i^j(k)$                              ▷ $J_i$ add, 1 assgn
    for $n = 1 : N$ do                                                      ▷ for $N$ times
        $SNR(i,n) = \frac{|\alpha_{i,n}|^2}{\zeta * \sigma_0^2}$            ▷ 5 mult, 1 assgn
    end for
end for

$\beta = \epsilon$                                                          ▷ 1 assgn
$\gamma_i = \epsilon$ for all $i \in \mathcal{I}$                           ▷ I assgn
$S_i = 0$ for all $i \in \mathcal{I}$                                       ▷ I assgn
loop                                                                        ▷ for $D$ times
    if $S_i = 1$ for all $i \in \mathcal{I}$ then                           ▷ I comp
        break
    end if
    \\ Perform subcarrier assignment
    for $i = 1 : I$ do                                                      ▷ for $I$ times
        for $n = 1 : N$ do                                                  ▷ for $N$ times
            $H(i,n) = \gamma_i \left[ \max\left(0, \log_2\left(\frac{SNR(i,n)\gamma_i}{\beta \ln 2}\right)\right)\right.$
                $\left. - \frac{1}{\ln 2} \max\left(0, 1 - \frac{\beta \ln 2}{SNR(i,n)\gamma_i}\right)\right]$
                                                      ▷ 2 add, 2 comp, 11 mult, 1 assgn
        end for
    end for
```

**Algorithm 5** KKT-CRA (Part II)

```
    for $n = 1 : N$ do                                              ▷ for $N$ times
        $i^*(n) = \arg\max_{i \in \mathcal{I}} H(i, n)$             ▷ $I$ comp, 1 assgn
        for $i = 1 : I$ do                                          ▷ for $I$ times
            if $i = i^*(n)$ then                                    ▷ 1 comp
                $\hat{a}_{i,n} = 1$                                 ▷ 1 assgn
            else
                $\hat{a}_{i,n} = 0$                                 ▷ 1 assgn
            end if
        end for
    end for
    $\backslash\backslash$ Use bisection to find $\beta$ and $p_{i,n}$
    $x = \epsilon$                                                  ▷ 1 assgn
    $y = 10^8$                                                      ▷ 1 assgn
    for $n = 1 : N$ do                                              ▷ for $N$ times
        $px(n) = \max(0, \frac{\gamma_{i^*(n)}}{x \ln 2} - \frac{1}{SNR(i^*(n), n)})$    ▷ 1 comp, 1 add, 4 mult, 1 assgn
        $py(n) = \max(0, \frac{\gamma_{i^*(n)}}{y \ln 2} - \frac{1}{SNR(i^*(n), n)})$    ▷ 1 comp, 1 add, 4 mult, 1 assgn
    end for
    $fx = \sum_{n \in \mathcal{N}} px(n) - P_{total}$               ▷ N+1 add, 1 assgn
    $fy = \sum_{n \in \mathcal{N}} py(n) - P_{total}$               ▷ N+1 add, 1 assgn
    while $|x - y| > \epsilon$ do                                   ▷ for $Q$ times
        $\lambda = \frac{x + y}{2}$                                 ▷ 1 add, 1 mult, 1 assgn
        for $n = 1 : N$ do                                          ▷ for $N$ times
            $p\lambda(n) = \max(0, \frac{\gamma_{i^*(n)}}{\lambda \ln 2} - \frac{1}{SNR(i^*(n), n)})$    ▷ 1 comp, 1 add, 4 mult, 1 assgn
        end for
        $f\lambda = \sum_{n \in \mathcal{N}} p\lambda(n) - P_{total}$    ▷ $N + 1$ add, 1 assgn
        if $fx * f\lambda > 0$ then                                 ▷ 1 comp, 1 mult
            $x = \lambda$                                           ▷ 1 assgn
            $fx = f\lambda$                                         ▷ 1 assgn
        else if $fy * f\lambda > 0$ then                            ▷ 1 comp, 1 mult
            $y = \lambda$                                           ▷ 1 assgn
            $fy = f\lambda$                                         ▷ 1 assgn
        else
            break
        end if
    end while
```

**Algorithm 5** KKT-CRA (Part III)

