UPLINK SCHEDULING ALGORITHMS FOR THE rtPS TRAFFIC CLASS FOR IEEE 802.16 NETWORKS

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Mustafa Cenk Ertürk
September 2008

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Nail Akar (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ezhan Karaşan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İbrahim Körpeoğlu

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet B. Baray
Director of Institute of Engineering and Sciences

ABSTRACT

UPLINK SCHEDULING ALGORITHMS FOR THE rtPS TRAFFIC CLASS FOR IEEE 802.16 NETWORKS

M. Cenk Ertürk
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Nail Akar
September 2008

IEEE 802.16 MAC provides extensive bandwidth allocation and QoS mechanisms for various types of applications. However, the scheduling mechanisms for the uplink and downlink are unspecified by the IEEE 802.16 standard and are thus left open for vendors' own implementations. Ensuring QoS requirements at the MAC level for different users with different QoS requirements and traffic profiles is also another challenging problem in the area. The standard defines five different scheduling services one of them being the real-time Polling Service (rtPS). In this thesis, we propose an uplink scheduler to be implemented on the WiMAX Base Station (BS) for rtPS type connections. We propose that the base station maintains a leaky bucket for each rtPS connection to police and schedule rtPS traffic for uplink traffic management. There are two scheduling algorithms defined in this study: one is based on a simpler round robin scheme using leaky buckets for QoS management, whereas the other one uses again leaky buckets for QoS management but also a proportional fair scheme for potential throughput improvement in case of varying channel conditions. The proposed two schedulers are studied via simulations using MATLAB to demonstrate their performance in terms of throughput, fairness and delay. We show that the leaky bucket based scheduler ensures the QoS commitments of each user in terms of a minimum bandwidth guarantee whereas the proportional fair algorithm is shown to opportunistically take advantage of varying channel conditions.

Keywords: IEEE 802.16, scheduling algorithms, quality of service, throughput and delay analysis, MATLAB.

ÖZET

IEEE 802.16 AĞLARI İÇİN YUKARI HAT PLANLAMA ALGORİTMALARI

M. Cenk Ertürk
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Nail Akar
Eylül 2008

IEEE 802.16 Ortam Erişim Yönetimi (MAC), kapsamlı bant genişliği dağılımı ve değişik tipteki uygulamalar için servis kalitesi (QoS) sağlamaktadır. Ancak, bu özellikler için planlama mekanizmaları standartta tanımlanmamış ve servis sağlayıcıların uygulamasına açık bırakılmıştır. Servis kalitesi isteklerini değişken trafik modelleri için MAC düzeyinde sağlamak bu alanda karşılaşılan zorlayıcı problemlerdendir. Standart bu problemleri planlama kapsamında değerlendirdiğinden standartta beş farklı planlama sınıfı tanımlanmıştır ve bunlardan biri de Gerçek Zamanda Seçilme Servisi'dir (GZSS). Bu tezde WiMAX baz istasyonlarının GZSS için yukarı hat planlamalarının nasıl tasarlanması gerektiği araştırılmıştır. Yukarı hat trafik yönetimi için baz istasyonu tarafından her GZSS bağlantısı için bir su sızdıran kovanın (leaky bucket) kullanılması önerilmiştir. Bu çalışmada iki adet planlama algoritması tanımlanmıştır: Birincisinde, yuvarlak robin (round robin) algoritması, su sızdıran kovalarla birlikte servis kalitesini sağlamak için tasarlanmıştır. İkincisinde su sızdıran kovalar yine servis kalitesini sağlamakla birlikte oransal adil (proportional fair) algoritması kullanılarak kanal durumlarının değişmesi durumunda potansiyel üretilen iş miktarlarının artırılmasına yönelik bir tasarım ortaya konulmuştur. Önerilen yöntemler MATLAB ortamında benzetim yapılarak gerçekleştirilmiş ve üretilen iş miktarları, adil olma özellikleri, gecikme karakteristikleri bazında performansları gösterilmiştir. Sonuç olarak, su sızdıran kovaların servis kalitesini kullanıcılara asgari bant genişliği sağlaması açısından uygun olduğu, oransal adil algoritmasının ise değişken kanal durumlarından faydalanarak üretrilen iş miktarını artırdığı ortaya konulmuştur.

Anahtar Kelimeler: IEEE 802.16, planlama algoritmaları, servis kalitesi, üretilen iş miktarı analizi, MATLAB

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Nail Akar, for suggesting that we take on this study and for his continuous support and interest to my M.Sc. study.

In addition to my supervisor, I address my extreme gratitude to Dr. Ezhan Karaşan and Dr. İbrahim Körpeoğlu for reading my thesis and for their invaluable comments. I would like to address my special thanks to Dr. Sinan Gezici for his valuable suggestions and helpful discussions.

I would also like to thank to my employer, the Scientific and Technological Research Council of Turkey, for supporting me throughout my graduate study and my colleague İdil Öncü for her valuable helps in this study.

This thesis study is a part of project that has been funded by the Scientific and Technological Research Council of Turkey under the grant number 106E046.

I am forever indebted to my parents and my brother for their constant support and encouragement throughout my life. This thesis devoted to them.

Table of Contents

List of Figures

List of Tables

Acronyms

AMC	Adaptive Modulation and Coding
AMR	Adaptive Multi Rate
ATM	Asynchronous Transfer Mode
BE	Best Effort
BPSK	Binary Phase Shift Keying
BRH	Bandwidth Request Header
BRI	Bandwidth Request Index
BS	Base Station
BWA	Broadband Wireless Access
CID	Connection Identifier
CP	Cyclic Prefix
CS	Convergence Sublayer
CPS	Common Part Sublayer
CSI	Channel State Information
DL	Downlink
DL MAP	Downlink Map
DSA	Dynamic Service Activate
DSC	Dynamic Service Change
DSD	Dynamic Service Delete
ertPS	extended real time Polling Service
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform
FUSC	Fully Utilized Subchannels
GPC	Grant per Connection
GPSS	Grant per Subscriber Station
IFFT	Inverse Fast Fourier Transform
IP	Internet Protocol

LAN	Local Area Network
LOS	Line Of Sight
MAC	Medium Access Control
ML	Maximum Latency
MRTR	Minimum Reserved Traffic Rate
MSH	MAC Subheader
MSTR	Maximum Sustained Traffic Rate
NLOS	Non Line Of Sight
nrtPS	non real time Polling Service
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PDU	Protocol Data Unit
PF	Proportional Fair
PFI	Proportional Fair Index
PHY	Physical Layer
PUSC	Partially Utilized Subchannels
P2MP	Point to Multipoint
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RR	Round Robin
rtPS	real time Polling Service
SDU	Service Data Unit
SFID	Service Flow Identification
TDD	Time Division Duplex
TJ	Tolerated Jitter
TP	Traffic Priority
UDP	User Datagram Protocol
UGS	Unsolicited Grant Service

UL	Uplink
UL MAP	Uplink Map
VoIP	Voice over IP
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network

Chapter 1

Introduction

1.1 Broadband Wireless Access

Wireless systems have a goal to support broadband access to Internet. IEEE 802.16, the so-called WiMAX (Worldwide Interoperability for Microwave Access), is the standard developed for the MAC and physical layers for broadband wireless metropolitan area networks. Since there is a rapid deployment of large-scale wireless infrastructures and a trend to support mobility, the popularity of WiMAX is increasing. In addition, setting up wireless systems such as WiMAX is much easier than constructing wireline systems, i.e. digging streets, setting up connections in houses or offices etc.

The IEEE standardization for WiMAX began in 1999 and the first standard is published in 2001. Several amendments, i.e. 802.16a, 802.16b, 802.16c are introduced but IEEE 802.16d 2004 standard [1] replaces all up to 2004. IEEE 802.16d 2004 (fixed WiMAX), IEEE 802.16e 2005 [2] (mobile WiMAX) are the most widely used standards for WiMAX. The most recent amendment 802.16e considers issues related to mobility and scalable OFDMA; in addition to given features in fixed WiMAX.

1.2 Ensuring the QoS and Scheduling

In recent years, people have become more familiar with new services based on multimedia applications, which require strict Quality of Service (QoS) guarantees. IEEE 802.16 MAC provides extensive bandwidth allocation and

QoS mechanisms for various types of applications. However, the specifications of the scheduling mechanisms to satisfy QoS requirements are unspecified by the standard and thus left open for vendors' implementations.

Ensuring QoS requirements at MAC level for different traffic sources is also another challenging problem in this area. The IEEE 802.16 standard addresses these problems with scheduling, i.e. five different QoS classes are defined in the standard [1], [2].

Scheduling in 802.16 is realized via Base Stations (BS). Scheduling structure should handle both downlink (from BS to Subscriber Station (SS)) and uplink (from SS to BS) flows. It can be suggested that for the overall QoS to be supported, fairness issue and QoS classes for both uplink and downlink should be taken into account by the BS scheduler. Since the information of the status of the real queues for SSs (i.e. actual backlog of each SS) is not available in the BS, uplink scheduling requires an additional step to get bandwidth requests - to learn the actual backlogs. Thus, uplink scheduling is somehow more complex compared to downlink scheduling.

1.3 Problem Definition

In this thesis, the IEEE 802.16 architecture is studied, the current research in the area is surveyed and potential research problems are laid. Particularly, we focus on the MAC architecture of WiMAX and introduce the capacity planning and scheduling problems for WiMAX.

In order to increase the overall throughput of the system while satisfying the QoS requirements of the users and achieving a level of fairness between users, scheduling algorithms have to be thoroughly studied. In this thesis, scheduling algorithms for rtPS type of connections are proposed. The scheduling problem for the downlink, where the backlog of each SS is known by the BS; is not much different than the scheduling problems for wireline networks. Therefore, our focus in this thesis would be on the uplink scheduling problem. The traffic patterns considered in all scenarios are the Voice over IP (VoIP) model, the near real time video streaming model and the full buffer model defined in [28]. Specifically, traffic patterns are solely used for defining uplink traffic.

1.4 Thesis Contributions

The contributions of this thesis mainly involve three perspectives:

- Ensuring the QoS requirements of SSs - assigning appropriate bandwidth to each user while considering their minimum bandwidth guarantees, maximum latency parameters with a relatively low complexity and practical scheduling algorithm,

- Developing channel aware scheduling algorithms by modifying the proportional fair algorithm defined in [4] and [28] in a way to encompass WiMAX systems and by implementing smart scheduling in terms of both QoS and channel awareness,

- Developing packet aware scheduling algorithms using traffic models defined in [28] for simulations.

Two schedulers are proposed; one is based on round-robin principles and the other based on the well-known proportional fair scheme. The first one is strictly in favor of fairness, whereas the latter considers both fairness and throughput maximization taking the channel conditions of the users into account. Several scenarios are considered to simulate the behavior of the schedulers in terms of throughput, fairness and delay characteristics. The advantages and disadvantages of both algorithms are discussed.

1.5 Thesis Outline

We discuss the 802.16 protocol model in Chapter 2. We describe the details of the MAC and PHY layer structures of the standard from the scheduler's point of view. Moreover, capacity planning for OFDM/OFDMA radios is presented. Chapter 2 also provides a brief literature survey on WiMAX schedulers. Several papers and theses on the topic are surveyed.

System design criteria, goals and decisions are introduced in Chapter 3. The scheduling parameters for our simulations and other details related to the simulation environment are presented. The traffic models used in the simulations are also described in this chapter. Chapter 3 also presents our scheduling algorithms and flow-charts along with their detailed explanations.

Chapter 4 is divided into two parts; each dealing with the same scenarios with different bandwidth request mechanisms. Throughput and delay analysis of the proposed schedulers are carried out for five different scenarios in the first part of the chapter. The second part of Chapter 4 deals with bandwidth request mechanisms and a simulation study of bandwidth request mechanisms is presented. Chapter 4 also includes an additional third part which discusses and compares the schedulers and gives a brief conclusion of simulation results.

Finally Chapter 5 concludes and provides a roadmap for future studies.

Chapter 2

IEEE 802.16 Standard and Related Work

2.1 IEEE 802.16 Standard

2.1.1. Overview

The IEEE 802.16 standard offers two operational modes: point-to-multipoint (P2MP) and mesh. In P2MP mode; Subscriber Stations (SS) i.e. laptop, PDA or an access point to a local area network (LAN) can only communicate with BSs but other SSs; whereas in mesh mode, SSs do communicate with each other and BSs. For the overall QoS to be achieved, mesh mode is somehow infeasible because when SSs have their own packets to send, they would probably tend not to send other SSs' data. This leads us to conclude that, QoS satisfaction in mesh topology is much harder than P2MP mode. From another point of view; using mesh mode, power could be saved due to decreased distance between hops and also more efficient routing could be done – channel conditions would possibly be better using another SS' access point to send. Most of current researches [5], [7], [8], [9], [10] , [15], [18], on WiMAX systems focus on the simpler P2MP mode; which will also be the scope of this thesis.

Uplink and downlink data transmissions are frame based in WiMAX standard, i.e., time is partitioned into frames of fixed duration. WiMAX frames are divided into two subframes; as downlink and uplink subframes in which data transmissions are done towards the SS and towards the BS, respectively. In a frame duration, the ratio of subframes can be dynamically varied for better scheduling.

Frame durations are partitioned into a number of slots. A slot can be defined as the smallest time and frequency unit of a frame that can be allocated for transmission. It is vital to note here that the term "slot" differentiates between OFDM and OFDMA radios; and even between uplink and downlink cases.

WiMAX subframes can be duplexed either by Frequency Division Duplexing (FDD); in which transmissions in each subframe can occur at the same time but at different frequencies, or by Time Division Duplexing (TDD); in which transmissions in each subframe can occur at the same frequency but at different times. SSs can be full duplex (transmit and receive simultaneously) or half duplex (either transmit or receive at a certain time) [4].

Bandwidth requests are always per connection; however, WiMAX standard specifies two allocation modes to those requests: grant per connection (GPC) or grant per SS (GPSS) [2]. In GPC, BS grants are per connection – allocated bandwidth is assigned to a connection which is under the management of an SS. However, in GPSS, grants are per SS – SS should be clever enough to deliver this grant to each connection. It can be inferred that rescheduling of the granted bandwidth by SSs in GPSS mode would be necessary.

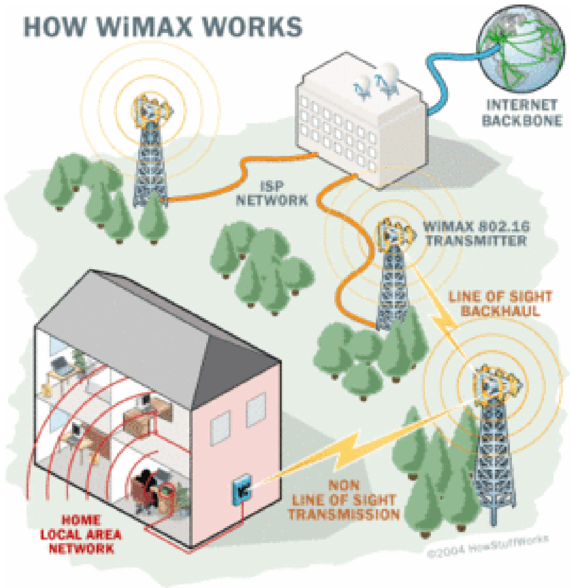


Figure 2.1 How WiMAX works [11]

In order to have a deeper understanding in WiMAX architecture, it is useful to analyze the structure given in Figure 2.1. Basically, local area networks i.e. Wi-Fi's, Ethernets enter the WiMAX network via an access point called the subscriber station (SS). It is important to note that Laptops, PDAs i.e. with a WiMAX adapter can also directly communicate with the BS without a usage of an access point. In P2MP mode, SSs, which are the houses' access points in Figure 2.1, cannot send their data to each other. BS controls the environment in terms of both downlink and uplink using scheduling algorithms. In P2MP mode, SSs send their data to BSs within the initially assigned time-frequency chunk of a frame and from an SS' point of view; the rest of the world could be connected through accessing the BS.

2.1.2. Physical Layer

In its former release; the 802.16 standard addressed applications in licensed bands in the 10 to 66 GHz frequency range. Line of sight (LOS) is necessary in this frequency band, since waves are comparable with millimeters. Waves in this band travel directly; therefore, BS has multiple antennas pointing to different sectors. Figure 2.2 illustrates the system for LOS structure. It is important to note that, even in line of sight structure; modulation schemes for SSs vary due to distance of SSs, thus path loss.

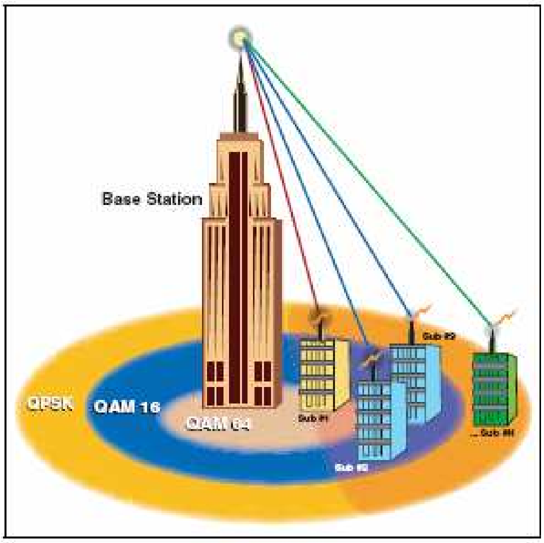
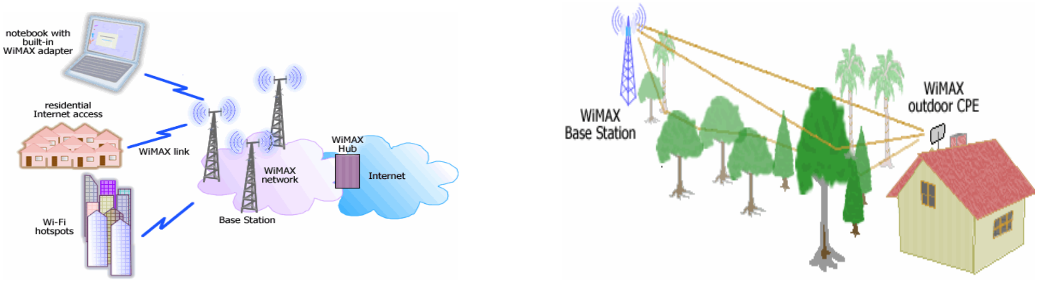


Figure 2.2 Illustration for LOS structure [17]

Subsequent amendments have extended the 802.16 2004 (Fixed WiMAX) air interface standard [1] to cover non-line of sight (NLOS) applications in licensed and unlicensed bands from 2 to 11 GHz bands. The latest amendment 802.16e (Mobile WiMAX) is designed to support mobility. The system illustration for mobile and NLOS structure is given in Figure 2.3.



a) Mobile structure b) NLOS structure

Figure 2.3 WiMAX Illustration [17]

2.1.2.1. Channel Sizes and Frequency Bands

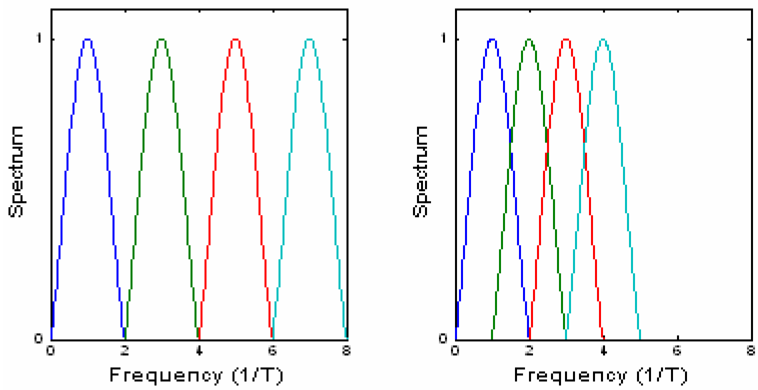
WiMAX standards - both fixed and mobile- do not specify the carrier frequency (2-11 GHz) for OFDM/OFDMA radios and define general limitations for channel sizes (1.25 – 20 MHz). Since neither worldwide spectrum band is allocated nor committed channel size is defined, WiMAX forum [12] defines system profiles for interoperability. Mobile WiMAX System Profile Release 1 is defined as follows: IEEE 802.16 2004, IEEE 802.16e and some optional and mandatory features.

