SCALABLE VIDEO STREAMING OVER WIMAX NETWORKS

by

Somsubhra Sharangi
M.Tech., Indian Institute of Technology Madras, India, 2005
B.Tech., University of Kalyani, India, 2003

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

MASTER OF SCIENCE

in the School
of
Computing Science

© Somsubhra Sharangi 2010
SIMON FRASER UNIVERSITY
Fall 2010

All rights reserved. However, in accordance with the *Copyright Act of Canada*, this work may be reproduced, without authorization, under the conditions for *Fair Dealing.* Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

APPROVAL

Name:	Somsubhra Sharangi
Degree:	Master of Science
Title of Thesis:	Scalable Video Streaming over WiMAX Networks

Examining Committee: **Dr. Qianping Gu**,
Professor of Computing Science
Chair

Dr. Ramesh Krishnamurti
Senior Supervisor
Professor of Computing Science

Dr. Mohamed Hefeeda
Senior Supervisor
Associate Professor of Computing Science

Dr. Binay Bhattacharya
Supervisor
Professor of Computing Science

Dr. Joseph G. Peters
Examiner
Professor of Computing Science

Date Approved: September 20, 2010

SFU SIMON FRASER UNIVERSITY
LIBRARY

Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the "Institutional Repository" link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.ca/handle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

Last revision: Spring 09

Abstract

The Multicast/Broadcast Service (MBS) feature of mobile WiMAX network is a promising technology for providing wireless multimedia, because it allows the delivery of multimedia content to large-scale user communities in a cost-efficient manner. In this thesis, we focus on two research problems in such networks: (i) maximizing the video quality and (ii) minimizing energy consumption for mobile receivers. We prove that these problems are NP-Complete, and propose constant factor approximation algorithms to solve them. Our algorithms intelligently select video layers from a scalable video stream to maximize video quality and construct burst transmission schedules that reduce receiver device energy consumption without sacrificing the video quality. Using extensive simulation and mathematical analysis we show that the proposed solution: (i) is efficient in terms of execution time, (ii) achieves high radio resource utilization, (iii) maximizes the received video quality, and (iv) minimizes the energy consumption for mobile receivers.

Keywords: multicast video streaming; mobile multimedia; wireless transmission scheduling; scalable video coding; WiMAX; energy efficiency

Acknowledgments

I am indebted to Dr. Ramesh Krishnamurti, my senior supervisor, for continuously supporting me over these years with his patience and knowledge. Anytime I needed help, he made his time available for me and helped me with valuable advice. I will also like to thank Dr. Mohamed Hefeeda, my senior supervisor, for his support. It was his guidance and encouragement that taught me how to do research, and without his support, completion of this thesis would not have been possible for me.

I would like to express my gratitude to Dr. Binay Bhattacharya, my supervisor, and Dr. Joseph Peters, my thesis examiner, for being on my committee and reviewing this thesis. I also would like to thank Dr. Qianping Gu for taking the time to chair my thesis defense.

Last but certainly not least, I owe my deepest gratitude to my parents for their love, encouragement and endless support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce features of WiMAX networks for scalable video multicast. We describe the challenges of providing high-quality streaming services in these networks under practical constraints. We present two research problems that we model to address these challenges and summarize our contributions on solving these problems. Then we describe the organization of rest of the thesis. Acronyms used in this thesis are listed in Appendix A.

1.1 Introduction

The demand for mobile multimedia streams has been increasing in the past few years as indicated by multiple market analysis studies [38, 30]. Multimedia streams can be delivered to mobile devices over a variety of wireless networks, including 3G (Third Generation mobile communication standards), WiFi (Wireless Local Area Network technologies based on IEEE802.11 set of standards) and WiMAX (Worldwide Interoperability for Microwave Access) networks. In this thesis, we focus on multimedia streaming over WiMAX networks, which are specified by the IEEE 802.16 standard [19]. Although some of the currently deployed WiMAX networks are mostly used to provide wireless Internet access to subscribers, the WiMAX standard supports various network services. One of these services is the Multicast and Broadcast Service (MBS), which can be used to deliver multimedia traffic to large-scale user communities. For example, Yota Telecom [41] has recently started a mobile TV service with 25 channels over its 10Mbps mobile WiMAX network, and UDCast [37] has announced plans for developing broadcast TV service supporting around 50 channels over mobile WiMAX. It is expected that more WiMAX deployments will offer mobile multimedia services in the near future. Although a considerable amount of work has been done to make these deployments a reality, several research problems remain to be addressed in order to optimize the quality of the offered multimedia services.

The Multicast/Broadcast Service feature of mobile WiMAX networks is a promising technology for providing wireless multimedia, because it allows the delivery of multimedia content to large-scale user communities in a cost-efficient manner. In this thesis, we consider WiMAX networks that transmit multiple video streams encoded in scalable manner to mobile receivers using the MBS feature. In addition, since many subscribers of the WiMAX multimedia services are expected to be mobile users with energy-constrained devices, such as smart phones, minimizing the energy consumption of these devices becomes an important problem in order to extend the viewing time. The WiMAX standard defines sleep mode operations for reducing energy consumption at the receiver. We focus on two research problems in such networks: (i) maximizing the video quality and (ii) minimizing the energy consumption for mobile receivers. We formulate and solve the substream selection problem to maximize the video quality, which arises when multiple scalable video streams are broadcast to mobile receivers with limited resources. We show that this problem is NP-Complete, and we design a polynomial time approximation algorithm to solve it. We prove that the solutions computed by our algorithm are always within a small constant factor from the optimal solutions. In addition, we extend our algorithm to reduce the energy consumption of mobile receivers.

In the WiMAX physical layer, data is transmitted over multiple carriers in Time Division Duplex (TDD) frames. As illustrated in Figure 1.1, each frame contains header information and upload/download maps followed by bursts of user data. Since video dissemination is expected to be a prevalent traffic pattern in future networks, the WiMAX standard defines Multicast and Broadcast Service in the MAC layer to facilitate broadcast and multicast. Although streaming video can be transmitted as a unicast service, broadcast and multicast are more resource efficient. Using MBS, a certain area in each TDD frame can be set aside for multicast-only or broadcast-only data, as shown in Figure 1.1. The entire frame can also be designated as a download-only broadcast frame. A major task of the MBS module is to allocate video data to the multicast/broadcast data area in each frame, such that, the real time nature of the video stream is maintained. This results in stringent QoS and efficiency demands on the allocation algorithm.

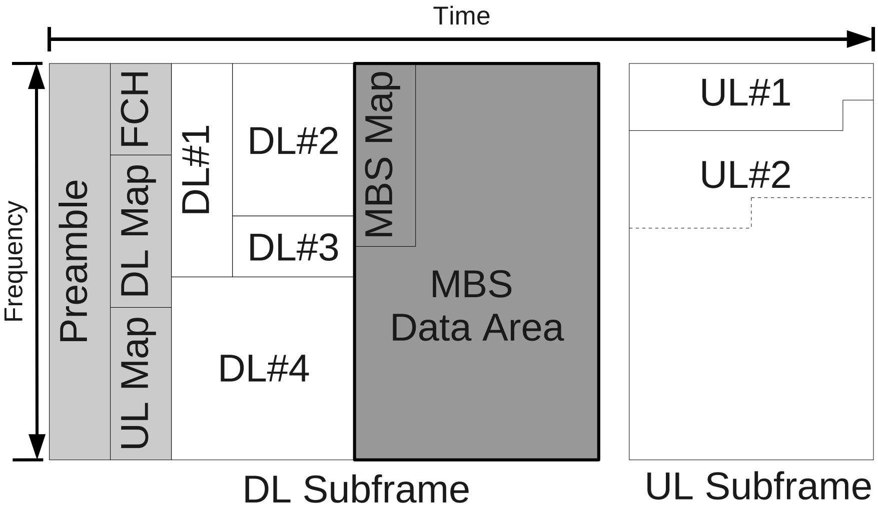


Figure 1.1: The Frame structure in WiMAX.

1.2 Problem Statement and Thesis Contributions

Our goal is to enhance a mobile user's experience of viewing video streams. This thesis studies two problems related to this goal: delivering high quality scalable video streams over bandwidth-limited channels to buffer-limited receivers and delivering video streams to the receiver in bursts to increase receiver energy efficiency.

1.2.1 Problem I: Scalable Video Streaming

We consider a scenario where a number of scalable video streams are available at a WiMAX base station. Each stream is to be broadcast using MBS to a group of mobile subscribers. At the WiMAX base station, the MBS module allocates a fixed-size data area in the download section of each TDD frame. All video streams are to be allocated only within this MBS data area. As per the mobile WiMAX standard, each MBS data area can transmit a different amount of data depending on the modulation scheme chosen, which is in turn selected based on the wireless channel conditions. For broadcast applications, a common modulation scheme is selected for a group of subscribers. Thus, each MBS area transmits a fixed amount of data, in effect, creating a fixed bandwidth broadcast channel.

We consider a scheduling window composed of a number of MBS data areas. Data from the video streams are to be allocated to the MBS areas in the scheduling window. Due to the variable bit rate (VBR) nature of the video streams, the aggregate data rates may exceed the broadcast channel capacity. Hence, in each scheduling window, we need to decide which layers to send for each stream. We assume that the base station has enough buffer space to hold the VBR traffic for one scheduling window. This way, the data rates can be assumed to be constant during a scheduling window, but they vary across scheduling windows. Since the bit rates and the receiver buffer states change in each scheduling window, the allocation has to be computed for every scheduling window. We also assume that all subscribers served by the base station have a fixed amount of buffer which is used to temporarily store the incoming video data before playing it out. Thus the optimal substream selection problem we need to solve can be stated as follows.

Problem 1. *[Optimal Substream Selection Problem] Select the optimal subset of layers from each scalable stream to broadcast over a WiMAX network such that: (1) the total data transmitted in a scheduling window does not exceed the window capacity, (2) the average quality of all selected substreams is maximized, and (3) the subscriber playout buffer does not suffer from overflow or underflow instance.*

1.2.2 Problem II: Energy Efficient Video Streaming

We further consider a scenario where the stream data transmission can be scheduled to provide better energy efficiency at mobile subscribers. Significant research studies have been dedicated for power management at mobile subscribers utilizing the sleep mode feature. When the device is idle, sleep mode will be activated so as to minimize the power consumption by using the minimal power to maintain the running system. However, frequent switching from sleep mode to normal mode can result in excessive power consumption. If a mobile station switches back and forth regardless of the amount of data to be received, unnecessary energy could be wasted. This effect is more severe when the mobile subscriber is watching a streaming video because video decoding and screen lighting already consume a lot of power. We propose a novel scheme to reduce the power consumption by minimizing the switching frequency of the receiver while still maintaining Quality of Service (QoS) requirements for streaming multimedia.

We consider the *Average Energy Efficiency (AEE)* metric which is defined as the ratio of energy consumption due to data transfer to the total energy consumption by a receiver.

Thus we have the following energy efficient scheduling problem:

Problem 2. *[Energy Efficient Scheduling Problem] Find a schedule for transmitting a set of video streams such that: (1) the average energy efficiency is maximized, and (2) the subscriber playout buffer does not suffer from overflow or underflow instance.*

1.2.3 Thesis Contributions

In this thesis we describe and solve two problems related to scalable video transmission over WiMAX networks. The main contributions of this thesis can be summarized as the following [34, 33].

- We formulate and solve the substream selection problem to maximize the video quality (Problem 1), which arises when multiple scalable video streams are broadcast to mobile receivers with limited resources. We show that this problem is NP-Complete, and we design a polynomial time approximation algorithm to solve it. We prove that the solutions computed by our algorithm are always within a small constant factor from the optimal solutions [34].
- We extend our algorithm to reduce the energy consumption of mobile receiver, which is described in Problem 2. This is done by transmitting the selected substreams in bursts, which allows mobile receivers to turn off their wireless interfaces to save energy. We show how our algorithm constructs burst transmission schedules that reduce energy consumption without sacrificing the video quality [33].
- Using extensive simulation and mathematical analysis we show that the proposed algorithm: (i) is efficient in terms of execution time, (ii) achieves high radio resource utilization, (iii) maximizes the received video quality, and (iv) minimizes the energy consumption for mobile receivers.

1.3 Thesis Organization

The rest of the thesis is organized as follows. We conduct an analysis and comparison of previous video streaming approaches over WiMAX in Chapter 2. In Chapter 3, we present, analyze, and evaluate our proposed scheme for streaming of scalable video streams over WiMAX networks. In Chapter 4, we present our solution to minimize the energy consumption at the receiver. In Chapter 5, we describe our simulation setup and results. Finally, we conclude this thesis and highlight future research directions in Chapter 6.

Chapter 2

Background and Related Work

In this chapter, we provide background about mobile video multicast techniques over broadband wireless access networks. We describe the network architecture of a typical video multicast service deployment over WiMAX networks. We describe the current mechanisms for streaming scalable video over WiMAX networks and current energy efficient scheduling algorithms.