```
    $\beta = \lambda$                                                       ▷ 1 assgn
    for $i = 1 : I$ do                                                       ▷ for $I$ times
        $C(i) = 0$                                                          ▷ 1 assgn
        for $n = 1 : N$ do                                                   ▷ for $N$ times
            $\hat{p}_{i,n} = \hat{a}_{i,n} \max(0, \frac{\gamma_i}{\beta \ln 2} - \frac{1}{SNR(i,n)})$   ▷ 1 comp, 1 add, 5 mult, 1 assgn
            $\hat{c}_{i,n} = \log_2(1 + SNR(i,n)\hat{p}_{i,n})$              ▷ 1 add, 2 mult, 1 assgn
            $C(i) = C(i) + \hat{c}_{i,n}$                                    ▷ 1 add, 1 assgn
        end for
        $b_i^{1,z} = 1$ for $z = 1 : \lfloor C(i) \rfloor$
        $b_i^{1,\lfloor C(i) \rfloor + 1} = C(i) - \lfloor C(i) \rfloor$
        $b_i^{1,z} = 0$ for $z = \lfloor C(i) \rfloor + 2 : B_i$             ▷ $B$ assgn
    end for
    for $i = 1 : I$ do                                                       ▷ for $I$ times
        if $S(i) = 0$ then                                                   ▷ 1 comp
            $S(i) = (\gamma_i + \epsilon) > \psi_i^{1,\lceil C(i) \rceil + 1}$   ▷ 1 comp, 2 add, 1 mult, 1 assgn
        else
            if $(\gamma_i + \epsilon) < \psi_i^{1,\lceil C(i) \rceil + 1}$ then   ▷ 1 comp, 2 add, 1 mult
                $S(i) = 0$                                                  ▷ 1 assgn
            end if
        end if
    end for
    \\ Update $\gamma_i$
    for $i = 1 : I$ do                                                       ▷ for $I$ times
        if $\lceil C(i) \rceil + 1 > B_i$ then                               ▷ 1 comp, 1 add, 1 mult
            $\psi^{HOL}(i) = 0$                                             ▷ 1 assgn
        else
            $\psi^{HOL}(i) = \psi_i^{1,\lceil C(i) \rceil + 1}$             ▷ 1 add, 1 mult, 1 assgn
        end if
        if $S(i) = 0$ then                                                   ▷ 1 comp
            $\gamma_i = (1 - \delta)\gamma_i + \delta \psi^{HOL}(i)$         ▷ 2 add, 2 mult, 1 assgn
        end if
    end for
end loop
```

## D.4 KKT-DRA

**Algorithm 6** KKT-DRA (Part I)

```
\\ *Compute bitQoS values*
**for** $i = 1 : I$ **do**                                              ▷ for $I$ times
    **for** $j = 1 : J_i$ **do**                                        ▷ for $J_i$ times
        **for** $z = 1 : B_i^j(k)$ **do**                               ▷ for $B_i^j$ times
            $w_i^{j,z}(k) = (k - b_i^{j,z}.arrivalTime) * T_s$          ▷ 1 add, 1 mult, 1 assgn
            $\psi_i^{j,z} = c_j * \pi_j * \gamma_j^{d_j*(w_i^{j,z}(k)-\eta_j)}$   ▷ 1 add, 4 mult, 1 assgn
        **end for**
    **end for**
**end for**

**for** $i = 1 : I$ **do**                                              ▷ for $I$ times
    $\psi_i$ = merge the bits from all application flows of user $i$ and
                    sort by $\psi_i^{j,z}$ in a descending order        ▷ $B \log B$ comp
    $B_i = sum_{j \in \mathcal{J}_i} B_i^j(k)$                          ▷ $J_i$ add, 1 assgn
    **for** $n = 1 : N$ **do**                                          ▷ for $N$ times
        $SNR(i,n) = \frac{|\alpha_{i,n}|^2}{\zeta * \sigma_0^2}$        ▷ 5 mult, 1 assgn
    **end for**
**end for**