In Release 1, Mobile WiMAX profiles cover 5, 7, 8.75, and 10 MHz channel bandwidths for licensed worldwide spectrum allocations in the 2.3 GHz, 2.5 GHz, 3.3 GHz and 3.5 GHz frequency bands. Among these spectrums, 3.5 GHz band is the mostly available one, except for US [13]. The channel sizes for this frequency band are therefore integer multiples of 1.75 MHz, i.e., 1.75 MHz, 3.5 MHz, 7MHz, 8.75 MHz, etc. Also it is important to note that, frequency reuse technique can be used in order to increase the overall capacity of the system in WiMAX [3].

2.1.2.2. OFDM vs. OFDMA

IEEE 802.16 [1], [2] specifies two types of Orthogonal Frequency Division Multiplexing (OFDM) systems: one of them is simply OFDM and the other is Orthogonal Frequency Division Multiple Access (OFDMA).

OFDM is a multi-carrier transmission technique that has been recently recognized as a method for high speed bi-directional wireless data communication [4]. Frequency Division Multiplexing (FDM) scheme uses multiple frequencies to transmit multiple signals in parallel. In FDM, the allocated spectrum is broken up into several narrowband channels known as "subcarriers". In FDM, frequency bands for each signal are disjoint; therefore simply, receiver demodulates the total signal and separates the bands using filters. In OFDM, frequency band is used more efficiently, since the subcarriers are overlapping. Figure 2.4 shows that the effect of spectral efficiency is obvious.



a) FDM spectra b) OFDM spectra

Figure 2.4 Spectra of FDM/OFDM

Since the subcarriers are *orthogonal* to each other, there is *no interference* between each data carrier [4]. Figure 2.5 illustrates how data is transmitted over OFDM. A number of signals are transmitted over the channel with orthogonal subcarriers. Receiver is able to demodulate the received signal, in which signals are overlapped in the frequency domain, using the orthogonality property.

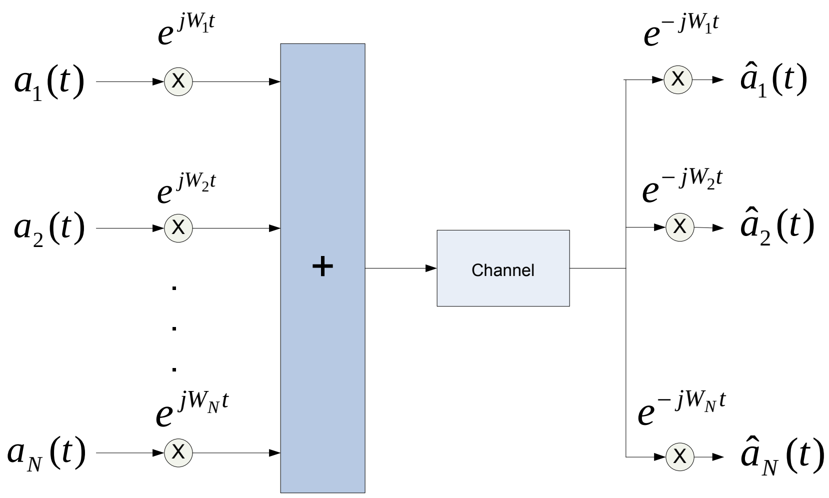


Figure 2.5 OFDM Structure

Table 2.1 gives the definitions and descriptions of the parameters used for OFDM/OFDMA schemes in WiMAX architecture.

Symbol	**Description**	**Symbol**	**Description**
C_{BW}	Channel bandwidth (in Hz)	$T_{frm,u}$	Uplink subframe time (in sec)
F_S	Sampling spectrum (in Hz)	N_{sub}	# of subchannels
n	Sampling factor (constant)	$N_{sub,u}$	# of subchannels for uplink
N_{FFT}	# of subcarriers	$N_{usubcar}$	# of useful subcarriers
Δf	Subcarrier spacing (in Hz)	$C_{subcar}(mod)$	Capacity of a subcarrier for modulation scheme (bits)

T_b	Useful symbol time (in sec)	C_{sym}	Number of bytes that can be carried in a symbol duration (byte)
T_S	Symbol time (in sec)	C_{chunk}	Number of bytes that can be carried in a chunk (byte)
G	Cyclic prefix index	C_{slot}	Number of bytes that can be carried in a slot (byte)
N_{sym}	Number of symbols per frame	C_{frame}	Number of bytes that can be carried in a frame (byte)
$N_{sym,u}$	Number of symbols per uplink subframe	$C_{frame,u}$	Number of bytes that can be carried in an uplink subframe (byte)
T_{frm}	Frame time (in sec)	$R_{d,u}$	Downlink uplink subframe ratio
CR	Coding rate - 64QAM (3/4, 2/3) 16QAM (3/4, 1/2) QPSK (3/4, 1/2) BPSK1/2.	$C_{channel,u}$	Capacity of uplink channel (in bps)

Table 2.1 Definitions of Symbols

Each subcarrier can be modulated with Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 16 Quadrature Amplitude Modulation (16QAM) or 64 Quadrature Amplitude Modulation (64QAM) [14]. Table 2.2 gives the capacity of subcarriers according to their modulation types.

Modulation Scheme	Capacity of a subcarrier (bits)
BPSK	1
QPSK	2
16 QAM	4
64 QAM	6

Table 2.2 Capacity of subcarriers for modulation schemes

In WiMAX OFDM PHY, there are a number of subcarriers spanning the sampling spectrum, meaning OFDM modulation can be realized with Inverse Fast Fourier Transform (IFFT). The standard defines the number of subcarriers as 256 for OFDM. It should be noted that in IEEE 802.16 2004, subcarriers cannot be allocated for different users i.e. subchannelization, which is to group subcarriers, is not defined for downlink but uplink. Therefore in terms of scheduling, according to 802.16 2004, minimum allocation unit of a frame is simply "one" OFDM symbol for downlink. 802.16 2004 allows up to 16 subchannels for uplink. For the OFDMA case, standard [1], [2] defines that a group of subcarriers can be assigned for different users in both uplink and downlink directions. Sampling spectrum (F_s) is defined as follows:

$$F_s = \left\lfloor \frac{n \times C_{BW}}{8000} \right\rfloor \times 8000 \qquad \text{Eq 2.1}$$

where C_{BW} is the channel bandwidth and n is the constant sampling factor which depends on channel size. The subcarrier spacing (Δf); which is the inverse of a useful symbol time (T_u), is defined as the ratio of sampling spectrum to the number of subcarriers. It can be observed that changing the channel bandwidth directly affects the subcarrier spacing. In scalable OFDMA, subcarrier spacing is set to the value of 10.94 kHz, resulting in fixed symbol durations and variable number of subcarriers.

$$\Delta f = \frac{F_s}{N_{FFT}}$$ Eq 2.2

$$T_u = \frac{1}{\Delta f}$$ Eq 2.3

For multipath channels, to cope with channel delay spreads and time synchronization errors, a paradigm called cyclic prefix (CP) is introduced [4]. Figure 2.6 illustrates the relationship between CP and symbol. CP is simply repeating a part of the useful symbol time.

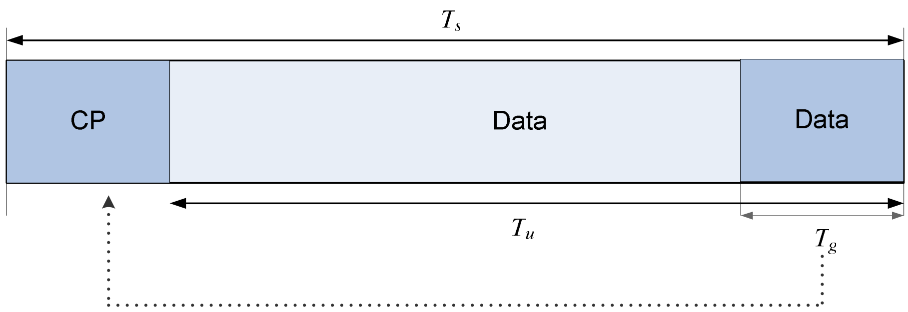


Figure 2.6 Symbol Structure

Therefore, overall symbol time can be defined as follows:

$$T_S = T_u + T_g ,$$ Eq 2.4

where

$$T_g = T_u \times G$$ Eq 2.5

and G is the CP index defined as:

$$G = 0.5^m , m \in \{2,3,4,5\}$$ Eq 2.6

Therefore via Eq 2.7, the number of symbols in a frame can be calculated as:

$$N_{sym} = \left\lfloor \frac{T_{frm}}{T_S} \right\rfloor = \left\lfloor \frac{T_{frm} \times \left\lfloor \frac{n \times C_{BW}}{8000} \right\rfloor \times 8000}{N_{FFT} \times \left(1 + 0.5^{m}\right)} \right\rfloor \quad \text{Eq 2.7}$$

It can be seen from Eq 2.7 that, if the symbol time is not a multiple of frame time, there can be a gap at the end of the frame. Since there cannot be data transmission in this gap (Figure 2.7), it can be defined as an overhead [25].

The number of useful subcarriers is not equal to the number of subcarriers since there are pilot, guard and DC subcarriers. For instance in OFDM, we have totally 256 subcarriers but not all of these subcarriers are energized. There are 28 lower, 27 upper guard subcarriers and a DC subcarrier that are never energized. Also, there are 8 pilot subcarriers that are dedicated for channel estimation purposes. Therefore, only 192 data subcarriers are left for data transmission [25]. For the OFDMA case where the number of subcarriers varies between 128 – 2048, the number of subcarriers which are not used for data transmission is also variable.

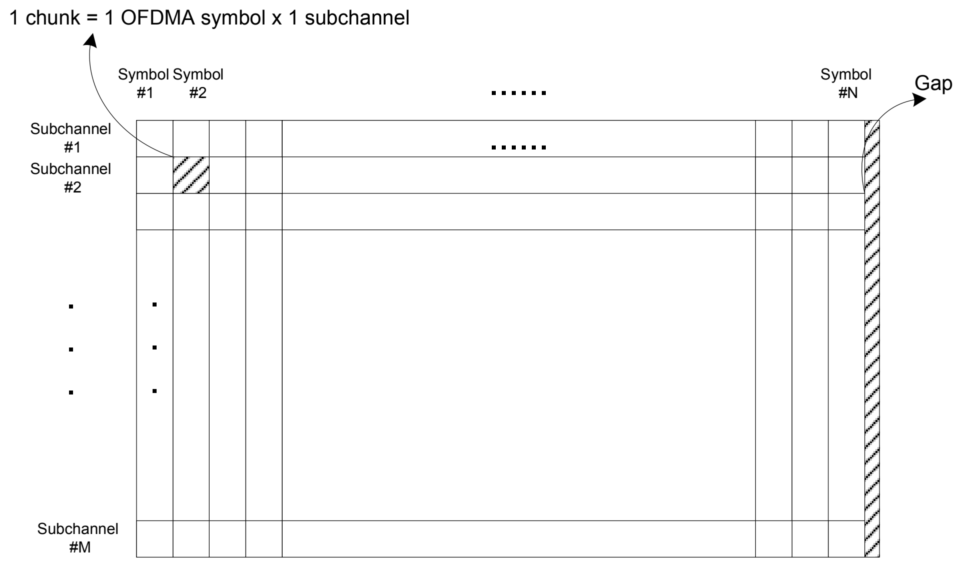


Figure 2.7 Frame Structure (OFDM)

For the OFDM case, in order to calculate the capacity of a chunk (the minimum frequency time unit of a frame), we first need to calculate the capacity of a symbol.

$$C_{sym} = N_{usubcar} \times C_{subcar}(\mathrm{mod}) \times CR \qquad \text{Eq 2.8}$$

Therefore, the capacity of a chunk is:

$$C_{chunk} = C_{sym} / N_{sub} \qquad \text{Eq 2.9}$$

A chunk (Figure 2.7) can be defined as the minimum allocation unit i.e. slot for the OFDM uplink case. On the other hand, for the downlink case C_{chunk} simply equals to C_{sym} since subchannelization is not available in downlink. Table 2.3 gives the capacity of chunks for different subchannel values in OFDM case.

Mod. scheme & coding rates \ $N_{sub,u}$	1	2	4
64 QAM 3/4	108	54	27
64 QAM 2/3	96	48	24
16 QAM 3/4	72	36	18
16 QAM 1/2	48	24	12
QPSK 3/4	36	18	9
QPSK 1/2	24	12	6
BPSK 1/2	12	6	3

Table 2.3 Capacity of a chunk (N_{FFT}=256, OFDM)

Therefore; calculation of the number of bytes that can be carried in a single uplink subframe and calculation of the capacity of the uplink channel are shown in Eq 2.10 and Eq 2.11 :

$$C_{frame,u} = \left(N_{sym,u} \times N_{sub,u}\right) \times C_{chunk} \qquad \text{Eq 2.10}$$

$$C_{channel,u} = {C_{frame,u}}\big/{T_{frm}} \qquad \text{Eq 2.11}$$

The relation between Eq 2.8 and Eq 2.11 shows that modulation schemes and coding rates of SSs directly affect the uplink channel capacity.

However; for OFDMA case, minimum allocation unit (slot) is defined rather different than OFDM case. It is defined that, for downlink Fully Utilized Subchannels (FUSC), a slot is 1 subchannel x 1 OFDMA symbols; yet, for downlink Partially Utilized Subchannels (PUSC) it is 1x2, for uplink PUSC 1x3 and for downlink and uplink adjacent subcarrier permutation 1x1 [2]. In particular, since [28] refers PUSC as the permutation scheme; we use PUSC in our simulations. In this permutation scheme, the subcarriers of a subchannel are spread over the spectrum, thus averaging out the frequency selective fading [16]. In case of PUSC; channel state information (CSI) for each SS for the whole spectrum is sufficient. On the other hand, in case of adjacent subcarrier permutation, CSI for each SS in each subchannel is necessary; since frequency selective nature of the band is still effective. However; adjacent subcarrier permutation allows opportunistic scheduling in terms of bands, since it could benefit from multi-user and frequency diversity in terms of subchannels. It is important to note that PUSC scheme could also take advantage of multi-user diversity in terms of the whole spectrum but not in terms of subchannels. For mobile applications, where channel conditions vary frequently, it is obvious that PUSC scheme will be more effective; since otherwise, CSI overhead would be higher. Contrarily, for fixed applications where channel conditions rarely vary, performing the band adaptive modulation and coding (AMC) will probably result in higher throughput [26].

Particularly, uplink slot definition and illustration is given in this thesis. Illustration of a slot for uplink PUSC is given in Figure 2.8. A slot is composed of 1 subchannel by 3 OFDMA symbols. A subchannel is composed of 6 tiles. Each tile is a region with 4 subcarriers by 3 OFDMA symbols; therefore, a tile is composed of 12 subcarriers i.e. 8 data, 4 pilot subcarriers.

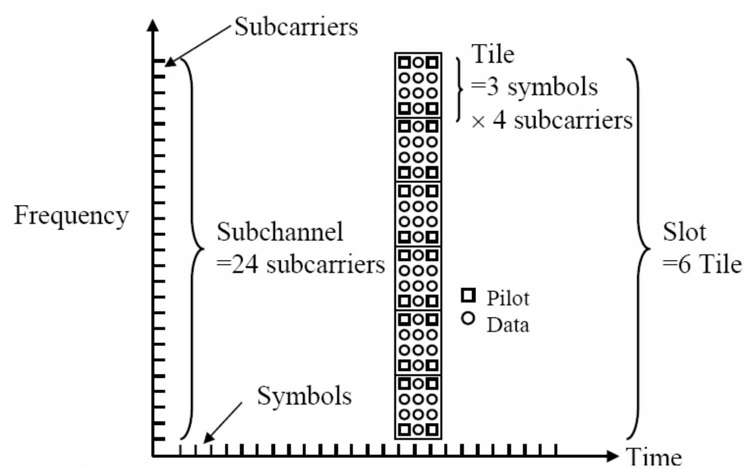


Figure 2.8 Slot definition for uplink PUSC [5]

The capacity of a slot for uplink PUSC is given in Table 2.4.

Modulation scheme and coding rates	**Capacity of a slot (bytes)**
64 QAM ¾	(48*6*(3/4)/8)=27
64 QAM 2/3	24
16 QAM ¾	18
16 QAM ½	12
QPSK ¾	9
QPSK ½	6
BPSK1/2	3

Table 2.4 Capacity of a slot (Uplink PUSC)

$$C_{frame,u} = N_{slot,u} \times C_{slot,u} \text{ ,where} \qquad \text{Eq 2.12}$$

$$N_{slot,u} = N_{sub,u} \times \left\lfloor \frac{N_{sym,u}}{3} \right\rfloor \qquad \text{Eq 2.13}$$

$$C_{channel,u} = {C_{frame,u}}/{T_{frm}} \qquad \text{Eq 2.14}$$

2.1.2.3. Uplink Capacity Illustrations for OFDM and OFDMA

Illustration of OFDM and OFDMA cases' capacity calculations are given in Table 2.5 and Table 2.6.

Parameter	**Value**	**Description**
C_{BW}	7 MHz	Chosen
F_S	8 MHz	Calculated via (Eq 2.1)
n	8/7	8/7 for CBW multiple of 1.75 MHz in OFDM. For OFDMA n=8/7 for all CBW
N_{FFT}	256	Defined by IEEE 802.16 2004
Δf	31250 Hz	Calculated via (Eq 2.2)
T_b	32 us	Calculated via (Eq. 2.3)
G	1/16	Chosen
T_S	34 us	Calculated via (Eq 2.4)
T_{frm}	5 ms	Chosen
N_{sym}	147 (67 DL - 80 UL)	Calculated via (Eq 2.5)
$N_{sub,u}$	4	Slot= 1 OFDM symbol x 1 subchannel for uplink. Slot= 1 OFDM symbol x all subcarriers for downlink.
$N_{usubcar}$	192	Calculated via (Eq 2.6)
$N_{slot,d}$	67	Nsym,u x Nsub,u (# of slots in downlink)
$N_{slot,u}$	80x4=320	Nsym,u x Nsub,u (# of slots in uplink)

Table 2.5 IEEE 802.16 2004 WirelessMAN OFDM illustration

Table 2.5 presents the number of slots that can be allocated for transmission both in uplink and downlink subframes. According to modulation scheme and coding rate parameters given in Table 2.3, the number of bytes that can be carried in an uplink subframe varies between 960 bytes and 8640 bytes; therefore the capacity of an uplink channel varies between 1.5 and 13.8 Mbps. Table 2.6 gives the system parameters for Scalable OFDMA case.

Parameter	**Downlink**	**Uplink**
C_{BW}	10 MHz	
N_{FFT}	1024	
Null Sub.	184	184
Pilot Sub.	120	280
Data Sub.	720	560
N_{Sub}	30	35
Δf	10.94 kHz	
T_b	91.4 ms	
T_bx(1/8) , (G=8)	11.4 ms	
T_s	102.9 us	
T_{Frm}	5 ms	
N_{sym}	48 (30 DL – 18 UL)	
$N_{slot,d}$	30 x (30/2)=450	
$N_{slot,u}$	35 x (18/3)= 210	

Table 2.6 S-OFDMA System Parameters with PUSC Subchannel [3]

Table 2.6 presents the number of slots that can be allocated for transmission both in uplink and downlink subframes. According to the modulation scheme and coding rate parameters given in Table 2.4, the number of bytes that can be carried in an uplink subframe varies between 630 bytes and 5670 bytes; therefore the capacity of an uplink channel varies between 1 Mbps and 9.21 Mbps.