2.1 Background

2.1.1 Video Multicast Standards

Conventional cellular wireless networks generally deliver data through unicast transmission and are not efficient for high volume multimedia delivery. Therefore, the wireless telephony companies consortium 3GPP (3rd Generation Partnership Project) [3] has proposed a *Multimedia Broadcast/Multicast Service (MBMS)* to deliver multimedia services over wireless cellular networks. MBMS includes specifications for a Mobile TV service over radio access networks like UMTS (Universal Mobile Telecommunications System) [2]. The specification declares features like support for electronic service guide (ESG), transparent handover and bounded maximum channel switching delays of 2 seconds. The unidirectional data rate for TV streaming is specified as 384kbps [3]. There can be only one broadcast session active at a time and there may be several such sessions in succession. The different sessions can have different quality of service. The main advantage of MBMS is the robust mobility performance of the cellular wireless network and that it can utilize some of the already existing GSM (Global System for Mobile) [15] infrastructure for its services. However, the low bandwidth is a limiting factor for multimedia rich applications. In comparison, WiMAX offers most of the facilities of cellular wireless networks while allowing significantly higher bandwidth. The MBS layer in WiMAX does not mention any application specific guidelines and is open to adaptation.

The broadcast industry has come up with service-specific broadcast network solutions like Digital Video Broadcasting - Handheld (DVB-H) [6] and MediaFLO [27]. In the DVB-H physical layer, data frames are transmitted in a time-division multiplexed manner with high data rate bursts of short duration followed by relatively long off times. This technique, known as time-slicing, saves energy at the receiver circuit by switching off the RF receiver during the idle period. Data bandwidth is a differentiating factor between DVB-H and WiMAX. Current WiMAX networks support 10Mbps [41] downlink bandwidth with achievable bandwidths of up to 40Mbps [13]. DVB-H deployments currently attain downlink bandwidth of around 7Mbps [29] with possibilities of increase up to 19Mbps [7] in the future. As the demand for wireless bandwidth grows, this higher bandwidth promises an excellent opportunity for WiMAX to succeed as a multimedia broadcasting technology. A major shortcoming of DVB-H is that it is a unidirectional broadcast-only standard and hence, by design, cannot support interactive multimedia content. Separate standards like IP Datacast [8] is required to add a feedback path through existing UMTS (Universal Mobile Telecommunications System) [2] networks for overcoming this. WiMAX is a bidirectional technology and can support interactive applications by itself. The WiMAX energy conservation scheme is based on the sleep mode operation for which the base station must inform the mobile subscriber about the sleep mode information in a header field. Using arbitrary sleep intervals, we can implement a mobile TV service over WiMAX that can achieve high energy savings for the mobile subscriber stations.

2.1.2 A Video Streaming System over WiMAX

In the rest of this section we describe the architecture of a video streaming system over WiMAX. We start with a system with four entities as displayed in Figure 2.1. The first entity is the content source. It can be a video content source of any type. The different types may include national TV broadcasters, local broadcasters, Internet TV operations and any other future video broadcast service. The content is assumed to be variable bit rate videos encoded using scalable coders. Generally, scalable video streams are implemented

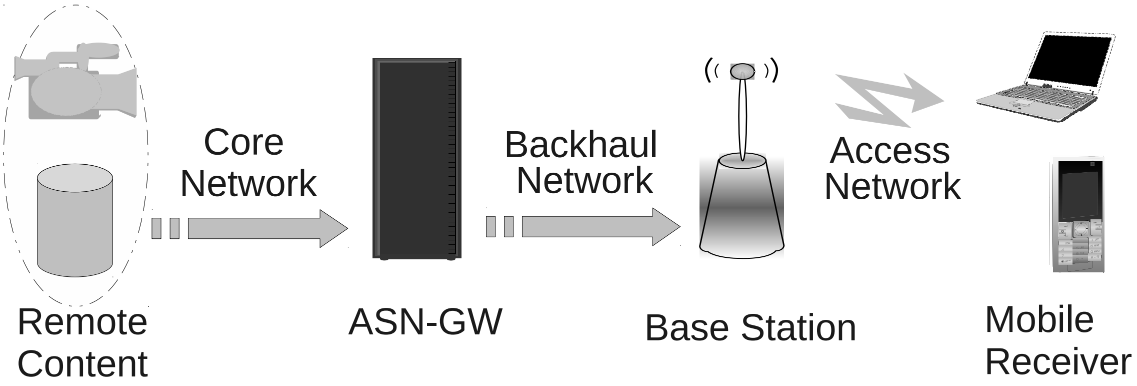


Figure 2.1: Mobile TV over WiMAX : System Architecture.

either using multiple description coding or layered coding. In multiple description coding, scalable layers are independent of each other and any subset of the layers can be selected for transmission. In contrast, the layered coding has dependency among the layers such that, if we select a layer, it is mandatory to select all layers below it. In recent times, layered coding with the standard H.264 AVC/SVC has become popular. So we chose this scheme for our scalable video representation. The remote content is delivered to the WiMAX network through the long distance high capacity content distribution core network. The encoding of the content is done at the head end while the authentication, authorization and accounting operations are performed at the core network. We assume that sufficient capacity is available in the core network to efficiently carry the video content from the content provider to the WiMAX operator's Access Service Network (ASN).

The next entity in Figure 2.1 refers to the WiMAX access network comprised of ASN-Gateways and links to base stations. For maximum utilization of resources, we consider stratifying the service area along the lines of [14] such that high capacity optical or copper wires are used for connecting the ASN Gateway to the base station in medium to low user-density areas. The high-density area wireless backhaul links are considered for speedy deployment and reduced costs. We also assume that the base station operates in a single frequency network (SFN) mode for easy handovers and better spectral efficiency. This means all connections between the ASN Gateway and the base stations are managed by the SFN adapter for synchronization. The SFN synchronization can be done with GPS (Global Positioning System) timers or using Network Time Protocol (NTP) over IP. Since the ASN Gateway is the entry point to the WiMAX network the centralized operations of the multimedia plane should be implemented there.

Electronic Service Guide (ESG) is a service that conveys information about the content of the TV channels. ESG can be obtained from an external ESG server. Other operations at the ASN-Gateway may include advertisement insertion, location based services, viewer poll and other interactivity operation management. Together with other enablers like subscriber management these services can provide the platform for an interactive and multimedia rich experience. The ESG and other push service data is carried to the base station through data streams separate from the video streams.

The third entity in Figure 2.1 is comprised of the base stations and the last mile wireless network to receiver devices. In this thesis we concentrate on the problem of managing broadcast transmissions at a base station. In our architecture, multiple video streams carrying the content and data streams containing the service information reach the base station through high speed backhaul links. The base station also receives a unique connection ID for each TV channel from the ASN Gateway. In each base station the TV channel video streams are stored in individual channel queues. After the data and video are packetized into channel specific MAC PDUs (Medium Access Control Protocol Data Unit) and security mechanisms applied they are handed over to the MBS manager. The MBS manager packs the MAC PDUs in the MBS block of the TDD (Time Division Duplex) frame and it is transmitted over OFDMA (Orthogonal Frequency-Division Multiple Access). For efficiently utilizing the MBS space, the MBS constructs a number of MBS areas at a time. This is our scheduling window. The scheduling window is similar to the superframe concept introduced by Cohen, Katzir and Rizzi [5]. A scheduling window consists of two types of TDD frames. The first few frames contain the ESG and other service data as well as the sleep intervals.

The data for a particular video channel is transmitted in a time-sliced manner to save energy. Energy conservation is achieved by setting the receiver device to sleep mode when it is not receiving any relevant frames. The start time and duration of each such sleep period is stored in the first frames. We assume that the overhead data area in each scheduling window is upper bounded by a fixed number of MBS blocks. In the course of scheduling the MAC PDUs the MBS controller must solve two problems. The first problem is to group the MAC PDUs to construct the MBS area of TDD Frames. In our design we only consider the case where in each frame the entire download area is denoted as an MBS broadcast zone. The second problem is to sequence the newly created MBS zones such that they maintain a real time service for all streams.

The objective of the scheduling algorithm is to maximize the throughput while satisfying various receiver constraints. First of all, the schedule should be such that none of the receiver buffers either overflow or underflow. The schedule should also conserve energy at the receiver device. Channel switching delay is an important metric in user experience of Mobile TV like applications. A longer channel switching delay is undesirable, so it is important that the delay is bounded within the acceptable time limits. Once we have arrived at a schedule we compute the sleep intervals for different channels and this information is then embedded in the first frames of the scheduling window in the form of a specific PDU type. These PDUs are called MOB_SLP_RSP type PDUs. At the start of transmission each mobile subscriber receives the first frame and reads the MOB_SLP_RSP PDU to know its sleep and active intervals. It then accordingly switches its receiver on or off to receive only the relevant frames. This procedure is continuously repeated to keep the broadcast service running.

The fourth and final entity of the architecture consists of the receiver devices. We mainly consider the mobile devices running on battery power. Mobile subscribers need to register with the base station for their service. Upon successful registration the subscriber looks for a frame that contains ESG information. Once the user selects a channel, the corresponding sleep interval information can be found from the data frames in the scheduling window. Also the subscriber needs to implement a mechanism to handle scalable video decoding.

2.1.3 Overview of WiMAX Sleep Mode

Reducing the power consumption of mobile subscriber (MS) devices is one of the major challenges in wireless multimedia systems. Keeping the wireless interface always active contributes significantly to the overall power consumption of mobile devices. This power consumption can be reduced by controlling the duration for which the wireless interface needs to be active. The WiMAX standard defines idle and sleep modes to reduce MS power consumption by temporarily suspending the operation of the wireless interface. In the idle mode, the MS periodically receives the down link broadcast data from base station. In the sleep mode, the MS turns itself off for a certain period of time to save power. This time is decided by the MS and the base station. Based on how the sleep period is determined, three power saving classes are defined in WiMAX standard as described in Table 2.1. We utilize the Class 3 sleeping mode since this allows us to change the sleep window size in an adaptable way. We algorithmically decide the sleep window size based on the multimedia

Table 2.1: Sleep modes defined in the WiMAX standard [19].

Type	Description
Class 1	The sleep period is recurring and the sleep window size is exponentially increased from minimum to maximum.
Class 2	The sleep period is recurring and the sleep window is of fixed length.
Class 3	There is only a one-time sleep window of a predefined duration.

data transmission requirements.

2.2 Related Work

In this section, we first summarize previous works on scalable video transmission over WiMAX networks. Then, we describe related works on energy efficient video transmission in WiMAX networks.

2.2.1 Video Streaming over WiMAX Networks

Wang et al. [39] discuss an architecture for video broadcasting in a multi-base-station WiMAX system. Their work focuses on coverage and spectral efficiency issues and considers only temporal video scalability. Cohen et al. [5] combine a group of TDD (Time Division Duplex) frames together into a super-frame. They describe a cost-based scheme where a cost function is associated with each user-channel pair. Three user interaction models are considered: (i) the user can be statically hooked to a channel, (ii) the user can choose to listen to a channel, or (iii) the user channel association can keep changing based on the transmission medium conditions. The work in [5] does not consider the delay requirements which are central to video streaming. Hosein [16] describes the frame allocation problem for broadcasting variable bit rate video over WiMAX, but does not consider scalable video content.

Jiang et al. [21] propose a scheme to transmit scalable video streams in which two layers of each video are transmitted separately. The base layer is transmitted as one stream over a reliable channel while the enhancement layer is transmitted as a different stream over a less reliable channel. Conceptually, this work implements a rate adaptive multiple description coding. However, it describes only one stream and it does not address the resource management problem arising in multi stream transmission scenarios. Reguant et al. [31] consider splitting a video stream into two streams and transmitting them over two different broadcast networks. The first stream is transmitted over the DVB-H network at all times while the second stream is transmitted over the WiMAX network most of the time. If the user wants to use some other non-video application in parallel, the stream going through WiMAX is degraded to accommodate that application. This ensures a minimum video quality at all times while maintaining the flexibility of using other applications. While this approach has its benefits, it is not very attractive from a deployment point of view since the service provider has to install and manage the infrastructure for two different kinds of networks. Also the solutions described in both [21] and [31] evaluate the performance of video streaming as an application along with other WiMAX applications and do not utilize MBS. In contrast, our approach considers a multimedia-intensive system with extensive use of MBS.