$\beta = \epsilon$                                                      ▷ 1 assgn
$\gamma_i = \epsilon$ for all $i \in \mathcal{I}$                       ▷ I assgn
$S_i = 0$ for all $i \in \mathcal{I}$                                   ▷ I assgn
**loop**                                                                ▷ for $D$ times
    **if** $S_i = 1$ for all $i \in \mathcal{I}$ **then**               ▷ I comp
        **break**
    **end if**
    \\ *Perform subcarrier assignment*
    **for** $i = 1 : I$ **do**                                          ▷ for $I$ times
        **for** $n = 1 : N$ **do**                                      ▷ for $N$ times
            $H(i,n) = \gamma_i \left[ \max\left(0, \log_2\left(\frac{SNR(i,n)\gamma_i}{\beta \ln 2}\right)\right) - \frac{1}{\ln 2} \max\left(0, 1 - \frac{\beta \ln 2}{SNR(i,n)\gamma_i}\right) \right]$
                                                ▷ 2 add, 2 comp, 11 mult, 1 assgn
        **end for**
    **end for**
```

**Algorithm 6** KKT-DRA (Part II)

```
    for $n = 1 : N$ do                                              ▷ for $N$ times
        $i^*(n) = \arg\max_{i \in \mathcal{I}} H(i, n)$              ▷ $I$ comp, 1 assgn
        for $i = 1 : I$ do                                          ▷ for $I$ times
            if $i = i^*(n)$ then                                    ▷ 1 comp
                $\hat{a}_{i,n} = 1$                                  ▷ 1 assgn
            else
                $\hat{a}_{i,n} = 0$                                  ▷ 1 assgn
            end if
        end for
    end for
    \\ Use bit-loading to find $\hat{p}_{i,n}$ and $\hat{b}_{i,n}^{j,z}$
    $p_{used} = 0$                                                   ▷ 1 assgn
    $p_{inc} = 0$                                                    ▷ 1 assgn
    $\hat{p}_{i,n} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{N}$        ▷ $NI$ assgn
    $r(n) = 0$ for all $n \in \mathcal{N}$                            ▷ $N$ assgn
    $C(i) = 0$ for all $i \in \mathcal{I}$                            ▷ $I$ assgn
    $\hat{b}_{i,n}^{1,z} = 0$ for all $i \in \mathcal{I}, n \in \mathcal{N}, z \in \{1, \ldots, B_i\}$   ▷ $IBN$ assgn
    while $p_{used} + p_{inc} \leq P_{total}$ do                      ▷ for $R$ times
        $p_{used} = p_{used} + p_{inc}$                               ▷ 1 add, 1 assgn
        for $n = 1 : N$ do                                          ▷ for $N$ times
            if $C(i^*(n)) \geq B_{i^*(n)}$ then                       ▷ 1 comp
                $p_{change}(n) = \infty$                              ▷ 1 assgn
                $\psi_{change}(n) = 0$                                ▷ 1 assgn
            else
                $p_{change}(n) = \frac{2^{r(n)+1} - 2^{r(n)}}{SNR(i^*(n), n)}$   ▷ 2 add, 3 mult, 1 assgn
                $\psi_{change}(n) = \psi_i^{1,C(i^*(n))+1}$           ▷ 1 add, 1 assgn
            end if
        end for
        $n^* = \arg\max_{n \in \mathcal{N}} \psi_{change}(n)/p_{change}(n)$   ▷ $N$ mult, $N$ comp, 1 assgn
        $p_{inc} = p_{change}(n^*)$                                  ▷ 1 assgn
        if $p_{used} + p_{inc} \leq P_{total}$ then                   ▷ 1 add, 1 comp
            $r(n^*) = r(n^*) + 1$                                    ▷ 1 add, 1 assgn
            $C(i^*(n^*)) = C(i^*(n^*)) + 1$                          ▷ 1 add, 1 assgn
            $\hat{p}_{i^*(n^*),n^*} = \hat{p}_{i^*(n^*),n^*} + p_{inc}$    ▷ 1 add, 1 assgn
            $\hat{b}_{i^*(n^*),n^*}^{1,C(i^*(n^*))} = 1$               ▷ 1 assgn
        end if
    end while
```

**Algorithm 6** KKT-DRA (Part III)