2.1.3. MAC Layer

Figure 2.9 shows the reference model, the scope of the standard and the management entities. The MAC layer of WiMAX is composed of three sublayers. The Service Specific Convergence Sublayer (CS) is defined so as to transform or map the external network data received through the CS Service Access Point (SAP) into MAC SDUs; and to send it through the MAC SAP to the MAC Common Part Sublayer (CPS). Briefly, what CS does, is to classify the MAC SDUs according to their associated connections with Connection Identifiers (CID) and Service Flow Identifiers (SFID). It is important to note that there are multiple CS specifications aiming to provide WiMAX to communicate with protocols (IP, ATM) through the CS interface. It may also include such functions as payload header suppression (PHS) [6].

The core functions of MAC layer such as bandwidth allocation, scheduling, QoS satisfaction, connection establishment; connection maintenance etc. are defined in MAC CPS. The MAC also contains a security sublayer providing authentication, secure key exchange, and encryption. Management of scheduling control messages, data and statistics which are transferred between the MAC CPS and the PHY (via the PHY SAP) is left open for vendor's implementation by the standard [2].

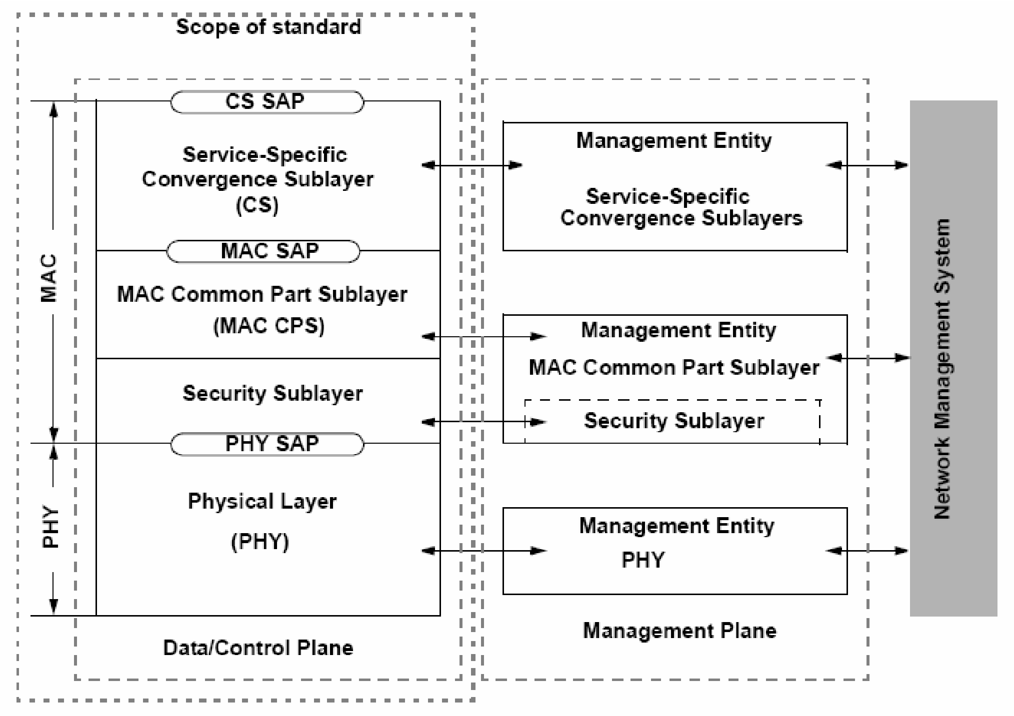


Figure 2.9 Reference Model for WiMAX [2]

Each SS shall have a 48-bit universal MAC address which uniquely identifies and distinguishes the SS from within the set of all possible vendors and equipment types. Since WiMAX is connection oriented, connectionless protocols such as UDP are also transformed into connection oriented flows. MAC associates all connections with a 16 bit CID. Also there is an SFID which identifies the QoS parameters of a flow associated with a CID.

In particular, MAC CPS will be discussed in this thesis. CPS performs construction and transmission of MAC protocol data units (PDUs) which are constituted by MAC service data units (SDUs). The scheduling and retransmission of MAC PDUs, the control signaling for the bandwidth request and grant mechanisms are done in this sublayer. The CPS also performs QoS control.

2.1.4. QoS

There are five service classes defined in the standard [2]: Unsolicited Grant Service (UGS), real-time Polling Service (rtPS), extended real-time Polling Service (ertPS), non-real-time Polling Service (nrtPS) and Best Effort (BE). UGS is designed to support Constant Bit Rate (CBR) applications and real-time service flows that generate fixed-size data packets on a periodic basis such as Voice over IP (VoIP) without silence suppression. On the other hand, rtPS is designed to support real time applications with variable size packets and with periodic nature such as compressed voice, video conferencing, Video on Demand (VoD). The ertPS service class is built on the efficiency of both UGS and rtPS and it is designed for real time traffic with variable data rate in an on-off manner such as VoIP with silence suppression. For data, the nrtPS class is designed to support non real time variable packet size applications such as File Transfer Protocol (FTP) but with QoS guarantees in terms of bandwidth per connection. BE is designed for applications that do not require any QoS commitments such as ordinary WEB surfing. Table 2.7 summarizes these five QoS classes and their parameters.

Class	**Minimum rate**	**Maximum rate**	**Latency**	**Jitter**	**Priority**
UGS		X	X	X	
rtPS	X	X	X		X
ertPS	X	X	X	X	X
nrtPS	X	X			X
BE		X			X

Table 2.7 Service Class Parameters

Each application of each SS has to register the network, where it will be assigned service flow classifications i.e. Minimum Reserved Traffic Rate (MRTR), Maximum Sustained Traffic Rate (MSTR), Maximum Latency (ML), Tolerated Jitter (TJ) and Traffic Priority (TP) with an SFID. QoS mapping and

SFID assignment of connections are done in CS. When a connection requires to send data packets, the service flow is mapped to a connection using a unique CID with its associated SFID. Dynamic Service Activate (DSA), Dynamic Service Change (DSC), Dynamic Service Delete (DSD) are the signaling functions for establishing and maintaining or deleting the service flows. Depending on the QoS needs and number of SSs, the BS sends control messages to SSs which contain the SFID, CID, and a QoS parameter set. The BS sends a control message called a DSA-REQ. The SS then sends a DSA-RSP message to accept or reject the service flow. This mechanism allows an application to acquire more resources when required.

2.1.5. Bandwidth Request Mechanisms

SSs send their requests to BSs using bandwidth request mechanisms. There are two kinds of bandwidth request mechanisms defined in standard. First type of request is realized via Bandwidth Request Header (BRH) and second via MAC Subheader (MSH).

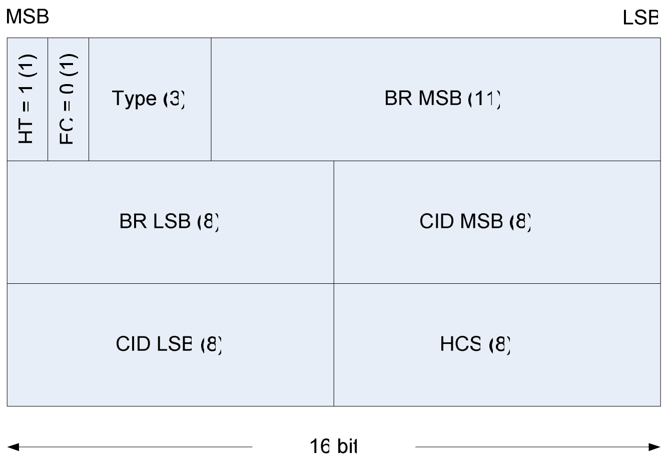


Figure 2.10 Bandwidth Request Header format [2]

Figure 2.10 shows BRH format. It contains 19 bits in order to specify bandwidth request length i.e. requests can be up to 512 bytes. Bandwidth requests with BRH can be contention based or non-contention based. In the non-contention based architecture, BS polls SSs by allocating bandwidth to them to send their bandwidth requests. Unsolicited requests and unicast poll response requests are the non-contention based bandwidth requests. UGS class uses unsolicited requests and rtPS, ertPS classes use unicast poll response requests. Moreover, nrtPS class also uses unicast polls to request bandwidth; but standard specifies a long time interval (500 ms) for unicast polls for this class. nrtPS and BE classes send contention based bandwidth requests. Contention based requests can be broadcast in which all SSs try to send their bandwidth request messages or multicast in which a group of SSs is able to send bandwidth request message. BS allocates contention slots for requesting bandwidth and it is obvious that contention based requests can collide when two or more SSs send requests in a slot. If a grant for a request is not assigned to an SS in a timeout period, the SS uses the exponential backoff algorithm and sends its requests less aggressively.

MAC subheaders in addition to a MAC header could also be used for sending requests i.e. piggybacking requests into MAC PDUs is specified in standard. Poll me bit is another option for requesting a unicast poll in order to send bandwidth request message. Additionally, it should be noted that bandwidth requests using MAC subheaders are optional in the standard. MAC header format is given in Figure 2.11.

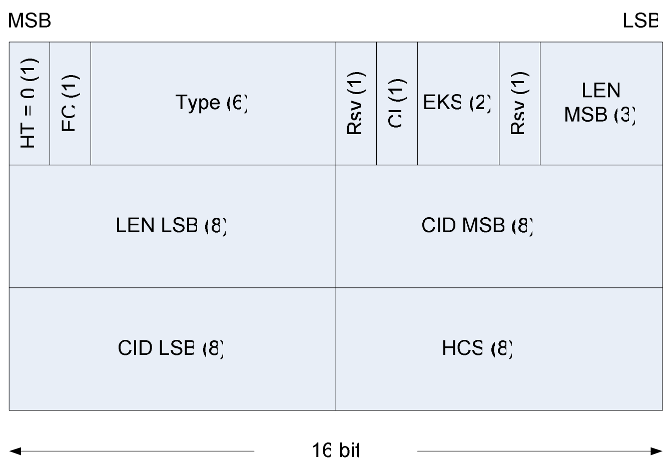


Figure 2.11 MAC Header format [2]

Figure 2.12 presents the signaling between BS and SS for the bandwidth request and grant mechanism.

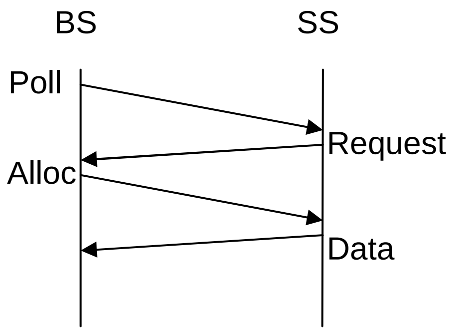


Figure 2.12 Signaling for Bandwidth Request Mechanism

Requests can be incremental or aggregate. Incremental requests indicate new bandwidth requirements whereas aggregate requests indicate the whole bandwidth requirement of a connection. Although bandwidth requests are always per connection, WiMAX standard specifies two modes for granting purposes:

- Grant per Connection (GPC): Bandwidth is granted to each connection explicitly. SS is only responsible to match the granted bandwidths to connections.

- Grant per Subscriber Station (GPSS): Bandwidth is granted to each SS as a whole. In this architecture, redistribution of allocated bandwidth to connections is the responsibility of SSs.

2.2 Related Work

In this subsection, a brief literature survey in the area of QoS scheduling algorithms is given. There are several studies [5],[7],[8],[9],[15],[18],[26] on the WiMAX scheduling that have presented architectures and scheduling disciplines.

One of the researches addressing WiMAX BS scheduling is [7]. The paper claims to propose a solution for the WiMAX base station that is capable of allocating the slots based on the QoS requirements, bandwidth request sizes and the WiMAX network parameters. WirelessMAN OFDM is the PHY layer of the system architecture. The authors have implemented the WiMAX MAC layer in the NS-2 simulator. Several scenarios are demonstrated in the simulator having proven the system ensures the QoS requirements for all service classes. P2MP mode is selected as the operational mode. GPSS is chosen as the mode for grant allocation.

The scheduling discipline for the base station is similar to the Weighted Round Robin in a way that the number of slots allocated to each SS connection, based on the QoS requirement of each station, is the weights of the WRR scheduler. According to the authors, WiMAX scheduling consists of three stages where the first stage is vital - allocation of the minimum number of slots i.e. calculating the minimum number of slots for each connection to ensure the basic QoS requirement. The second stage is the allocation of unused slots, meaning to assign free slots to some connections to avoid the non-work conserving behavior. The authors have defined this stage as inevitable also, since the provider would try to realize this stage to maximize the profit anyhow. The third one is selecting the order of slots; to interleave the slots to decrease the maximum jitter and delay values. The first and the second stages are effective approaches to the scheduler, however; the third stage may have a drawback. Interleaving slots which are assigned to a particular SS will probably increase the overhead of the MAP messages and the effect of interleaving the slots to the MAP messages should be investigated. In addition, the paper does not consider the overloaded cases in terms of number of SSs. The scheduling proposal will become infeasible for the service classes which use non-contention based bandwidth request mechanisms, in case there are greater number of SSs (for instance 80 SSs) using the VoIP model defined in the paper. Since all SSs are assigned at least 1 slot in each and every frame (80 slots consist a frame) in order to send bandwidth request message, the capacity of the system will entirely be used for bandwidth request mechanisms for the case of 80 VoIP users.

In [9], the authors focus on mechanisms that are available in 802.16 systems to support QoS and whose effectiveness is evaluated through simulation. It is suggested that 802.16 technology addresses the market segment of high-speed internet access for the residential customers where broadband services based on DSL or cable are not available; such as rural areas or developing countries. For the SME market, 802.16 will provide a cost effective alternative to existing solutions based on very expensive leased-line services. The task for QoS support in wireless networks is challenging, since the wireless medium is highly variable and unpredictable, both on time dependent and location dependent basis. Authors review and analyze the mechanisms for supporting QoS at the IEEE 802.16 MAC layer. Two application scenarios are simulated to demonstrate the effectiveness of the 802.16 MAC protocol in providing differentiated services to applications with different QoS requirements such as VoIP, videoconference and Web. P2MP mode is used in the study.

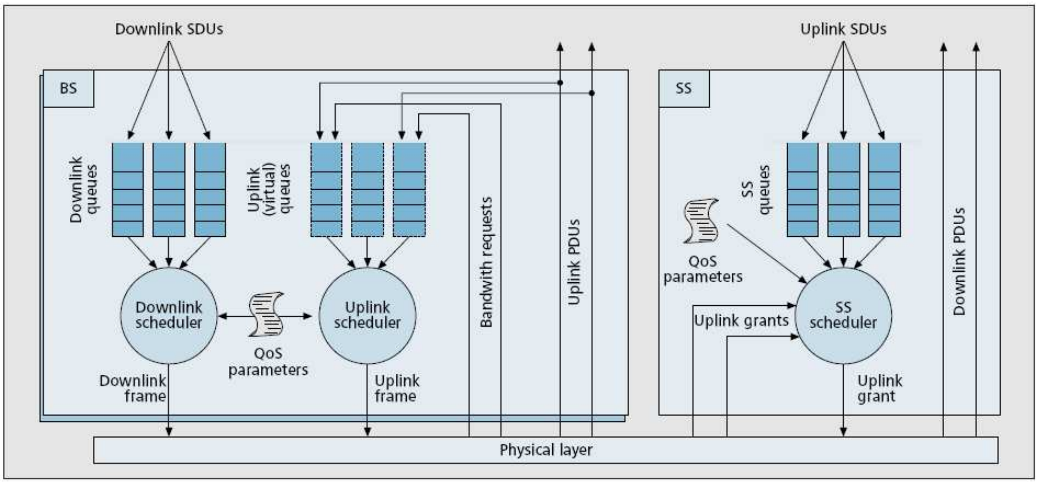


Figure 2.13 BS and SS model for [9]

Figure 2.13 summarizes the system described in the paper. In Figure 2.13, each downlink connection has a queue at the BS. In accordance with the QoS parameters and the status of the queues, the BS downlink scheduler selects from the downlink queues, on a frame basis; the next SDUs to be transmitted to SSs. Uplink connection queues reside at SSs. Based on the amount of bandwidth requested and granted so far, the BS uplink scheduler estimates the residual backlog at each uplink connection. Uplink grants are allocated according to the QoS parameters and the virtual status of the queues. It is also important to note that GPSS mode is used in the study.

DRR is selected as the downlink scheduler, since the size of the head-of-line packet is known at each packet queue. Since estimation of the overall amount of backlog of each connection is done at BS for uplink direction, but not size of each backlogged packet; it is impossible to use DRR as uplink scheduler. Therefore, the authors selected WRR as the uplink scheduler in their 802.16 simulator. Also DRR is selected as the SS scheduler since SS knows the sizes of the head-of-line packets of its queues. Channel conditions and their effects on the overall performance are not studied in the paper. Delay and delay variations are the performance metrics of the analysis. IEEE 802.16 MAC layer is implemented by the authors using the C++ program. The authors of this paper do not study the case with dynamic channel conditions. Delay performances of SSs are given but packet drop rates of SSs are not given. It is assumed that all packets of the SSs are being delayed until they are sent. Outage probabilities of SSs are not considered. Bandwidth request mechanisms for BE and rtPS types of SSs are considered, however the effect of unicast polling (for variable values) intervals to the overall system is not taken into account. Instead, the unicast polling interval for both VoIP and videoconference are fixed to the value of 2 frame times (20 ms). The throughput analyses of the SSs are not given in the paper. Therefore, studying the maximization of the throughput is out of the scope of the paper.

Authors aim at verifying, via simulation, the ability of the WiMAX MAC to manage traffic generated by multimedia and data applications in [8]. Conclusions are drawn for an IEEE 802.16 wireless system working in P2MP mode with Frequency Division Duplex (FDD) and with full-duplex SSs. Three types of traffic sources are used in the simulation scenarios. The data traffic is modeled as a Web source, multimedia traffic sources are chosen as videoconference and VoIP. The downlink scheduler is DRR and uplink scheduler is WRR at BS. SS scheduler is DRR. An SS sends a contention-based bandwidth request to the BS for a BE or nrtPS connection when it becomes busy. It may happen that new SDUs are buffered at a connection while it is busy. Piggybacking type of request is made in this case. Reservation of a minimum amount of contention slots for broadcast polls is a must in their algorithm. Also for rtPS connections unicast polling periods are matched to the SDU interarrival time of multimedia traffic.

Throughput, delay and load partitioning analysis for different scenarios are investigated. Bandwidth request analysis is done for uplink data traffic. Evaluation of multimedia traffic in terms of delay analysis is done. In this paper, authors do not study with dynamic channel conditions as in [9] . Throughput maximization is not realized in variable channel conditions and OFDMA structure is not investigated.

In [27], authors consider the uplink traffic management for rtPS type of connections. They propose a round robin based scheduler which uses leaky bucket principals for QoS management. The proposed scheduler is studied for a various number of scenarios via MATLAB. WirelessMAN OFDM is the PHY structure and near real time video streaming model is the traffic pattern of the proposed architecture. The results show that BS protects SSs who need higher Minimum Reserved Traffic Rate parameters from other SSs which offer traffic to the system much above of their MRTR parameter.

Bandwidth request mechanisms are briefly investigated and the throughput gain for less aggressive bandwidth request mechanisms are shown. It is proven that presented scheduling mechanism satisfies the QoS parameters of SSs even in variable channel conditions. Finally, we show that after satisfying all other service class parameters, making opportunistic scheduling for remaining slots for those connections which have greater modulation schemes and coding rates increases the overall throughput.