2.2.2 Receiver Energy Efficiency in WiMAX Networks

Power aware scheduling schemes for general WiMAX networks have been proposed [32, 22, 35, 23]. For example, Seo et al. [32] propose a scheme that utilizes subscriber information available at the base station. They describe a sleep interval algorithm based on queuing analysis of the packet arrival rate of subscribers. In contrast, Kim et al. [22] describe a sleep interval scheme based on the remaining battery life of a mobile subscriber device. Shi et al. [35] propose a burst scheduling algorithm for energy minimization on a per subscriber basis for unicast data. The algorithm arranges the mobile subscribers in ascending order based on the ratio of the current data arrival rate to the required data rate. If the current rate is significantly higher than the required rate, the mobile subscriber can go to sleep for some interval. After computing the sleep intervals for all mobile subscribers the bursts are scheduled in a longest interval first manner. After transmission of each burst, the algorithm checks to ensure that the data requirements of all mobile subscribers are being satisfied. The work in [35] is designed for unicast streaming of video and does not consider the multicast/broadcast service. Also the algorithm requires maintaining the state information of all mobile subscribers served by a base station. Liao and Lee [23] suggest a scheduling scheme where the uni-cast data is clustered around the multicast data bursts for increased energy efficiency. They assume that the burst length and positions for a particular stream is the same in all super-frames. Then they present an enhancement to the longest-virtual-buffer-first scheduling algorithm proposed by Shi et. al. [35] by clustering the unicast data around the multicast data bursts.

Chapter 3

Proposed Scalable Video Streaming Scheme

In this chapter, we solve the optimal substream selection problem. We formulate this problem into an optimization problem and show its hardness. We propose an approximation algorithm to solve this problem, which computes solutions close to the optimal in a time-efficient manner. For quick reference, we list all symbols used in the formulation in Table 3.1.

3.1 Problem Statement

We consider a scenario where a number of scalable video streams are available at a WiMAX base station. Each stream is to be broadcast using MBS to a group of mobile subscribers. At the WiMAX base station, the MBS module allocates a fixed-size data area in the download section of each TDD frame. All video streams are to be allocated only within this MBS data area. As per the mobile WiMAX standard, each MBS data area can transmit a different amount of data depending on the modulation scheme chosen, which is in turn selected based on the wireless channel conditions. For broadcast applications, a common modulation scheme is selected for a group of subscribers. Thus, each MBS area transmits a fixed amount of data, in effect, creating a fixed bandwidth broadcast channel. We consider a scheduling window composed of a number of MBS data areas. Data from the video streams are to be allocated to the MBS areas in the scheduling window. Due to the variable bit rate (VBR) nature of the video streams, the aggregate data rates may exceed the broadcast channel capacity. Hence, in each scheduling window, we need to decide which layers to send for each stream. We assume that the base station has enough buffer space to hold the VBR traffic for one scheduling window. Then, the data rates can be assumed to be constant during a scheduling window, but varying across scheduling windows. Since the bit rates and the receiver buffer states change in each scheduling window, the allocation has to be computed for every scheduling window. We also assume that all subscribers served by the base station have a fixed amount of buffer which is used to temporarily store the incoming video data before playing it out. Thus the optimal substream selection problem we need to solve can be stated as follows.

Problem 1. *[Optimal Substream Selection Problem] Select the optimal subset of layers from each scalable stream to broadcast over a WiMAX network such that: (1) the total data transmitted in a scheduling window does not exceed the window capacity, (2) the average quality of all selected substreams is maximized, and (3) the subscriber playout buffer does not overflow or underflow.*

3.1.1 Problem Hardness

Let us assume that for a given radio modulation scheme, the MBS data area in each frame can accommodate F amount of data and the TDD frame takes τ time to be transmitted. Let the scheduling window consist of P such frames. Then, the maximum amount of data that can be transmitted within the scheduling window is given as $C = PF$.

We have S scalable video streams. Each scalable stream s, $1 \leq s \leq S$, has a number of layers. The number of layers can be different for each stream. We define L as an upper bound on the number of layers any stream can have. Therefore, for each stream we have L substreams to choose from, where a substream l includes layer l and all layers below it. Let the data rates and quality values for selecting substream l of stream s be r_{sl} and q_{sl} respectively. Here r_{11} denotes the data rate of the base layer of the first stream. Thus, we have the problem of choosing the substreams such that the average quality across the video streams is maximized subject to the following constraints. The first constraint is that the total data to be transmitted must fit into the MBS area in the current scheduling window. The second constraint is that the buffers at the subscribers must not run out of data anytime during the scheduling window, and the third constraint is that the base layer of each stream must be transmitted to guarantee a basic service level agreement.

Theorem 1. *[Hardness]* The Optimal Substream Selection Problem is NP-Complete.

Proof. First, we consider a relaxed version of the problem with no buffer overflow or underflow constraints. Thus, we are left with the problem of selecting the substreams such that the average quality is maximized. We assume that in each scheduling window at least all the base layer streams have to be transmitted due to service level agreement. Thus, we further modify the problem by eliminating the base layer constraints, which can be trivially done, by reducing the scheduling window capacity by the sum of data rates of all base layers. Therefore, the modified data capacity can be given as $C' = C - \sum\limits_{s \in S} r_{s1}$.

Now we are left with the problem of deciding which substreams to choose from each stream. We show that this problem is equivalent to the NP-Complete 0-1 Multiple Choice Knapsack Problem (0-1-MCKP) [10], which is defined as follows. There are M classes $N_1, \ldots, N_M$ of items to pack in a knapsack of capacity W. Each item (i, j), where $i \in M, j \in N_i$, has a profit $p(i, j)$ and a weight $w(i, j)$. The problem is to choose at most one item from each class such that the profit sum is maximized without having the total weight exceed W. We reduce the 0-1-MCKP problem to the Optimal Substream Selection problem in polynomial time as follows. The data rate of a particular substream corresponds to the item weight and the quality value of the same substream correspond to the utility of the item. Each stream corresponds to an item-class and the scheduling window capacity corresponds to the knapsack capacity. The constraint of choosing only one substream per stream corresponds to the constraint of choosing only one item per item-class. Thus, we have an MCKP instance with S classes, $L - 1$ items per class and a knapsack capacity of C'. This means that an efficient solution for the simplified Optimal Substream Selection Problem could be employed to efficiently solve the NP-Complete 0-1-MCKP problem. In other words, the substream selection problem is NP-Hard. In addition, clearly a solution for the simplified Optimal Substream Selection Problem can be verified in polynomial time. Thus the simplified Optimal Substream Selection Problem is NP-Complete. Consequently, the more general Optimal Substream Selection Problem subject to buffer overflow and underflow constraints is also NP-Complete. □

3.1.2 Mathematical Formulation

We assume that all subscribers have B amount of buffer available for the video streaming application and the data rate and quality values for all substreams of each stream are known

Table 3.1: List of symbols used in this thesis.

Symbol	**Description**
S	Number of streams
L	Number of layers
q_{sl}	PSNR of substream sl
r_{sl}	Data rate of substream sl
b_{sl}	Number of frame sized blocks of substream sl
n_{sl}	Number of frame bursts for substream sl
t_{sl}^k	Start of burst k of substream sl
w_{sl}^k	Width of burst k of substream sl
τ	Duration of a TDD frame
F	Capacity of MBS data in a TDD frame
P	Number of frames in scheduling window
C	Data capacity of scheduling window
B	Buffer size at the receiver
u_s	Initial buffer level for stream s
E_a	Receiver energy consumption in active state
E_w	Energy consumption for wake up from sleep state
ϵ	Approximation Parameter

ahead of the scheduling window. This information can be obtained as a separate meta data for each stream. Alternatively, if the scalable video is encoded using H.264/SVC [40] and the base station is media-aware, this information can be obtained directly from the encoded video stream itself using the Supplementary Enhancement Information (SEI) messages. Let the data rate values of substreams of stream s be $\{r_{s1}, r_{s2}, \ldots, r_{sL}\}$ and the corresponding quality values be $\{q_{s1}, q_{s2}, \ldots, q_{sL}\}$. Each scheduling window is of duration τP. If substream l of stream s is selected, the amount of data to be transmitted during a scheduling window can be given as $\tau P r_{sl}$. Let binary variables x_{sl} take the value 1 if substream l of stream s is selected for transmission in the current scheduling window and 0 otherwise. For a substream, we define a burst as a consecutive set of MBS data areas allocated to the substream in the scheduling window. For any schedule, let n_{sl} be the number of bursts for substream l of stream s. Let t_{sl}^k define the starting frame number and w_{sl}^k define the number of MBS data areas in burst k for substream l of stream s.

The solution procedure for the optimum substream selection problem should generate a list $< l, n, < t_{sl}^1, w_{sl}^1 >, \ldots, < t_{sl}^n, w_{sl}^n >>$ for each stream. In the list, l denotes the selected substream, n denotes the number of bursts required for transmitting substream l, and $< t_{sl}^k, w_{sl}^k >$ denote the starting point and width of burst k, respectively. For a subscriber receiving channel s, let the buffer level at the beginning of the scheduling window be u_s. We need to ensure that all data received during a scheduling window is also consumed in the same window. In other words $F\sum_{k=1}^{n_{sl}} w_{sl}^k = \tau P r_{sl}$. At the same time we need to ensure that buffer overflow and underflow do not occur. At the end of each burst, the total data received is given by $F\sum_{i=1}^{k} w_{sl}^i$. During that period the total data consumed is given by $\tau(t_{sl}^k + w_{sl}^k)r_{sl}$. Now in order to avoid underflow, the difference of these two terms must be greater than zero for all bursts. Similarly, the overflow conditions can be applied by constraining the difference to be never greater than B. Our objective is to maximize the average video quality over all the streams. We use the Peak Signal to Noise Ratio (PSNR) values of the streams to denote quality and take an arithmetic average of the PSNRs of the selected streams to denote the average video quality. Let us assume that the data to be transmitted for each substream can be divided into b_{sl} number of F sized data blocks. In other words, $r_{sl} = b_{sl}F$. Consequently, we have the following optimization problem.

$$\text{Maximize} \quad \frac{1}{S}\sum_{s=1}^{S}\sum_{l=1}^{L} x_{sl}q_{sl} \tag{P1}$$

$$\text{such that} \quad \sum_{s=1}^{S}\sum_{l=1}^{L} x_{sl}b_{sl} \quad \leq \quad P \tag{1a}$$

$$\sum_{l=1}^{L} x_{sl} \leq 1, \qquad \forall s \in S \tag{1b}$$

$$u_s + \sum_{i=1}^{k} w_{sl}^i F - \tau(t_{sl}^k + w_{sl}^k)r_{sl} \leq B, \qquad \forall s \in S, \forall l \in L \tag{1c}$$

$$u_s + \sum_{i=1}^{k-1} w_{sl}^i F - \tau(t_{sl}^{k-1} + w_{sl}^{k-1})r_{sl} > 0, \qquad \forall s \in S, \forall l \in L \tag{1d}$$

$$[t_{sl}^k \ldots t_{sl}^k + w_{sl}^k] \cap [t_{sl}^{\bar{k}} \ldots t_{sl}^{\bar{k}} + w_{sl}^{\bar{k}}] = \emptyset, \qquad \forall s \in S, \forall l \in L \tag{1e}$$

$$\sum_{k=1}^{n_{sl}} w_s^k = x_{sl}b_{sl}, \qquad \forall s \in S, \forall l \in L \tag{1f}$$

In the above formulation the constraint (1a) makes sure that the selected substreams can be transmitted within the broadcast bandwidth. Constraint (1b) ensures that at most one substream is selected for each stream. Constraints (1c) and (1d) represent the buffer overflow and underflow constraints, respectively. Constraint (1e) implies that no two bursts of data blocks should be allocated to the MBS area of the same TDD Frame. Here the $\bar{k}$ represents any burst other than burst k and operator $[\ldots]$ denotes integer interval. This constraint is

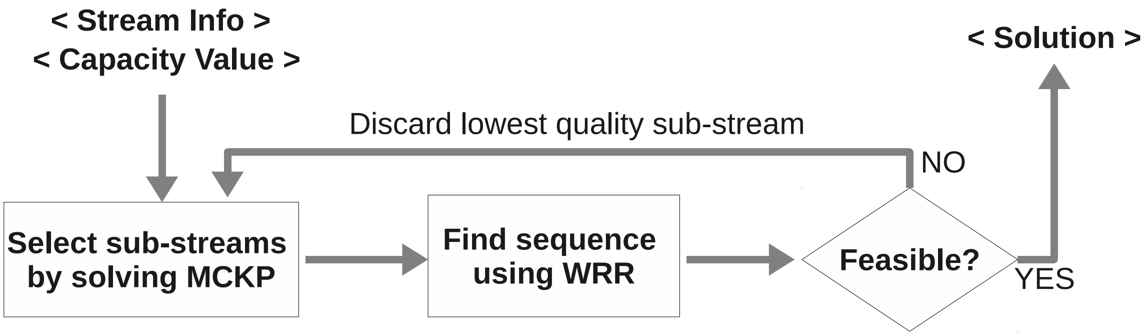


Figure 3.1: High-level diagram of the Substream Scheduling Algorithm (SSA).

required because the streams are transmitted over a time-shared, multiple-access wireless channel where only one burst can be transmitted at a time. Constraint (1f) implies that if a layer is selected, then all the data blocks corresponding to the layer must be allocated in the schedule.