```
\\ *Use bisection to find* $\beta$
$x = \epsilon$                                                            ▷ 1 assgn
$y = 10^8$                                                                ▷ 1 assgn
**for** $n = 1 : N$ **do**                                                ▷ for $N$ times
    $px(n) = \max(0, \frac{\gamma_{i^*(n)}}{x \ln 2} - \frac{1}{SNR(i^*(n), n)})$    ▷ 1 comp, 1 add, 4 mult, 1 assgn
    $py(n) = \max(0, \frac{\gamma_{i^*(n)}}{y \ln 2} - \frac{1}{SNR(i^*(n), n)})$    ▷ 1 comp, 1 add, 4 mult, 1 assgn
**end for**
$fx = \sum_{n \in \mathcal{N}} px(n) - P_{total}$                         ▷ N+1 add, 1 assgn
$fy = \sum_{n \in \mathcal{N}} py(n) - P_{total}$                         ▷ N+1 add, 1 assgn
**while** $|x - y| > \epsilon$ **do**                                     ▷ for $Q$ times
    $\lambda = \frac{x + y}{2}$                                           ▷ 1 add, 1 mult, 1 assgn
    **for** $n = 1 : N$ **do**                                            ▷ for $N$ times
        $p\lambda(n) = \max(0, \frac{\gamma_{i^*(n)}}{\lambda \ln 2} - \frac{1}{SNR(i^*(n), n)})$   ▷ 1 comp, 1 add, 4 mult, 1 assgn
    **end for**
    $f\lambda = \sum_{n \in \mathcal{N}} p\lambda(n) - P_{total}$         ▷ $N + 1$ add, 1 assgn
    **if** $fx * f\lambda > 0$ **then**                                   ▷ 1 comp, 1 mult
        $x = \lambda$                                                     ▷ 1 assgn
        $fx = f\lambda$                                                   ▷ 1 assgn
    **else if** $fy * f\lambda > 0$ **then**                              ▷ 1 comp, 1 mult
        $y = \lambda$                                                     ▷ 1 assgn
        $fy = f\lambda$                                                   ▷ 1 assgn
    **else**
        **break**
    **end if**
**end while**
$\beta = \lambda$                                                         ▷ 1 assgn
**for** $i = 1 : I$ **do**                                                ▷ for $I$ times
    **if** $S(i) = 0$ **then**                                            ▷ 1 comp
        $S(i) = (\gamma_i + \epsilon) > \psi_i^{1,C(i)+1}$                ▷ 1 comp, 2 add, 1 assgn
    **else**
        **if** $(\gamma_i + \epsilon) < \psi_i^{1,C(i)+1}$ **then**       ▷ 1 comp, 2 add
            $S(i) = 0$                                                    ▷ 1 assgn
        **end if**
    **end if**
**end for**
```

**Algorithm 6** KKT-DRA (Part IV)

```
    \\ Update $\gamma_i$
    for $i = 1 : I$ do                                        ▷ for $I$ times
        if $C(i) + 1 > B_i$ then                              ▷ 1 comp, 1 add
            $\psi^{HOL}(i) = 0$                               ▷ 1 assgn
        else
            $\psi^{HOL}(i) = \psi_i^{1,C(i)+1}$               ▷ 1 add, 1 assgn
        end if
        if $S(i) = 0$ then                                    ▷ 1 comp
            $\gamma_i = (1 - \delta)\gamma_i + \delta\psi^{HOL}(i)$   ▷ 2 add, 2 mult, 1 assgn
        end if
    end for
end loop
```

## D.5 WF

**Algorithm 7** WF (Part I)

```
for $n = 1 : N$ do                                              ▷ for $N$ times
    $i^*(n) = \arg\max_{i \in \mathcal{I}} \alpha_{i,n}$                    ▷ $I$ comp, 1 assgn
    $SNR(n) = \frac{|\alpha_{i^*(n),n}|^2}{\zeta * \sigma_0^2}$                  ▷ 5 mult, 1 assgn
    for $i = 1 : I$ do                                          ▷ for $I$ times
        if $i = i^*(n)$ then                                    ▷ 1 comp
            $\hat{a}_{i,n} = 1$                                 ▷ 1 assgn
        else
            $\hat{a}_{i,n} = 0$                                 ▷ 1 assgn
        end if
        $\hat{p}_{i,n} = 0$                                     ▷ 1 assgn
    end for
end for