In this work authors do not study WirelessMAN OFDMA systems. Their scheduler is based on round robin principals to show that bandwidth allocations are done fairly. Although their scheduler takes channel conditions into account, the architecture does not provide an entire structure taking advantage of variable channel conditions; therefore throughput maximization issue is considered less significantly. Throughput analysis for different scenarios is done but delay variations of packets are not shown. The simulations are done with a particular attention to only one traffic pattern.

In the thesis [26], two types of system architecture, the cellular and the relayed system, envisioned for the next generation wireless system, are considered. For each system, the main target is to produce radio resource allocation and scheduling algorithms that provide good performance with low complexity, making them desirable for practical implementation. The objective of the authors to propose the algorithms is to enhance the fairness among users and reduce service delays, without sacrificing the system throughput. Channel State Information (CSI) is analyzed in terms of scheduling and system overhead. The higher the amount of CSI, the better the scheduling performance is, but the larger the amount of signaling. Adaptive CSI reduction schemes are also developed by the authors. It is important to note that the thesis considers P2MP networks with a PHY description of OFDMA.

Allocation algorithms are developed with a particular attention towards Proportional Fair Scheduling (PFS). While optimal PFS in the MC case is prohibitively complex, the proposed method provides extremely tight bounds with reduced complexity. In this thesis, a group of adjacent subcarriers is defined as the subcarrier permutation and therefore the algorithms given in this thesis benefit from multi-user diversity.

Results show that the proposed algorithms achieve great throughput/fairness trade–off and reduce service delays. Moreover, CSI feedback schemes are proposed, characterized by their flexibility to adapt to the required CSI which varies depending on the scheduler.

Chapter 3

Scheduling Proposals and Environment

In this chapter, the scheduling polices and the simulation environment are given in details. Capacity planning for the simulation types and traffic of connections are also discussed.

In this thesis, among 5 service classes, rtPS type of service class is considered and studied in detail. BS provides periodic unicast bandwidth request opportunities to the rtPS connections. Using these opportunities, the SSs send their bandwidth requests to the BS and they do not use contention request opportunities. Some of the key mandatory traffic parameters for the rtPS service class that are key to our work are Minimum Reserved Traffic Rate (MRTR) (in bps), Maximum Sustained Traffic Rate (MSTR) (in bytes per frame), and Maximum Latency (ML) (in seconds).

MRTR specifies the average bandwidth commitment given to the connection over a large time window. On the other hand, MSTR determines the maximum number of bytes an SS can request in one single frame. The parameter ML specifies the maximum latency between the entrance of a packet to the Convergence Sublayer of the MAC and the epoch at which the corresponding packet is forwarded to the WiMAX air interface [4]. A good rtPS implementation is to ensure the QoS requirements of all rtPS connections, including those that are negotiated at connection setup; such as MRTR, MSTR, and ML. The goal of this thesis is to design an rtPS scheduler for uplink traffic for IEEE 802.16 WiMAX networks.

3.1 System Design Goals and Decisions

Main design goals of this thesis are as follows:

- To propose new low-complexity scheduling algorithms for uplink rtPS type of connections.

- Develop scheduling algorithms such that they can be extended to other service classes and downlink.

- To provide MRTR guarantees for connections using leaky buckets under different channel conditions.

- Introducing packet structure and realistic traffic models into simulations.

- (In addition to satisfaction of each connection's QoS requirements) Using opportunistic and/or fair scheduling in order to maximize the throughput and/or ensure the fairness criteria.

Main design decisions for this thesis are as follows:

- P2MP mode is chosen as wireless network topology since QoS satisfaction for P2MP mode is simpler compared to mesh mode. Figure 3.1 illustrates the designed P2MP mode.

- The scheduling problem for the downlink where the backlog of each SS is known by the BS is not much different than the scheduling problems for wireline networks. Therefore, our focus in this study is the uplink scheduling problem.

- Among 5 different service classes, rtPS class is considered.

- Every SS in Figure 3.1 is assigned one uplink connection; therefore, load partitioning is not studied in this thesis. We do not differentiate between two grant allocation modes i.e. GPC and GPSS, in this study; since we assign only one connection to each SS.

- BS allocates uplink bandwidth to each SS depending on their virtual queues (bandwidth requests) at BS side.

- No matter what the channel condition is, BS calculates the appropriate number of slots to be granted and allocates bandwidth in order to satisfy QoS parameters. It is assumed that there is perfect channel estimation so that BS estimates the true modulation schemes and coding rates of SSs.

- After satisfying all SSs' MRTR parameter, remaining bandwidth is distributed fairly among all users. Additionally, it could be inferred that in order to achieve high bit rates, remaining bandwidth could be scheduled opportunistically to the SSs which have better channel conditions.

- Round Robin (RR) algorithm is used to build up a fair bandwidth allocation mechanism whereas Proportional Fair (PF) algorithm [4], [28] is used to build up a structure such that it considers both fairness and throughput criteria together.

- QoS awareness and channel awareness are both considered in PF algorithm, whereas only QoS awareness is considered in RR algorithm.

- IEEE 802.16m Evaluation Methodology Document [28] baseline assumptions are used in our system level simulation assumptions, traffic models, OFDMA air interface parameters and test scenarios.

- EMD specifies Partially Used Subcarriers (PUSC) for the subchannel permutation in which subcarriers of one subchannel are spread over the whole spectrum, averaging out the frequency selective fading. With this mode, all SSs experience similar channel qualities in all subchannels, therefore scheduling can operate blindly to link qualities in the frequency band. Only time direction (not frequency) channel qualities of SSs are sufficient in such a scheme. It is important to note that, this permutation scheme does not benefit from frequency diversity; however, cost of channel state information is lower.

3.2 Simulation Environment

The simulations are implemented in MATLAB. All simulations are run for a duration of 30 seconds. Not all the procedures and functions of WiMAX environment are implemented; since this study is a concept demonstration and the scope of this thesis is basically on the uplink scheduler and the basic frame structure. DL and UL MAP messages are assumed to be sent in the downlink frame. There is no loss or overhead due to channel conditions and CRC field is not implemented in the simulation. The service class chosen is rtPS and WirelessMAN OFDMA is the physical (PHY) layer of the system. Figure 3.1 defines the environment in terms of functions defined for BS and SSs. If an SS has one or more packets to send when a polling is done by BS; SS sends its bandwidth request to the BS. Bandwidth requests of SSs are maintained by the virtual queues at the BS side. If BS schedules a bandwidth to an SS, SS sends its uplink PDU to the BS through the WiMAX OFDMA PHY.

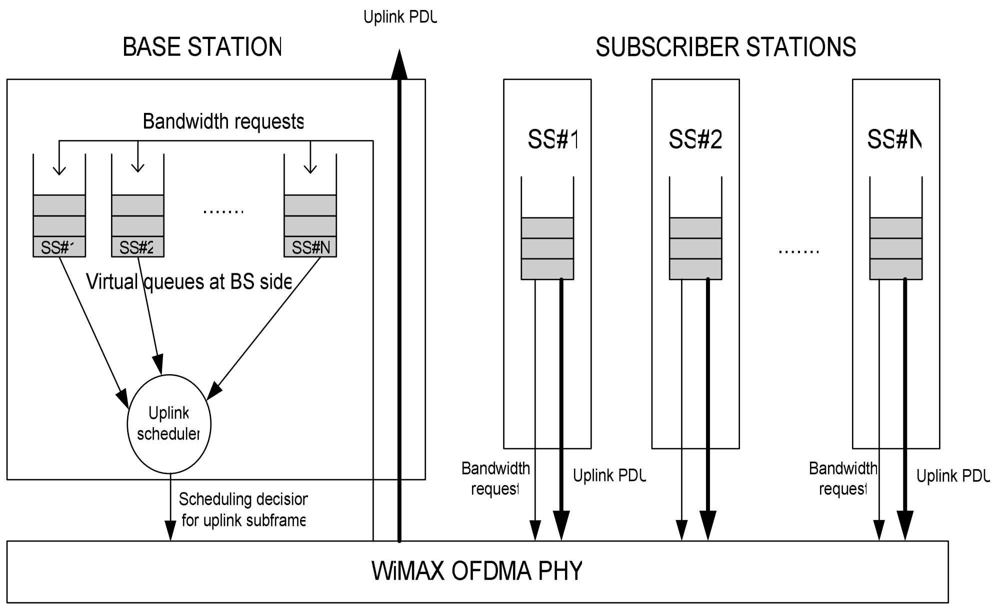


Figure 3.1 Uplink Functions within BS and SSs.

3.3 Capacity Planning Parameters

Capacity of an OFDMA system can be calculated using Eq 2.1 - 2.7 and Eq 2.12 - 2.14. Selected and calculated parameters for the simulations considered in this thesis are given in Table 3.1.

Capacity calculation for the overall system depends on the modulation scheme and coding rates of connections. Table 3.2 provides how the capacity of a slot (minimum frequency time unit of a frame) can be calculated. It is important to note that we assume uplink PUSC as the subcarrier permutation. Therefore, the definition of a slot is similar with the one given in Figure 2.8 i.e. six tiles are defined as one slot. The capacity of a slot (in bytes) for various modulation schemes, coding rates and number of subchannels are given in Table 3.2.

Parameter	Value	Description
C_{BW}	10 MHz	Given by [28]
F_S	11.2 MHz	Calculated via Eq 2.1
N	8/7	8/7 for CBW multiple of 1.75 MHz in OFDM. For OFDMA n=8/7 for all CBW
N_{FFT}	1024	Given by [28]
Δf	10937,5 Hz	Calculated via Eq 2.2
T_b	91.43 us	Calculated via Eq 2.3
G	1/8	Chosen
T_S	102.86 us	Calculated via Eq 2.4-6
T_{frm}	5 ms	Given by [28]
N_{sym}	47 (D:29-U:18)	Calculated via Eq 2.7 (Uplink 15 symbol for data, given by [28])
$N_{sub,u}$	35	(48 data subcarriers = 1 subchannel, [3]) Partially used subcarriers (PUSC) (slot = 3 OFDMA symbols x 1 subchannel)
$N_{slot,u}$	(15/3)x35=175	(15/3) x Nsub,u (# of slots in uplink)

Table 3.1 Parameters for simulation [28]

Table 3.1 presents the number of slots that can be allocated for transmission in the uplink subframe. 15 OFDMA symbols which are assigned for uplink data transmissions should also be used for bandwidth request mechanisms, ranging etc. According to modulation scheme and coding rate parameters given in [28], Table 3.2 presents the number of bytes that can be carried by a single slot.

Modulation scheme and coding rates	**Capacity of a slot (bytes)**
16 QAM ¾	(48*4*(3/4)/8)18
16 QAM ½	12
QPSK ¾	9
QPSK ½	6

Table 3.2 Capacity of a slot in Uplink PUSC

Therefore, number of bytes that can be carried in an uplink subframe varies between 1050 (6*175) bytes and 3150 (18*175) bytes. The capacity of uplink channel vary between 1.68 Mbps (1050*8/(5*10^-3)) and 5.04 Mbps. It is important to note that, [28] specifies the modulation scheme as 16 QAM and QPSK with a coding rate of 1/2 and 3/4.

3.4 Traffic Related Parameters

In order to deal with realistic simulation scenarios and develop a packet aware scheduling algorithm, we consider real traffic models given in [28]. Traffic models used in the simulations are VoIP model, near real time video streaming model and the full buffer model.

3.4.1. VoIP Traffic Model Parameters

Voice over IP (VoIP) refers to real-time delivery of voice packet across networks using Internet protocols. There are a variety of encoding schemes for voice (i.e., G.711, G.722, G.722.1, G.723.1, G.728, G.729, and AMR) that result in different bandwidth requirements. In particular, we use AMR in our simulations. Illustration of a phone call which is composed of active talking periods and silence periods is given in Figure 3.2.

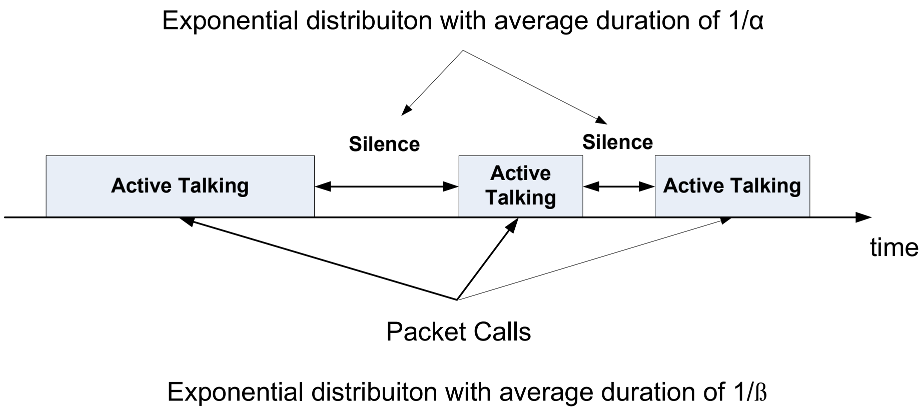


Figure 3.2 Illustration of a phone call [28]

Figure 3.3 shows the Markovian model of 2 state (active and silent) voice session. During each call (session), a VoIP user will be in the active or silence state. The duration of each state is exponentially distributed with a mean of 1.25 seconds.

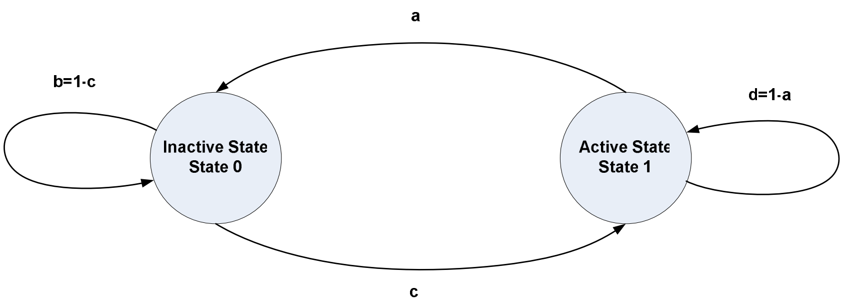


Figure 3.3 Markovian model for state transition [28]

In the model, the conditional probability of transitioning from state 1 (the active speech state) to state 0 (the silent state) while in state 1 is equal to 0.016 (which is denoted as “a” in the Figure 3.3), while the conditional probability of transitioning from state 0 to state 1 while in state 0 is also 0.016 (denoted as “c”). The model is assumed to be updated at the speech encoder frame rate R=1/T, where T is the encoder frame duration (20 ms). The probabilities given above result in concluding that each state duration is exponentially distributed with a mean of 1.25 seconds.

Without header compression, an AMR payload of 33 bytes is generated in the active state every 20 + τ ms and an AMR payload of 7 bytes is generated in the inactive state every 160 + τ ms, where τ is the DL network delay jitter. For the UL, τ is equal to 0. Assuming IP version 4 and uncompressed headers, the resulting VoIP packet size is 81 bytes in the active mode and 55 bytes in the inactive mode [28]. Since this traffic pattern has random packet inter-arrival times and variable packet sizes for each state; it is suitable for rtPS class.

VoIP traffic rate can be calculated as:

$$R_{VoIP} = (1/2)\times(81*8)/(20\times10^{-3}) + (1/2)\times(55*8)/(160\times10^{-3}) = 17.575kbps$$

In this model, a user is defined to have experienced voice outage; if more than 2% of VoIP packets are dropped, erased or not delivered successfully within a delay bound of 50ms.

3.4.2. Near Real Time Video Streaming Model Parameters

Near real time video streaming model is another traffic pattern used in scenarios. This model is the modified version of the model defined in [28]. The main reason for choosing the near real time video model is that this traffic pattern has variable packet lengths and random packet inter-arrival times; hence, it is suitable for the rtPS class. Figure 3.4 illustrates the video streaming model.

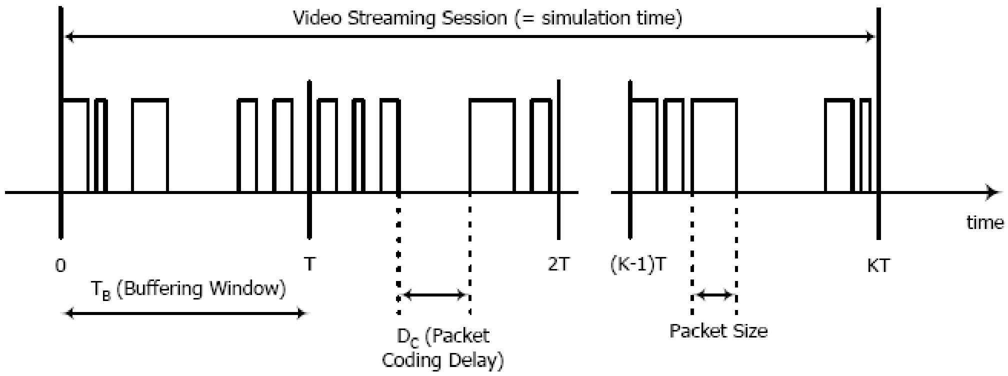


Figure 3.4 Video streaming traffic model [28]

The video streaming model is frame based and generates a deterministic number of variable length packets in a video frame. The traffic model parameters are given in Table 3.3 .

Component	**Distribution**	**Parameters**
Inter-arrival time between the beginning of each frame	Deterministic	100ms
Number of packets in a frame	Deterministic	8 packets per frame
Packet size	Truncated Pareto	Mean = 100 Bytes Mean = 106 bytes (with MAC header) Min = 40 Bytes Max = 250 Bytes
Inter-arrival time between packets in a frame	Truncated Pareto	Mean = 6 ms Min = 2.5 ms Max = 12.5 ms

Table 3.3 Parameters for video traffic model

The mean of the generated traffic per SS, denoted by R_{Video} is calculated as follows:

$$R_{Video} = \frac{8 \times 106 \times 8}{100 \times 10^{-3}} = 67.84 kbps$$

In this model, a user is defined to have experienced outage if more than 2% of video frames (8 packets) are dropped, erased or not delivered successfully within a delay bound of 500ms.

3.4.3. Full Buffer Traffic Model Parameters

Full buffer model is also another traffic pattern considered in the simulations. In the full buffer model, it is assumed that there are infinitely many packets waiting for transmission with a constant packet size of 250 bytes. The full buffer model is implemented to present the effect of other traffic models.

3.5 Scheduling Policies

In WiMAX environment, the BS scheduler assigns slots i.e. bandwidth, to the SSs in each and every frame with a scheduling algorithm. In rtPS class, SSs send their bandwidth requests to the BS in response to the unicast polls for uplink transmission purpose. Since we assume that neither fragmentation nor packing is enabled, the requests of SSs result either in a whole grant for each request or nothing. Therefore, the length of the bandwidth request becomes a critical issue, since there is a higher probability that smaller requests fit into the frame. On the other hand, SSs which send larger requests, have opportunity to send more from their backlog. This tradeoff and the choice of optimal bandwidth request size is not considered in this study. Instead, MSTR parameter is set to 500 bytes which means SSs cannot send bandwidth requests more than the MSTR parameter.

Figure 3.5 illustrates the inner side of uplink scheduler block given in Figure 3.1. Slot assignment block in Figure 3.5 calculates and assigns the appropriate number of slots which are needed for SSs to send their packets in the uplink direction. To build up a good scheduler, slot assignment block; which is the most important part of the scheduler, need to keep in touch with QoS parameter blocks and virtual queues (bandwidth requests) at the BS side.

Minimum Reserved Traffic Rate (MRTR) is a parameter defined in [2], in order to satisfy minimum bandwidth guarantees of SSs which have been initially negotiated between SSs and BS. MRTR specifies the average bandwidth commitment given to a connection over a large time window. In this study, a leaky bucket algorithm is proposed in order to satisfy MRTR requirements of SSs. Maximum Sustained Traffic Rate (MSTR), on the other hand, determines the maximum number of bytes an SS can send in a single frame. Maximum Latency (ML) specifies the maximum latency between the entrance of a packet to the convergence Sublayer of the MAC and the epoch at which the corresponding packet is forwarded to the WiMAX air interface. SSs drop and do not send bandwidth request messages for packets not transmitted within the ML value. Priority parameter is optional and can be used for general purposes.