3.2 Proposed Solution

3.2.1 Overview of the Proposed Algorithm

The proposed algorithm is called *Substream Selection Algorithm* and is denoted by SSA. The high level idea of the algorithm is depicted in Figure 3.1 and is described as follows. We first find a set of near optimal substreams given the data capacity of a scheduling window. Then, we allocate them to the MBS areas in the frames of the scheduling window. If no feasible allocation is found, we reduce the problem instance by discarding the substream with the lowest quality among all substreams. We solve the optimal substream selection problem again for the reduced set of substreams. This cycle is repeated until either a feasible solution is found, or none of the substreams is selected. Once a solution is found the frame allocation is done in a modified weighted round robin manner.

As shown in Theorem 1, the problem of selecting the optimal substreams is equivalent to solving the 0-1 Multiple Choice Knapsack Problem. This problem has been studied in the mathematical programming community and several near optimal solution schemes exist. The reader is referred to the survey by Lin [24] for a summary of the main results. Dynamic

Substream Selection Algorithm (SSA)

Input: Substreams, MBS capacity, Frame duration, Scheduling window size
Output: Data burst allocation in the MBS area of the current scheduling window

```
 1. For each enhancement layer $i$ across all streams do
 2.         Compute $\rho_i = r_{sl} - r_{sl-1}$ and $\phi_i = (q_{sl} - q_{sl-1})$
 3. Select $k$ largest $\phi_i/\rho_i$ such that $\sum_{i \in k} \rho_i < PF - \sum r_{s1}$
 4. Determine lower bound $Q_0 = \sum \phi_i + \sum q_{s1}$
 5. Compute scale factor $K = \epsilon Q_0 / S$
 6. Scale the quality values such that $q'_{sl} = \lceil q_{sl}/K \rceil$
 7. For $q = 1$ to $2Q_0$ do
 8.       For $s = 1$ to $S$ do
 9.             If $s$ is 1, Compute $R(s,q)$ using equation (2a)
10.             Else, Compute $R(s,q)$ using equation (2b)
11. Backtrack table $R(s,q)$ to find the substreams $s^*$
12. Until all streams are allocated do
13.         Arrange substreams in ascending order of $B_s/r_s$
14.         Allocate $\sigma_s = \min\{B_s/\tau r_s\}$ frames to stream $s$
15.         Update $B_s = B_s + \sigma_s * F - \sigma_s \tau r_s$
16. If no valid allocation found do
17.         Find substream $(\hat{l}, \hat{s})$ such that $q_{\hat{s}\hat{l}} = \min_{s \in S, l \in L} \{q_{sl}\}$
18.         Discard substream $(\hat{l}, \hat{s})$
19.         Go to step 3
```

Figure 3.2: The proposed Substream Selection Algorithm.

programming is one of the techniques used for designing approximation algorithms for the 0-1 Multiple Choice Knapsack Problem. However, dynamic programming solutions are often memory intensive and may involve large constants. In our algorithm we derive both an upper bound and lower bound on the value of the solution. These bounds significantly reduce the solution search space.

3.2.2 Details of the Proposed Algorithm

Figure 3.2 summarizes the pseudo-code of the SSA algorithm. There are three main steps of the algorithm: (i) Finding an approximate solution to the substream selection problem, (ii) Allocating the selected substreams to the MBS data areas of the scheduling window, and (iii) Validating the schedule to confirm that there are no buffer underflow or overflow conditions. We describe each step in the following.

Step 1: Approximate Substream Selection

In a naive dynamic programming solution we construct a table of all possible data rates for the given streams (i.e., $1 \ldots \sum r_{SL}$) and their resulting quality values. We note that multiple quality values can result for a single aggregate data rate value depending on the composition of the substreams selected. Then, we search for the highest quality entry in the table such that the aggregate data rate is less than the scheduling window capacity. In our proposed algorithm, we first derive bounds on the solution value which will reduce the size of the search space. Then, we construct a dynamic programming table for all quality values within the bounds and find the solution substreams using backtracking.

Bounding the Optimum Solution Value: In this step we consider the relaxed Optimal Substream Selection Problem where the buffer constraints are relaxed and we only need to find the optimal substreams. We call it the Relaxed Substream Selection problem or RSS. The value of the optimal solution to the linear programming relaxation of RSS is an upper bound on the value of optimal solution of the RSS. Now we derive a lower bound on the optimal solution of the RSS problem as follows. A solution x_c^* to the linear relaxation has the following two properties: (1) x_c^* contains at most two fractional values and (2) when there are two fractional values in x_c^*, they belong to substreams of the same stream. For a proof of these properties the reader is referred to [36]. Let z_c^* be the value of the objective function corresponding to x_c^*. Let Q_0 be the maximum of (a) the objective function value when both the fractional values are dropped from the solution and (b) maximum of the quality values of the fractional variables. If the optimal solution for the integer problem is Q^*, it is evident that $Q_0 \leq Q^*$. From the properties of x_c^* it is evident that at most two variables are dropped. Since at most two variables are dropped, z_c^* can be bounded as $z_c^* \leq 2Q_0$. Also, since the solution obtained by the linear relaxation must be greater than or equal to the solution obtained by the integer program, we have an upper bound on the optimum integer solution as $Q_0 \leq Q^* \leq z_c^* \leq 2Q_0$. We note that although the bound is obtained from linear programming theory, we do not require an LP solver to calculate Q_0. Q_0 can be calculated using the median finding algorithms in linear time [42].

Recursive Table Generation: Now that we know the bounds of the optimal solution value we define a dynamic programming formulation as follows. For all streams

$s \in \{1, \ldots, S\}$ and all quality values $q \in, \{0 \ldots, 2Q_0\}$, we define $V(s, q)$ as the set of substreams from streams $1, \ldots, s$ such that no two substreams are selected from the same stream and the total quality of the selected substreams is q. If for a quality value q the *at most one substream per stream* constraint is violated we set the corresponding sum of weights to infinity. Let $R(s, q)$ denote the sum of data rates selected in $V(s, q)$. We assume that the sum of data rates to produce zero quality is zero, i.e., $R(s, 0) = 0$. Also, for the first stream, the data rate values can be computed easily as just the data rate of the substream, or the minimum of the data rates if more than one substream has the same quality, i.e., $R(1, q) = \min_l \{r_{sl}\}$ where $q_{sl} = q$. In mathematical terms, the first stream data rates can be expressed as in equation (2a). The data rates for the other quality values and other streams can be computed by the recursive definition described in (2b) and the optimum quality can be expressed by equation (2c).

$$R(1,q) = \begin{cases} \min\limits_{l}\{r_{sl}\}, & \text{where } l \in L \text{ and } q_{sl} = q, \\ \infty, & \text{otherwise.} \end{cases} \tag{2a}$$

$$R(s,q) = \begin{cases} \min\{R(s-1,q), \min\limits_{l \in L}\{r_{sl} & \\ \quad +R(s-1, q-q_{sl})\}\}, & \text{when } q_{sl} \leq q, \\ R(s-1,q), & \text{otherwise.} \end{cases} \tag{2b}$$

$$Q^* = \max\{q | R(s,q) \leq PF\}. \tag{2c}$$

However, the size of the table can still be very large as it is bounded only by Q_0. Therefore we select a scaling factor $K = \frac{\epsilon Q_0}{S}$, and scale down the quality values to $q'_{sl} = \frac{q_{sl}}{K}$. This operation considerably reduces the table size while admitting only a small error factor. We bound the quality degradation due to scaling in our mathematical analysis in Theorem 3.

Backtracking: Once we have computed the dynamic programming table, the solution quality value is obtained by a simple scanning of the table as in equation (2c). The solution substream vectors are found using a backtracking mechanism as follows. While constructing the recursive table, we store the composition of substreams leading to the data rates $R(s, q)$ as a list for each table cell. The solution substream vector is found using the additional information by backtracking from the cell containing the solution quality value.

Step 2: Data allocation

Once the substreams are selected, it remains to allocate them to the MBS data area such that the subscriber's playback buffers do not overflow or underflow. We use a modified version of the *weighted round robin* algorithm to allocate data to frames. The weighted round robin has been used for scheduling constant bit rate traffic before [17]. However, for a variable bit rate stream the stream priorities are not static. We derive the priority of a stream based on its buffer level at the subscriber. At the beginning of the scheduling window, for a stream s let the data rate of the selected substream be r_s and the buffer level be B_s. Then stream s is assigned priority B_s/r_s. A lower value of B_s/r_s denotes higher priority. We also need to allocate the number of frames to a stream in the current round, that is, the length of the burst. The burst length is chosen such that none of the other streams suffer from starvation, nor does it cause overflow or underflow at the receiver buffer. For a stream s the length of the burst is given by $\min\{B_s/\tau r_s\}$.

Step 3: Buffer State Validation

After the schedule is constructed, we check if any buffer constraint is violated. This can be easily determined by verifying the buffer overflow and underflow constraints described in equations (1c) and (1d). If the buffer constraints are violated, the current substreams cannot be allocated within the current scheduling window. Hence, we reduce the problem size and re-compute substreams. The problem size is reduced by discarding the substream with minimum quality value among all substreams. This process is repeated until a feasible solution is found or none of the substreams is selected. We note that even though the scheduler is located at the base station, it is aware of the subscriber buffer size and stream data rates. From this information it can calculate the change in buffer states for a given schedule without any involvement from the subscribers.

3.2.3 Correctness and Performance Analysis

We start by describing the concept of non-dominated solutions for our problem. Let l and l' be two substreams of a given stream s. Substream l is said to be dominated by substream l' if including l' in the solution always leads to better quality than including substream l. For example, let r_{sl}, r'_{sl} be the data rates and q_{sl}, q'_{sl} be the quality values of substreams l and l'. If $r_{sl} > r'_{sl}$ and $q_{sl} < q'_{sl}$ then l is dominated by l'.

We first show in Lemma 1 that the data rate and quality values of substreams of a scalable stream constitute a non-dominated set. In Lemma 2, we prove the correctness of the recursive formulation described in equations (2a)-(2b). Using Lemma 1 and Lemma 2 we prove the correctness of SSA.

Lemma 1: Data rates and quality values of substreams extracted from scalable streams constitute non-dominated set.

Proof. Greet et al. [11] have shown that, for the H.264 PSNR scalability, when there is sufficient variability in a video, its rate-distortion characterization will be close to a quadratic function which is concave. In our problem, since the streams are variable bit rate videos and layer encoded, the data rate and quality value pair of the layers within a stream can be assumed to form a concave set. □

Therefore, Lemma 1 indicates that our problem instances are already in non-dominated form and we can easily solve the linear relaxation of the best quality substream selection problem. An efficient solution to the linear relaxation will help us to efficiently compute the final solution value. We empirically validate the assumption of rate variability and concavity. In Figure 3.3a we plot the sizes of frames of one of our test streams to show the high data rate variability. The high variability can be attributed to the different types of frames used to encode the video. The I frames are almost unchanged picture frames and contain the maximum details about the video. They are therefore of higher data size. The long spikes in the figure correspond to I frames. The other frames contain differential information between two frames and are smaller in size. In Figure 3.3b we plot the data rate and PSNR values of three test streams and validate that they indeed form a concave envelope.

Lemma 2: The recurrence relations described in equations (2a)-(2c) produce a near optimal substream selection solution.

Proof. According to Lemma 1, all instances consist of only non-dominated substreams. Thus we only need to prove the correctness of the recurrence relation. We can prove the correctness of the recursive expression by induction. The basis step where $s = 1$ is true since it will lead to the selection of the maximum quality substream such that the data rate is less than the scheduling window capacity. Now let us assume that it is also true for the case of $s - 1$ streams. For stream s the expression $R(s-1, q - q_{sl})$ retrieves the weight of the solution and updates it by adding the current data rate. Then all such data rates are compared which can result in quality q and only the minimum is chosen. Since $R(s-1, q)$ is already minimum, this results in $R(s, q)$ also being minimum for every quality value. Since only selections from the set $V(s, q)$ can have non-infinite values is ensures that only one substream per stream is selected. Thus, the solution produced by the recurrence relation produce an optimal solution for the scaled down instances of RSS problem. Since some quality values are rounded during the scaling down procedure, the solution produced by the recurrence relation is a near optimal solution for the optimal substream selection problem. □

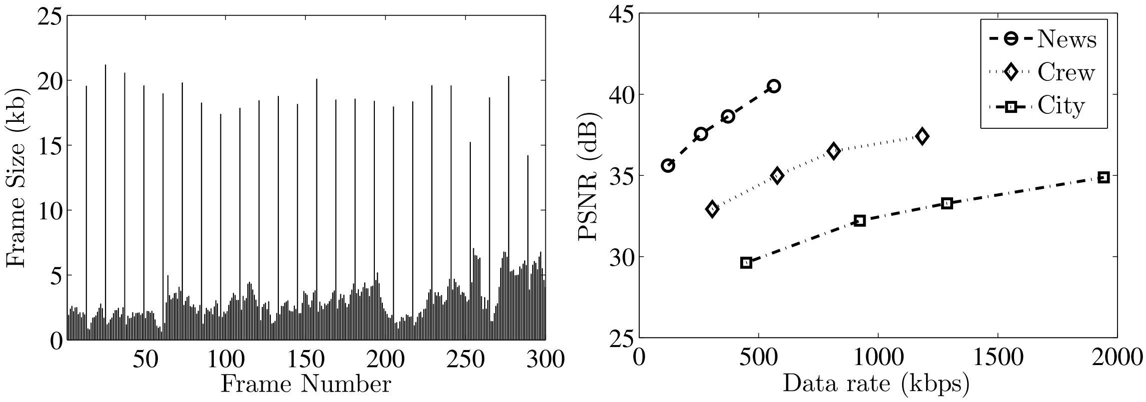
Theorem 2. *[Correctness] The Substream Selection Algorithm described in Figure 3.2 returns a valid solution for the Substream Selection Problem.*

Proof. By Lemma 2, the solution to the dynamic programming formulation selects substreams such that the average quality is close to the optimal and the total data requirement is less than the scheduling window capacity. Therefore, the capacity constraint and the *at most one substream per stream* constraint are satisfied. The round robin algorithm assigns MBS data areas to streams one frame at a time. Thus, it guarantees that no two frame bursts are assigned to the same frame. Finally, the buffer state validation step of the algorithm ensures no buffer overflow or underflow instances occur in the schedule. Hence, the SSA algorithm generates a valid solution for the substream selection problem. □

Next we analyze the approximation factor and time complexity. We define an approximation algorithm for maximization problems as the following. For every instance I of a maximization problem Π, let $\Pi(I : OPT)$ be the optimal objective function value and let $\Pi(I : ALG)$ be the objective function value obtained by the algorithm ALG. ALG is an approximation algorithm for the problem Π with an approximation factor α if for any input size, $\Pi(I : OPT)/\Pi(I : ALG) \leq \alpha$. For the problem under discussion Π corresponds to the substream selection problem.