$x = 0$                                                         ▷ 1 assgn
$y = P_{total} + \max_{n \in \mathcal{N}} \frac{1}{SNR(n)}$               ▷ $N$ comp, 1 add, 1 assgn
for $n = 1 : N$ do                                              ▷ for $N$ times
    $px(n) = \max(0, x - \frac{1}{SNR(n)})$                     ▷ 1 comp, 1 add, 1 mult, 1 assgn
    $py(n) = \max(0, y - \frac{1}{SNR(n)})$                     ▷ 1 comp, 1 add, 1 mult, 1 assgn
end for
$fx = \sum_{n \in \mathcal{N}} px(n) - P_{total}$                       ▷ N+1 add, 1 assgn
$fy = \sum_{n \in \mathcal{N}} py(n) - P_{total}$                       ▷ N+1 add, 1 assgn
while $|x - y| > \epsilon$ do                                   ▷ for $L$ times
    $\lambda = \frac{x + y}{2}$                                 ▷ 1 add, 1 mult, 1 assgn
    for $n = 1 : N$ do                                          ▷ for $N$ times
        $\hat{p}_{i^*(n),n} = \max(0, \lambda - \frac{1}{SNR(n)})$  ▷ 1 comp, 1 add, 1 mult, 1 assgn
    end for
    $f = \sum_{n \in \mathcal{N}} \hat{p}_{i^*(n),n} - P_{total}$           ▷ $N + 1$ add, 1 assgn
```

**Algorithm 7** WF (Part II)

```
    if $fx * f > 0$ then                                    ▷ 1 comp, 1 mult
        $x = \lambda$                                        ▷ 1 assgn
        $fx = f$                                             ▷ 1 assgn
    else if $fy * f > 0$ then                               ▷ 1 comp, 1 mult
        $y = \lambda$                                        ▷ 1 assgn
        $fy = f$                                             ▷ 1 assgn
    else
        break
    end if
end while

for $i = 1 : I$ do                                           ▷ for $I$ times
    $HOL = 0$                                                ▷ 1 assgn
    for $n = 1 : N$ do                                       ▷ for $N$ times
        if $\hat{a}_{i,n} = 1$ then                          ▷ 1 comp
            $\hat{c}_{i,n} = \log_2(1 + SNR(n) * \hat{p}_{i,n})$    ▷ 1 add, 2 mult, 1 assgn
            $b_{i,n}^{1,z} = 1$ for $z = HOL + 1 : HOL + \hat{c}_{i,n}$    ▷ $c_{i,n}$ assgn
            $HOL = HOL + \hat{c}_{i,n}$                      ▷ 1 add, 1 assgn
        end if
    end for
end for
```

## D.6 MDU

**Algorithm 8** MDU (Part I)

```
for $n = 1 : N$ do                                                        ▷ for $N$ times
    $p'(n) = \frac{P_{total}}{N}$                                         ▷ 1 mult, 1 assgn
    $i^*(n) =$ a random number from $[1, I]$                              ▷ 1 assgn
    for $i = 1 : I$ do                                                    ▷ for $I$ times
        $SNR(i,n) = \frac{|\alpha_{i,n}|^2}{\zeta * \sigma_0^2}$          ▷ 5 mult, 1 assgn
        if $i = i^*(n)$ then                                              ▷ 1 comp
            $\hat{a}_{i,n} = 1$                                           ▷ 1 assgn
        else
            $\hat{a}_{i,n} = 0$                                           ▷ 1 assgn
        end if
    end for
end for
for $i = 1 : I$ do                                                        ▷ for $I$ times
    $r_i = 0$                                                             ▷ 1 assgn
    $w(i) =$ UtilityFunc($\frac{Q_i}{\overline{r}_i}$, flow type of $i)/\overline{r}_i$   ▷ 2 comp, 2 add, 4 mult, 2 assgn
    $\gamma_i = w_i * (r_i < Q_i)$                                        ▷ 1 comp, 1 mult
end for
loop                                                                      ▷ for $\kappa$ times
    for $n = 1 : N$ do                                                    ▷ for $N$ times
        for $i = 1 : I$ do                                                ▷ for $I$ times
            $c(i,n) = \log_2(1 + SNR(i,n) * p'(n)) * \hat{a}_{i,n}$       ▷ 1 add, 3 mult, 1 assgn
        end for
        $i^*(n) = \arg\max_{i \in \mathcal{I}} \gamma_i * c(i,n)$         ▷ I comp, I mult, 1 assgn
    end for