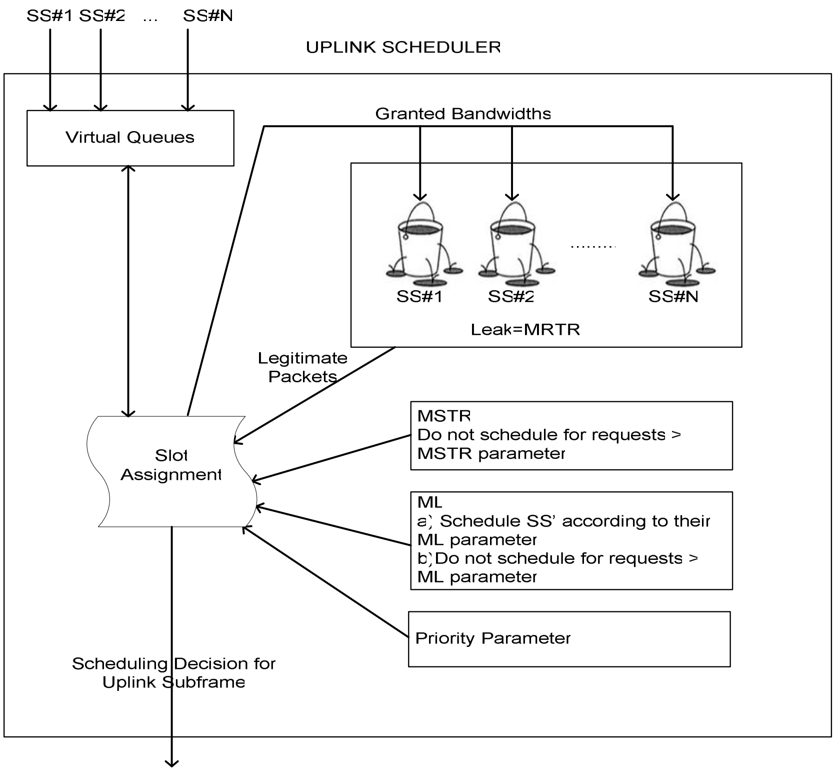


Figure 3.5 Uplink Scheduler

In this thesis, two uplink scheduling algorithms are proposed: First is QoS aware scheduling algorithm which is based on Round Robin principals and the second is both QoS and channel aware scheduling algorithm, based on Proportional Fair policy.

3.5.1. QoS Aware Scheduling Algorithm

QoS aware scheduling algorithm uses the efficiency of leaky bucket and round robin algorithms altogether. This algorithm is developed to serve QoS architecture defined in [2], [28]. Bandwidth guarantees of SSs are satisfied with leaky bucket algorithm whereas fairness issue is considered with round robin scheme. The architecture of the algorithm is given in Figure 3.6.

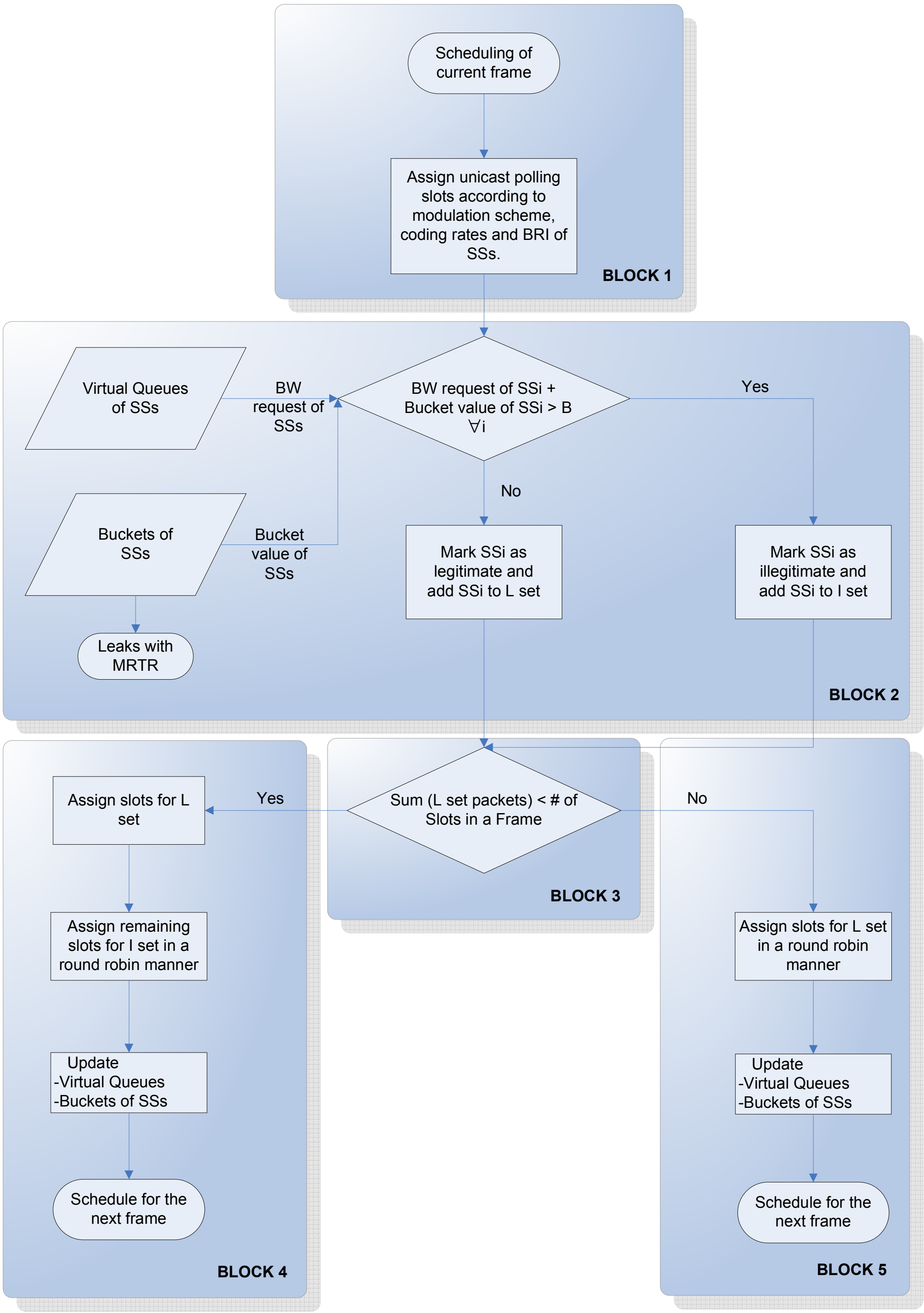


Figure 3.6 QoS aware Scheduling Algorithm

Detailed explanation for the Figure 3.6 is given in Table 3.4. In particular, each row of Table 3.4 illustrates the work done by each block in Figure 3.6.

BS scheduling is done in the following manner for every frame for QoS aware structure:
BLOCK 1- Depending on the initial negotiation by each SS, which determines how frequently unicast polling is done; first, BS assigns unicast polling slots to each SS. The modulation scheme and coding rate of SSs determine the slot size in bytes. According to their slot size, the number of unicast polling slots are calculated and assigned to each SS. For example, for a slot size of 6 bytes, one slot is sufficient for bandwidth requests; since the request's MAC header is 6 bytes without the CRC field. Unicast polls are assigned to each user depending on their bandwidth request index (BRI). When BRI is "n" for an SS, then unicast polling for that SS is done once in every "n" frame.
BLOCK 2- The BS maintains a leaky bucket of a certain size 'BL' for each SS. When an SS has the chance to send its requested data in a certain frame, then the bucket is incremented by the length of the granted data. Moreover, the bucket leaks in each frame by a number of bytes dictated by the connection's MRTR parameter. When a bandwidth request message arrives at the BS, and the sum of the current bucket value and the new bandwidth request exceeds the bucket limit BL; then the bandwidth request is marked 'illegitimate' otherwise 'legitimate'. Then set L, composed of SSs which have legitimate packets and set I, composed of SSs which have illegitimate packets are built up and forwarded to the next block in the Figure 3.6.
BLOCK 3- Let τ_i and B_i denote slot size (in bytes) and bandwidth request (in bytes) of the i^{th} SS, respectively. Then $T_i = \left\lceil \frac{B_i}{\tau_i} \right\rceil$, where T_i is the bandwidth request of the i^{th} SS in number of slots. If $\sum_{i \in L} T_i <= N_{slot}$ do Block 4 otherwise do Block 5.

BLOCK 4- Schedule slots for the set L. It is important to note that there can be additional slots which are not assigned to any SS. This may happen for two reasons: First, there may be no packets in SS' virtual queue at the BS side, second, their bucket may be full; therefore, they may not be eligible for bandwidth assignment in this frame. Those slots ($N_{slot} - \sum_{i \in L} T_i$) are the remaining slots since MRTR parameter for each SS is satisfied. Remaining slots are eligible for all SSs and are distributed fairly among all users in a round robin manner. Figure 3.7 shows how round robin algorithm is inserted to the scheme. In each and every frame, the last SS which has the opportunity to send its backlog is kept in memory and in the newly coming frame, the allocation for remaining bandwidth is started from the SS kept in memory. This structure satisfies the fairness criteria in terms of bandwidth allocation to each SS but it's obvious that SSs having greater packet sizes will experience greater throughputs. (Moreover, to maximize the throughput, remaining slots can directly be assigned for the SSs which have higher modulation schemes and coding rates rather than applying a round robin scheme. But if remaining slots are assigned firstly to the SSs which have greater modulation scheme and coding rate then it is obvious that fairness criteria would not be satisfied.)
BLOCK 5- Schedule the first K legitimate SS' backlog such that $\sum_{i=1}^{K} T_i < N_{slot}$ but $\sum_{i=1}^{K+1} T_i > N_{slot}$ where $i \in L$. And search through the rest of the SSs in order to schedule other SS' packets that would fit into the remaining bandwidth. In addition, first K SSs are changed in each frame in a round robin manner described in Figure 3.7. It is important to note that, the round robin scheme described in Block 5 is the same with Block 4 while their memories (the last SS which had the opportunity to send its backlog) are different.

Table 3.4 Detailed Explanation for QoS aware Algorithm

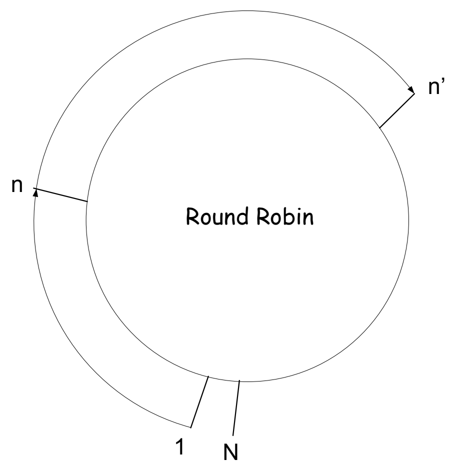


Figure 3.7 Round Robin Scheme

3.5.2. QoS and Channel Aware Scheduling Algorithm

QoS and channel aware scheduling algorithm use the efficiency of leaky bucket and proportional fair algorithms together. This algorithm is developed to serve the QoS architecture defined in [2], [28] and it also considers the channel quality of SSs. Bandwidth guarantees of SSs are satisfied with leaky bucket algorithm; whereas fairness and throughput maximization issues are considered with proportional fair algorithm. The architecture of the algorithm is given in Figure 3.8.

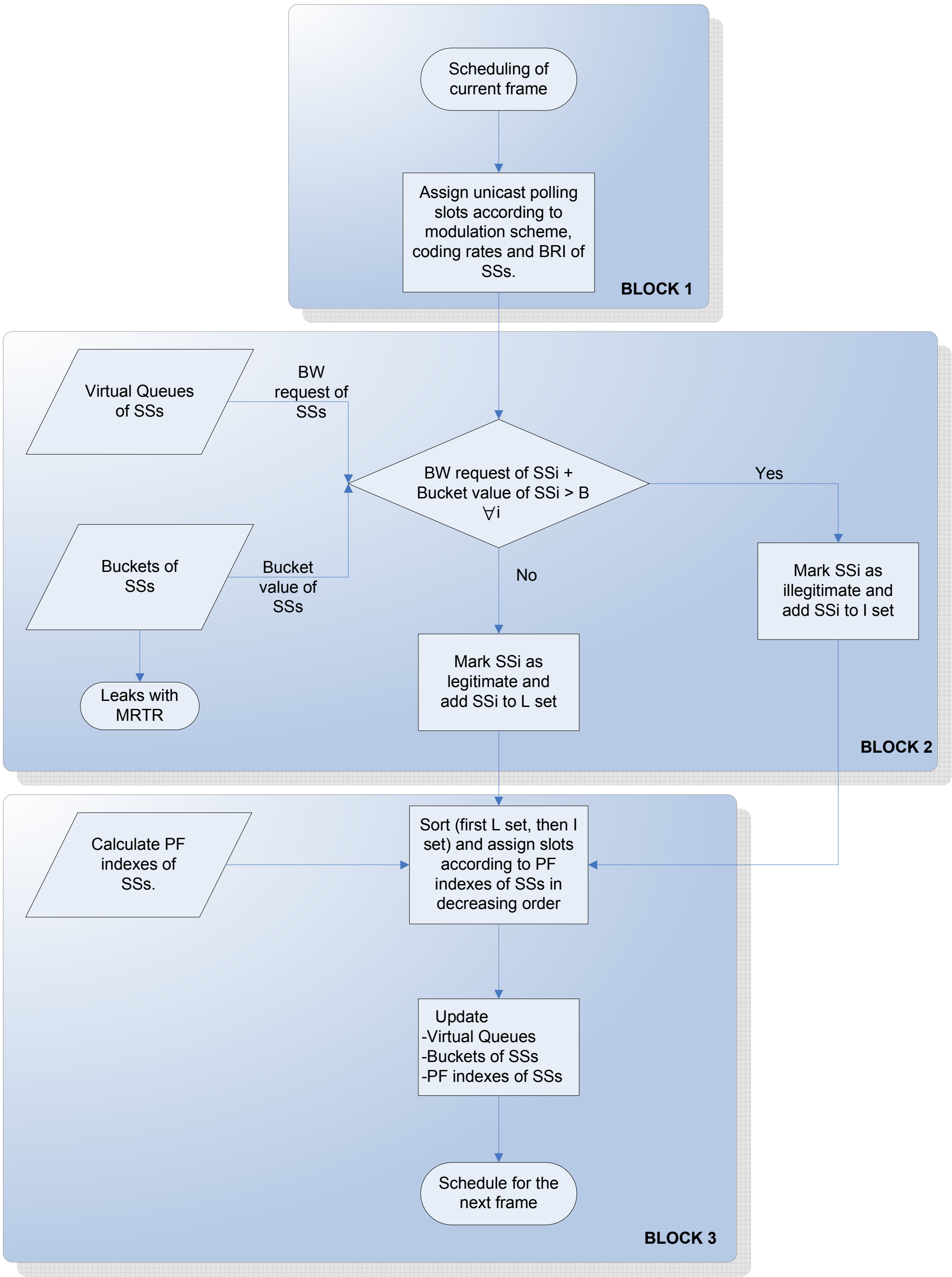


Figure 3.8 QoS and Channel-aware Scheduling Algorithm

Detailed explanation for the Figure 3.8 is given in Table 3.5. Each row of Table 3.5 illustrates the work done by each block in Figure 3.8.

BS scheduling is done in the following manner for every frame for QoS and channel aware structure:
BLOCK 1 - Depending on the initial negotiation by each SS; which determines how frequently unicast polling is done, first, BS assigns unicast polling slots to each SS. The modulation scheme and coding rate of SSs determine the slot size in bytes. According to their slot size, the number of unicast polling slots are calculated and assigned to each SS. For example, for a slot size of 6 bytes, one slot is sufficient for bandwidth requests; since the request's MAC header is 6 bytes without the CRC field. Unicast polls are assigned to each user depending on their bandwidth request index (BRI). When BRI is "n" for an SS, then unicast polling for that SS is done once in every "n" frame.
BLOCK 2 - The BS maintains a leaky bucket of a certain size B for each SS. When an SS has the chance to send its requested data in a certain frame, then the bucket is incremented by the length of the granted data. Moreover, the bucket leaks in each frame by a number of bytes dictated by the connection's MRTR parameter. When a bandwidth request message arrives at the BS, and the sum of the current bucket value and the new bandwidth request exceeds the bucket limit B; then the bandwidth request is marked 'illegitimate', otherwise 'legitimate'. Then set L, composed of SSs which have legitimate packets and set I, composed of SSs which have illegitimate packets are built up and forwarded to the third block in the Figure 3.8.
BLOCK 3 - In each frame after calculating the resultant proportional fair indexes (PFI) of SSs, we sort the SSs according to their PFIs in decreasing order. It is important to note that, we first sort the set L (which correspond to the SSs having legitimate packets since their MRTR parameter is not satisfied yet.) and then set I (which corresponds to the SSs having illegitimate packets). We start assigning slots to SSs according to the order described above; therefore SSs

having legitimate packets are assigned first and then what remains in terms of slots, after legitimate packets are served, is assigned to set I according to their PFIs in decreasing order.

- PFI of each SS is calculated in each frame as follows:

$$PF^{i}(k)=\frac{(R^{i}(k))^{\beta}}{W^{i}(k)}$$

where PF^i (k) is proportional fair index of the i^{th} SS in the k^{th} frame, $R^i(k)$ is the rate of the i^{th} SS in the k^{th} frame, $W^i(k)$ is the long term average rate of the i^{th} SS in the k^{th} frame. β is an index that tunes the fairness of the scheduler. B is assumed to be equal to 1 unless it is it is dictated.

- Updating of Wi (k) in each frame is done as follows:

Let B_i denote the bandwidth request (in bytes) of the i^{th} SS. Then;

$$W^{i}(k+1)=(1-\alpha)\times W^{i}(k)+\alpha\times S^{i}(k)$$

$$S^{i}(k)=\begin{cases}\dfrac{B^{i}(k)\times 8}{T_{frame}} & ,i\in \mathit{Admitted}\\ 0 & ,i\notin \mathit{Admitted}\end{cases}$$

where T_{frame} is the length of the frame (in seconds) and α is the memory index which adjusts the memory of the $W^i(k)$ [4], [28]. In particular α is set to a value of 0.1 and never changed for simulations and W^i (0) = 10 kbps.

Table 3.5 Detailed Explanation for QoS aware Algorithm

Chapter 4

Simulation Results

In this section, we present and discuss the results of simulations. There is one BS and 30 SSs in the area. In all simulation scenarios, 30 SSs (defined in [28]) generate traffic according to the VoIP traffic model, the near real time video streaming model and the full buffer model defined in [28].

All simulations are done within the same environment given in Figure 4.1. The bandwidth and the delay performance of the link between the BS and backhaul is assumed to be greater than the P2MP network; so that the latter would be the bottleneck of the system.