Theorem 3. *[Approximation Factor] The Substream Selection Algorithm described in Figure 3.2 is a constant factor approximation algorithm.*

Proof. In Section 3.2.2 we derived the upper bound and lower bound of the optimal solution in terms of Q_0. For a small nonzero constant ϵ we selected a scaling factor K as $\epsilon Q_0/S$. Let us consider an instance I of problem Π which has data rate values r_{sl} and quality values



(a) Foreman sequence (b) Non-dominating property of scalable videos.

Figure 3.3: Rate-distortion characteristics of scalable videos.

q_{sl}. We obtain a scaled down instance I' from I by dividing each quality value q_{sl} by K. Therefore I' has quality values q'_{sl} which are are obtained as :

$$q'_{sl} = \lceil \frac{q_{sl}}{K} \rceil.$$

All other aspects of I and I' remain identical. Let $\Pi(I' : ALG)$ be the best solution obtained by any algorithm ALG on the scaled down instance I' and $\Pi(I' : SSA)$ be the solution obtained by our SSA algorithm. Since we rounded up the quality values during the scaling operation we have

$$\Pi(I : ALG) \leq \Pi(I' : ALG).$$

Also, since our algorithm finds the optimum solution for the scaled down problem, we have

$$\Pi(I' : ALG) \leq \Pi(I' : SSA).$$

From the above two equations we have

$$\Pi(I : ALG) \leq \Pi(I' : SSA).$$

Now, since each quality value in the solution of $\Pi(I' : SSA)$ is at most K times bigger than the quality values in solution of $\Pi(I : SSA)$, and we can have only S number of such quality values in a valid solution we have

$$\Pi(I : ALG) \leq \Pi(I' : SSA) \leq \Pi(I : SSA) + SK.$$

Replacing the value of K we have

$$\Pi(I : ALG) \leq \Pi(I : SSA) + \epsilon Q_0.$$

Now, since Q_0 is a lower bound to our solution we have

$$\Pi(I : SSA) + \epsilon Q_0 \leq \Pi(I : SSA) + \epsilon \Pi(I : SSA).$$

From the above two equations we have,

$$\Pi(I : ALG) \leq (1 + \epsilon)\Pi(I : SSA).$$

ALG can be any algorithm including OPT, the optimal algorithm. Therefore we have,

$$\frac{\Pi(I : OPT)}{\Pi(I : SSA)} \leq (1 + \epsilon).$$

In other words, the solution obtained by the SSA algorithm is always within a factor of $(1+\epsilon)$ of the optimal algorithm for every instance I of the problem Π. Therefore the SSA algorithm is a constant factor approximation algorithm with approximation factor $(1 + \epsilon)$. ☐

The parameter ϵ is a small non-negative constant passed as input to our algorithm SSA. It represents the gap between the average quality value obtained by our solution to that obtained by an optimal algorithm. The running time of our algorithm SSA is inversely proportional to the value of ϵ. This means, to find solutions very close to optimal the algorithm will incur longer execution times. For our application, solutions with a value of 0.01 for ϵ were found to be feasible. The impact of different values of ϵ is discussed in more detail in Section 5.2.3. Next, in Theorem 4, we prove time complexity results for the algorithm SSA.

Theorem 4. *[Time and Space Complexity] The SSA algorithm in Figure 3.2 has a time complexity of* $O((nS/\epsilon) + Pn \log n)$*, where* $n = O(\sum L)$ *is the total number of substreams,* L *is the maximum number of substreams within a stream,* P *is the scheduling window size, and* $\epsilon > 0$ *is a small constant. The space complexity is* $O(SQ^*)$*, where* S *is the number of streams and* Q^* *is the optimum quality value.*

Proof. The dynamic programming algorithm computes $S \times 2Q_0$ entries for constructing the $R(s, q)$ table. Computing each entry takes $O(L)$ time. Hence the table can be completely constructed in $O(L \cdot S \cdot 2Q_0)$ time or $O(nQ^*)$ time. Computation of Q_0 takes $O(n \log n)$

time, leading to a total time complexity of $O(n \log n + nQ^*)$. This is not polynomial in n since the value of Q^* may not be bounded polynomially in n. For a small constant ϵ, we selected a scale factor K and scaled quality value for each substream q'_{sl} according to equations (3a)-(3b).

$$K = \frac{\epsilon Q_0}{S} \tag{3a}$$

$$q'_{sl} = \lceil \frac{q_{sl}}{K} \rceil \tag{3b}$$

Since $Q^* \leq 2Q_0$ we have $Q^*/K \leq 2S/\epsilon$. Thus the table computation can now be computed in $O(nS/\epsilon)$ time. The round robin allocation takes $O(n \log n)$ time for sorting the buffer levels in each round. The number of rounds depends on the size of the scheduling window P. Hence the total time complexity is $O((nS/\epsilon) + Pn \log n)$.

Since the optimum solution value Q^* is upper bounded by $2Q_0$, there can be at most $2Q_0$ columns in the dynamic programming table. Also since there is one row for each stream, the number of rows in the table is S. Thus, table requires $O(SQ^*)$ space for storing the minimum aggregate data rate values pertaining to each quality value. Each cell in the table also needs a constant amount of space to store the backtracking information but this does not increase the asymptotic space complexity. Hence, the space complexity of the SSA algorithm is $O(SQ^*)$. □

Chapter 4

Proposed Energy Efficient Allocation Scheme

In this chapter, we formally describe and formulate the energy efficient scheduling problem for video streaming to WiMAX mobile-subscribers. We then show its hardness and propose an approximation algorithm for the problem. We then analyze our solution and derive its approximation factor.

4.1 Problem Statement

Let the energy consumption for a mobile subscriber while receiving a burst be E_a per TDD frame. Additionally let us assume E_w amount of additional power consumption every time a subscriber has to wake up to receive bursts. We consider the *Average Energy Efficiency (AEE)* metric which is defined as the ratio of energy consumption due to data transfer to the total energy consumption. This AEE metric has been used in previous works such as Shi et.al. [35]. Our goal is to maximize the AEE metric across all mobile subscribers receiving different streams. To achieve this goal, we add a secondary objective function to the optimization problem given in (P1). This secondary objective function is given by:

$$\text{Maximize} \quad \frac{1}{S}\sum_{s=1}^{S}\frac{b_s E_a}{b_s E_a + n_s E_w}. \tag{P2}$$

Clearly, adding the secondary objective function does not reduce the hardness of the problem, it is still NP-Complete. Thus, we propose an approximation algorithm to solve this problem. For the rest of this section we omit the substream subscripts l from the corresponding terms (e.g. b_s instead of b_{sl} etc.) for simplicity.

4.2 Proposed Approximation Algorithm

4.2.1 Overview of the Proposed Algorithm

The proposed approximation algorithm is called *Energy Efficient Substream Allocation* and is denoted by EESA. The EESA algorithm is executed after determining the substreams to be transmitted to the subscribers using the SSA algorithm. Thus, instead of sending the selected substreams in a continuous manner, the EESA algorithm will transmit them in bursts in order to save energy for mobile subscribers. The high level idea of the algorithm is as follows. We first assume that the receiver buffer B can be divided into two buffers of size $B/2$ each and the two buffers can be accessed in parallel. This is known as the double-buffering scheme [18]. Since one half of the buffer can be drained while the other half is being filled up in parallel, the scheme always has one buffer for receiving the current burst. Thus if we stipulate the burst sizes to $B/2$ the buffer overflow problem is resolved. Now if we construct the frames in such a way that the data received in the previous burst is equal to the data consumed during the current burst, then we can avoid buffer underflow. Thus our problem is reduced to finding the number and size of bursts for each stream.

4.2.2 Details of Proposed Algorithm

Figure 4.1 summarizes the pseudo-code of the EESA algorithm. We calculate the number and size of the bursts for each stream as follows. Since the buffer size is $B/2$, it can accommodate a burst size of at most $\lfloor B/2F \rfloor$. Therefore we divide the data blocks of each streams into bursts of length $B/2F$. There has to be at least $\lceil 2b_sF/B \rceil$ number of bursts for each stream. We note that the last burst in a scheduling window might have less than $B/2F$ data blocks. We fix the number of bursts n_s to be $\lceil 2b_sF/B \rceil$. For each burst $k \in n_s$ there is a starting frame number, the length of burst, and a deadline. These are denoted by x_s^k, y_s^k and z_s^k respectively. The starting frame number ensures that no data is transmitted before there is sufficient buffer space available in the receiver. Thus it eliminates the possibility of an

Energy Efficient Substream Allocation (EESA)

```
Input: Selected substreams, Initial buffer values, Scheduling window size
Output: Data burst allocation in the MBS area of the current scheduling window
For $s = 1$ to $S$ do
    Determine $n_s = \lceil 2b_sF/B \rceil$
    For $k = 1$ to $n_s$ do
        Determine $x_s^k$, $y_s^k$, and $z_s^k$ using Equations (4a)-(4c)
Let $\Lambda = \emptyset$
For each decision point do
    Add a burst from frames $t_c$ to $t_n$ to stream $s$, where $\mathbf{w}_s^k$ has the smallest $z_s^k$ among outstanding bursts, $t_c$ is current time, and $t_n$ is time of the next decision point
Let $e_s^k$ be the actual finish time of burst $k$
If $\max\{e_s^k - z_s^k\} \leq 0$ // complete on time
    Return $\Lambda$
Return no feasible schedule
```

Figure 4.1: Energy Efficient Substream Allocation Algorithm

overflow. Similarly the deadline frame numbers ensure that the current burst is transmitted before the receiver buffer runs out of data, eliminating the underflow possibilities. The values of x_s^k, y_s^k and z_s^k can be derived using the following equations.

$$x_s^k = \begin{cases} \max\{0, \frac{u_s}{r_s\tau}\}, & \text{for } k = 1, \\ \lfloor \frac{(k-1)B}{2r_s\tau} \rfloor, & \text{for } 1 < k \leq n_s. \end{cases} \tag{4a}$$

$$y_s^k = \begin{cases} \lfloor \frac{B}{2F} \rfloor, & \text{for } 1 \leq k < n_s, \\ b_s - (k-1)\lfloor \frac{B}{2F} \rfloor, & \text{for } k = n_s. \end{cases} \tag{4b}$$

$$z_s^k = \begin{cases} \lfloor \frac{kB}{2r_s\tau} \rfloor, & \text{for } 1 \leq k < n_s, \\ P, & \text{for } k = n_s. \end{cases} \tag{4c}$$

We also define e_s^k as the actual completion frame number for burst k of stream s. When the buffers are initially empty the starting frame number of the first burst of all streams is zero. However if the buffers are not empty, the bursts need to wait until the buffers are drained. Each subsequent burst starting point is $\lceil B/2r_s\tau \rceil$ frames away, since this is the time required for the previous burst to be consumed. The end time of each burst can also be derived in a similar way. We define decision points as the time instances at which either a new burst starts, or all frames of a burst have been allocated. These are the two points when we need to decide which burst to send next. At each such decision point we keep on allocating the burst which has the smallest z_s^k among outstanding bursts.

4.3 Correctness and Performance Analysis

In the EESA algorithm described in Figure 4.1, we use the double buffering scheme and fixed length bursts to satisfy the buffer overflow and underflow constraints. Also, since the final schedule is constructed by appending the bursts one after the other, there can be no overlap between bursts. Thus the algorithm produces a valid solution to the energy efficient burst allocation problem.