    $x = \epsilon$                                                        ▷ 1 assgn
    $y = \max_{n \in \mathcal{N}} \gamma_{i^*(n)} * SNR(i^*(n), n)$       ▷ $N$ mult, $N$ comp
    for $n = 1 : N$ do                                                    ▷ for $N$ times
        $px(n) = \max(0, \frac{\gamma_{i^*(n)}}{x} - \frac{1}{SNR(i^*(n),n)})$   ▷ 1 comp, 1 add, 2 mult, 1 assgn
        $py(n) = \max(0, \frac{\gamma_{i^*(n)}}{y} - \frac{1}{SNR(i^*(n),n)})$   ▷ 1 comp, 1 add, 2 mult, 1 assgn
    end for
```

**Algorithm 8** MDU (Part II)

```
    $fx = \sum_{n \in \mathcal{N}} px(n) - P_{total}$    ▷ $N + 1$ add, 1 assgn
    $fy = \sum_{n \in \mathcal{N}} py(n) - P_{total}$    ▷ $N + 1$ add, 1 assgn
    while $|x - y| > \epsilon$ do    ▷ for $L$ times
        $\lambda = \frac{x+y}{2}$    ▷ 1 add, 1 mult, 1 assgn
        for $n = 1 : N$ do    ▷ for $N$ times
            $p'(n) = \max(0, \frac{\gamma_{i^*(n)}}{\lambda} - \frac{1}{SNR(i^*(n), n)})$    ▷ 1 comp, 1 add, 2 mult, 1 assgn
        end for
        $f = \sum_{n \in \mathcal{N}} \hat{p}_{i^*(n),n} - P_{total}$    ▷ $N + 1$ add, 1 assgn
        if $fx * f > 0$ then    ▷ 1 comp
            $x = \lambda$    ▷ 1 assgn
            $fx = f$    ▷ 1 assgn
        else if $fy * f > 0$ then    ▷ 1 comp
            $y = \lambda$    ▷ 1 assgn
            $fy = f$    ▷ 1 assgn
        else
            break
        end if
    end while
    for $i = 1 : I$ do    ▷ for $I$ times
        $r_i = 0$    ▷ 1 assgn
        for $n = 1 : N$ do    ▷ for $N$ times
            if $\hat{a}_{i,n} = 1$ then    ▷ 1 comp
                $r_i = r_i + \log_2(1 + SNR(i, n) * p'(n))$    ▷ 2 add, 2 mult, 1 assgn
            end if
        end for
    end for
    for $i = 1 : I$ do    ▷ for $I$ times
        $\gamma_i = (1 - \mu) * \gamma_i + \mu * w_i * (r_i < Q_i)$    ▷ 1 comp, 2 add, 3 mult, 1 assgn
    end for
    if $\sum_{i \in \mathcal{I}} w_i * (r_i < Q_i) * (r_i^{old} - r_i) \leq \epsilon$ then    ▷ $I$(1 comp, 3 mult, 2 add), 1 comp
        break
    end if
end loop
```

**Algorithm 8** MDU (Part III)

```
**for** $i = 1 : I$ **do**                                          ▷ for $I$ times
    $HOL = 0$                                                       ▷ 1 assgn
    **for** $n = 1 : N$ **do**                                      ▷ for $N$ times
        **if** $\hat{a}_{i,n} = 1$ **then**                         ▷ 1 comp
            $\hat{c}_{i,n} = \log_2(1 + SNR(i,n) * \hat{p}_{i,n})$  ▷ 1 add, 2 mult, 1 assgn
            $b_{i,n}^{1,z} = 1$ for $z = HOL + 1 : HOL + \hat{c}_{i,n}$   ▷ $c_{i,n}$ assgn
            $HOL = HOL + \hat{c}_{i,n}$                             ▷ 1 add, 1 assgn
        **end if**
    **end for**
**end for**

**function** UTILITYFUNC($x$, flow type)
    **if** flow type is BE **then**                                 ▷ 1 comp
        **if** $x < \eta_{BE}$ **then**                             ▷ 1 comp
            $f = x^{0.5}$                                           ▷ 1 mult, 1 assgn
        **else**
            $f = \eta_{BE}^{0.5}$                                   ▷ 1 mult, 1 assgn
        **end if**
    **else if** flow type is EF **then**                            ▷ 1 comp
        **if** $x < \eta_{EF}$ **then**                             ▷ 1comp
            $f = x$                                                 ▷ 1 assgn
        **else**
            $x^{1.5} - \eta_{EF}^{1}.5 + \eta_{EF}$                 ▷ 2 add, 2 mult, 1 assgn
        **end if**
    **end if**
    **return** $f$
**end function**
```