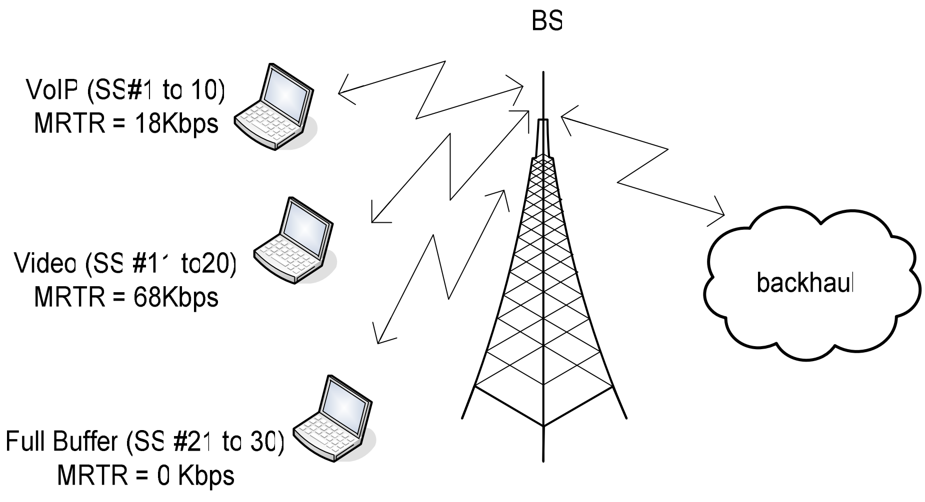


Figure 4.1 Simulation Environment

Minimum reserved traffic rate (MRTR) for the first 10 of 30 SSs (SSs #1 to #10) is 18 kbps. MRTR of SSs #11 to #20 is 68 kbps and MRTR of SSs #21 to #30 is 0 bps. The reason for these choices is to ensure that the scheduler protects or isolates these two sets of SSs from the other SSs. Maximum sustained traffic rate (MSTR) parameter is chosen as 500 bytes per frame; BS does not grant more than 500 bytes for an SS in a frame; therefore SSs do not send more than 500 bytes in a frame and therefore they do not request more than 500 bytes. Maximum latency for VoIP packets and video frames are 50 ms and 500 ms, respectively [28]. The SS drops the packet if a packet's delay is greater than 50 ms for VoIP and it drops the entire video frame if its first packet's delay is larger than 500 ms. It is worth noting that if a video frame starts transmission within 500ms, then the successive packets of that video frame are never dropped and always wait for transmission; a cross layer work is required to implement such a scheme. Subsections 4.1 and 4.2 present the simulation results of the proposed algorithms. Subsection 4.1 considers the QoS aware scheduling and QoS & channel aware scheduling algorithms. Simulations for five different scenarios are carried out in this subsection. Subsection 4.2 presents the simulation results for different Bandwidth Request Indexes (BRI). It is important to note that Subsection 4.1 assumes that SSs send their bandwidth requests to the BS in each and every frame (BRI=1). Subsection 4.3 discusses the simulation results and presents a comparison between two proposed algorithms.

4.1 Performance Evaluation

In this subsection of the thesis, QoS aware scheduling algorithm (Scheduler 1) and QoS & channel aware scheduling algorithm (Scheduler 2) are considered and compared. The performance of the system in terms of throughput and delay is analyzed under different scenarios. The sub – subsections present the analysis under static and dynamic channel conditions.

4.1.1. Static Channel Conditions

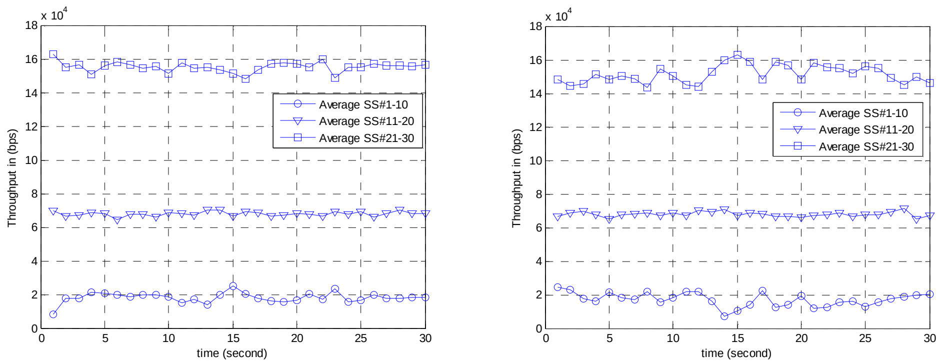
We propose two different scenarios for the analysis of static channel conditions under Scheduler 1 and Scheduler 2. In the first scenario, all SSs use same modulation scheme and coding rate, which do not change in the given simulation time. SSs' modulation schemes and coding rates do not vary in the second scenario either; however, in that, SSs use different modulation schemes and coding rates.

4.1.1.1. Scenario 1

In this scenario, BS always sends unicast polls for each and every SS in every frame, so that BS is informed about the requests of each user immediately after the packets are generated by SSs. All SSs in this scenario use 16 QAM 1/2; therefore, the capacity of a slot is 12 bytes (see Table 3.2), the number of bytes that can be carried in an uplink subframe is 2100 (175*12) bytes and capacity of uplink channel is 3.36 Mbps. Yet, since 30 slots are assigned for bandwidth requests, in each uplink subframe, there are 145 (175-30) slots available for data transmission. Thus, the available capacity for uplink data transmission is 1740 (145*12) bytes in an uplink subframe, which amounts to an uplink rate of 2.784 Mbps. It is important to note the following: After satisfying all SSs' MRTR requirement (10x68 kbps+ 10x18 kbps=0.86 Mbps), there exists a remaining bandwidth. This remaining bandwidth will be used for SSs which are characterized with full buffer traffic models.

Figure 4.2 presents the simulation results of Scheduler 1 and Scheduler 2 under Scenario 1. In particular, Figure 4.2 gives "throughput versus time" graphics for all SSs, with grouping SSs that offer the same traffic patterns. The subfigures (a) and (b) of Figure 4.2 show that all SSs with QoS requirements possess a throughput indicative of their MRTR parameters. The first 10 SSs which generate traffic according to a VoIP session, have the throughput around

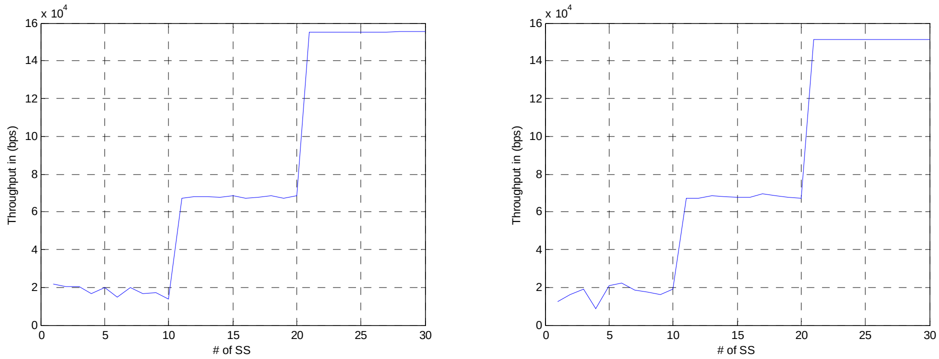
1.8 kbps each; whereas second 10 SSs (SS#11 to 20), which generate video streaming traffic model, have the throughput around 68 kbps each. It is vital to say that both Scheduler 1 and Scheduler 2 protect the first 20 SSs from the rest that is offering a much larger amount of traffic than their MRTR value (which is zero) using the leaky bucket mechanism. Figure 4.2 also shows that when SSs # 1 – 20 have less packets to send, then it is obvious that the remaining bandwidth for SS # 21- 30 increases; therefore their throughput is higher in these cases. This situation could be observed in the 23rd second of the simulation in Figure 4.2 (a) and 14-17th seconds of Figure 4.2 (b).



a) Scheduler 1 b) Scheduler 2

Figure 4.2 Throughput vs. time Scenario 1

Figure 4.3 presents the “Throughput versus # of SSs” plots for Scheduler 1 and Scheduler 2. In Figure 4.3, both (a) and (b) show that the remaining slots are distributed fairly among the last 10 SSs (SS #21 - 30). It is important to note that in this scenario when using Scheduler 1, the overall throughput of the system is 2.3985 Mbps; overheads due to (i) bandwidth request headers, (ii) partial fitting of bandwidth requests to an integer number of slots (because of ceiling T_i) and (iii) unused slots are 0.576 Mbps, 0.0381 Mbps, and 0.3474 Mbps respectively. On the other hand, when using Scheduler 2, the overall throughput of the system is 2.3498 Mbps; overheads due to (i) bandwidth request headers, (ii) partial fitting of bandwidth requests to an integer number of slots and (iii) unused slots are 0.576 Mbps, 0.0512 Mbps, and 0.3828 Mbps respectively.

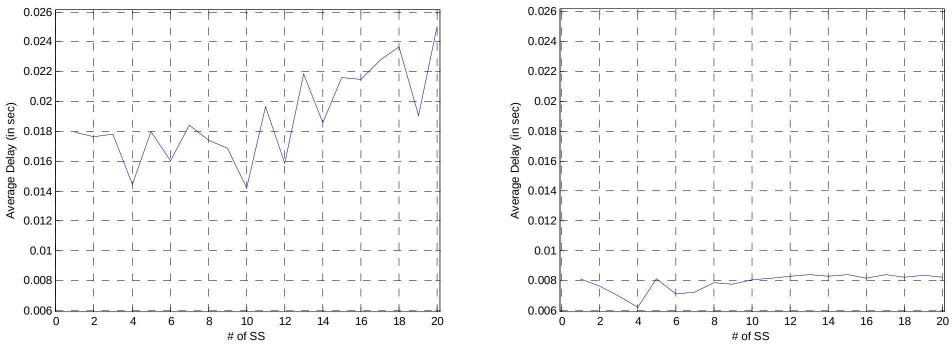


a) Scheduler 1 b) Scheduler 2

Figure 4.3 Throughput vs. SSs Scenario 1

Average delay variations of SS # 1-20 are given in Figure 4.4 for both Scheduler 1 and Scheduler 2. It is obvious that, the time between entrance of a packet to MAC layer of an SS and the epoch at which a grant for that packet is scheduled, is greater than the frame duration (5 ms); since there should be a bandwidth request message sent to the BS before an uplink grant occurs. Figure 4.4 also shows that, Scheduler 2, which uses a variant of the proportional fair algorithm, performs better in overall, compared to Scheduler 1 in terms of delay variations. The fluctuations in average delay variations of results for Scheduler 1 can be explained as follows: Since SSs with full buffer traffic model have always illegitimate packets at the virtual queues at the BS side; most probably the last SS kept in the memory when the round robin scheme is performing, would be one of the last 10 SSs. First 20 SSs have usually legitimate packets but from time to time they may have illegitimate packets. In those cases illegitimate packets with VoIP and video traffic model SSs are mostly stacked (since the last SS kept in the memory usually being one of the last 10 SSs) and should have to wait for a turn around for all SSs. The reason for the fluctuations is that this situation may or may not occur for some SSs due to randomness of traffic patterns. The average delay variations of SSs for Scheduler 1 converge to the Scheduler 2 when SSs with an MRTR parameter (video or VoIP) never have illegitimate packets.

Figure 4.4 (b) shows that average delay variations for SSs using VoIP traffic model also have some fluctuations, since packet interarrival times for VoIP traffic are not random during an active or passive state; therefore during each state, packets experience constant delays. The length of the active and passive states are exponentially distributed, thus some SSs may experience longer delays in a simulation time. It can be inferred that these fluctuations would be greater for greater bandwidth request indexes, since the variation of constant packet delay values will have a greater range.



a) Scheduler 1 b) Scheduler 2

Figure 4.4 Average Delay vs. SSs Scenario 1

4.1.1.2. Scenario 2

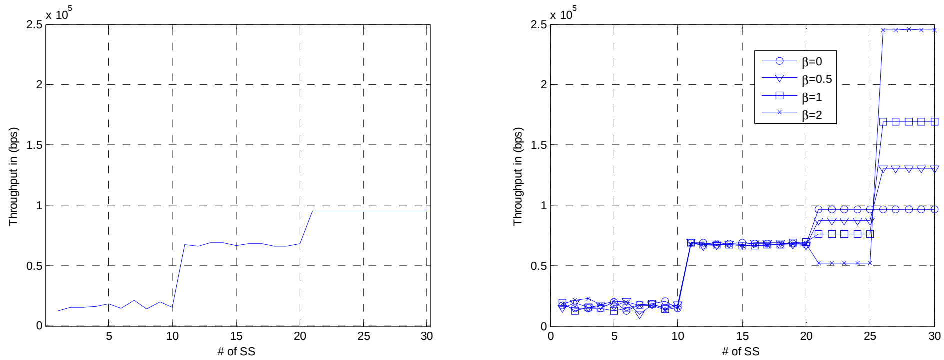
In this scenario, SS #1 to 20 use 16 QAM 1/2, SS #21 to 25 use QPSK 1/2 and SS #26 to 30 use 16QAM 3/4 as the modulation schemes and coding rates during the simulation. The modulation schemes and coding rates of SSs in this scenario are similar to Scenario 1 except for the last 10 SSs.

Figure 4.5 presents the simulation results of Scheduler 1 and Scheduler 2 under Scenario 2 and gives throughput versus SS number plots. We show that all SSs receive a throughput corresponding to their MRTR parameters in this scenario as well. Throughput vs. time and average delay vs. time graphics for this scenario are similar in manner to the first scenario; hence, for the sake of brevity, we do not give these figures.

Scheduler 1 in Figure 4.5 (a) shows that, the remaining slots are distributed fairly among the last 10 SSs (SS #21 - 30). On the other hand, Scheduler 2 in Figure 4.5 (b) shows that, the remaining slots are distributed somehow fairly in order to increase the throughput (when $\beta \neq 0$). The throughput of the last 10 SSs in Figure 4.5 (a) is less then the throughput of SS #26 to 30 in Figure 4.5 (b), on the other hand, it is greater than the throughput of SS #21 to 25 in Figure 4.5 (b) (when $\beta \neq 0$). The proportional fair scheme yields an increase in the overall throughput; therefore, remaining bandwidth has been mostly used for SSs having higher modulation schemes and coding rates.

The proportional fair algorithm parameter β tunes the fairness of the scheduler. Figure 4.5 (b) presents the throughput of Scheduler 2 for various β values. We show that if β is equal to 0, then the Scheduler 2 does not consider the channel conditions of SSs. Therefore, the results given in Figure 4.5 (a) and Figure 4.5 (b) (β =0) are very similar to each other. The larger the parameter β is, the more throughput the system offers but at the expense of a relatively less fair bandwidth allocation.

In this scenario, when using Scheduler 1, the overall throughput of the system is 1.787 Mbps; on the other hand, when using Scheduler 2 (β =1), the overall throughput of the system is 2.055 Mbps; thus higher. For the Scheduler 2, when β =0, β =0.5, β =2 the overall throughput of the system is 1.814 Mbps, 1.923 Mbps, 2.342 Mbps, respectively.



a) Scheduler 1 b) Scheduler 2

Figure 4.5 Throughput vs. SS number for Scenario 2

4.1.2. Dynamic Channel Conditions

We propose three different scenarios for the analysis of QoS aware and QoS & channel aware scheduling algorithms under dynamic channel conditions. In the third scenario, SSs' modulation schemes and coding rates vary according to a proposed pattern; whereas in the fourth scenario, it is assumed that the modulation schemes and coding rates of all SSs change in each and every frame with respecto to a uniform distribution. In the fifth scenario, a large scale fading model is proposed to analyze the performance of the schedulers.

4.1.2.1. Scenario 3

In this scenario, SSs use one of the following modulation schemes and coding rates during a simulation run according to Table 4.1: 16QAM 3/4, 16QAM 1/2, QPSK 3/4, QPSK 1/2.

SS numbers	Simulation Time			
	0-7.5 sec	7.5-15sec	15-22.5sec	22.5-30sec
SS # 1-10	18 bytes (16 QAM3/4)	18 bytes (16 QAM3/4)	12 bytes (16 QAM1/2)	12 bytes (16 QAM1/2)
SS # 11-20	18 bytes (16 QAM3/4)	18 bytes (16 QAM3/4)	12 bytes (16 QAM1/2)	12 bytes (16 QAM1/2)
SS # 21-30	6 bytes (QPSK1/2)	9 bytes (QPSK3/4)	6 bytes (QPSK1/2)	9 bytes (QPSK3/4)

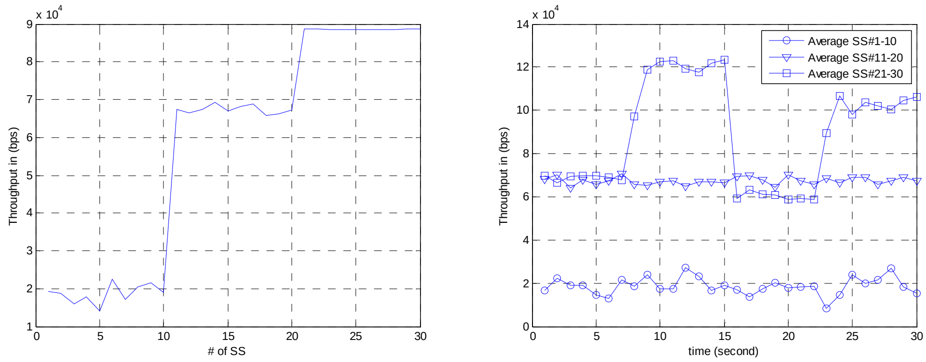
Table 4.1 Slot sizes of SSs according to Simulation Time

Figure 4.6 presents the simulation results of Scheduler 2 under Scenario 3. The results for Scheduler 1 are similar in manner to the results of Scheduler 2, hence they are not presented.

"Throughput versus # of SSs" plot (Figure 4.6 (a)) shows that all SSs have the throughput of their MRTR parameter. The first 10 SSs which generate traffic according to a VoIP session have the throughput around 1.8 kbps each, whereas the second 10 SSs (SS#11 to 20) which generate video streaming traffic model, have the throughput around 68 kbps each. We conclude that the BS scheduler protects the first 20 SSs from the rest; even in dynamic channel conditions.

In accordance with the results in "Throughput versus time" graphics (Figure 4.6 (b)), we conclude that if SSs' modulation schemes and coding rates are higher; then the number of slots for satisfying their minimum bandwidth guarantees will be less. Hence, the remaining bandwidth for transmissions of the last 10 SSs is greater in those cases.

The throughput values of Scheduler 1 and Scheduler 2 are roughly the same contrary to the expectations that Scheduler 2 must have performed better due to its channel aware structure. From the results; it can be deducted that when the channel conditions change slowly, Scheduler 1 and Scheduler 2 perform similarly.



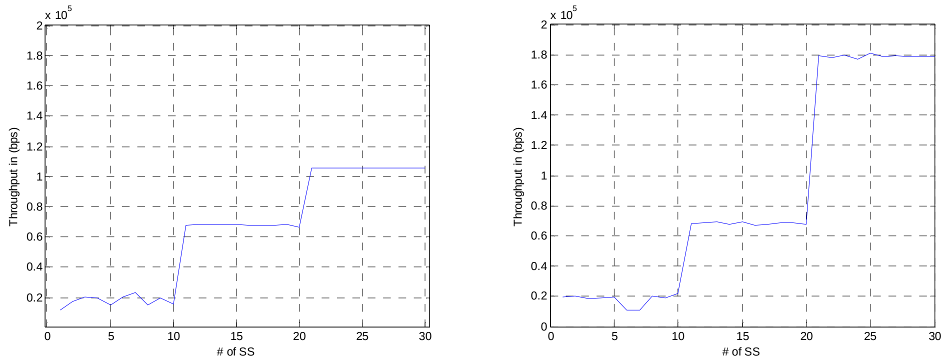
a) Throughput vs. SSs b) Throughput vs. time

Figure 4.6 Simulation Result for Scenario 3

4.1.2.2. Scenario 4

In this scenario, we assumed that, the modulation scheme and coding rates of all SSs are changing in each and every frame with a uniform distribution. Every SS chooses one of the following modulation schemes and coding rates in each and every frame by a uniform distribution with a space of: 16QAM 3/4, 16QAM 1/2, QPSK 3/4, QPSK 1/2.