Theorem 5. *[Approximation Factor] The EESA algorithm described in Figure 4.1 is a constant factor approximation algorithm.*

Proof. To compute the approximation factor, we first determine the number of bursts in the optimal schedule and in the schedule produced by the EESA algorithm. To prevent buffer underflow instances, any optimal burst schedule must have at least $\lceil b_s F/B \rceil$ number of bursts for stream s, that is, $\lceil b_s F B \rceil \leq n_s^*$. On the other hand, because of applying double buffering the minimum number of bursts required by the EESA algorithm is at least $\lceil 2b_s F/B \rceil$. Now we derive an upper bound on the maximum number of bursts that can be constructed by algorithm the EESA. From the definition of the decision points, each burst derived by equations (4a)-(4c) can cause at most one interruption in rest of the bursts. This happens when all frames of the current burst are not allocated but the current frame number is the starting frame number of a different burst. Therefore these interruptions may cause an additional $\lceil 2b_s F/B \rceil$ number of bursts for the EESA algorithm. Therefore the maximum number of bursts created by the EESA algorithm is $2(\lceil 2b_s F/B \rceil)$, in other words, $n_s \leq 4\lceil b_s F/B \rceil$. Comparing this with the value of n_s^* we have $n_s \leq 4n_s^*$. The approximation factor can be given as the AEE achieved by the optimal algorithm to that achieved by our EESA algorithm. Ignoring the common terms in both expressions we have the following approximation factor.

$$\frac{\text{AEE (OPT)}}{\text{AEE (EESA)}} = \sum_{s=1}^{S} \frac{b_s E_a + 4n_s^* E_w}{b_s E_a + n_s^* E_w}. \tag{5}$$

We note that since the number of bursts in the optimal solution n_s^* is a constant, the approximation factor is also constant. $\square$

The allocation algorithm for mobile subscribers, EESA, involves a sorting of the bursts to assign their priorities. Since the number of bursts depends on the length of the scheduling window, sorting takes $O(P \log P)$ time. Therefore the overall running time of the algorithm EESA is $O(SP \log P)$.

Chapter 5

Evaluation using Simulation

This chapter provides the details of our simulation setup and results. We use this setup to evaluate the effectiveness of algorithms proposed in Chapter 3 and Chapter 4.

5.1 Simulation Setup

We have implemented a point-to-multipoint WiMAX multimedia broadcast simulator and evaluated our algorithm in it using actual scalable video traces. For the WiMAX network parameters we use the 16-QAM modulation scheme with 3/4 convolution turbo coding and 10MHz channel. Since each TDD frame is 5ms, for a one second scheduling window we will have to allocate data to 200 TDD frames. Also we assume that within each TDD frame we have an MBS data area of 50kb. This gives us a broadcast channel bandwidth of 10Mbps [13]. At the receiver side we assume a buffer limit of 512kb. For generating the video traffic we use 10 raw (YUV files in 4:2:0 format) video files from the video trace repository of Arizona State University [4]. For each video we generate a 10 minute workload by starting from a random initial frame and then repeating the frame sequences. Then we encode the videos into H.264/SVC format using the JSVM reference software version 9.18 [9]. We encode each stream into four PSNR scalable layers using the medium grain scalability (MGS) feature of the H.264/SVC coding standard [40]. We tune the encoding parameters such that the substreams have an average bit rate between 100kbps and 2.5Mbps. In Table 5.1, we summarize the information of the data rates and quality values of each layer of the different video files.

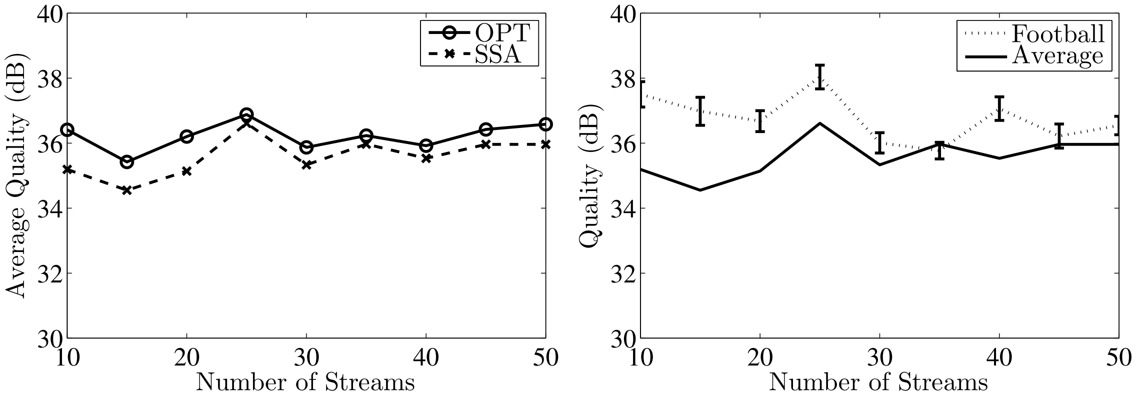
Table 5.1: Data Rates (kbps) and PSNR Values (dB) of the Scalable Videos Used in Evaluation.

Name	1 Layer		2 Layers		3 Layers		4 Layers	
	r_1	q_1	r_2	q_2	r_3	q_3	r_4	q_4
CREW	306	32.92	578	34.99	814	36.5	1184	37.41
FOOTBALL	442	30.50	827	32.91	1114	33.98	1621	35.55
MOBILE	189	35.76	322	37.87	442	38.93	649	40.36
CITY	448	29.62	923	32.21	1288	33.28	1943	34.88
FOREMAN	170	32.9	407	34.86	589	36.0	890	37.43
BUS	185	33.17	390	35.43	567	36.41	857	37.65
HARBOUR	577	31.8	1025	33.46	1379	34.67	1929	36.25
NEWS	121	35.6	259	37.55	372	38.63	564	40.5
SOCCER	385	29.92	795	32.18	1095	33.13	1651	34.68
ICE	277	32.35	548	34.71	767	35.82	1123	37.32

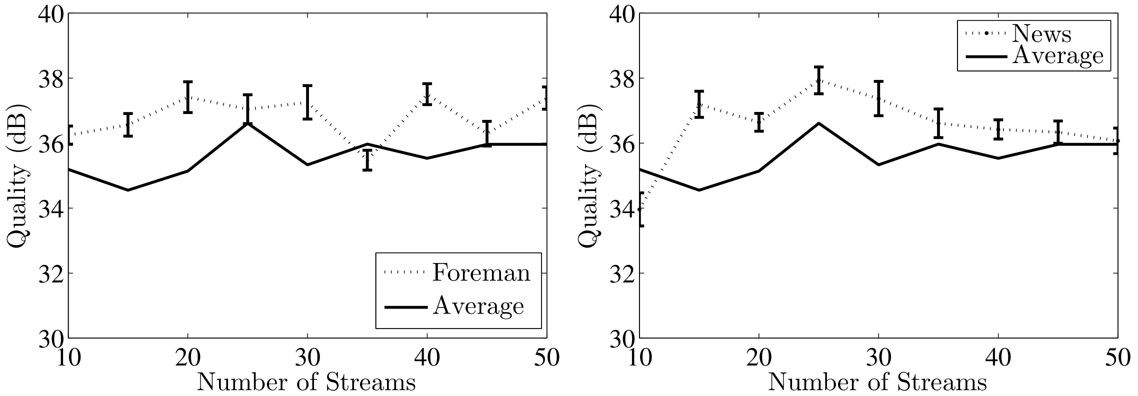
5.2 Simulation Results

5.2.1 Video Quality

In our first experiment, we compare the performance of the SSA algorithm versus the optimum algorithm in terms of video quality. We perform this comparison over a period of 100 consecutive scheduling instances. We keep the receiver buffer size fixed at 512kb, the scheduling window size at 1 sec and vary the number of streams from 10 to 50. Sample results are given in Figure 5.1; other results are similar. Figure 5.1a shows the average quality across all the video streams, where as Figure 5.1b, Figure 5.1c and Figure 5.1d show individual qualities for the Football, Foreman and News video sequences respectively. The figures show that our SSA algorithm produces near optimal solutions, which are less than 1dB from the optimal solution. Optimal solutions are computed using the optimization software GLPK [12]. We note that although there are other methods of computing the optimal solution (e.g. branch and bound) they too are not suitable for real-time computation of the solutons. We also observe that the proposed SSA algorithm scales well when the number of streams increase. In another experiment we keep the number of streams fixed at 20 and vary the scheduling window from 1 to 10 sec. As we can see from the results of this experiment in Figure 5.2, the solution quality improves as the scheduling window grows. Again, Figure 5.2a shows the average quality across all the video streams, where as Figures 5.2b, Figure 5.2c and Figure 5.2d show individual qualities for the Football, Foreman and News



(a) Average (b) Football



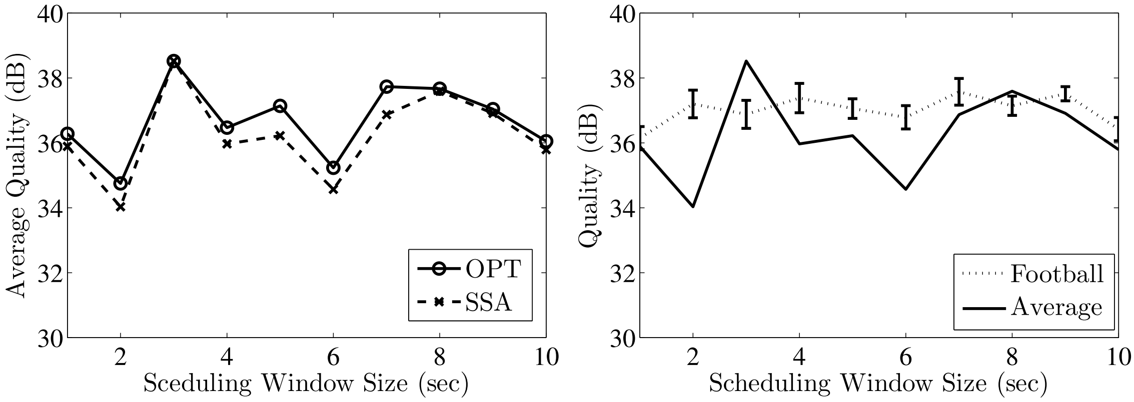
(c) Foreman (d) News

Figure 5.1: Near optimality of the solutions obtained by the SSA algorithm.

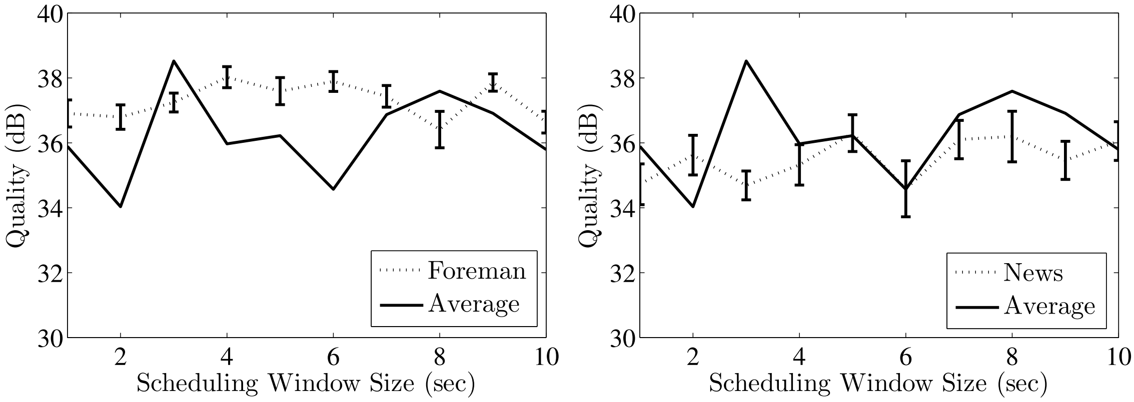
video sequences. These two experiments show that our algorithm produces close to optimal solutions and it scales well, which means that it can support large scale WiMAX streaming services.

5.2.2 Time Efficiency

We evaluate the running time of our algorithms by changing the problem size in two ways. First we keep the scheduling window capacity fixed and increase the number of streams. In a second experiment we keep the number of streams fixed and increase the scheduling window. We compare the execution times of our algorithms to that of the optimum. For deriving the

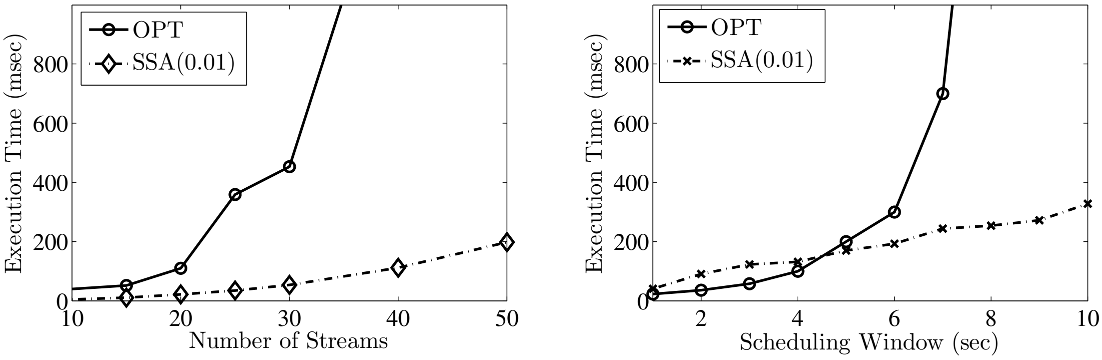


(a) Average (b) Football



(c) Foreman (d) News

Figure 5.2: Effect of scheduling window size on the solutions obtained by the SSA algorithm.