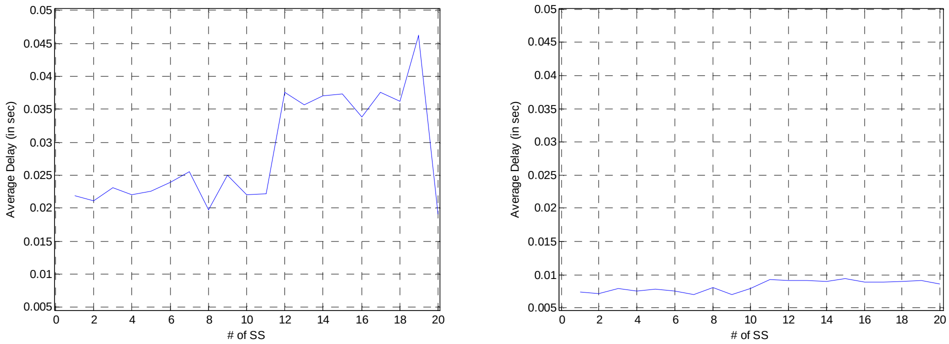
Figure 4.7 presents the simulation results of Scheduler 1 and Scheduler 2 under Scenario 4, and it gives “throughput versus # of SSs” graphics. Figure 4.7 (a) and (b) show that, all SSs have the throughput of their MRTR parameters. “Throughput vs. time” graphics for this scenario are similar in manner to the first scenario.



a) Scheduler 1 b) Scheduler 2

Figure 4.7 Throughput vs. SSs Scenario 4

When we consider the last 10 SSs, Scheduler 2 performs better than Scheduler 1. Moreover, since the average slot size in a simulation run is similar for the last 10 SSs, the remaining bandwidth is distributed fairly among them. It is important to note that in this scenario when using Scheduler 1, the overall throughput of the system is 1.898 Mbps, on the other hand, using Scheduler 2, the overall throughput of the system is 2.638 Mbps. Intuitively, this result was expected; since the second scheduler considers the channel conditions i.e. modulation schemes and coding rates of SSs, while making a scheduling decision. Figure 4.8 considers the average delay performances of SSs. Besides the better performance in throughput, proportional fairness scheme also performs better in terms of average delay compared to the round robin scheme.



a) Scheduler 1 b) Scheduler 2

Figure 4.8 Throughput vs. SSs Scenario 4

4.1.2.3. Scenario 5

In this scenario, we propose a model to show the effect of large scale path loss to the system. A cellular configuration is assumed and the simulation is run for only one cell defined in [28]. The dimensions and structure of the cell is presented in Figure 4.9.

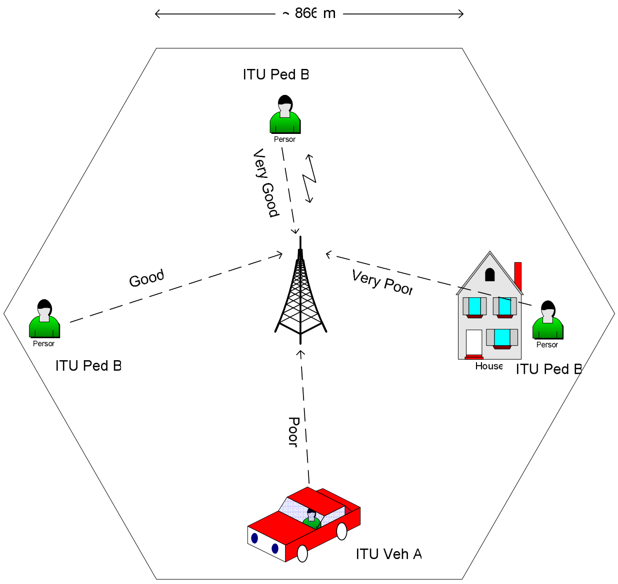


Figure 4.9 Structure of the cell

Similar to the other scenarios, there are 30 SSs (10 users with VoIP, 10 users with video and 10 users with full buffer traffic model) in this scenario as well. 30 SSs are assumed to be randomly distributed inside the cell before the beginning of the simulation run.

According to [28], there are two kinds of SSs: ITU Pedestrian B (3 km/hr) and ITU Vehicular A (120 km/hr). Half of the each 10 SSs with different traffic model types is assumed to be ITU Ped B and the other half is assumed to be ITU Veh A. The simulation environment for this scenario described above is presented in Figure 4.10.

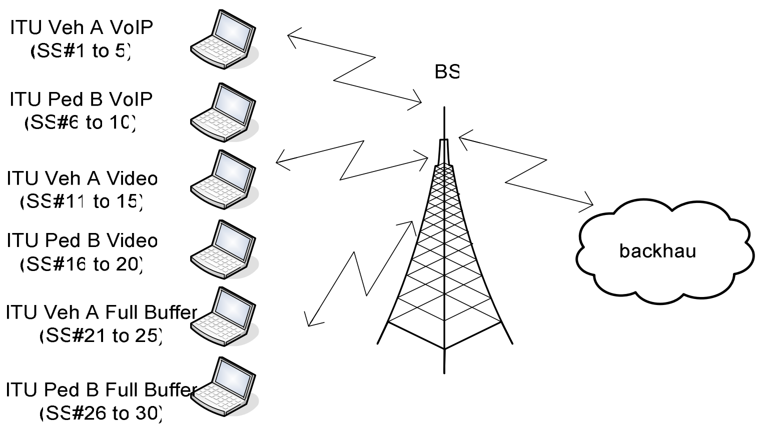


Figure 4.10 Simulation Scenario 5

Each SS is assumed to be moving in one of the 8 directions (horizontally, vertically and diagonally) from the initially randomly assigned starting point. The moving direction is also randomly assigned from the possible set of moving directions. ITU Veh A with the speed of 120 km/hr moves 1000 meters from the starting point within the simulation time, on the other hand, ITU Ped B with 3 km/hr speed can only move 25 meters from the starting point of its own. Distance of SSs from BS during a simulation run is calculated in each frame according to this scheme.

Path loss model (given in Eq 4.1) defined in [28] is used to calculate the large scale path loss for each user in each frame according to their distance from BS (R).

$$PathLoss(dB) = 130.62 + 37.6\log_{10}(R) + X_{\sigma} \qquad \text{Eq 4.1}$$

Shadowing effect is modeled and inserted into the system (X_σ) to consider the surrounding environmental clutter, thus to distinguish the path loss of two different points with same distance to BS [20]. Lognormal distribution is used to model the shadowing effects with a standard deviation of 8dB [28]. It is important to note that shadowing effect is updated in every 50 meters [28]. Eq 4.2 shows the received power in dB after the large scale fading model where P_R is the received power and P_{SS} is the transmitted power of an SS (23 dBm [28]).

$$P_R(dB) = P_{SS}(dB) - Pathloss(db) \qquad \text{Eq 4.2}$$

$$SNR(dB) = 10\log_{10}\left(\frac{T_s \times P_R}{N_0}\right) - IL(dB) - NF(dB) \qquad \text{Eq 4.3}$$

SNR of each user is calculated with Eq 4.3, where Ts is the symbol time, N_0 is the power spectral density of the noise (-174 dBm/Hz), IL is implementation loss (5 dB), NF is noise figure (8dB). The modulation scheme and coding rates of SSs are calculated in each frame according to their SNR value via Table 4.2. The modulation scheme and coding rates of SSs are kept in a two dimensional vector on a frame and SS basis.

Modulation Scheme and Coding Rate	Receiver SNR (dB)
QPSK 1/2	5
QPSK 3/4	8
16 QAM 1/2	10.5
16 QAM 3/4	14

Table 4.2 Receiver SNR assumptions [2],[28]

After channel simulator is run, slot sizes (which corresponds to the modulation schemes and coding rates) of SSs are generated for each and every frame. The average slot sizes for SSs for a whole simulation run is given in Figure 4.11.

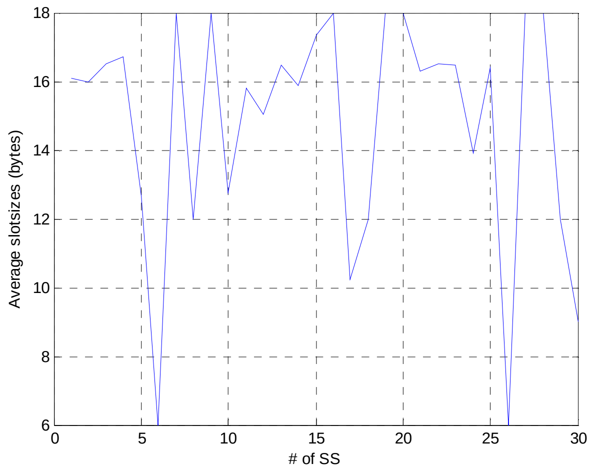
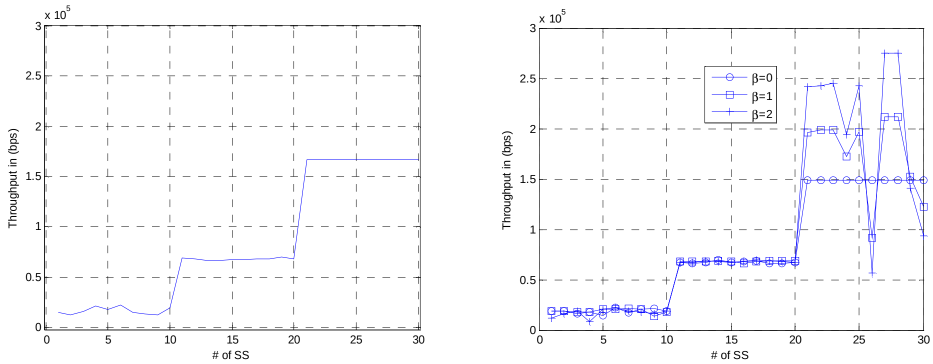


Figure 4.11 Average slotsizes vs. SSs Scenario 5

Simulation results for Scheduler 1 and Scheduler 2 in terms of Throughput vs. SSs, Throughput vs. Simulation time and Average delay vs. SSs are given in Figure 4.12, Figure 4.13 and Figure 4.14 respectively.

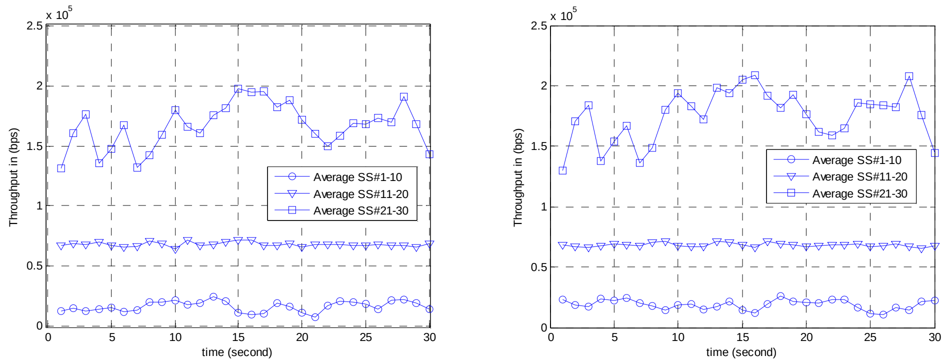


a) Scheduler 1 b) Scheduler 2

Figure 4.12 Throughput vs. SSs Scenario 5

Figure 4.12 shows that both Scheduler 1 and Scheduler 2 satisfy the MRTR parameters of SSs. The overall throughput of Scheduler 1 is 2.47 Mbps, on the other hand, the overall throughput of Scheduler 2 when β=0, β=1 and β=2 is 2.34 Mbps, 2.55 Mbps and 2.72 Mbps respectively. We observe that, when using Scheduler 1, the remaining bandwidth is distributed fairly among the last 10 SSs that do not have MRTR parameters. However, Scheduler 2 (when β=1 and β=2) favors SSs with better channel conditions. Therefore, SSs having higher modulation schemes and coding rates experience higher throughputs compared to Scheduler 1.

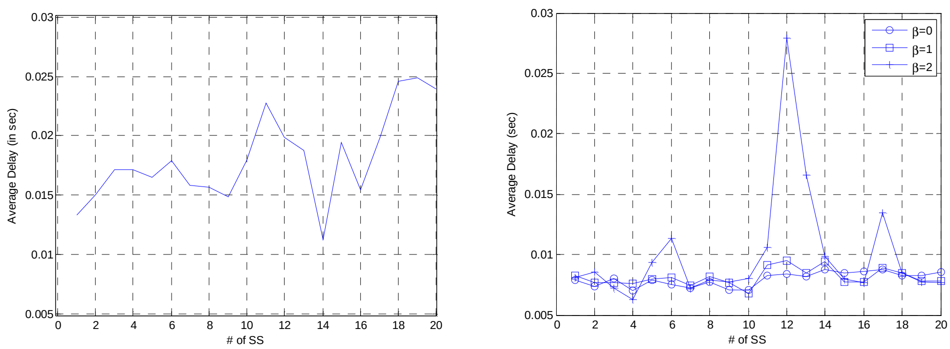
Figure 4.13 presents the effect of large scale fading in the time domain. When SSs have higher modulation schemes and coding rates; the throughput of the last 10 SSs increases, otherwise they decrease.



a) Scheduler 1 b) Scheduler 2(β=1)

Figure 4.13 Throughput vs. time for Scenario 5

Figure 4.14 presents the effect of path loss model to the average delay. It can be seen from Figure 4.14 that, Scheduler 2 (β=0 and β=1) performs better compared to Scheduler 1 in terms of delay performance as well. The fluctuations in simulation results for Scheduler 1 can be explained with the same argument given in Scenario 1 and also with the variations of channel conditions. We could conclude that SSs which have relatively bad channel conditions, experience dramatically higher delays when β is increasing for Scheduler 2. In addition, for larger β values, SSs having relatively bad channel conditions start to experience outage. It is important to note that the randomness of traffic pattern is also another parameter that affects the delay performance of SSs.



a) Scheduler 1 b) Scheduler 2

Figure 4.14 Average delay vs. SS number under Scenario 5

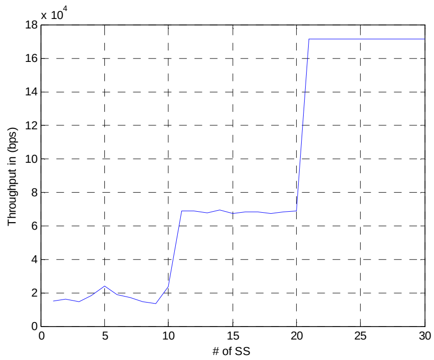
4.2 Bandwidth Request Indexes

This subsection is discussed in order to prove that less aggressive bandwidth request mechanisms will increase throughput but they can reduce delay performance. In this scenario, SSs do not send their bandwidth request in each frame i.e. unicast polling is not done on a frame by frame basis but once in every "n" frame. Each SS is assigned a Bandwidth Request Index (BRI) "n" meaning that its bandwidth request messages can be sent in every "n" frame. We note that the simulation scenarios of this subsection are all the same with the scenarios given in Subsection 4.1. BRI parameter has been introduced to each scenario and the results are collected. However, it is enough to show the effect of BRI to the overall system by using only one of the scenarios. Therefore, Scenario 1 is chosen to demonstrate the idea behind the proposal, and for the sake of brevity, the results for the rest are not given.

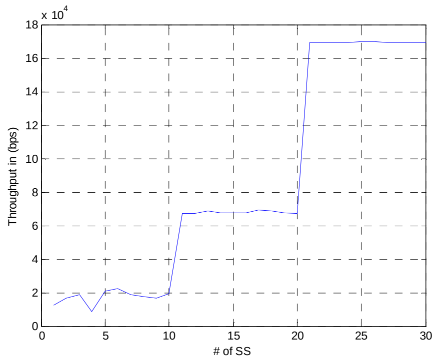
4.2.1. Effect of Bandwidth Request Index

In this scenario, BS sends unicast polls for each and every SS in every "n" frame, where "n" is different for every SS. In this scenario, BRI for VoIP SSs (SS #1 to 10) is chosen as 4 (therefore, they are able to send their bandwidth requests in every 20 ms i.e. 4*5 ms), since their packet interarrival times are 20 ms (active) and (silence) 160 ms. For video streaming model, "n" is chosen as 2 and their packets' mean interarrival time is 6ms. In order to make a fair comparison with Scenario 1 defined in Subsection 4.1, BRI for full buffer model is chosen as 1. All SSs in this scenario use 16 QAM 1/2; therefore the capacity of a slot is 12 bytes, the number of bytes that can be carried in an uplink subframe is 2100 (175*12) bytes and the capacity of uplink channel is 3.36 Mbps. But since 70 (10 + 20 + 40, for SS #1 to 10, SS #11 to 20 and SS # 21 to 30 respectively) slots are assigned for bandwidth requests in every 4 frame; there are 630 (700-70) slots available for data transmission. Thus, the available number of bytes to be carried in 4 frames for data transmission is 7560 (630*12) bytes, therefore 3.024 Mbps. It is important to note that after satisfying all SSs' MRTR parameter (10x68 kbps+ 10x18 kbps=0.86 Mbps), there exists a remaining bandwidth. This remaining bandwidth is used for SSs which generate full buffer traffic. It is obvious that there exists more remaining bandwidth for full buffer applications compared to the Scenario 1 in the Subsection 4.1.1 due to effective bandwidth request mechanism.

Figure 4.15 presents throughput versus SSs graphics of Scheduler 1 and Scheduler 2 under Scenario 1. Figure 4.15 shows that all SSs have the throughput of their MRTR parameter. The first 10 SSs which generate traffic according to a VoIP session have the throughput around 1.8 kbps each, whereas the second 10 SSs (SS#11 to 20) which generate video streaming traffic model have the throughput around 68 kbps each. Figure 4.15,compared to Figure 4.3 in Subsection 4.1, shows that the last 10 SSs (SS # 21 to 30) have more throughputs. The main reason of this increase in throughput, while other parameters remaining the same, is BRI. It is important to note that in this scenario when using Scheduler 1, the overall throughput of the system is 2.552 Mbps; overheads due to (i) bandwidth request headers, (ii) partial fitting of bandwidth requests to an integer number of slots (because of ceiling T_i) and (iii) unused slots are 0.336 Mbps, 0.037 Mbps and 0.434 Mbps respectively. On the other hand, when using Scheduler 2, the overall throughput of the system is 2.529 Mbps; overheads due to (i) bandwidth request headers, (ii) partial fitting of bandwidth requests to an integer number of slots and (iii) unused slots are 0.336 Mbps, 0.045 Mbps and 0.450 Mbps respectively.



a) Scheduler 1



b) Scheduler 2

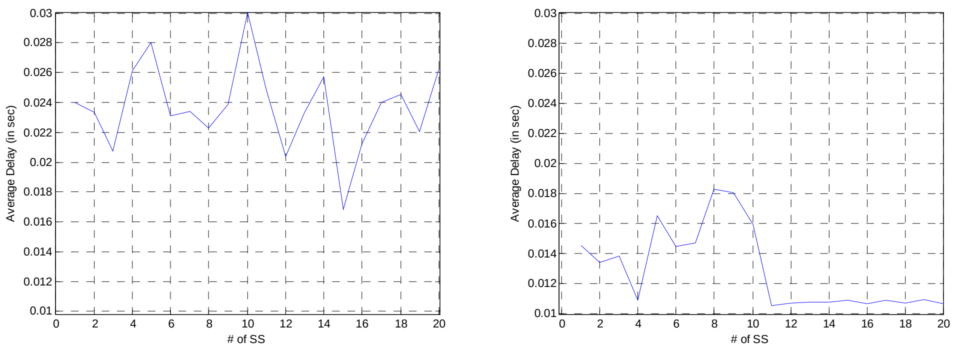
Figure 4.15 Throughput vs. SSs

Average delay variations of SS # 1-20 are given in Figure 4.16 for both Scheduler 1 and Scheduler 2. It is obvious that, for larger BRI values, the waiting time of SSs for sending their bandwidth requests would be larger and therefore the packet delay would be higher. Figure 4.4 in Subsection 4.1 compared to Figure 4.16, demonstrates better performance, since the latter has more delay because of less aggressive bandwidth request mechanism. Figure 4.16 also shows that Scheduler 2, which uses the proportional fair algorithm, performs better in overall compared to Scheduler 1.

The fluctuations in simulation results for Scheduler 1 can be explained with the same argument given in Scenario 1: Since SSs with full buffer traffic model have always illegitimate packets at the virtual queues at the BS side; most probably the last SS kept in the memory when round robin scheme is performing, would be one of the last 10 SSs. First 20 SSs have usually legitimate packets but from time to time they may have illegitimate packets. In those cases, illegitimate packets with VoIP and video traffic model SSs are mostly stacked (since the last SS kept in the memory usually being one of the last 10 SSs) and should have to wait for a turn around for all SSs. The reason for the fluctuations is that this situation may or may not occur for some SSs due to randomness of traffic patterns. The average delay variations of SSs for

Scheduler 1 converge to the Scheduler 2 when SSs with a MRTR parameter (video or VoIP) never have illegitimate packets.

Figure 4.16 (b) shows that average delay variations for SSs using VoIP traffic model also have some fluctuations since packet interarrival times for VoIP traffic are not random during an active or passive state; therefore during each state, packets experience constant delays. The length of the active and passive states are exponentially distributed thus some SSs may experience longer delays in a simulation time. It can be inferred that from Figure 4.16 (b) that these fluctuations are greater for greater bandwidth request indexes (compared to Figure 4.4 (b)) since the variation of constant packet delay values have a greater range i.e. the delay variation is between 5ms and 20ms for a BRI (VoIP) value of 4 in this case.



a) Scheduler 1 b) Scheduler 2

Figure 4.16 Average Delay vs. SSs

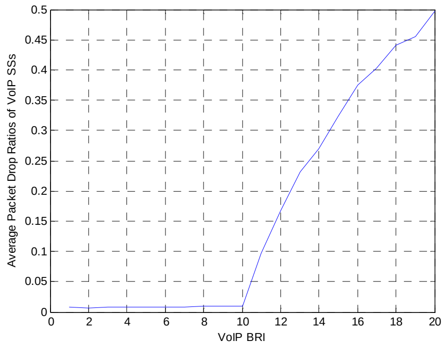
4.2.2. Extensive Study of Bandwidth Request Index

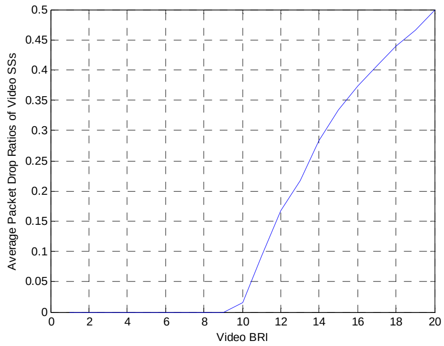
We did an extensive bandwidth request analysis for both Scheduler 1 and Scheduler 2. In particular, we repeated the same analysis given in Subsection 4.1 (Scenario 1) for different BRIs, i.e. while VoIP BRI (VoIP BRI stands for bandwidth request index of SSs using VoIP traffic model) and video BRI (video

BRI stands for bandwidth request index of SSs using video traffic model) vary in range of [1 : 20]. The results for Scheduler 1 and Scheduler 2 are similar in manner; therefore, only the results for Scheduler 2 are given in this part.

It is given in Subsection 3.4 that, a user is defined to have voice outage if more than 2% of the VoIP packets are dropped, erased or not delivered successfully within the delay bound of 50 ms; on the other hand, a user is defined to have outage if more than 2% of video frames (8 packets) are dropped, erased or not delivered successfully within the delay bound of 500 ms [28]. It is obvious that, using higher BRI for users increases the overall throughput (since larger bandwidth remains for data or payload transmission); however, one should consider the delay bounds of SSs, since increasing the BRI directly affects the transmission delay of packets. For video sessions, up to BRI = 10; video frames are never dropped; but for BRI values larger than 10, SSs are always having outage. Similarly, for VoIP SSs, they are not experiencing outage up to BRI=10. Therefore, it can be inferred for this scenario that the largest throughput in which all parameters for video and VoIP sessions are satisfied, will be achieved with VoIP BRI =10 and video BRI =10.

Figure 4.17, (a) gives the packet drop ratio of SSs using VoIP model, according to different VoIP BRIs, while BRI of SSs using video model is equal to one. Figure 4.17 (b) gives the packet drop ratios of SSs which generate video traffic for different values of video BRIs , while BRI of SSs using VoIP model is equal to one. Both VoIP and video model SSs do not experience outage up to BRI = 10, yet for BRI greater than 10, SSs always have outage.



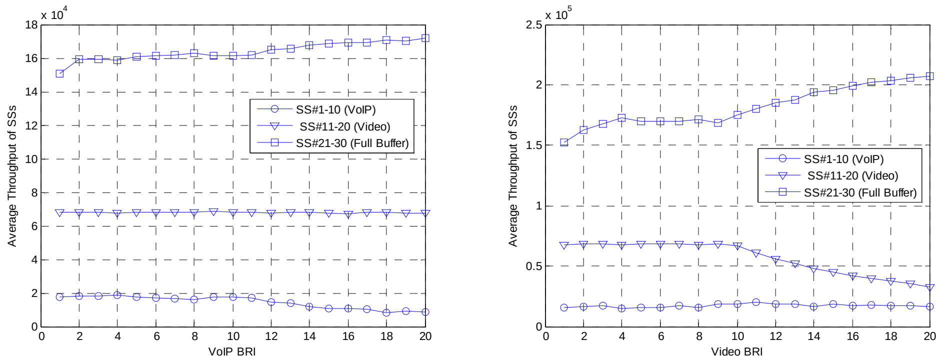


a) VoIP BRI impact b) Video BRI impact

Figure 4.17 Average Packet Drop Ratios of SSs vs. BRI

Figure 4.18 (a) presents the throughput of SSs, according to different VoIP BRIs, while BRI of SSs using video model is equal to one. Figure 4.18 (b) presents the throughput of SSs, for different values of video BRIs, while BRI of SSs using VoIP model is equal to one.

Figure 4.18 (a) shows that the throughput of the SSs using full buffer model is always increasing with the VoIP BRI; on the other hand, for SSs using video model, throughput value is neither increasing nor decreasing. The throughput value of SSs which generate traffic according to VoIP model does not change up to VoIP BRI=10. For VoIP BRI greater than 10, VoIP SSs start to experience outage, therefore their throughput decreases. Similarly, Figure 4.18 (b) shows that, full buffer SSs' throughput is increasing with the video BRI and throughput of VoIP SSs does not change. Up to video BRI=10, SSs using video model do not experience packet drops, however their throughput decreases for video BRI greater than 10.



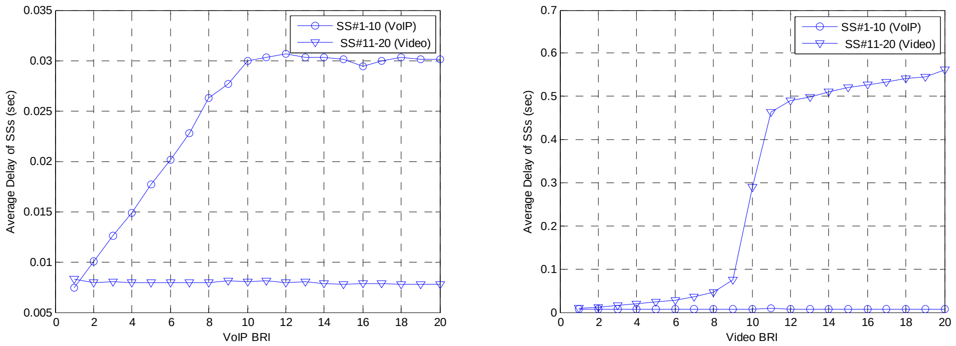
a) VoIP BRI impact b) Video BRI impact

Figure 4.18 Throughput of SSs vs. VoIP and Video BRI

Figure 4.19 (a) presents the delay values of SSs, according to different VoIP BRIs while BRI of SSs using video model is equal to one and Figure 4.19 (b) presents the delay of SSs, for different values of video BRIs, while BRI of SSs using VoIP model is equal to one.

Figure 4.19 (a) and (b) show that the delay values of SSs using video model in VoIP BRI and SSs using VoIP model in video BRI are never changing. Figure 4.19 (a) shows that the delay value of SSs using VoIP model is linearly increasing with the VoIP BRI value up to 10, however it does not change for BRI greater than 10. The main reason for this result can be explained as follows: The average delay value of VoIP SSs in Figure 4.4 (b) is about 8 ms. The maximum delay value of a VoIP packet is 50 ms (packet delay bound for VoIP). When VoIP BRI is greater than 10, the delay value of VoIP SSs in Figure 4.19 (a) is roughly the average value of 8 ms and 50 ms and does not change.

Figure 4.19 (b) presents that the delay value of SSs using video model is increasing linearly with the video BRI value up to 9. This linear increase is similar to the VoIP model, however, for video BRI greater than 9, it can be said that the packet delays of SSs are increasing dramatically. The main reason for this situation can be described by the cross layer work done for the video frames: If a video frame is started to be sent within 500ms, then the packets of that video frame are never dropped and always wait for transmission. After the first packet of a video frame is sent, the other packets of this frame is supposed to be sent, however, they should wait until the forthcoming bandwidth request time; therefore i.e. for BRI = 20 they should wait at least 100 ms.



a) VoIP BRI impact

b) Video BRI impact

Figure 4.19 Delay of SSs. vs. VoIP and Video BRI

4.3 Discussion and Comparison of Simulation Results

In this subsection, we make comparisons between simulation results. Discussion of the simulation results and the expected outcomes are given.

First and foremost, it is obvious that Scheduler 2 (QoS & channel aware scheduler) performs better in terms of both throughput and delay performances than Scheduler 1. The throughput performance of Scheduler 2 converges to the Scheduler 1 for static channel conditions (in which SSs have same modulation schemes and coding rates). This result could have been expected intuitively, since Scheduler 2 benefits from dynamic channel conditions while making a decision. If there is no difference between users' channel conditions or a relatively slow change in the channel conditions during a simulation run, the throughput performance of Scheduler 2 converges to the Scheduler 1.

Even though the throughput performance of Scheduler 2 converges to the Scheduler 1 for static channel conditions (in which SSs modulation scheme and coding rates are similar), its delay performance is still greater. It has been dictated that Scheduler 1 combines the effectiveness of round robin and leaky bucket algorithms, whereas Scheduler 2 combines proportional fair and leaky bucket algorithms. Scheduler 1 needs to take a turn among all SSs before the next scheduling is done. This scheme decreases the delay performance of SSs compared to proportional fair algorithm; in which a scheduling could be done for each SS in each and every frame.

Sub-subsection 4.1.1.2 also considers a static channel condition; however in that, SSs have different modulation schemes and coding rates. This scenario proves that Scheduler 1 does better scheduling for SSs in terms of fairness criteria. On the other hand, although Scheduler 2 does somehow less fair scheduling compared to Scheduler 1; it performs better in terms of throughput. The results prove that even in static channel conditions, Scheduler 2 achieves higher throughputs if the modulation scheme and coding rates are different for each SSs in simulation duration. It has been demonstrated that as the tuning factor for Scheduler 2 varies, different throughput and fairness performances could be achieved. Also it should be noted that tuning factor directly affects the SSs' delay variations, thus it should be chosen carefully for the SSs not to experience outage.

Subsection 4.2 proves that less aggressive bandwidth request mechanisms increase the system performance in terms of throughput; however, it decreases the delay performance. It is clear that, less aggressive bandwidth requests (or polling) increase the bandwidth to be used for data transmission purposes. Nevertheless, bandwidth requesting process for packets is slower. Therefore, the time gap between the newly coming packet time and a corresponding request time for that packet becomes wider. Hence, the delay performance of SSs decreases. After a packet (video frame) delay time exceeds the delay bound, SS drops the corresponding packet (video frame). Outage bounds are defined in order to characterize the QoS parameters of SSs by traffic models. Given that QoS parameters are satisfied for each SS (packets are not dropped), one can easily say that the larger the BRI is, the more throughput the system offers.

Chapter 5

Conclusions

This thesis presents WiMAX P2MP structure from the BS scheduler's perspective. Both MAC and PHY layers of IEEE 802.16 standards are discussed from scheduler's point of view. In particular, two BS scheduling algorithms are proposed: QoS aware and QoS & channel aware scheduling algorithms. These algorithms extended and combined the efficiency of leaky bucket principles, and the well-known proportional fair and round robin scheduling algorithms.

A brief overview for the capacity of WiMAX OFDM/OFDMA systems is given. It is important to note that the capacity of channel in WiMAX depends on a vast number of variables but only one of them is not BS (operator) dependent: modulation schemes and coding rates of SSs. Additionally, all scenarios, parameters and traffic models are gathered from [28]. By using realistic traffic models defined in [28], a packet aware scheduling structure is proposed.

We have shown that the proposed scheduling algorithms satisfy the QoS requirements of SSs regardless of their channel conditions. The throughput and delay analysis of scheduling algorithms are also carried out. The results show that the BS protects SSs having QoS requirements, from other SSs which offer traffic to the system much above of their MRTR parameters by taking advantage of the leaky bucket mechanism.

After satisfying all SSs' QoS parameters by the leaky bucket algorithm; two schemes are proposed for distributing the remaining bandwidth: proportional fair and round robin mechanisms. The round robin scheme only considers the fairness issue and does not take the channel variations and throughput maximization issues into account. On the other hand, throughput and fairness issues are both considered in our proposed proportional fair scheme. The scheduler using the proportional fair scheme is shown to achieve higher throughput values for scenarios involving different user channel conditions and variable channels, when compared with the basic scheduler using round robin only. Moreover, with a proper tuning of the parameters of the proposed proportional fair scheme, one can play out the tradeoff between throughput and fairness.

The BRI index is introduced to study the effects of bandwidth mechanisms to the overall system. We conclude that less aggressive bandwidth request mechanisms increase system throughput but they reduce the delay performance of the system. There appears to be an optimal value of BRI above and below which the system performs poorly and this BRI can be calculated in advance using the delay requirements of each SS.

For future work, we list the extension of the scheduling algorithms to the downlink case and also using the BE service class that does not use the unicast polling scheme. It may also be worthwhile to study the problem of imperfect CSI. The case that the scheduler does not assign appropriate number of slots (i.e. fewer slots) to an SS due to imperfect CSI and a BS triggered fragmentation scheme for such cases also appear to be interesting problems from the scheduler’s perspective. Latency dependent proportional fair algorithm could be considered - introducing latency indexes to proportional fair indexes could be discussed.

BIBLIOGRAPHY

[1] *IEEE 802.16-2004, IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems*, IEEE, Oct. 1,2004.

[2] *IEEE Standard 802.16e, Air Interface for Fixed Broadband Wireless Access Systems – Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, IEEE, December 2005.

[3] Mobile WiMAX – Part I: *A Technical Overview and Performance Evaluation*, www.wimaxforum.org, 2006.

[4] J.G.Andrews, A. Ghosh and R. Muhamed, *Fundamentals of WiMAX-understanding Broadband Wireless Networking*, Prentice Hall, Feb, 2007

[5] Chakchai So-In, Raj Jain, and Abdel-Karim Al Tamimi, Scheduling in WiMAX: Baseline Multi-class Simulations, *WiMAX Forum Meeting*, Washington DC, 19th November 2007

[6] Mark C. Wood, *An Analysis of the Design and Implementation of QoS over IEEE 802.16*, Washington University in St. Louis, Department of Computer Science and Engineering CSE 574, 2006.

[7] Sayenko, O. Alanen, J. Karhula, T. Hamalainen, “Ensuring the QoS Requirements in 802.16 Scheduling,” *Telecommunication laboratory, MIT Department, University of Jyvaskylia, Finland, MSWiM’06*, October 2–6, 2006, Torremolinos, Malaga, Spain.

[8] C. Cicconetti, A. Erta, L. Lenzini, and E. Mingozzi, Performance Evaluation of the IEEE 802.16 MAC for QoS Support, *IEEE Transactions On Mobile Computing*, Vol. 6, No. 1, January 2007.

[9] C. Cicconetti, L. Lenzini, E. Mingozzi, and C. Eklund, "Quality of Service Support in IEEE 802.16 Networks," *IEEE Network*, Vol. 20, Issue 2, pp. 50 - 55, 2006.

[10] K. Wongthavarawat and A. Ganz, "IEEE 802.16 Based Last Mile Broadband Wireless Military Networks with Quality of Service Support," *IEEE Military Communications Conference 2003 (MILCOM '03)*, Vol. 2, pp.779 - 784, Monterey, CA, USA, 2003.

[11] Mobilewhack.com Web site (www.mobilewhack.com)

[12] WiMAX Forum Web site (www.wimaxforum.com)

[13] Adlane Fellah, *The WiMAX Spectrum Picture*, Marvedis Inc., Montreal, Canada, 4.25.05.

[14] John G. Proakis, Masoud Salehi, *Communications Systems Engineering* Prentice Hall, 2002.

[15] H. Lee, T. Kwon, and D. Cho, "An Efficient Uplink Scheduling Algorithm for VoIP Services in IEEE 802.16 BWA Systems," *IEEE 60th Vehicular Technology Conference 2004 (VTC '04)*, Vol. 5, pp. 3070 - 3074, Los Angeles, CA, USA, 2004.

[16] *Intel Technology Journal on WiMAX*, Online article at http://download.intel.com/technology/itj/2004/volume08issue03/vol8_iss03.pdf , ISSN 1535-864X, Vol. 8, Issue 3, August 20, 2004, p.215

[17] Intel Corporation Web site (www.intel.com)

[18] Supriya Maheshwari, *An Efficient QoS Scheduling Architecture for IEEE 802.16 Wireless MANs*, Master's thesis, Indian Institute of Technology, Bombay, 2005.

[19] A.M. Law and W.D. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill, 2000.

[20] Theodore S. Rappaport, *Wireless Communications Principals and Practice*, Prentice Hall Communications Engineering and Emerging Technologies Series, 2002.

[21] James F. Kurose, Keith W. Ross, *Computer Networking, A Top Down Approach Featuring The Internet*, Addison Wesley, 2005.

[22] Henry Stark, John W. Woods, *Probability and Random Processes with Applications to Signal Processing*, Prentice Hall, 2002.

[23] C. Eklund et al., "IEEE standard 802.16: A Technical Overview of the WirelessMAN Air Interface for Broadband Wireless Access," *IEEE Comm. Mag*., vol. 40, no. 6, June 2002, pp. 98–107.

[24] T. Huovila, P. Lassilay, J. Manner, and A. Penttinen, "State of the Art Analysis of Wireless Mesh Technologies", *ABI Project Technical Report*, Helsinki, November 2006.

[25] SR Telecom, *WiMAX Capacity White Paper*, 033-100743-001, Issue 1, Canada, 2006.

[26] Megumi Kaneko, *Wireless Multi–Carrier Systems:Resource Allocation, Scheduling and Relaying*, PhD Thesis, Aalborg University, Aalborg, Denmark, November 2007.

[27] M. Cenk Ertürk, Nail Akar, "rtPS Uplink Scheduling Algorithms for IEEE 802.16 Networks", *International Symposium on Computer Networks 08*, İstanbul, Turkey, 2008.

[28] IEEE 802.16 Broadband Wireless Access Working Group, *Draft IEEE 802.16m Evaluation Methodology*, 2007-12-14.