(a) Fixed window size at 1 sec

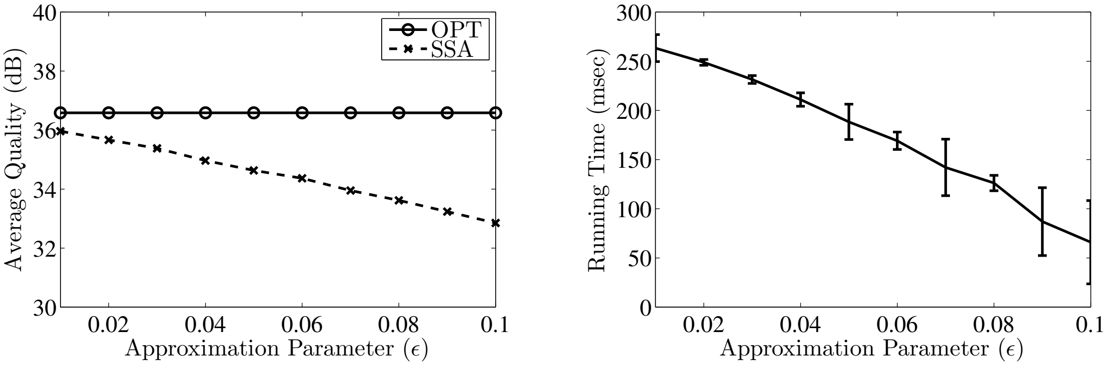
(b) Fixed number of streams at 20

Figure 5.3: Running time of the SSA and the optimal algorithms.

optimum solution we first use the GLPK LP solver [12] to determine the substreams that can be scheduled and then sequence the frames within a scheduling window in a weighted round robin manner. For the SSA algorithm, we compute the running time with approximation parameter $\epsilon = 0.01$. For a one second scheduling window, we vary the number of streams from 10 to 50 and observe their behaviour. The execution times of these algorithms are measured on a computer with 1.6GHz dual-core processor and 1GB of memory. The results of the first experiment are shown in Figure 5.3a. As expected, computing the optimum solution using GLPK takes much longer as the number of streams increase. The SSA algorithm runs well within the time window even for large problem instances. For the second experiment we keep the number of streams fixed at 20 and vary the scheduling window size from 1 sec to 10 sec. From the results of the second experiment, depicted in Figure 5.3b, we can see that the SSA algorithm scales efficiently with increase in window size. For real time operation, the algorithm needs to compute the solution within the scheduling window duration. From Figure 5.3b, we see that with every one second increment in the scheduling window size the execution time increases by only a few milliseconds. This shows that the SSA algorithm scales well for large problem instances under real-time constraints.

5.2.3 Significance of the Approximation Parameter

Next we investigate the effect of the approximation parameter ϵ on the quality of solution and also on the time efficiency of the SSA algorithm. The approximation parameter can be



(a) Fixed number of streams at 50 (b) Fixed number of streams at 50

Figure 5.4: Effect of the approximation parameter on quality and running time.

thought of as a knob for tuning the trade-off between solution quality and solution computation time. We first vary the approximation parameter from 0.01 to 0.10 in steps of 0.01 and as seen in Figure 5.4a, the average quality of the received videos degrades as the approximation parameter increase. In Figure 5.4b we see that when the approximation parameter is increased the running time of the algorithm decreases. This means that approximate schedules for very large scale problems can be computed, which can be used for preliminary analysis of network deployments. However, in all cases, our algorithm can easily run in real time even with very small approximation parameters. This is shown in Figure 5.4b for ϵ =0.01, and the schedule is computed in less than 0.3 seconds for a scheduling window of 1 second.

5.2.4 Resource Utilization and Buffer Validation

Next, we evaluate the resource utilization of the SSA algorithm in terms of the scheduling window capacity used. Let the total schedulable data capacity of a scheduling window be PF. If $\{\bar{r_1}, \dots, \bar{r_S}\}$ are the data rates of the chosen substreams, the total data sent in the schedule is $\sum \tau P \bar{r_s}$. The capacity utilization is then given by $\sum \tau P \bar{r_s} / PF$. As seen in Figure 5.5, the resource utilization of the SSA algorithm remains close to optimal for different scheduling window capacity sizes.

Next we verify that the buffer conditions are satisfied by the SSA algorithm. That is, we check if the SSA algorithm causes any buffer overflow or underflow. We found neither overflow nor underflow occurrences for any of the streams. In Figure 5.6, we display the

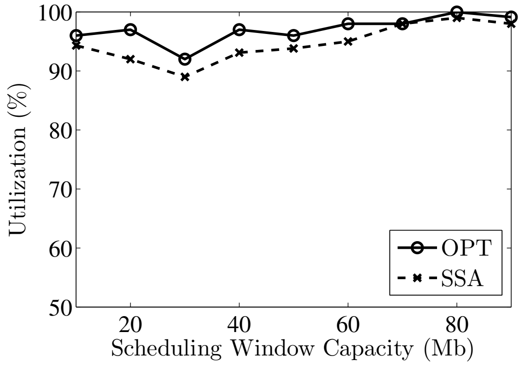
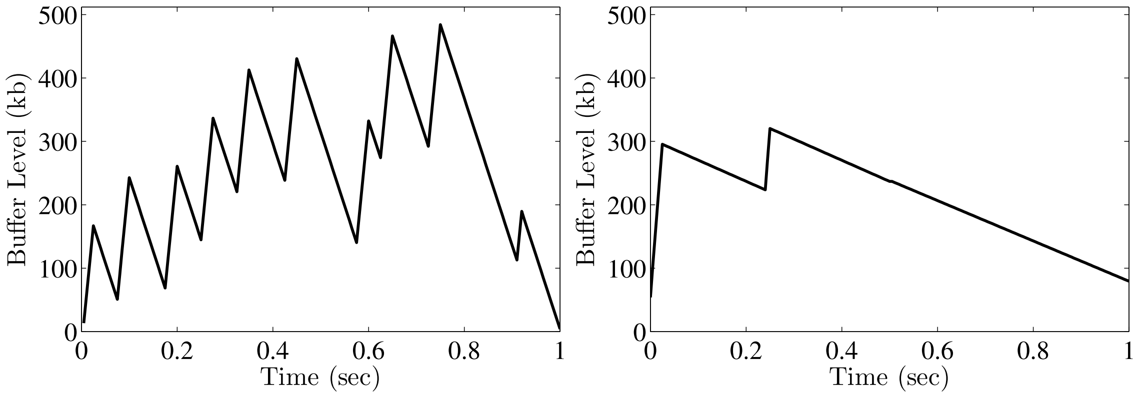


Figure 5.5: Performance of the SSA Algorithm in terms of resource utilization.

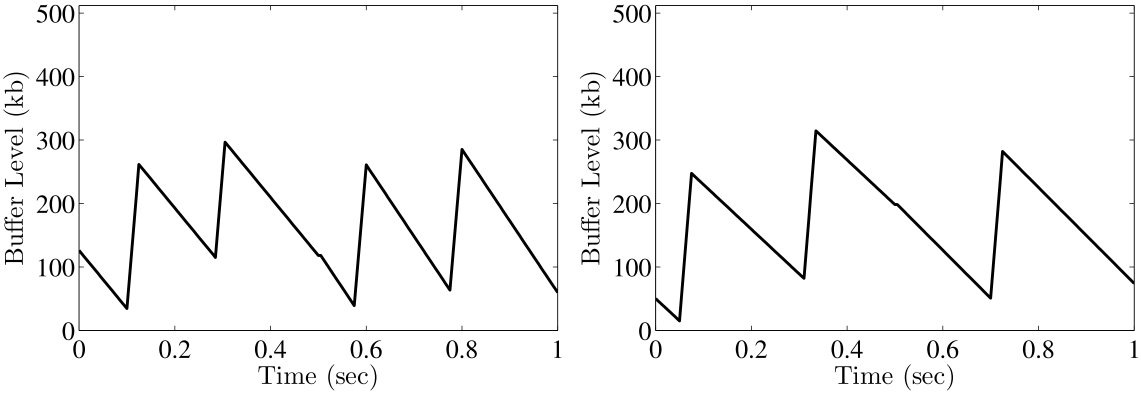
buffer level dynamics of subscribers receiving different streams, which show that the buffer level never exceeds 500kb, that is, there are no overflow instances. It also shows the the buffer level never goes below zero, which means there are no underflow instances. Similar results were obtained for subscribers receiving other video streams.

5.2.5 Energy Savings

We evaluate and compare the energy savings resulting from our EESA algorithm to that of the Weighted Round Robin Allocation (WRRA) scheduler. We assume equal number of mobile stations are receiving each stream. For evaluating the energy saving we use the Average Energy Efficiency metric. In Figure 5.7a and Figure 5.7b, we display the comparison between the two algorithms when the number of streams and window size are varied respectively. The figures show that the EESA algorithm achieves high values for the AEE metric (close to 1) and remains significantly more efficient than the WRR allocation when the window size increases. The results of another simulation is displayed in Figure 5.8 where the impact of receiver buffer size on energy efficiency is measured. We see that the energy efficiency increases when the buffer size is increased. This is directly linked to the fact that the number of times switching occurs in a schedule has a high impact on energy savings, and large buffers can absorb larger bursts, reducing the total number of bursts needed.

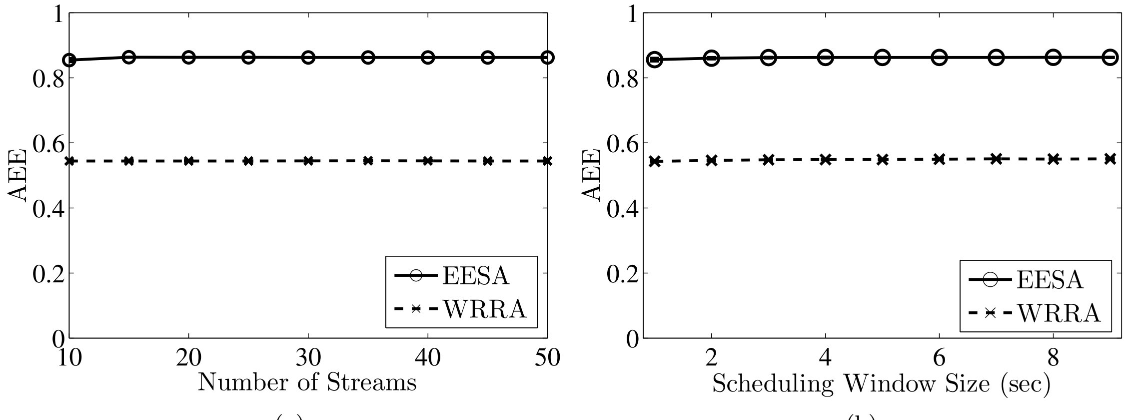


(a) Football (b) Foreman



(c) News (d) Soccer

Figure 5.6: Performance of the SSA Algorithm in terms of receiver buffer dynamics.



(a) (b)

Figure 5.7: Energy efficiency of EESA algorithm.

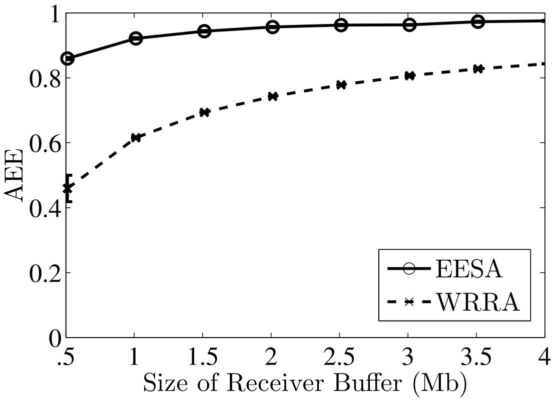


Figure 5.8: Effect of buffer size on energy efficiency.

Chapter 6

Conclusions and Future Work

In this chapter, we first summarize the contributions in this thesis. Then, we outline several future research directions in which this work can be extended.

6.1 Conclusion

We studied a framework for multicasting scalable video streams over mobile WiMAX networks. We mathematically analyzed the problem of selecting the optimal substreams of scalable video streams under limited buffer and bandwidth constraints. We also explored different scheduling schemes to enhance the energy efficiency at the mobile subscriber device.

Solving these two problems is important for the following reasons. Improving video quality at the subscriber device results in increased user satisfaction and in turn increases operator revenue. Alternatively our solution can be used to transmit a larger number of videos at acceptable levels of quality. This also increases user satisfaction by offering the user more options in terms of content. Our energy efficient solution reduces energy consumption at the receiver, providing longer viewing times and better user experience.

We showed that the substream selection problem in the presence of limited bandwidth is NP-Complete. We proposed a novel approximation algorithm for this problem. We proved that our algorithm has a small approximation factor of $(1+\epsilon)$, and a time complexity of $O(nS/\epsilon)$, where $n = O(\sum L)$ is the total number of layers, L is the maximum number of layers in a scalable stream, and ϵ is a small constant.

We extended our formulation to consider the energy constraints of mobile receivers. We presented an algorithm to transmit the data in bursts in order to conserve the energy of mobile receivers. We showed that our energy efficient algorithm achieves an approximation factor of $\sum_{s=1}^{S}(b_s E_a + 4n_s^* E_w)/(b_s E_a + n_s^* E_w)$, compared to the optimum burst construction algorithm. Here n_s^* is the optimum number of bursts for stream s.

We implemented and validated our algorithms in a simulation setup and studied the impact of a wide range of parameters using multiple video traces. Our simulation results show that the performance of the proposed substream selection algorithm is very close to the optimal for practical scenarios. The average video quality achieved by our solution remains within 1dB of the optimal achievable quality for most practical purposes. We also verified that our algorithm can run in real time and that it scales well to large scheduling problems. Finally, our simulations also show that the energy efficient burst scheduling algorithm achieves higher energy efficiency as compared to the weighted round robin schedules and remains significantly more efficient when the window size increases.

6.2 Future Work

The work in this thesis can be extended in many directions. We summarize some of them bellow for future exploration.

- Our current work considers only single hop WiMAX networks. However with increased capacity demand, wireless networks are becoming more complex in terms of topology. An associated standard, 802.16j [20] provides the specifications for deploying multiple relay stations within the range of a base station for improving coverage. While our algorithm can be applied to such deployments it may be improved further by incorporating the capacity fluctuations due to different channel conditions between the base station and different relay stations.

- The upcoming WiMAX Release 2 or 802.16m standard [1] has several proposed facilities for improving video streaming services while still being backward compatible with the current 802.16e standard. The new standard has enhanced MBS mode such that it can dynamically switch between multicast and unicast mode. It will be interesting to see how our algorithm can be adapted to this enhancement such that the video streaming performance can be improved even in the presence of variable-bit-rate unicast traffic demands. The 802.16m standard also has improved sleep mode operations to further improve energy efficiency. We can evaluate the performance of our energy efficient algorithm under the new mechanism.

- We can extend our solution to consider the probability distribution of device profiles of active receivers. The algorithm may take into account diverse parameters such as buffer size, display resolution and energy consumption profiles to not only optimizes the video quality but also enhance the quality of experience across multiple dimensions.
- Our solution can be applied to other similar networks like LTE (3GPP Long Term Evolution) [26] and LTE-Advanced [25] with minimal changes. Since it does not depend on any input from the receiver our solution is particularly suitable for broadcast only networks like DVB-H, MediaFLO or ATSC M/H [28]. However, due to the different physical layer mechanism and data encapsulation techniques, the performance of the algorithm may vary. A comparative evaluation of it's performance may reveal the technology for which is best suited and how it can be improved for other technologies.
- In out current problem, we only consider unidirectional delivery of video. However, future multimedia applications are expected to be interactive. Interactivity imposes many challenges including lower latency requirements, very low channel switching delay and integration of other services like community viewing. Accommodating these requirements while still maintaining the video streaming QoS is an exciting open area of research.
- For video streaming applications, the hand-off performance at the border of a base station coverage area is an important case study. We can explore new scheduling algorithms for improving performance for mobile devices in the multi-base-station overlap region. In particular, we can try to improve the performance of our algorithm by associating the mobile subscriber with the base station with the strongest signal, or even try to deliver different layers of video through different base stations according to their channel conditions to the subscriber.

Appendix A

List of Acronyms

3G	3rd Generation standards for mobile communication
3GPP	3rd Generation Partnership Project
AEE	Average Energy Efficiency
ASN	Access Service Network
ATSC M/H	Advanced Television Systems Committee - Mobile/ Handheld
AVC	Advanced Video Coding
DVB-H	Digital Video Broadcasting - Handheld
EESA	Energy Efficient Substream Allocation
ESG	Electronic Service Guide
GLPK	GNU Linear Programming Kit
GPS	Global Positioning System
GSM	Global System for Mobile Communications: originally from Groupe Spcial Mobile
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
JSVM	Joint Scalable Video Model
LP	Linear Programming
LTE	Long Term Evolution
MAC	Medium Access Control
MBMS	Multimedia Broadcast and Multicast Services
MBS	Multicast and Broadcast Services
MGS	Medium Grain Scalability

NTP	Network Time Protocol
OFDMA	Orthogonal Frequency-Division Multiple Access
PDU	Protocol Data Unit
PSNR	Peak Signal-to-Noise Ratio
QAM	Quadrature amplitude modulation
QoS	Quality of service
RF	Radio frequency
RSS	Relaxed Substream Selection
SFN	Single Frequency Network
SSA	Substream Selection Algorithm
SVC	Scalable Video Coding
TDD	Time Division Duplex
UMTS	Universal Mobile Telecommunications System
VBR	Variable Bit Rate
WiFi	A Synonym for IEEE 802.11 standard related technology
WiMAX	Worldwide Interoperability for Microwave Access
WRR	Weighted Round Robin

Bibliography

[1] WiMAX and the IEEE 802.16m Air Interface Standard. WiMAX Forum, White Paper, April 2010. available online at `http://www.wimaxforum.org/sites/wimaxforum.org/files/document_library/wimax_802.16m.pdf`.

[2] Multimedia Broadcast/Multicast Service (MBMS) in GERAN; Stage 2 (Release 6). Third Generation Partnership Project (3GPP) Standard TS 43.246 Ver. 9.0.0, 2009.

[3] Multimedia Broadcast/Multicast Service (MBMS); stage 1: User services. Third Generation Partnership Project (3GPP) Standard TS 22.246 Ver. 9.0.0, 2009.

[4] Arizona State University: Video Traces Research Group. `http://trace.eas.asu.edu`.

[5] R. Cohen, L. Katzir, and R. Rizzi. On the trade-off between energy and multicast efficiency in 802.16e-like mobile networks. *IEEE Transactions on Mobile Computing*, 7(3):346–357, March 2008.

[6] Digital Video Broadcasting - Handheld (DVB-H) home page, 2009. `http://www.dvb-h.org/`.

[7] DVB-H Implementation Guidelines;European Telecommunications Standards Institute (ETSI) Standard EN 102 377 Ver. 1.4.1, April 2009.

[8] Digital Video Broadcasting (DVB); IP datacast over DVB-H: Content delivery protocol. European Telecommunications Standards Institute (ETSI) Standard EN 102 472 Ver. 1.1.1, November 2006.

[9] Fraunhofer HHI: JSVM H.264 AVC/SVC Reference Software. `http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm`.

[10] M. Garey and D. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[11] V. Geert, P. David, M. Reisslein, and L. Karam. Traffic and quality characterization of the H.264/AVC scalable video coding extension. *Advances in MultiMedia*, 2008(2):1–27, January 2008.

[12] GNU Linear Programming Kit. `http://www.gnu.org/software/glpk/glpk.html`.

[13] D. Gray. Mobile WiMAX Part I - Overview and Performance. WiMAX Forum, White Paper, August 2006. available online at `www.wimaxforum.org/technology/downloads/Mobile_WiMAX_Part1_Overview_and_Performance.pdf`.

[14] D. Gray. A comparative analysis of mobile WiMAX deployment alternatives in the access network. `http://www.wimaxforum.org/sites/wimaxforum.org/files/document_library/mobile_wimax_deployment_alternatives.pdf`, May 2007. White Paper.

[15] GSM Home Page. `http://www.gsmworld.com/`.

[16] P. Hosein. Broadcasting VBR traffic in a WiMAX network. In *Proc. of IEEE VTC'08*, pages 1–5, Calgary, Canada, September 2008.

[17] P. Hosein and T. Gopal. Radio resource management for broadcast services in OFDMA-based networks. In *Proc. of IEEE International Conference on Communications Workshops*, pages 271–275, Beijing, China, May 2008.

[18] C. Hsu and M. Hefeeda. Time Slicing in Mobile TV Broadcast Networks with Arbitrary Channel Bit Rates. In *Proc. of IEEE INFOCOM'09*, pages 2231–2239, Rio de Janeiro, Brazil, April 2009.

[19] IEEE Standard. Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Broadband Wireless Metropolitan Area Network . `http://standards.ieee.org/getieee802/download/802.16-2009.pdf`.

[20] IEEE Standard. Local and metropolitan area networks Part 16: Amendment 1: Multihop Relay Specification . `http://standards.ieee.org/getieee802/download/802.16j-2009.pdf`.

[21] H. Juan, H. Huang, C. Huang, and T. Chiang. Scalable video streaming over mobile WiMAX. In *Proc. of International Symposium on Circuits and Systems*, pages 3463–3466, New Orleans, LA, May 2007.

[22] M. Kim, M. Kang, and J. Choi. Remaining Energy-Aware Power Management Mechanism in the 802.16e MAC. In *Proc. of IEEE Consumer Communications and Networking Conference (CCNC'08)*, pages 222–226, Las Vegas,NV, January 2008.

[23] W. Liao and N. Lee. An Integrated Power Saving Scheduling Algorithm in 802.16e Wireless Metropolitan Area Networks. In *Proc. of International Conference on Mobile Technology, Applications & Systems*, pages 1–8, Yilan, Taiwan, 2008.

[24] E. Lin. A bibliographical survey on some well-known non-standard knapsack problems. *Information Systems and Operation Research*, 36:274–317, November 1998.

[25] LTE Advanced homepage from the 3GPP website. `http://www.3gpp.org/article/lte-advanced`.

[26] LTE homepage from the 3GPP website. `http://www.3gpp.org/article/lte`.

[27] MediaFLO home page, 2009. `http://www.mediaflo.com/`.

[28] Mobile DTV standard homepage from the ATSC website. `http://www.atsc.org/cms/index.php/standards/published-standards/163-a153-atsc-mobile-dtv-standard-parts-1-8`.

[29] DVB-H Deployment in Austria. `http://www.dvb-h.org/Services/services-austria.htm`.

[30] OMVC Mobile TV Web Page. `http://www.openmobilevideo.com/resources/omvc-materials/reports/`.

[31] V. Reguant, F. Prats, R. de Pozuelo, F. Margalef, and G. Ubiergo. Delivery of H.264 SVC/MDC streams over WiMAX and DVB-T networks. In *Proc. of IEEE International Symposium on Consumer Electronics (ISCE'08)*, pages 1–4, Algarve, Portugal, April 2008.

[32] J. Seo, S. Lee, N. Park, H. Lee, and C. Cho. Performance analysis of sleep mode operation in IEEE 802.16e. In *Proc. of IEEE VTC'04*, pages 1169–1173, Los Angeles, CA, September 2004.

[33] S. Sharangi, R. Krishnamurti, and M. Hefeeda. Energy-efficient Multicasting of Scalable Video Streams over WiMAX Networks. *IEEE Transactions on Multimedia*, 2010. (Accepted with minor revision).

[34] S. Sharangi, R. Krishnamurti, and M. Hefeeda. Streaming Scalable Video over WiMAX Networks. In *Proc. of IEEE Workshop on Quality of Service (IWQoS'10)*, pages 1–9, Beijing, China, June 2010.

[35] J. Shi, G. Fang, Y. Sun, J. Zhou, Z. Li, and E. Dutkiewicz. Improving Mobile Station Energy Efficiency in IEEE 802.16e WMAN by Burst Scheduling. In *Proc. of IEEE GLOBECOM'06*, pages 1–5, San Francisco, CA, December 2006.

[36] P. Sinha and A. Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, June 1979.

[37] UDCAST WiMAX TV Home Page. `http://www.udcast.com/products/udcast_wimax_tv_products.htm`.

[38] Visiongain report on the Mobile TV market. `http://www.visiongain.com/Report/478/Mobile-TV-Market-Analysis-and-Opportunities-2010-2015`.

[39] J. Wang, M. Venkatachalam, and Y. Fang. System architecture and cross-layer optimization of video broadcast over WiMAX. *IEEE Journal on Selected Areas in Communications*, 25(4):712–721, May 2007.

[40] T. Wiegand, G. Sullivan, G. Bjøntegaard, and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, July 2003.

[41] Yota Mobile WiMAX Home Page. `http://www.yota.ru/en/info/main/`.

[42] E. Zemel. An $O(n)$ Algorithm for the linear multiple choice knapsack problem and related problems. *Information Processing Letters*, 18(3):123–128, March 1984.